Arbitrary pattern formation by opaque fat robots on infinite grid

Manash Kumar Kundu, Pritam Goswami, Satakshi Ghosh & Buddhadeb Sau

Taylor & Francis
Taylor & Francis Group

Check for updates

Arbitrary pattern formation by opaque fat robots on infinite grid

Manash Kumar Kundu[a], Pritam Goswami[b], Satakshi Ghosh[b] and Buddhadeb Sau[b]

[a]Department of Science and Humanities, Gayeshpur Government Polytechnic, Kalyani, West Bengal, India;
[b]Department of Mathematics, Jadavpur University, Kolkata, West Bengal, India

ABSTRACT

Arbitrary Pattern formation ($\mathcal{APF}$) by a swarm of mobile robots is a widely studied problem in the literature. Many works regarding $\mathcal{APF}$ have been proposed on plane and infinite grid by point robots. But in practical application, it is impossible to design point robots. In Bose et al. [Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. Theor Comput Sci. 2020;815:213–227], the robots are assumed opaque fat robots but the environment is plane. To the best of our knowledge, no work till now ever considered the $\mathcal{APF}$ problem assuming opaque fat robots on infinite grid where movements are restricted. In this paper, we have provided a collisionless distributed algorithm and solved $\mathcal{APF}$ using 9 colours.

1. Introduction

Nowadays, the distributed system is gaining popularity among researchers due to its many positive aspects. Designing a centralised system and also maintaining its robustness is not at all cost-effective. But these factors can be handled easily and effectively in a distributed system. Swarm robotics is an example of such distributed system. In swarm robotics, more than one robot is considered in an environment (plane, network etc.). The robots are considered to be autonomous (i.e. they do not have any central control), homogeneous (i.e. all robots execute the same algorithm) and identical (they are not indistinguishable by their physical appearance). Before the study of swarm robotics, designing a robot to do a specific task was costly as it would have needed many strong capabilities. But designing a swarm of robots is cheaper than using such robot with many capabilities as the goal now become to design the robots with minimal capabilities such that they can do the same task autonomously. Among many applications of swarm robots military operations, border surveillance, cleaning of a large surface, rescue operations, disaster management etc. are the ones that use the swarm robots vividly in present days. So, it is evident why swarm robotics has gained such popularity in the industry and among researchers in the current scenario.

Among many problems (eg. gathering, scattering, exploration) Arbitrary Pattern Formation ($\mathcal{APF}$) is a classical problem in the field of swarm robotics. In this problem, a swarm of robots which are deployed in an environment (plane, graph etc.), need to form an already decided pattern which is given as input to the robots. The robots can move freely in the plane but in the case of a graph, they always move through an edge of the graph. There are mainly four models of robots depending upon

CONTACT Manash Kumar Kundu ✉ manashkrkundu.rs@jadavpuruniversity.in ▣ Department of Science and Humanities, Gayeshpur Government Polytechnic, Kalyani, West Bengal 741234, India; Pritam Goswami
✉ pritamgoswami.math.rs@jadavpuruniversity.in ▣ Department of Mathematics, Jadavpur University, Kolkata, West Bengal 700032, India; Satakshi Ghosh ✉ satakshighosh.math.rs@jadavpuruniversity.in ▣ Department of Mathematics, Jadavpur University, Kolkata, West Bengal 700032, India; Buddhadeb Sau ✉ buddhadeb.sau@jadavpuruniversity.in ▣ Department of Mathematics, Jadavpur University, Kolkata, West Bengal 700032, India

their capabilities. These models are $\mathcal{OBLOT}$, $\mathcal{FST\,A}$, $\mathcal{FCOM}$ and $\mathcal{LUMI}$. In all of these models, robots are considered to be autonomous, homogeneous, identical and anonymous (i.e. the robots do not have any unique identifiers). In the $\mathcal{OBLOT}$ model, the robots are considered to be oblivious (i.e. the robots do not have any persistent memory to remember any previous state) and silent (i.e. the robots do not have any means of communication among themselves). In the $\mathcal{FST\,A}$ model, the robots are silent but not oblivious. In the $\mathcal{FCOM}$ model, the robots are oblivious but not silent. And in the $\mathcal{LUMI}$ model, the robots are neither silent nor oblivious. There are many works of $\mathcal{APF}$ which has considered the $\mathcal{OBLOT}$ model in literature [1–5].

Activation time of the robots plays an important role to design algorithms for a swarm of robots. It is assumed that a scheduler controls the activation of robots during the execution of any algorithm. Mainly, there are three types of schedulers that have been considered in many previous works. These schedulers are $\mathcal{FSYNC}$ or a fully synchronous scheduler, $\mathcal{SSYNC}$ or a semi-synchronous scheduler and $\mathcal{ASYNC}$ or an asynchronous scheduler. In the case of a $\mathcal{FSYNC}$ scheduler, time is divided into global rounds of the same duration and each robot is activated at the beginning of each round. $\mathcal{SSYNC}$ scheduler is a more general version of the $\mathcal{FSYNC}$ scheduler. In the case of the $\mathcal{SSYNC}$ scheduler, time is divided into global rounds of the same duration as it has been done for $\mathcal{FSYNC}$ scheduler. But at the beginning of each round, the set of activated robots can be a proper subset of the set of all robots (i.e. all robots may not get activated at the beginning of each round). Now, in the case of the $\mathcal{ASYNC}$ scheduler, there is no sense of global rounds. Any robot can get activated at any time. So, the $\mathcal{ASYNC}$ scheduler is more general and realistic among all the scheduler models.

In any model, the robots can be considered as transparent or opaque. In the case of transparent robots, a robot can see another robot even if there are other robots between them. But in the case of opaque robots, a robot can not see another robot if there are other robots between them. There are many works where both these models have been considered [1–9]. Opaque robots can be considered to be dimensionless (i.e. point robots) [2–5,9–13] or they can have some dimension (i.e. fat robots) [1]. In the literature on $\mathcal{APF}$, there are many works which have considered the robots to be dimensionless (i.e. point robots) and opaque [2,5]. But in practical application, it is impossible to design a point robot as any physical object must have some dimensions. So in our work, we have considered the robots to have some dimension. In fact, we have considered the robots to be a disc of radius 'rad', where $rad \leq \frac{1}{2}$.

In this paper, we are interested in the problem of $\mathcal{APF}$ on an infinite grid where the robots are considered to be fat and opaque and are placed on distinct vertices of the grid. The goal is to design an algorithm $\mathcal{A}$ such that the robots after executing $\mathcal{A}$ form a pattern which is provided to each of the robots as input. In this paper, we have provided such an algorithm that solves the problem of $\mathcal{APF}$ under an $\mathcal{ASYNC}$ scheduler.

1.1. Earlier works

The arbitrary pattern formation problem was introduced first in [14] and it has become a popular topic for research. It has been vastly studied under different types of environments and different types of settings [1–8,10,13,15–18]. In most of these works, the basic assumption was that the robots are points and they do not have obstructed visibility. But in a practical application-based scenario designing a point robot is impossible because every physical object has a certain dimension. So in [1], authors have considered a swarm of fat and opaque robots and shown that this swarm can form any given pattern from any asymmetric configuration without collision under the $\mathcal{LUMI}$ model using 10 colours in a plane. This luminous model was first introduced in [19] by Peleg. The visible lights can be used as a means of communication and persistent memory. Designing a collision-free algorithm in plane is easier than handling collision in discrete domain. This is because, in plane the robots can move freely in any direction avoiding other robots but in discrete domain, there can be only one single path to reach from one point to another point. This is why many researchers became interested to study the problem of $\mathcal{APF}$ in discrete domain. In [3], an algorithm for $\mathcal{APF}$ has been provided for a swarm of point robots on an infinite grid but considering full and Unobstructed visibility. Now in [5], considering

obstructed visibility model the authors have shown that a circle can be formed on an infinite grid from any initial configuration if the opaque point robots in the swarm have one-axis agreement and 7 colours. Then, in [6] authors have presented another algorithm where a swarm of opaque point robots on an infinite grid can form the given pattern in finite time using one-axis agreement and 8 colours. But none of these works considered fat robots on infinite grid and solve the problem of $\mathcal{APF}$.

To the best of our knowledge, there is no work till now which has considered fat robots on infinite grid and provided any algorithm for arbitrary pattern formation on the grid. So, in this paper, we have considered a swarm of opaque fat robots on an infinite grid and provided an algorithm (APFFATGRID) where the swarm can form a predefined given pattern on the grid using 9 colours.

1.2. Problem description and our contribution

This paper deals with the problem of arbitrary pattern formation on an infinite grid using luminous opaque fat robots with 9 colours. The robots are considered to be a disk having a fixed radius '*rad*', which is less or equal to $\frac{1}{2}$. The robots manoeuvre in a LOOK-COMPUTE-MOVE (LCM) cycle under an adversarial asynchronous scheduler. The robots are autonomous, anonymous, identical and homogeneous. The robots only move to one of its four adjacent grid points and their movement is considered to be instantaneous (i.e. a robot can only be seen on a grid point). The robots have one-axis agreement. Here, it is assumed that the robots do not agree upon any global coordinate though all robots agree on the direction and orientation of the *x*-axis. Initially, the centre of each robot is on a grid point of the infinite grid and a target pattern is provided to each of them. The robots are needed to agree on a global coordinate system and embed the target pattern according to the global coordinate and then move to the target locations to form the target pattern.

The main difficulty of $\mathcal{APF}$ lies in the problem of Leader Election problem. For that, the initial configuration is assumed to be asymmetric or there is at least one robot on some line of symmetry. Even with this assumption, it is quite hard to elect a leader as the vision of the robots becomes obstructed since the robots are opaque and fat. So, the main challenge of this problem is to elect a leader depending on the local view of each robot. The algorithm described in this paper does so. Another massive challenge of this problem is to avoid collision during the movement of robots on the grid. Our algorithm handles this by providing sequential movement of the robots and for this purpose $\mathcal{LUMI}$ model has been used.

The problem, we have considered in this paper, is very practical in nature. Restricted movement, robots with dimension and obstructed visibility all these assumptions are very much practical in terms of designing robots. The algorithm presented in this paper solves the $\mathcal{APF}$ on infinite grid with a swarm of luminous, opaque and fat robots with finite time. A comparison table is provided below which will help readers to compare our work to the previous such works.

Paper	Environment	Visibilty	Robot Type	#Colours
Bose et al. [3]	Grid	Unobstructed	Point	0
Bose et al. [2]	Plane	Opaque	Point	6
Kundu et al. [6]	Grid	Opaque	Point	8
Bose et al. [1]	Plane	Opaque	Fat	10
This paper	Grid	Opaque	Fat	9

2. Model and definitions

2.1. Model

Grid: The infinite two-dimensional grid $\mathcal{G}$ is a weighted graph $\mathcal{G} = (V, E)$ such that each node $v \in V$ has four adjacent nodes v_0, v_1, v_2 and $v_3 \in V$ and the edges vv_i (mod 4) $\in E$ is perpendicular to the edge

$vv_{i+1 \pmod 4} \in E$. Also, the weight of each edge $e \in E$ is basically the length of the edge e which is considered to be 1 unit in this work.

Robots: In this work, a set of n robots $R = \{r_0, r_1, \ldots, r_{n-1}\}$ are considered to be autonomous, anonymous, homogeneous and identical. This means that the robots do not have any central control, they do not have any unique identifiers such as IDs and they are indistinguishable by their physical appearance. The robots are also considered to have some dimension, i.e. the robots are considered to be a disk of radius 'rad' ($rad \leq \frac{1}{2}$) rather than points. The robots are deployed on a two-dimensional infinite grid $\mathcal{G}$, where each of them is initially positioned in such a way that their centre is on distinct grid points of $\mathcal{G}$. The robots are considered to have an agreement over the direction and orientation of x-axis i.e, all the robots have an agreement over left and right but the robots do not have any agreement over the y-axis. Also, they do not have knowledge of any global coordinate system other than their agreement over the direction of x-axis. Here in this paper, we have considered the robots to have light. A light of any robot can have $\mathcal{O}(1)$ distinct colours. A robot $r \in R$ can see the colour of its own light and the colour of the lights of other robots that are visible to r. In this work, we have assumed that the light of each robot has nine distinct colours namely `off`, `terminal1`, `candidate`, `call`, `moving1`, `reached`, `leader1`, `leader` and `done`.

Look-Compute-Move cycles: A robot $r \in R$, when active, operates according to the LOOK-COMPUTE-MOVE (LCM) cycle. In the LOOK phase, a robot takes the snapshot of the configuration to get the positions represented in its own local coordinate system and the colours of the light of all other robots visible to it. Then, r performs the computation phase where it decides the position of the adjacent grid point where it will move next and changes the colour of its light if necessary depending on the input it got from the LOOK phase. In the MOVE phase, r moves to the decided grid point or makes a null move. The movements of robots are restricted only along grid lines from one grid point to one of its four adjacent grid points. The movements of robots are assumed to be instantaneous in discrete domain. Here, we assume that the movements are instantaneous, i.e. they are always seen on grid points, not on edges.

Scheduler: We assume that the robots are controlled by an asynchronous adversarial scheduler. That implies the duration of the three phases LOOK, COMPUTE and MOVE are finite but unbounded. So, there is no common notion of round for this asynchronous scheduler.

Visibility: The visibility of robots 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 there is a point p_{r_j} on the boundary of r_j and p_{r_i} on the boundary of r_i such that the line segment $\overline{p_{r_i}p_{r_j}}$ does not intersect with any point occupied by other robots in the configuration. Now, it follows from the definition that r_i can see r_j implies r_j can see r_i.

Configuration: We assume that the robots are placed on the infinite two-dimensional grid $\mathcal{G}$. Next we define a function $f : V \to \{0, 1\}$, where $f(v)$ is the number of robots placed on a grid point v. Then $\mathcal{G}$ together with the function f is called a configuration which is denoted by $\mathbb{C} = (\mathcal{G}, f)$. For any time $T, \mathbb{C}(T)$ will denote the configuration of the robots at T.

2.2. Notations and definitions

We have used some notations throughout the paper. A list of these notations is mentioned in the following table.

   Terminal Robot: A robot r is called a terminal robot if there is no robot below or above r on $\mathcal{L}_V(r)$.

$\mathcal{L}_1$	First vertical line on left that contains at least one robot.
$\mathcal{L}_V(r)$	The vertical line on which the robot r is located.
$\mathcal{L}_H(r)$	The horizontal line on which the robot r is located.
$\mathcal{L}_l(r)$	The left immediate vertical line of robot r which has at least one robot on it.
$\mathcal{R}_l(r)$	The right immediate vertical line of robot r which has at least one robot on it.
$H_L^O(r)$	Left open half for the robot r.
$H_L^C(r)$	Left closed half for the robot r (i.e. $H_L^O(r) \cup \mathcal{L}_V(r)$).
$H_B^O(r)$	Bottom open half for the robot r.
$H_B^C(r)$	Bottom closed half for the robot r (i.e. $H_B^O(r) \cup \mathcal{L}_H(r)$).
$H_U^O(r)$	Upper open half for the robot r.
$H_U^C(r)$	Upper closed half for the robot r (i.e. $H_U^O(r) \cup \mathcal{L}_H(r)$).
K	The horizontal line passing through the middle point of the line segment between two robots with light `candidate` or `call` or `reached` on the same vertical line.
$l_{next}(r)$	The next vertical line on the right of $\mathcal{L}_V(r)$.
$\mathcal{H}_{last}$	The lowest horizontal line having a robot with colour `done`.

Symmetry of a vertical line L w.r.t K: Let λ be a binary sequence defined on a vertical line L such that ith term of λ is defined as follows:

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

Since there are two ith grid points from $K \cap L$ on the line L (above K and below K), there are two such values of λ, say λ_1 and λ_2. If $\lambda_1 = \lambda_2$, then L is said to be symmetric with respect to K. Otherwise, it is said to be asymmetric with respect to K. Henceforth, whenever the symmetry of a line is mentioned, it means the symmetry of the line with respect to K.

Dominant half: A robot r is said to be in the dominant half if for $\lambda_1 > \lambda_2$ (lexicographically) on $\mathcal{R}_l(r)$, r and the portion of $\mathcal{R}_l(r)$ corresponding to λ_1 lie on same half-plane delimited by K.

3. The algorithm

The main result of the paper is Theorem 3.1. The proof of the 'only if' part is the same as in the case for point robots, proved in [2]. The 'if' part will follow from the algorithm presented in this section.

Theorem 3.1: *For a set of opaque fat robots having one-axis agreement, $\mathcal{APF}$ 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 $\mathbb{C}(0)$ does not admit the unsolvable symmetry stated in Theorem 3.1. Our Algorithm executes in two phases. In the first phase, a leader is elected and in the second phase, the robots form the target pattern embedded on the grid using the location of the leader as an agreement to the origin of a global coordinate system. The phases are described in detail in the following subsections.

3.1. Phase 1

Initially, at $\mathbb{C}(0)$ all the robots are on the grid $\mathcal{G}$ with colour `off`. Note that in $\mathbb{C}(0)$, there are at least one and at most two robots that see no robot on its left open half and no robot either above or below it on the same vertical line (i.e. these robots are terminal on their own vertical line). These robots change their colours to `terminal1`. A robot with colour `terminal1` changes its colour to `candidate` and moves if it sees it has its left open half empty. Also, if a robot r with colour `candidate` is a singleton in $H_L^C(r)$ and all robots in $\mathcal{R}_l(r)$ are `off`, it changes its colour to `leader1`. Note that due to the asynchronous scheduler, it might happen that r is a singleton in $H_L^C(r)$ with colour `candidate` and there

is another robot r' on $\mathcal{R}_l(r)$ with colour `terminal1`. In this case, if r awakes, it does not change its colour to `leader1` as it does not see all robots on $\mathcal{R}_l(r)$ have colour `off`. Also if r' awakes, it sees r with colour `candidate` in $\mathcal{L}_l(r')$ and turns its colour to `off`. In this scenario, r becomes singleton in $H_L^C(r)$ and sees all robots on $\mathcal{R}_l(r)$ have colour `off`. So, r changes its colour to `leader1`. Now consider that both r and r' are on the same vertical line $\mathcal{L}_V(r)$ with colour `candidate` such that there is no robot between $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r')$. Note that in this configuration, all robots on $\mathcal{R}_l(r)$ (i.e. $\mathcal{R}_l(r')$) have colour `off`. In this case, both r and r' check the symmetry of $\mathcal{R}_l(r)$ with respect to the line K (i.e. the horizontal line which is equidistant from both r and r'). If $\mathcal{R}_l(r)$ is not symmetric with respect to K, then one of r or r' whichever is on the dominant half changes the colour to `leader1`. On the other hand, if $\mathcal{R}_l(r)$ is symmetric, both the robots r and r' change their colours to `call` from `candidate`. Now all the robots on $\mathcal{R}_l(r)$ have colour `off` and all of them can see exactly two robots with colour `call` on their left immediate line. Note that since all the robots on $\mathcal{R}_l(r)$ can see both r and r', all of them also know the line K. Now if there is any robot on K, it changes its colour to `leader1`. Otherwise, the robots on $\mathcal{R}_l(r)$ (at least one and at most two robots) which are closest to K change the colours to `moving1`. A robot with colour `off` on $\mathcal{R}_l(r)$ which is not closest to K, changes its colour to `moving1` when it sees another robot with colour `moving1` on the same vertical line. Note that after a finite time, at least all robots either above or below K which are on $\mathcal{R}_l(r)$ change their colours to `moving1` if $\mathcal{R}_l(r)$ is symmetric with respect to K. Now suppose a robot with colour `moving1` say r_1, is terminal on $\mathcal{R}_l(r)$. Also, note that r_1 can see at least one of r and r' on $\mathcal{L}_l(r_1)$. Now, if r_1 sees another robot r_2 on $\mathcal{L}_V(r_1)$ and no robot with colour `reached` on $\mathcal{L}_l(r_1)$, it moves vertically opposite to r_2. Otherwise, if it is singleton on $\mathcal{L}_V(r_1)$ and sees no robot with colour `reached` on $\mathcal{L}_l(r_1)$, it moves vertically according to its positive y-axis until there is no robot either in $H_U^C(r_1) \cap \mathcal{L}_l(r_1)$ or in $H_B^C(r_1) \cap \mathcal{L}_l(r_1)$ and then towards left until it reaches $\mathcal{L}_V(r)$. Now when r or r' with colour `call` sees all robots on $\mathcal{R}_l(r)$ have colour `off` and sees a robot with colour `moving1` or `reached` on the same vertical line, then it changes its colour to `reached`. Due to the asynchronous environment, it might happen that one of r or r' does not see a robot with colour `moving1` on the same vertical line, but it is guaranteed that it will see a robot with colour `reached` on the same vertical line after a finite time. So, r or r' can change their colours to `reached` if all robots on $\mathcal{R}_l(r)$ have colour `off` and there is a robot on $\mathcal{L}_V(r)$ with colour `reached`. So after a finite time, both the robots with colour `call` change their colours to `reached` (when there was no robot on $K \cap \mathcal{R}_l(r)$). Now a robot say r_3 with colour `moving1` on $\mathcal{L}_V(r)$ moves to the left if all robots on $\mathcal{R}_l(r_3)$ are with colour `off` and it can see a robot with colour `reached` on $\mathcal{L}_V(r_3)$. Due to asynchrony, it may happen that r and r' changed their colours to `reached` and after that a robot say r_4, on $\mathcal{R}_l(r)$ changes its colour to `moving1`. Observe that in this case, the robot r_4 changes its colour to `off` whenever it sees at least one robot with colour `reached` on $\mathcal{L}_l(r_4)$, otherwise the robots with colour `moving1` on $\mathcal{L}_V(r)$ will not move left. So, after a finite time, all robots with colour `moving1` on $\mathcal{L}_V(r)$ move to $\mathcal{L}_1$ and at this moment r and r' will be the only two robots with colour `reached` on $\mathcal{L}_V(r)$ that are terminal also. In this situation, r and r' change their colours to `candidate`. Now due to asynchrony, it may happen that r and r' changed their colours to `candidate` and after that a robot say r_5, on $\mathcal{R}_l(r)$ changes its colour to `moving1`. In this case, the robot r_5 changes its colour to `off` whenever it sees at least one robot with colour `candidate` on $\mathcal{L}_l(r_5)$. Therefore, then r and r' are with colours to `candidate` and all robots on $\mathcal{R}_l(r)$ have colour `off`. So, they again check the symmetry of the new $\mathcal{R}_l(r)$ repeating the whole process. Thus after a finite time, a robot with colour `off` or a robot r or r' with colour `candidate` whoever is on dominant half changes its colour to `leader1`. A robot say r_l with colour `leader1` always moves to the left when it sees other robot in $H_L^C(r_l)$ or $l_{\text{next}}(r_l)$, no robot with colour `call` on $\mathcal{L}_l(r_l)$ and no robot with colour `candidate` on $\mathcal{L}_V(r_l)$.

Note that, a robot with colour `call` changes its colour to `off` if it sees a robot with colour `leader1` on its right immediate line. Also, a robot with colour `candidate` changes the colour to `off` when it sees a robot with colour `leader1` on the same vertical line. r_l moves to the left until it becomes the singleton robot on the leftmost line of the configuration and there is no robot on $l_{\text{next}}(r_l)$ and then moves according to its positive y-axis until either one of $H_U^C(r_l)$ and $H_B^C(r_l)$ has no other robot. Note that

Algorithm 1: APFFATGRID: Phase 1

```
1   Procedure PHASE1()
2        r ← myself. L_i is a horizontal line with i ≥ 2. L_r = horizontal line where r resides.
3        L_1 = lowest horizontal line.
4        if r.light = off then
5            if there is no robot in H_L^O(r), no robot with light leader1 in R_I(r) ∪ L_V(r) and r is terminal on L_V(r) then
6                r.light ← terminal1
7            else if there are exactly two robots in L_I(r) and their lights are call and r is closest to K then
8                if r is on K then
9                    r.light = leader1
10               else
11                   r.light = moving1
12           else if there is a robot with light moving1 in L_V(r) then
13               r.light = moving1
14       else if r.light = terminal1 then
15           if there is no robot in H_L^O(r) then
16               r.light ← candidate
17               move left
18           else if there is a robot with light candidate in L_I(r) then
19               r.light = off
20       else if r.light = candidate then
21           if r is singleton in H_L^C(r) and all robots in R_I(r) are off then
22               r.light ← leader1
23           else if there is a robot with light candidate or call on L_V(r), r is terminal on L_V(r) and all robots in R_I(r) are off then
24               if R_I(r) is symmetric with respect to K then
25                   r.light = call
26               else
27                   if r is in the dominant half then
28                       r.light = leader1
29           else if there is a robot with light leader1 on L_V(r) then
30               r.light = off
31       else if r.light = moving1 then
32           if there is at least one robot with light call and no robot with light reached in L_I(r) and r is terminal on L_V(r) then
33               if there is other robot both in H_U^C(r) ∩ L_I(r) and H_B^C(r) ∩ L_I(r) then
34                   if there is a robot r' on L_V(r) then
35                       move opposite to r'
36                   else
37                       move according to its positive y-axis
38               else
39                   move left
40           else if there is a robot with light reached on L_V(r) and all robots in R_I(r) are off then
41               move left
42           else if there is at least one robot with light reached or candidate in L_I(r) then
43               r.light = off
44       else if r.light = call then
45           if there is a robot with light moving1 or, reached on L_V(r) and all robots in R_I(r) are off then
46               r.light = reached
47           else if there is a robot with light leader1 in R_I(r) then
48               r.light = off
49       else if r.light = reached then
50           if there is a robot with light reached or candidate on L_V(r), r is terminal on L_V(r) and all robots in R_I(r) are off then
51               r.light = candidate
52       else if r.light = leader1 then
53           if there is other robot in H_L^C(r) or I_next(r), no robot with light call in L_I(r) and no robot with light candidate on L_V(r) then
54               move left
55           else
56               if there is other robot both in H_U^C(r) and H_B^C(r) then
57                   move vertically according to its positive y-axis
58               else
59                   r.light = leader
```
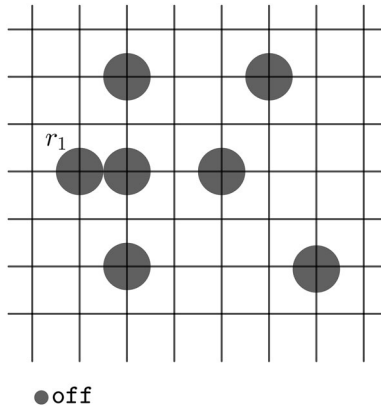
Figure 1. r_1 is singleton robot on $\mathcal{L}_1$.

it may happen due to the asynchronous environment that another robot with colour `candidate` moves to $l_{\text{next}}(r_l)$ while r_l is on $\mathcal{L}_1$. In this case, when r_l activates again it finds out it has non-empty $l_{\text{next}}(r_l)$ and moves left again even it was moving vertically in the previous activation. In this situation, when r_l reaches a point where either one of $H_U^C(r_l)$ and $H_B^C(r_l)$ has no other robot, it changes its colour to `leader` and *Phase 1* ends.

The following Theorem 3.2 and Lemmas 3.3–3.15 justify the correctness of the Algorithm 1.

Theorem 3.2: *For any initial configuration $\mathbb{C}(0), \exists T > 0$ such that $\mathbb{C}(T)$ have exactly two robots with light* `candidate` *or exactly one robot with light* `leader1` *in L_1.*

Proof: Observe that there can be at least one and at most two robots in $\mathbb{C}(0)$ such that they have their left open half empty and are terminal on $\mathcal{L}_1$. Let there is only one robot r_1, who has $H_L^O(r_1)$ empty and is terminal on $\mathcal{L}_1$ (Figure 1). This implies r_1 is singleton on $\mathcal{L}_1$. In this case, r_1 changes its colour to `terminal1` at some time $T' > 0$ and eventually changes to `leader1` at a time $T > T'$.

Now let us consider the case where there are two robots r_1 and r_2 such that both r_1 and r_2 are terminal on $\mathcal{L}_1$ in $\mathbb{C}(0)$. Now if any one of r_1 or r_2 awakes, it changes its colour to `terminal1`. A robot with colour `terminal1` moves left after changing its colour to `candidate` if it has its left open half empty. Due to asynchronous environment, the following cases may occur.

Case-I: Let us consider the case where r_1 already changed its colour to `candidate` from `terminal1` and moved to $\mathcal{L}_1$ at a time $T_1 > 0$ and r_2 wakes after T_1 (Figure 2). Then r_2 remains with colour `off` as it sees it is not on $\mathcal{L}_1$ anymore. Then r_1 during the next activation sees it is singleton on $H_L^C(r_1)$ and all robots on $\mathcal{R}_l(r_1)$ have colour `off`. So, it changes its colour to `leader1`.

Case-II: Let us consider the case where r_1 already changed its colour to `candidate` from `terminal1` and moved to $\mathcal{L}_1$ at a time $T_1 > 0$ and r_2 wakes before T_1 and changes its colour to `terminal1` at a time $T_2 \geq T_1$ (Figure 3). Now if again r_1 wakes between the times T_1 and T_2 (in this scenario $T_1 > T_2$), then it sees all robots on $\mathcal{R}_l(r_1)$ have colour `off` and r_1 is singleton on $H_L^C(r_1)$. So, r_1 changes its colour to `leader1` and r_2 does not change its colour as $H_L^C(r_2)$ has other robots. Now if r_1 wakes at a time T_3 where $T_3 > T_2$ and r_2 has not woke again, then it does not change its colour to `leader1` as it sees r_2 with colour `terminal1` on $\mathcal{R}_l(r_1)$. Now when r_2 wakes again at a time $T_4 > T_2$, it changes its colour to `off` as it sees r_1 with colour `candidate` on $\mathcal{L}_l(r_2)$. Now when r_1 wakes after T_4 again, it sees it is singleton on $H_L^C(r_1)$ and have all robots with colour `off` on $\mathcal{R}_l(r_1)$ and so changes its colour to `leader1`.

Case-III: Let us consider the case where r_1 already changed its colour to `candidate` from `terminal1` and moved to $\mathcal{L}_1$ at a time $T_1 > 0$ and r_2 wakes before T_1 and changes its colour to `terminal1` at a time $T_2 < T_1$. Now, let r_2 wakes again at a time T_3.
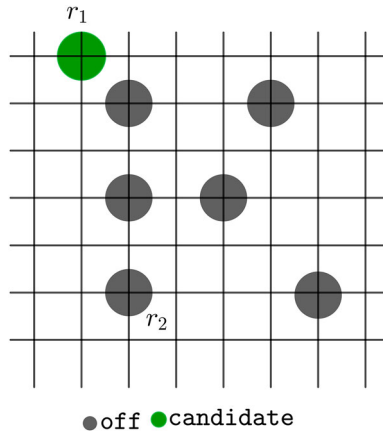
Figure 2. r_1 changes its colour to candidate and moves to $\mathcal{L}_1$ at time T_1 and r_2 wakes after time T_1.
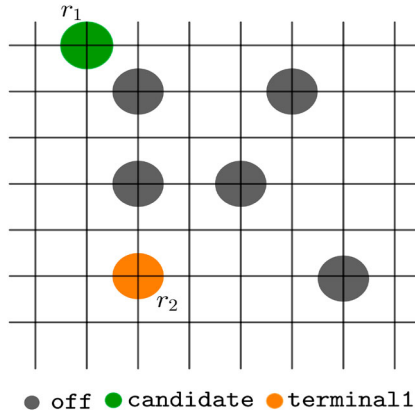

Figure 3. r_1 changed its colour to candidate and moves to $\mathcal{L}_1$ at time T_1 and r_2 changes its colour to terminal1 at time $T_2 \geq T_1$.

Case-III(a): Now if $T_3 > T_1$, then even if r_1 wakes again between T_3 and T_1, it sees r_2 with colour terminal1 on $\mathcal{R}_l(r_1)$. So, it does not change its colour to leader1. Now r_2 at time T_3 wakes and sees r_1 with colour candidate on $\mathcal{L}_l(r_2)$ and so changes its colour to off. Now $\exists T_4$ such that r_1 wakes at $T_4 > T_3$ and sees it is singleton on $H_L^C(r_1)$ and have all robots on $\mathcal{R}_l(r_1)$ with colour off. So, r_1 changes its colour to leader1.

Case-III(b): Now if $T_3 = T_1$, then both r_1 and r_2 changes its colour to candidate and moves to $\mathcal{L}_1$. In this case, there will be two robots with colour candidate on $\mathcal{L}_1$ (Figure 4). **Case-III(c):** Now when $T_3 < T_1$, it is similar as case-III(a).

Note that above all cases are exhaustive and in each case there is a time $T > 0$ such that in $\mathbb{C}(T)$, there is either one robot with colour leader1 or two robots with colour candidate on $\mathcal{L}_1$. ∎

Lemma 3.3: *Any robot r with colour* candidate *or,* call *or,* reached *always can see all robots with colour* off *in* $\mathcal{R}_l(r)$ *(if exist) and vice versa.*

Proof: Let us consider that there is exactly one robot *r* and no other robot is on $\mathcal{L}_V(r)$. In this case, it is obvious that *r* can see all robots including the robots with colour off on $\mathcal{R}_l(r)$ and vice versa.
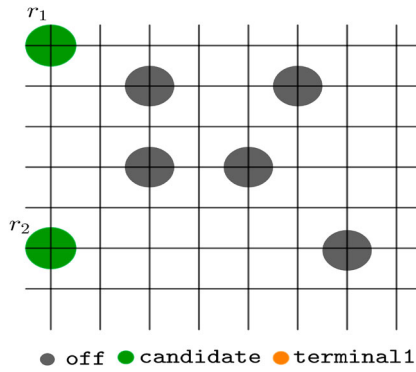
Figure 4. r_2 is with colour `terminal1` and r_1 changes its colour to `terminal1` and both r_1 and r_2 move to $\mathcal{L}_1$ changing their colour to `candidate` at the same time.
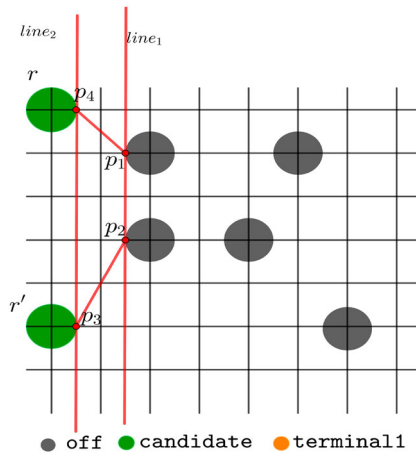

Figure 5. Area bounded by the quadrilateral $\overline{p_1 p_2 p_3 p_4}$ is convex.

Now let us consider that there are at least two robots r and r' on $\mathcal{L}_V(r)$ where colour of r and r' can be any one of `candidate`, `call` or, `reached` and r is above r'. Now there are two cases.

Case-I: There are other empty vertical lines between $\mathcal{L}_V(r)$ and $\mathcal{R}_I(r)$. In this case, let us take the common tangent $line_1$ of all robots on $\mathcal{R}_I(r)$ which is parallel to the line $\mathcal{R}_I(r)$ and nearest to $\mathcal{L}_V(r)$ and similarly take the common tangent $line_2$ of the robots on line $\mathcal{L}_V(r)$ parallel to $\mathcal{L}_V(r)$ and nearest to $\mathcal{R}_I(r)$. Let us denote the points where $line_1$ touches the terminal robots on $\mathcal{R}_I(r)$ as p_1 and p_2 respectively (p_1 is above p_2) and the points where $line_2$ touches r' and r as p_3 and p_4 respectively. Now let us draw a line segment say $line_3 = \overline{p_4 p_1}$ and $line_4 = \overline{p_3 p_2}$. Observe that the area bounded by the lines $line_1, line_2, line_3$ and $line_4$ is a trapezoid which is a convex set containing no other robot (Figure 5). Let r_1 be any robot with colour `off` on $\mathcal{R}_I(r)$. Let $line_1$ touches the robot r_1 at a point say P. Then the line segments $\overline{Pp_3}$ and $\overline{Pp_4}$ contains no robot on them. So, each of r and r' can see r_1. Thus r and r' can see all robots with colour `off` on $\mathcal{R}_I(r)$ and all robots with colour `off` can see both of r and r'.

Case-II: Next let there is no other vertical line between $\mathcal{L}_V(r)$ and $\mathcal{R}_I(r)$. Note that all robots with colour `off` must be between the lines $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r')$. In this scenario, let us draw the common tangent $line_1$ of the robots on $\mathcal{R}_I(r)$ which is parallel to $\mathcal{R}_I(r)$ and nearest to $\mathcal{L}_V(r)$. Let us denote the points where $line_1$ touches the terminal robots on $\mathcal{R}_I(r)$ as p_1 and p_2 (p_1 is above p_2). Let us now denote the points where boundary of r and r' intersect the line $\mathcal{L}_V(r)$ which is nearest to r' and r respectively as p_4 and p_3. Let us call the line segment $\overline{p_3 p_4}$ as $line_2$, $\overline{p_4 p_1}$ as $line_3$ and $\overline{p_3 p_2}$ as $line_4$. Then the area
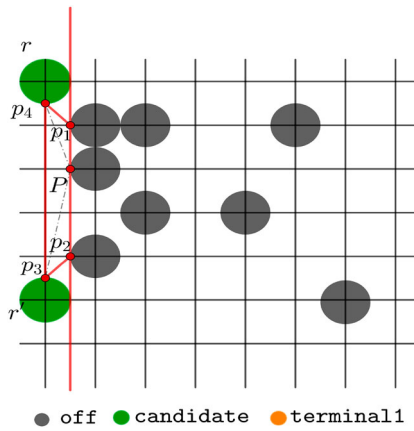
Figure 6. Area bounded by the quadrilateral $\overline{p_1p_2p_3p_4}$ is convex.

bounded by these four lines is convex and there is no robot inside this area. 0For any robot r_1 on $\mathcal{R}_l(r)$ with colour off, let us denote the point where $line_1$ touches r_1 as P. Then both the line segment $\overline{Pp_3}$ and $\overline{Pp_4}$ do not contain any other robot (Figure 6). So, both r and r' can see r_1 and similarly r_1 sees both r and r'. Thus r and r' can see all robots with colour off on $\mathcal{R}_l(r)$ and all robots with colour off can see both of r and r'.

So, we can conclude the lemma. ∎

Lemma 3.4: *If r_1 and r_2 be two robots with colour* call *or* reached *or* candidate *on the same vertical line, then any terminal robot r with colour* moving1 *on $\mathcal{R}_l(r_1)(= \mathcal{R}_l(r_2))$ always can see at least one of r_1 and r_2.*

Proof: Without loss of generality, let us assume that r_1 is above r_2 and r is above K (i.e. the horizontal line which is equidistant from both $\mathcal{L}_H(r_1)$ and $\mathcal{L}_H(r_2)$). Also, let there is no other vertical line between $\mathcal{L}_l(r)$ and $\mathcal{L}_V(r)$, otherwise with the same argument as Lemma 3.3 we can say that r can see both r_1 and r_2. Now there are three cases.

Case-I: r is below $\mathcal{L}_H(r_1)$. In this case, by similar argument in Case-II of Lemma 3.3, we can conclude that r can see both r_1 and r_2.

Case-II: r is on $\mathcal{L}_H(r_1)$. Let us draw the tangents of r, $line_1$ parallel to $\mathcal{L}_V(r)$ and nearest to $\mathcal{L}_l(r)$ and tangent of r_1, $line_2$ parallel to $\mathcal{L}_V(r_1)$ and nearest to $\mathcal{R}_l(r_1)$. Now let $line_1$ touches r at point p_1 and $line_2$ touches r_1 at point p_2 (Figure 7). Since $\overline{p_1p_2}$ does not contain any other robot, r can see r_1. Note that $line_1$ and $line_2$ can be same if the robots are of radius $\frac{1}{2}$. Now let $line_1$ touches both r and r_1 at a point p. Hence r can see r_1. **Case-III:** r is above $\mathcal{L}_H(r_1)$. In this case, we claim that if there is any other robot with colour moving1 on $\mathcal{L}_V(r)$, it must be below $\mathcal{L}_H(r_1)$. If possible let, there is another robot r' on $\mathcal{L}_V(r)$ which is above $\mathcal{L}_H(r_1)$ but below $\mathcal{L}_H(r)$ with colour moving1. Since a robot turns its colour to moving1 from off, there exists a time T when r' had colour off. So, in $\mathbb{C}(T)$, r' must be located below $\mathcal{L}_H(r_1)$. Now, r is already located on $\mathcal{L}_V(r')$ and above r'. So, if r' is terminal, it moves opposite to r and never reaches above $\mathcal{L}_H(r_1)$. And if r' is not terminal, then it never moves until r moves left. So, if r is above $\mathcal{L}_H(r_1)$ with colour moving1 and is terminal, then there is no other robot on the grid points on $\mathcal{L}_V(r)$ between $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r_1)$. Let the tangent of r which is parallel to $\mathcal{L}_V(r)$ and nearest to $\mathcal{L}_l(r)$ touches r at point p_1 and intersects $\mathcal{L}_H(r_1)$ at p_3. Also, boundary of r_1 touches the line $\mathcal{L}_H(r_1)$ at a point nearest to $\mathcal{L}_V(r)$ (say p_2) (Figure 8). Since p_1, p_2 and p_3 form a triangle and the area bounded by the triangle is a convex set containing no other robot, r can see r_1. ∎

Lemma 3.5: *A robot changes its colour to* leader1 *only from light* candidate *or* off.
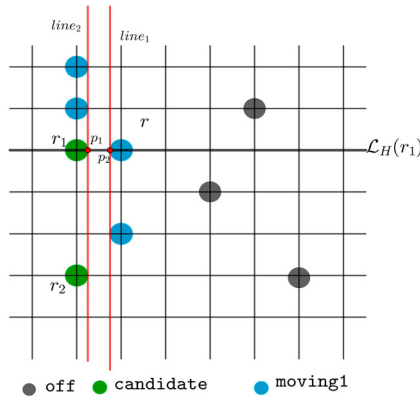
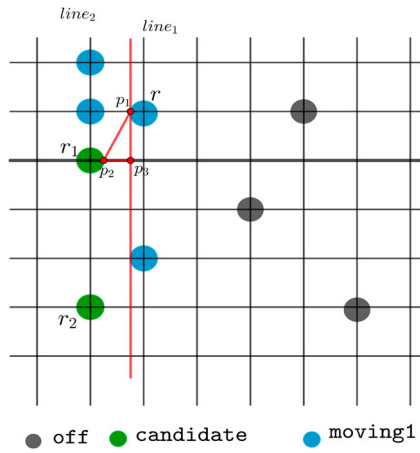**Figure 7.** r is on $\mathcal{L}_H(r_1)$.

Figure 8. r is above $\mathcal{L}_H(r_1)$.

Proof: From Algorithm 1, it follows directly that a robot can change its colour to `leader1` only if it was either with colour `candidate` or with colour `off`. ∎

Lemma 3.6: *A robot with light* `leader1` *always has empty grid point in its left.*

Proof: If at some tome $T > 0$, there is only one robot say r, on $\mathcal{L}_1$ with colour `candidate` and no robot with colour other than `off` on $\mathcal{R}_l(r)$, then r changes its colour to `leader1`. Observe that since r is on $\mathcal{L}_1$, it will have its left grid point empty.

Now, consider there are two robots r and r' with colour `candidate` on $\mathcal{L}_V(r)$ (i.e. $\mathcal{L}_V(r')$). Now by Lemma 3.5, it is evident that a robot with colour `candidate` or `off` can only change its colour to `leader1`. So, let us consider these cases.

Case-I: Let a robot r_1 with colour `off` changes its colour to `leader1`. That implies only r and r' is on $\mathcal{L}_l(r_1)$ having colour `call` and r_1 is on $K \cap \mathcal{R}_l(r)$. Note that if r and r' are adjacent on $\mathcal{L}_V(r)$, then K can not be a horizontal line of the grid $\mathcal{G}$. So, r and r' are not adjacent on $\mathcal{L}_V(r)$ (Figure 9). Now note that even if there are robots other than r and r' on $\mathcal{L}_V(r)$ or, $\mathcal{L}_l(r)$, they are not on or between the line $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r')$. So, $\mathcal{L}_H(r_1)$ lies between $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r')$ and r_1 is on $\mathcal{R}_l(r)$. So, we can say that $\mathcal{L}_H(r_1) \cap H_L^O(r_1)$ is always empty. As r_1 moves only on left until it becomes singleton on $\mathcal{L}_1$ and has $l_{next}(r_1)$ empty, r_1, the robot with light `leader1` always has its left grid point empty.
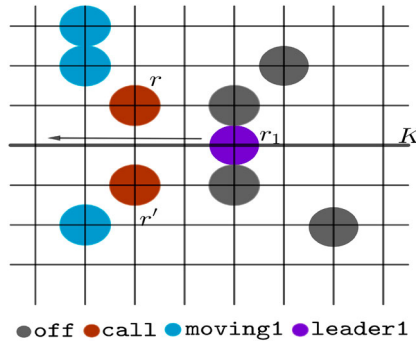
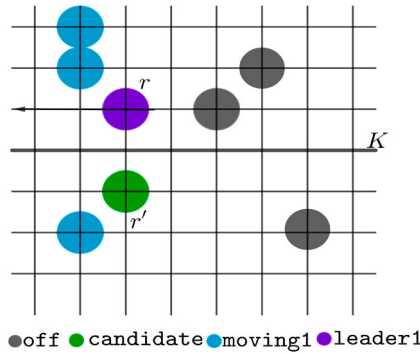**Figure 9.** r_1 has colour leader1 and has $\mathcal{L}_H(r_1) \cap H_L^O(r_1)$ empty.

Figure 10. r is on $\mathcal{L}_2$ with colour leader1 and has $\mathcal{L}_H(r) \cap H_L^O(r)$ empty.

Case-II: Without loss of generality, let r be the robot with colour candidate that changes the colour to leader1. Note that, either r is on $\mathcal{L}_1$ or it is on $\mathcal{L}_2$. If r is on $\mathcal{L}_1$ then it finds its left grid point is empty and moves to left and become singleton on $\mathcal{L}_1$. So, left grid point of r is empty. Now if r was on $\mathcal{L}_2$ (Figure 10). Then $\mathcal{L}_1 \cap \mathcal{L}_H(r)$ is empty as there are no robots between the line $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r')$ on $\mathcal{L}_I(r)$. So, r can move left and become singleton on $\mathcal{L}_1$. Hence r always has its left grid point empty. ∎

Lemma 3.7: *If a robot r with colour* call *does not see another robot with colour* leader1 *on* $\mathcal{R}_I(r)$, *then there is a time T when* $\mathcal{L}_V(r)$ *will always have a robot with colour* moving1 *and two robots with colour* call *in* $\mathbb{C}(T)$.

Proof: r is a robot with colour call on $\mathcal{L}_V(r)$. This implies $\mathcal{R}_I(r)$ is symmetric with respect to K, where K is known because there is another robot say r' on $\mathcal{L}_V(r)$ with colour call or candidate. Note that if r' has colour candidate, it changes the colour to call after a finite time. In this situation, if there is a robot say r_1 on $K \cap \mathcal{R}_I(r)$, then r_1 changes its colour to leader1 from off. And also, r sees r_1 on $\mathcal{R}_I(r)$. Since it is assumed that r is not seeing any robot with colour leader1 on $\mathcal{R}_I(r)$, it is evident that there is no robot on $K \cap \mathcal{R}_I(r)$. In this scenario, the robots on $\mathcal{R}_I(r)$ see that there are exactly two robots r and r' with colour call on left immediate vertical line. So, the robots on $\mathcal{R}_I(r)$, which are closest to K change their colours to moving1 upon activation and all the robots who can see a robot with colour moving1 on their vertical line eventually change their colours to moving1. Observe that in this way, after a finite time there will be at least one robot on $\mathcal{R}_I(r)$ which has colour moving1 and also will be terminal on $\mathcal{R}_I(r)$. Let r_2 be that robot. Now r_2 will move vertically in one fixed direction until at least one of $H_U^C(r) \cap \mathcal{L}_I(r)$ and $H_B^C(r) \cap \mathcal{L}_I(r)$ has no other robot and then it moves left to $\mathcal{L}_V(r)$
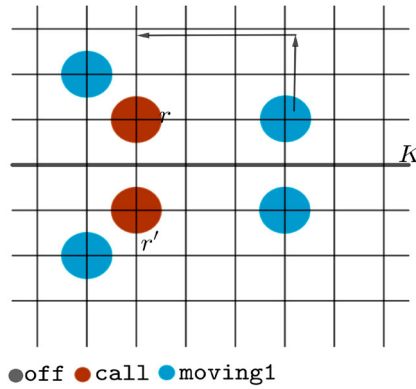
Figure 11. Terminal robot on $\mathcal{R}_l(r)$ see r with colour call and move according to the path shown by the arrow.
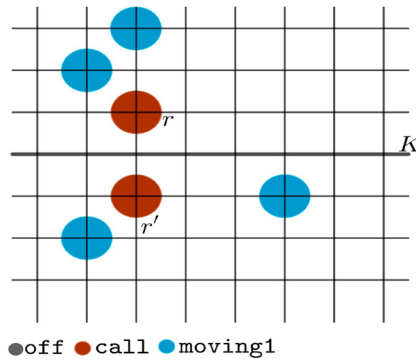

Figure 12. The robot with colour moving1 reaches above r. In this moment, $\mathcal{L}_V(r)$ has two robots r and r' with colour call and one robot with colour moving1.

(Figure 11). Also, note that r and r' do not change their colours until r_2 reaches $\mathcal{L}_V(r)$. So, after a finite time say T, there will be a robot r_2 with colour moving1 and two robots r and r' with colour call on $\mathcal{L}_V(r)$ in $\mathbb{C}(T)$ (Figure 12).

∎

Lemma 3.8: *During movement of robots with colour* moving1 *in Phase 1, no collision occurs.*

Proof: A robot r with colour moving1 can have two type of moves, horizontal to the left and vertical. r moves vertically on $\mathcal{L}_V(r)$ only when it sees at least one robot with colour call on $\mathcal{L}_l(r)$. Note that during vertical movement of r, no other robot on $\mathcal{L}_V(r)$ moves vertically in the same direction as r. This is because if another robot say r' moves on $\mathcal{L}_V(r)$, it must be terminal on $\mathcal{L}_V(r)$ and has colour moving1. But since r is already on $\mathcal{L}_V(r)$, r' moves opposite of r. So, as long as r moves vertically on $\mathcal{L}_V(r)$ no collision occurs. Note that r moves vertically in such a way such that at least one of $H_U^C(r) \cap \mathcal{L}_l(r)$ or $H_B^C(r) \cap \mathcal{L}_l(r)$ has no other robot and then it moves left towards $\mathcal{L}_l(r)$ (i.e. the same vertical line were the robot with colour call is located). Now let there is a non-terminal robot r_1 which is nearest to r and below r on $\mathcal{L}_V(r)$ with colour moving1. Now observe that r_1 only moves when r reaches the vertical line of the robot with colour call. In this scenario, r_1 moves vertically in such a way such that it has either $H_U^C(r_1) \cap \mathcal{L}_l(r_1)$ or $H_B^C(r_1) \cap \mathcal{L}_l(r_1)$ has no other robot and then moves left to the empty grid point. So, during horizontal or vertical movement of robots with colour moving1, no collision occurs. Hence the result.
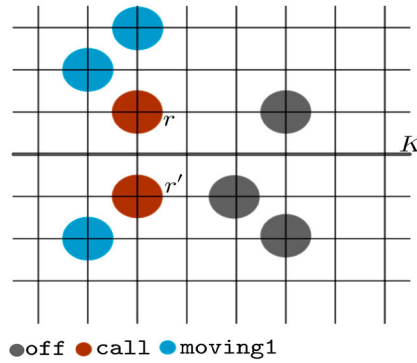
∎

Figure 13. r sees a robot with colour `moving1` on $\mathcal{L}_V(r)$ and sees all robots on $\mathcal{R}_I(r)$ with colour `off`. r' does not see any robot with colour `moving1` or `reached` on $\mathcal{L}_V(r')$.

Lemma 3.9: *If at time T, two robots r and r' have colour* `call` *on the same vertical line and there is no robot on* $K \cap \mathcal{R}_I(r)$, *then there exist* $T' > T$ *such that both r and r' are with colour* `reached` *at* $\mathbb{C}(T')$.

Proof: Let r and r' be two robots with colour `call` at time T on same vertical line $\mathcal{L}_V(r)$ (i.e. $\mathcal{L}_V(r')$). Then $\mathcal{R}_I(r)$ must be symmetric with respect to K. Also, there is no robot on $K \cap \mathcal{R}_I(r)$. So, no robot with colour `off` on $\mathcal{R}_I(r)$ changes its colour to `leader1`. Now in this scenario, the robots which are closest to K on $\mathcal{R}_I(r)$ change their colours to `moving1`. Note that a robot with colour `off` also can change its colour to `moving1` if it sees another robot with colour `moving1` on the same vertical line. Also, no robot with colour `moving1` moves unless it is terminal on the same vertical line. Hence we can say that at least all robots of above or below K on $\mathcal{R}_I(r)$ change their colours to `moving1`. Now by Algorithm 1, the terminal robots with colour `moving1` move to $\mathcal{L}_V(r)$. Then next robot becomes terminal and do the same. So, after a finite time say $T_1 > T$, all robots with colour `moving1` on $\mathcal{R}_I(r)$ move to $\mathcal{L}_V(r)$. In this moment, all robots of $\mathcal{R}_I(r)$ have colour `off`. Note that in this scenario, at least one of r or r' must see a robot with colour `moving1` on the same vertical line upon activation. Without loss of generality, let r sees a robot with light `moving1` on $\mathcal{L}_V(r)$ and all robots on $\mathcal{R}_I(r)$ have colour `off` (Figure 13). Then r changes its colour to `reached` at time say $T_2 \geq T_1 \geq T$ (Figure 14). Now when r' activates, it sees r with colour `reached` on $\mathcal{L}_V(r')$ and changes its colour to `reached` at a time $T_3 \geq T_2$ (here $T' = T_3$) (Figure 15). Now it may be possible due to asynchronous environment that after r changes its colour to `reached` at time T_2, a robot say r_1, on $\mathcal{R}_I(r)$ changes its colour to `moving1`. Then r' will not change its colour to `reached` now, even after seeing r with colour `reached` as all robots on $\mathcal{R}_I(r')$ now do not have colour `off`. Now when r_1 wakes again at a time say $T_4(\geq T_2)$, it sees r with colour `reached` on $\mathcal{L}_I(r)$ and changes its colour to `off`. Now when r' wakes again at some time $T' \geq T_4 \geq T_2 \geq T_1 > T$, it changes its colour to `reached`. Note that r does not change its colour from `reached` to `candidate` before r' wakes and changes its colour to `reached` as it will not see any other robot with colour `reached` or `candidate` on $\mathcal{L}_V(r)$ before r' wakes. So, we can conclude that $\exists T' > T$ when both r and r' have colour `reached`. ∎

Lemma 3.10: *If a robot r changed its colour to* `reached` *at some time T > 0, then* $\exists T' \geq T$ *such that all robot in* $\mathcal{R}_I(r)$ *in* $\mathbb{C}(T')$ *have colour* `off`.

Proof: During the look phase, r must have seen robots on $\mathcal{R}_I(r)$ have colour `off`. Now if no robot on $\mathcal{R}_I(r)$ change its colour to `moving1` in between the completion of look phase of r and time T, then $T' = T$. Now if a robot, say r_1 changes its colour to `moving1` in between completion of look phase of r and time T, then there exists $T' > T$ when r_1 sees a robot with colour `reached` on $\mathcal{L}_I(r_1)$ and so changes its colour to `off`. Note that before r_1 changes its colour to `off`, r does not change its colour
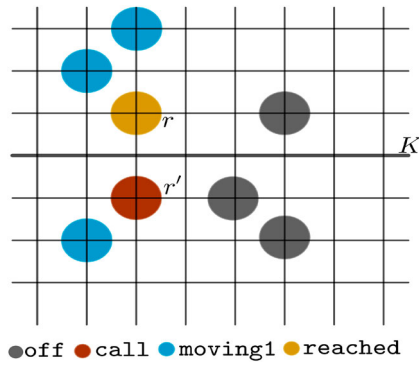
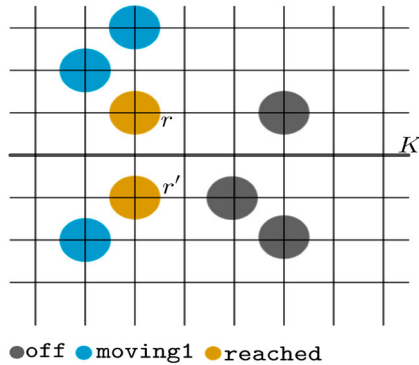**Figure 14.** r changes its colour to `reached`.

Figure 15. Now r' sees r with colour `reached` on $\mathcal{L}_V(r')$ and all robots on $\mathcal{R}_I(r')$ with colour `off`. So, r' changes its colour to `reached`.

as it sees r_1 with colour `moving1` on $\mathcal{R}_I(r)$. So, we can conclude $\exists T' \geq T$ such that all robots on $\mathcal{R}_I(r)$ have colour `off` in $\mathbb{C}(T')$. ∎

Lemma 3.11: *If at time T, a robot changes its colour to* `leader1` *from* `off`*, then* $\mathbb{C}(T')$ *has no robot with colour* `candidate` *or* `terminal1`*, where* $T' \geq T$*.*

Proof: If r changes its colour to `leader1` from `off` at some time T, then it must have seen exactly two robots say r_1 and r_2 with colour `call` on $\mathcal{L}_I(r)$ at a time T_1 where $T_1 < T$ (Figure 16). Note that a robot can only have colour `call` at some time T_2 if it had colour `candidate` at some time $T_3 < T_2$. Also, a robot can change its colour to `candidate` from `terminal1` only if it sees there is no other robot on its left open half. Also, a robot with colour `off` changes to colour `terminal1` only if its left open half empty, there is no robot with colour `leader1` on $\mathcal{R}_I(r_1)$ or on $\mathcal{L}_V(r_1)$ and it is terminal on $\mathcal{L}_V(r)$. Since during the whole execution of *Phase 1*, no other robot having colour `off` except r_1 and r_2 can see its left open half empty and find themselves to be terminal, no other robot except r_1 and r_2 can change their colours to `terminal1`. Now upon activation again at any time $T_4 > T$, both r_1 and r_2 sees r on $\mathcal{R}_I(r_1)$ with colour `leader1` and change their colours to `off` (Figure 17). Observe that after time T_4, r_1 and r_2 can never change their colour to `terminal1` and hence to `candidate` as they will see r with colour `leader1` on $\mathcal{R}_I(r_1)$ or on $\mathcal{L}_V(r_1)$ or r_1 and r_2 would have its left open half non-empty. So, we can conclude the lemma. ∎

Lemma 3.12: *If at a time T, a robot r changed its colour to* `leader1`*, then there will be no robot with colour* `reached` *in* $\mathbb{C}(T')$*, where* $T' \geq T$*.*
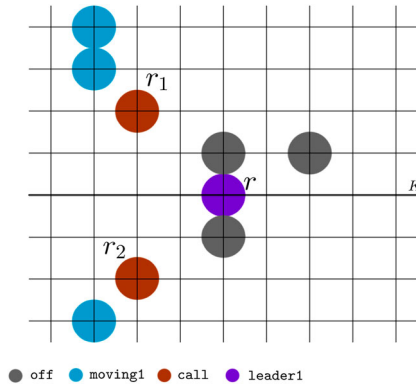
Figure 16. r_1 and r_2 with colour call both see r with colour leader1 on $\mathcal{R}_l(r_1) \cap K$.
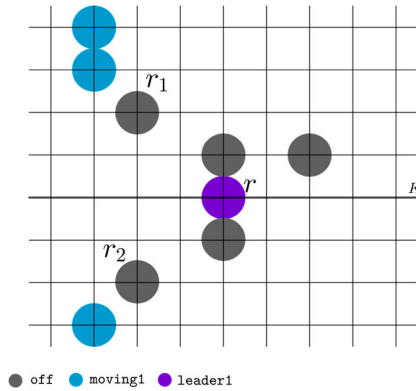

Figure 17. Both r_1 and r_2 change their colours to off after seeing r.

Proof: Note that a robot can only change its colour to reached at a time T if $\exists T_1 < T$ such that the robot had colour candidate in $\mathbb{C}(T_1)$. Now if r changed its colour to leader1 from candidate, then even if there is another robot say r' with colour candidate on $\mathcal{L}_V(r)$ (Figure 18), r' will change its colour to off upon first activation at a time $T_2 > T$. So, the configuration now has no robot with colour candidate or reached (as both r and r' with colour candidate who could have changed their colour to reached changed it to leader1 and off) (Figure 19). Also, note that during the period between T and T_2, the configuration does not have colour reached as in this time r has colour leader1 and r' has colour candidate. Also, no other robot with colour off will ever change its colour to candidate after time T_2 as a robot say r_1 with colour off or terminal1 either sees r with colour leader1 on $\mathcal{L}_V(r_1)$ or on $\mathcal{R}_l(r_1)$ or it has its left open half non-empty. And since a robot can only change its colour to reached when it had colour candidate before, there will be no robot with colour reached in $\mathbb{C}(T')$, where $(T' \geq T)$.

Now, if r has changed its colour to leader1 from off at time T, then r must have seen two robots say, r_1 and r_2 on $\mathcal{L}_l(r)$ with colour call at some time $T_1 < T$. Now upon activation after time T, both r_1 and r_2 see r on $\mathcal{R}_l(r_1)$ and turn their colours to off. Now for r_1 and r_2 to ever have the colour reached again must have colour candidate first. But by Lemma 3.11, after r changes its colour to leader1, the configuration can never have a robot with colour candidate. Hence, $\mathbb{C}(T')$ $(T' \geq T)$ has no robot with colour reached if r changed its colour to leader1 at time T.   ∎

Lemma 3.13: *At any time T, there can be at most one robot with colour* leader1 *and at most one robot with colour* leader *in the configuration.*

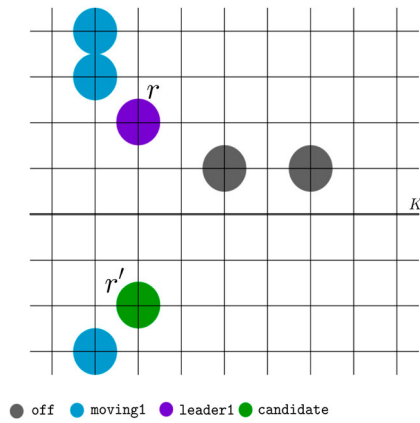**Figure 18.** r' with colour candidate sees r with colour leader1 on $\mathcal{L}_V(r')$.
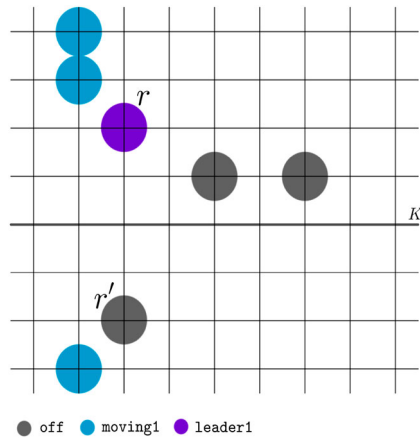

Figure 19. r' changes its colour to off. Now, no other robot changes colour to terminal1 and hence to candidate and hence to reached.

Proof: Note that by Lemma 3.5, a robot can change its colour to leader1 only from the colour off or candidate. Let us consider the following cases:

Case-I: Consider the case where a robot changes its colour to leader1 from the colour candidate. Now from Theorem 3.2, for any initial configuration $\mathbb{C}(0)$, there exist a time T such that $\mathbb{C}(T)$ has either one robot with colour leader1 who has changed its colour to leader1 from candidate or two robots with colour candidate on the same vertical line.

Case-I(a): Let r is a robot with colour leader1 in $\mathbb{C}(T)$ who has changed its colour from candidate. We claim that in $\mathbb{C}(T')$ where $T' \geq T$, there is no other robot who changes its colour to leader1. For this, we first show that no other robot with colour terminal1 ever change their colour to candidate. This is because no other robot with colour terminal1 will find its left open half empty (as the robot with colour leader1 is there) (Figure 20). So $\mathbb{C}(T')$, where $T' \geq T$, will not have any robot with colour candidate who can change further to leader1. Also observe that after at $T' \geq T$, no other robot with colour off changes its colour to leader1 as they will not see any robot with colour call on their left immediate occupied vertical line. This is also for the reason that $\mathbb{C}(T')$ where $T' \geq T$ will not have any other robot with colour candidate who can change its colour to call further. So, there will be exactly one robot with colour leader1 which eventually changes its colour to leader.
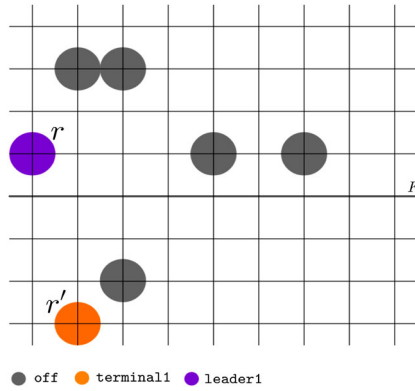
Figure 20. r' with colour `terminal1` sees r with colour `leader1` on $\mathcal{L}_l(r')$. So, r' does not change its colour as it does not have its left open half empty.
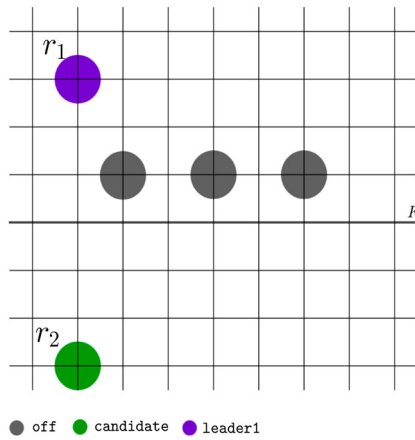

Figure 21. r_2 with colour `candidate` sees r with colour `leader1` on $\mathcal{L}_V(r_2)$. No robot with colour `call` in the configuration.

Case-I(b): Let us now assume the case where there are two robots r_1 and r_2 with colour `candidate` both on the same vertical line $\mathcal{L}_V(r_1)$(i.e. $\mathcal{L}_V(r_2)$). In this case, we will first show that both r_1 and r_2 can not change their colour to `leader1`. Then we will show if one of r_1 or r_2 changes its colours to `leader1`, then no other robot with colour `off` changes its colour to `leader1`.

In this case, r_1 and r_2 check the symmetry of the line $\mathcal{R}_l(r_1)$(i.e. $\mathcal{R}_l(r_2)$). If $\mathcal{R}_l(r)$ is asymmetric, then the robot (r_1 or, r_2) whichever is on the dominant half changes its colour to `leader1`. Without loss of generality. let at some time T_1, r_1 changes its colour to `leader1` from `candidate`. Then r_2 must have colour `candidate` in $\mathbb{C}(T_1)$ (Figure 21). Now when r_2 wakes again at a time say $T_2 > T_1$, it sees r_1 with colour `leader1` on $\mathcal{L}_V(r_2)$ and changes its colour to `off` (Figure 22). Note that between time T_1 and T_2 even if r_1 awakes again, it does not move as it sees r_2 with colour `candidate` on $\mathcal{L}_V(r_1)$. Now even if r_2 is terminal with colour `off` and has left open half empty, it would not change its colour to `terminal1` and then to `candidate` again as it sees r_1 with colour `leader1` on the same vertical line. So, between two robots with colour `candidate` only one can change its colour to `leader1`.

Now a robot say r with colour `off` can never change its colour to `leader1` as it would not see exactly two robots with colour `call` on $\mathcal{L}_l(r)$. This is because a robot can only change its colour to `call` from colour `candidate` and no other robot with colour `off` will ever change its colour to `terminal1` and then to `candidate` as in this case even if a robot with colour `off` has its left open half empty and is terminal on its vertical line, it will see r_1 with colour `leader1` on the same vertical
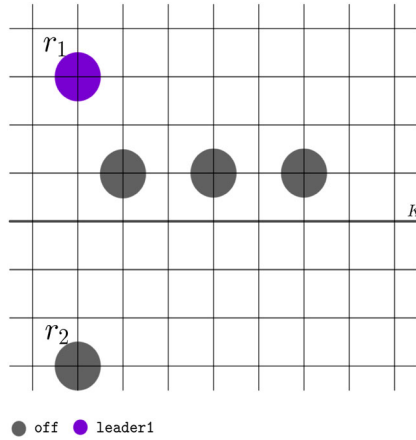
Figure 22. r_2 changes its colour to off. No robot with colour call in the configuration and no other robot with colour off changes colour to terminal1 and hence to candidate and hence to leader1 or call.

line (Figure 22). So, it would not change its colour to terminal1. So, if a robot changes its colour to leader1 from candidate, then no other robot will change its colour to leader1.

Case-II: Next we show that if a robot say r has changed its colour to leader1 from off, then no other robot with colour candidate or off ever changes its colour to leader1.

Let r changed its colour to leader1 from colour off at some time T_2. This implies r must have seen exactly two robots say r_1 and r_2 with colour call on $\mathcal{L}_l(r)$ and r is on $K \cap \mathcal{L}_V(r)$. Also $\mathbb{C}(T_2)$ has no robot with colour candidate (Figure 23). We will now show that no robot will ever change its colour to candidate again. Now when r_1 and r_2 wake again (lets say at time $T_2' > T_2$), it sees r with colour leader1 on $\mathcal{R}_l(r_1)$ and so change their colours to off (Figure 24). Observe that, in this scenario there is no robot with colour reached and terminal1 in the configuration $\mathbb{C}(T_2')$ and no robot with colour off will ever change its colour to terminal1 and then to candidate eventually. This is because even if a robot with colour off finds its left open half empty and it is terminal on its vertical line, it sees r with colour leader1 on its right immediate vertical line or on its own vertical line. So after time T_2', the configuration has no robot with colour candidate, so no robot with colour call or reached. Thus if r changed its colour to leader1, no other robot can change its colour to leader1 from candidate.

Again this scenario, all robots with colour off are either on $H_R^C(r)$ or on $\mathcal{L}_l(r)$. Note that all robots with colour off that are on $\mathcal{L}_l(r)$, never change their colours as they either see r on their right immediate vertical line or on the same vertical line or they find their left open half is non-empty. Similarly, the robots with colour off on $H_R^C(r)$ never change their colours again as they find their left open half non-empty, never see exactly two robots with colour call on their left immediate vertical line and never sees a robot with colour moving1 on their same vertical line. So, we have proved if a robot has changed its colour to leader1 from colour off, no other robot ever changes its colour to leader1 again.

So, from all the cases, it is evident that in *Phase 1* any configuration can have at most one robot with colour leader1 and since a robot changes its colour to leader from leader1 only, there can be at most one robot with colour leader in any configuration during *Phase 1*. ∎

Lemma 3.14: *If a robot r changes its colour to* leader1 *at a time T and no robot with colour* terminal1 *changes its colour to* candidate *at T' where $T' \geq T$, then robots on $\mathcal{R}_l(r)$ in $\mathbb{C}(T)$ never move in Phase 1.*
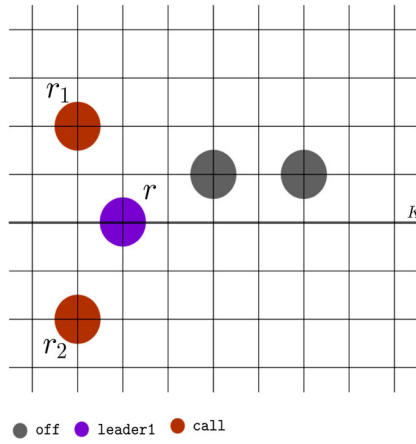
Figure 23. r_1 and r_2 with colour `call` see r with colour `leader1` on $\mathcal{R}_I(r_1)$. No robot with colour `candidate` in the configuration.
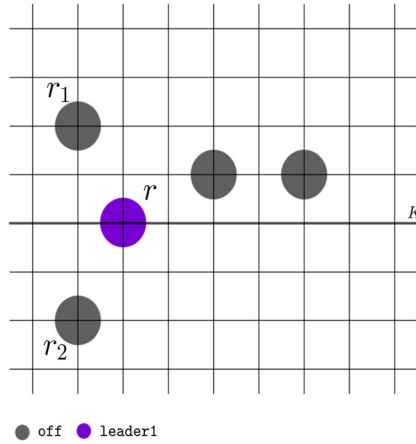

Figure 24. r_1 and r_2 change their colour to `off`. No robot with colour `reached` or `terminal1` or `candidate` or `call` in the configuration and no other robot with colour `off` changes colour to `terminal1` and hence to `candidate` and hence to `leader1` or `call` or directly to `leader1`.

Proof: Let r changes its colour to `leader1` from colour `off` at a time T. Observe that in this case, r must have seen two robots say r_1 and r_2 with colour `call` at some time $T_1 < T$ and it is on $K \cap \mathcal{R}_I(r_1)$. Note that in this case, all robots on $\mathcal{R}_I(r)$ have colour `off`, so they do not move. Now upon activation after time T, both r_1 and r_2 change their colours to `off` and never change their colours again as they see r with colour `leader1` on $\mathcal{L}_V(r_1)$ or on $\mathcal{R}_I(r_1)$ or other robots on their left open half throughout completion of *Phase 1*. So, robots on $\mathcal{R}_I(r)$(at T) never see any robot with colour `call` and also, they do not see their left open half empty after time T. Thus robots on $\mathcal{R}_I(r)$ at time T never change their colours and never move until completion of *Phase 1*.

Now if r changes its colour from `candidate` to `leader1` at time T and no robot with colour `terminal1` changes its colour to `candidate` at some time T' where $T' \geq T$, then all robots which are on $\mathcal{R}_I(r)$ in $\mathbb{C}(T)$ can have colour either `moving1` or `off` or, `terminal1`. Note that the robots with colour `moving1` will not move as it does not see any robot with colour `call` on its left immediate vertical line or, robot with colour `reached` on its same vertical line and on its left immediate vertical line. Similarly robots with colour `off` on $\mathcal{R}_I(r)$ does not change its colour if it wakes after T as it finds

out that it has its left open half non-empty and there is no robot with colour `call` on its left immediate vertical line. Note that a robot with colour `terminal1` may change its colour to `off` but after that this robot with colour `off` will not move by similar argument above. So, no robot on $\mathcal{R}_I(r)$ ever moves after time T until *Phase 1* is complete. ∎

Lemma 3.15: *If a robot r changes its colour to* `leader1` *from* `candidate` *at some time T and another robot r′ changes its colour to* `candidate` *at a time T′ where T′ ≥ T, then no collision occurs even if both r and r′ move.*

Proof: Let r changes its colour to `leader1` from `candidate` at a time T and r' changes its colour to `candidate` from colour `terminal1` at a time $T' \geq T$. Note that at time T, r must be singleton on $\mathcal{L}_V(r)$. This implies there is a time $T_1 < T$ when r had colour `off` and was terminal on $\mathcal{L}_1$ in $\mathbb{C}(T_1)$. Now when r wakes at a time say T_2, where $T_1 \leq T_2 < T$, it changes its colour to `terminal1` and there exist a time $T_3 > T_2 \geq T_1$ and $T_3 < T$ such that r changes its colour to `candidate` and moves left and become singleton on $\mathcal{L}_V(r)$. We claim that r' changes its colour to `candidate` only if r' was on $\mathcal{L}_V(r)$ in $\mathbb{C}(T_2)$ and it was also terminal on $\mathcal{L}_V(r)$ (i.e. $\mathcal{L}_V(r')$) as otherwise r' can not see its left open half empty. Now, let r' wakes before time T_2 and decide to changes its colour to `candidate` but it changes its colour at a time $T' \geq T$ and has a pending move. Then observe that now r and r' are on two different vertical lines and r' has a pending move to the left. So, if there are other vertical lines in between $\mathcal{L}_V(r)$ and $\mathcal{L}_V(r')$, then even if both of them move, no collision occurs as r can only move either vertically or on left and they are on different horizontal line. So, let us consider r' is on $l_{\text{next}}(r)$. Then r can not move vertically as $l_{\text{next}}(r)$ is non-empty. Hence, both r and r' move left and no collision occurs as r and r' are on different horizontal lines. ∎

Now, from the above lemmas and the discussions, we can conclude the following theorem.

Theorem 3.16: *For any initial configuration* $\mathbb{C}(0)$, *there exists a* $T > 0$ *such that* $\mathbb{C}(T)$ *has exactly one robot with colour* `leader` *and it's left closed half and one of upper and bottom closed half have no other robots.*

3.2. Phase 2

After completion of *Phase 1*, the configuration has exactly one robot r_0 with colour `leader` such that r_0 is singleton on $H_L^C(r_0)$ and also singleton on $\mathcal{L}_H(r_0)$ and there is no other robot on either below or above $\mathcal{L}_H(r_0)$. Note that in this configuration, all the robots who can see r_0 can agree on a global coordinate. Let r_1 be a robot which can see r_0. Then it assumes the position of r_0 as the coordinate $(0, -1)$. Now since all robots agree on the direction and orientation of the *x*-axis (i.e. the horizontal lines), r_1 can think of the horizontal line let's say $\mathcal{H}$ which is just above r_0 as the *x*-axis where right half of the line of $\mathcal{L}_V(r_0)$ correspond to the positive direction of *x*-axis. Now r_1 agrees on the the vertical line $\mathcal{L}_V(r_0)$ as *y*-axis. Note that r_1 can also know the orientation of *y*-axis by assuming its own *y*−*coordinate* to be greater or equals to the *y*−*coordinate* of r_0 (i.e. if $H_U^O(r_0)$ has robots, then the direction of $\mathcal{L}_V(r_0)$ from r_0 towards $H_U^C(r_0) \cap \mathcal{L}_V(r_0)$ is the direction of positive *y*−*axis* and similar for the case if $H_B^C(r_0)$ has robots) (Figure 25). In this phase, robots first form a line and then from that line move to their corresponding target positions which are embedded in the grid assuming the global coordinate which has been agreed upon by robots after seeing r_0. Thus the pattern formation is done. We provided a detailed description of the algorithm for *Phase 2*.

3.2.1. Line formation

In the beginning of *Phase 2*, if a robot r sees r_0 with colour `leader1`, it agrees on a global coordinate as mentioned above and find that r_0 is on $H_B^O(r)$. Now, if r sees there is no robot in $H_B^O(r) \cap H_U^O(r_0)$ (i.e. there is no horizontal line containing any robot between $\mathcal{L}_H(r_0)$ and $\mathcal{L}_H(r)$) and r is leftmost on $\mathcal{L}_H(r)$ and also if it finds there are i other robots on $(1, -1), (2, -1), \ldots, (i, -1)$ and no robot with colour `done`,
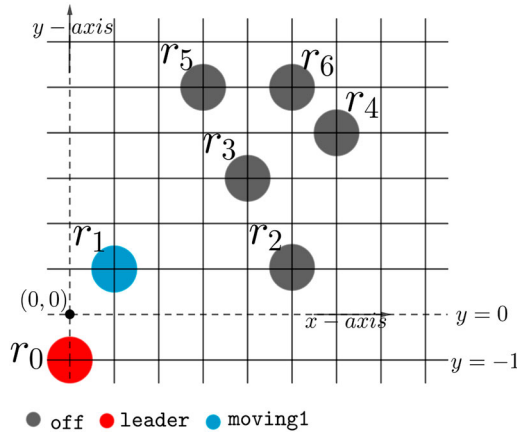
Figure 25. r_0 with colour `leader` is at $(0, -1)$. Any robot that sees r_0 can agree on a global coordinate system as shown in this diagram.

Algorithm 2: APFFATGRID: Phase 2

---

1 $r \leftarrow$ myself
2 $r_0 \leftarrow$ the robot with light `leader`
3 **if** $r.light = moving1$ or $candidate$ or $terminal1$ **then**
4   **if** $(r_0 \in H_B^O(r))$ and $(r$ is leftmost on $\mathcal{L}_H(r))$ and (there is no robot in $H_B^O(r) \cap H_U^O(r_0))$ **then**
5   **if** there are i robots on $\mathcal{L}_H(r_0)$ other than r_0 at $(1, -1), \ldots, (i, -1)$ **then**
6 $r.light \leftarrow$ off
7   move to an empty grid point towards $(i + 1, -1)$ by GOTOLINE

8 **else if** $r.light = off$ **then**
9   **if** $(r_0 \in H_B^O(r))$ and $(r$ is leftmost on $\mathcal{L}_H(r))$ and (there is no robot in $H_B^O(r) \cap H_U^O(r_0))$ **then**
10   **if** there are no robots on $\mathcal{L}_H(r_0)$ other than r_0 **then**
11   **if** there is a robot with light $done$ **then**
12   **if** r is at t_{n-2} **then**
13 $r.light \leftarrow$ done
14   **else**
15   move to an empty grid point towards t_{n-2} by GOTOTARGET
16   **else**
17   move to $(1, -1)$ by GOTOLINE
18   **else if** there are i robots on $\mathcal{L}_H(r_0)$ other than r_0 at $(1, -1), \ldots, (i, -1)$ **then**
19   move to a empty grid point towards $(i + 1, -1)$ by GOTOLINE
20   **else if** there are i robots on $\mathcal{L}_H(r_0)$ other than r_0 at $(n - i, -1), \ldots, (n - 1, -1)$ **then**
21   **if** r is at t_{n-i-2} **then**
22 $r.light \leftarrow$ done
23   **else**
24   move to an empty grid point towards t_{n-i-2} by GOTOTARGET
25   **else if** $r_0 \in \mathcal{L}_H(r)$ and $H_U^O(r)$ has no robots with light off **then**
26   **if** r is at $(i, -1)$ **then**
27   move to $(i, 0)$

28 **else if** $r.light = leader$ **then**
29   **if** all robots, which are visible to r, have light $done$ **then**
30   **if** r is at t_{n-1} **then**
31 $r.light \leftarrow$ done
32   **else**
33   move to an empty grid point towards t_{n-1} by LEADERMOVE
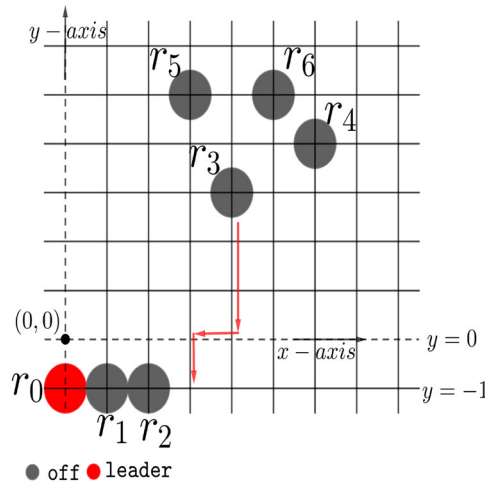

Figure 26. Movement of r_3 to $\mathcal{L}_H(r_0)$ by executing the method GOTOLINE().

then it changes its colour to off (if r has colour off, it would not change the colour) and moves to $(i+1, -1)$ by a method GOTOLINE(). The method GOTOLINE() is described as follows. In this method, if r is above the horizontal line where $y = 0$, then it moves vertically downwards until it reaches the line where $y = 0$. Note that no collision occurs during this vertical movement as there are no robots between $\mathcal{L}_H(r)$ and $\mathcal{L}_H(r_0)$ and no other robot will move until it reaches at $(i+1, -1)$. This is because other robots even if gets activated before r reaches $(i+1, -1)$, sees r between its horizontal line and $\mathcal{L}_H(r_0)$. Now, when r is at the horizontal line where $y = 0$, it moves horizontally to the position $(i+1, 0)$. Note that the horizontal line $y = 0$ may not be initially empty at the beginning of *Phase 2*. Also, initially all robots on line $y = 0$ have $x-coordinate > 0$. Now let at some time T, the robot r which is leftmost on the line $y = 0$ sees robots on $(1, -1), (2, -1), \ldots (i, -1)$. Now if there is no other robot except r on line $y = 0$ and r is not on $(i+1, 0)$, then r can simply move horizontally without collision to reach $(i+1, 0)$. Note that during this movement, no other robot from above moves as they see r in $H_B^O(r) \cap H_U^O(r_0)$. Now if there are other robots on the line $y = 0$ other than r, then this implies r has $x-coordinate > i$. This is because the i robots say $r_1, r_2, \ldots, r_i$ must have reached their current position at $(1, -1), (2, -1), \ldots (i, -1)$ from the line $y = 0$ and all robots on line $y = 0$ have $x - coordinate > 0$. Now since r has $x-coordinate > i$, r will move to its left until it reaches $(i+1, 0)$. Note that during this movement, no collision occurs as r is leftmost robot on $\mathcal{L}_H(r)$ and no other robot on $\mathcal{L}_H(r)$ moves as they are not leftmost on $\mathcal{L}_H(r)$. Next after r reaches the position $(i+1, 0)$, it moves vertically downward to $(i+1, -1)$ (Figure 26). So after a finite time, all robots will reach on the line $y = -1$ in such a way that there is no empty grid point between two robots on the line $y = -1$ (Figure 27). From the above discussion, we have the following Lemmas 3.17 and 3.18.

Lemma 3.17: *During movement of a robot r, that is executing the method* GOTOLINE() *in Phase 2, r does not collide with any other robot in the configuration.*

Lemma 3.18: *There exists a $T > 0$ such that $\mathbb{C}(T)$ has one robot with light* leader *and all other robots with light* off *in same horizontal line.*

Lemma 3.19: *The leftmost robot r of a horizontal line can always see all the robots on the horizontal line $y = -1$ if there is no robot in $H_B^O(r) \cap H_U^O(r_0)$, where r_0 is the robot with colour* leader *on the line $y = -1$.*

Proof: Let r be the leftmost robot on a horizontal line $y = s$ where $s > 0$. Then it is obvious that r will see all robots on $y = -1$. Now if r is on $y = 0$ and is singleton, then also it is obvious that r can
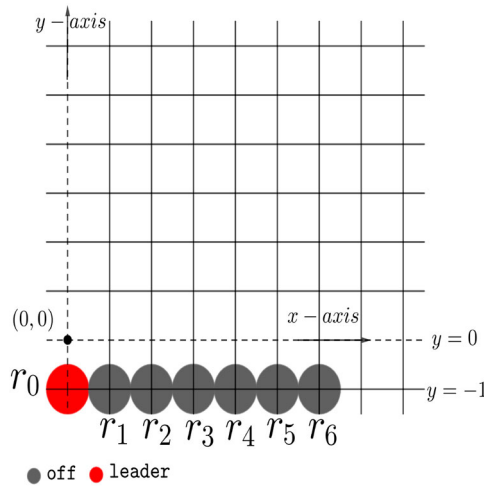
Figure 27. All robots formed a line on $\mathcal{L}_V(r_0)$. There is no empty grid point between any two robots on the line.
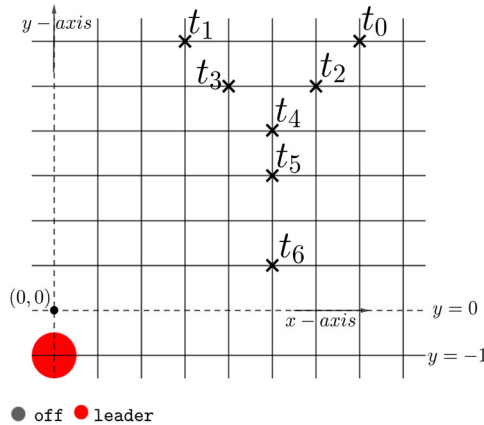

Figure 28. The target pattern embedded in the coordinate system.

see all robots on $y = -1$. Now if r is not singleton on $y = 0$. Then from the discussions above, it is clear that $x-coordinate$ of $r > x-coordinate$ of rightmost robot on $y = -1$. So, r can see all robots on $y = -1$. ∎

3.2.2. Target pattern formation

After the line is formed, all the robots except r_0 with colour `leader` on $(0, -1)$ have colour `off` and they are all placed on the horizontal line $y = -1$ in such a way that r_0 is the leftmost robot on $y = -1$ and any two robots do not have any unoccupied grid points between them. Now, the target pattern is embedded on the grid (based on the global coordinate system described above) such that $x-coordinate$, $y-coordinate$ of any target position say, t_i is greater than 0. Also, the target position of robot with colour `leader`, t_{n-1} is on line $y = 1$. And for any two other robots, let there be two target positions t_i and t_j. If both t_i and t_j are on the same horizontal line and $i < j$, then t_i is at the right of t_j. Also, if t_i and t_j are on two different horizontal lines and $i < j$, then t_i is on the above horizontal line (Figure 28).

Note that a robot say r who can see the robot r_0 with colour `leader` can agree on the global coordinate system as described earlier. So, r can agree on its target position which is embedded target
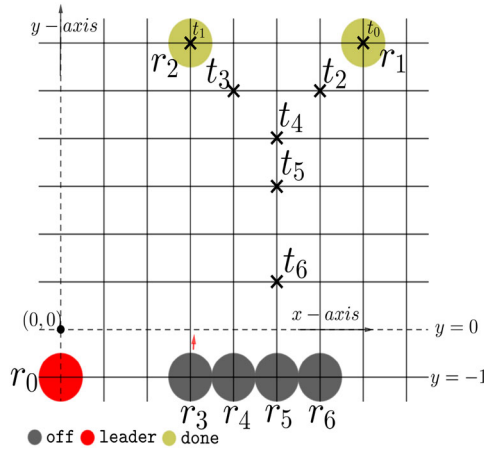
Figure 29. r_1 and r_2 already moved to their target position and changed their colour to done. r_3 sees r_0 with colour leader and moves to line $y = 0$ by moving vertically.

positions on the grid. Now a robot r, who can see r_0 and has coordinate $(s, -1)$ on $\mathcal{L}_H(r)$ and sees $H_U^O(r)$ has no robot with colour off, moves to position $(s, 0)$ (Figure 29). Now, r can see all robots on $\mathcal{L}_H(r_0)$ as r is singleton on $y = 0$. Now, if r finds out that there are i robots on $\mathcal{L}_H(r_0)$ other than r_0 at the positions $(n - i, -1), (n - i - 1, -1), \ldots (n - 1, -1)$, then r moves to t_{n-i-2} by a method GOTO-TARGET() (Figure 30) and changes its colour to done. The method GOTOTARGET() is described as follows. When executing the method GOTOTARGET(), a robot r first moves vertically to the horizontal line that is just below the horizontal line of its target location say t_r. Note that during this movement, no other robot on $\mathcal{L}_H(r_0)$ moves even if they see r_0 as they see r with colour off on their upper open half. Now, let coordinate of t_r be (x_{t_r}, y_{t_r}). Note that after the vertical movement, r is now singleton on the line $y = y_{t_r} - 1$. Now, if r is not already on the position $(x_{t_r}, y_{t_r} - 1)$, it moves horizontally to the position $(x_{t_r}, y_{t_r} - 1)$. Note that since r is singleton on $y = y_{t_r} - 1$ and no robot from $\mathcal{L}_H(r_0)$ has started its vertical movement (this is because they still see r with colour off on their upper open half), no collision occurs during this movement by r. Now when r reaches the position $(x_{t_r}, y_{t_r} - 1)$, it moves above once and reaches the designated target position t_r of r. Observe that the last robot say r_{n-1} on $(n - 1, -1)$ (i.e. on $\mathcal{L}_H(r_0)$) sees no other robot except r_0 on $\mathcal{L}_H(r_0)$ after it starts moving vertically above. Now, the problem is if it can distinguish whether r_{n-1} is meant to execute GOTOLINE() or GOTOTARGET() when it is above the line $\mathcal{L}_H(r_0)$. Note that r_{n-1} will see at least one robot having colour done above it or on the same line while it is meant to execute GOTOTARGET() as $n > 2 \implies n - 1 > 1$ which implies $\mathcal{L}_H(r_0)$ had at least one other robot r_{n-2} between r_0 and r_{n-1} which already executed GOTOTARGET() and changed its colour to done before r_{n-1} started executing GOTOTARGET(). So in this case, r_{n-1} sees at least one robot with colour done and moves to its designated target location $t_{r_{n-1}} = t_{n-2}$. So, we can conclude that after a finite time, all robots except the robot r_0 with colour leader move to their designated target locations embedded on the grid as described earlier.

From the above discussion, we can conclude the following lemma.

Lemma 3.20: *During the execution of the method* GOTOTARGET(), *a robot r never collides with another robot in the configuration.*

Now the robot r_0 with colour leader sees that all the visible robots have colour done. So, it now moves to its designated target location t_{n-1} by a method LEADERMOVE(). The method LEADERMOVE() is described as follows. In this method, r_0 first moves to $(0, 0)$. Now, let the lowest horizontal line be $\mathcal{H}_{last}$ having a robot with colour done. Now, note that r_0 can always see the leftmost robot r_{n-1} on the horizontal line $\mathcal{H}_{last}$. So, r_0 can always know its own position on the global coordinate as it knows
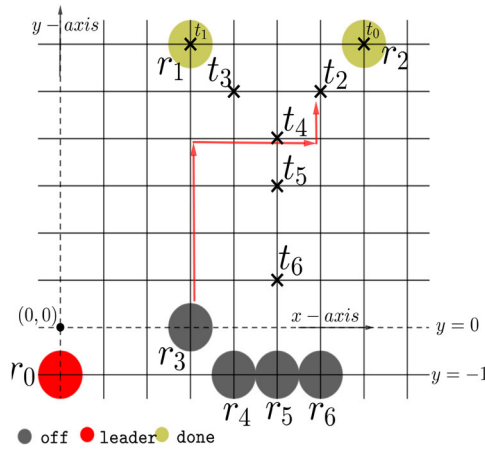
Figure 30. From $(3, 0)$, r_3 can see 3 robots on $(4, -1), (5, -1), (6, -1)$ and moves to t_2 by executing the method GOTOTARGET().
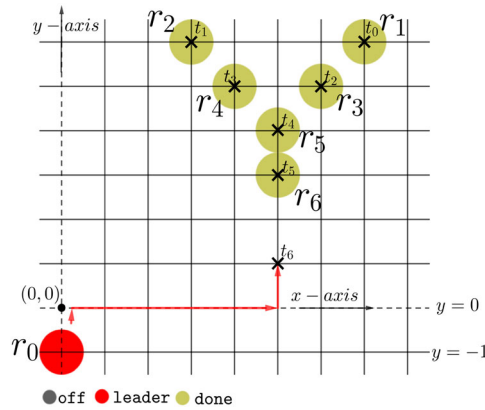

Figure 31. When all other robots except r_0 with colour `leader` reach their corresponding target positions, r_0 moves to t_6 executing the method LEADERMOVE().

the target position of r_{n-1} from the input even if it is not at $(0, -1)$. Now the target was embedded in such a way that the target position t_{n-1} of r_0 is on line $y = 1$. Let $(x_{t_0}, 1)$ be the target position of r_0. Now from $(0, 0)$, r_0 moves horizontally to the location $(x_{t_0}, 0)$ and then moves vertically once to $t_{n-1} = (x_{t_0}, 1)$ and changes the colour to `done`. Note that below $y = 1$, there is no other robot while r_0 starts moving (Figure 31). So, we can conclude the following lemma.

Lemma 3.21: *While executing the method* LEADERMOVE(), *a robot r never collides with other robots in the configuration.*

So, from Lemmas 3.17, 3.20 and 3.21, we can directly conclude the following result.

Lemma 3.22: *During movement of robots in Phase 2, no collision occurs.*

Now from the above lemmas and the discussions, we can now finally conclude the following theorem.

Theorem 3.23: *There exists a T > 0 such that* $\mathbb{C}(T)$ *is a final configuration similar to the given pattern and has all robots with light* `done` *(Figure 32).*
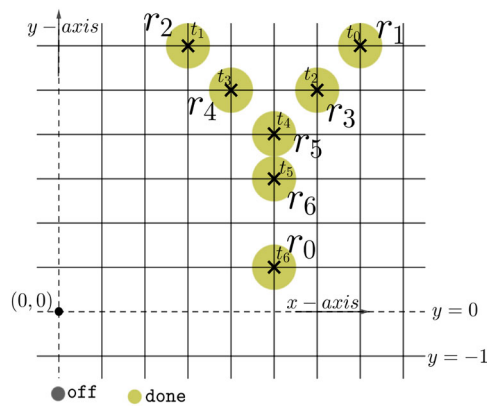
Figure 32. After r_0 reaches t_6, it changes its colour `done` and the target pattern has been formed.

4. Conclusion

The problem of arbitrary pattern formation ($\mathcal{APF}$) is a widely studied area of research in the field of swarm robotics. It has been studied under various assumptions on plane and discrete domain (e.g. infinite regular tessellation grid). With obstructed visibility model, this problem has been considered on plane and infinite grid using luminous opaque robots. But using fat robots (i.e. robots with certain dimensions), it is only done in plane. In this paper, we have taken care of this. We have shown that with a swarm of luminous opaque fat robots having one-axis agreement on an infinite grid, any arbitrary pattern can be formed from an initial configuration which is either asymmetric or has at least one robot on the line of symmetry using one light having 9 distinct colours which are less than the number of colours used to form an arbitrary pattern on plane using opaque and fat robots with one-axis agreement. For future courses of research, it would be interesting to see if the same problem can be solved using less number of colours under the same assumptions.

Acknowledgments

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

References

[1] Bose K, Adhikary R, Kundu MK, et al. Arbitrary pattern formation by opaque fat robots with lights. In: Changat M, Das S, editors. Algorithms and discrete applied mathematics – proceedings of the 6th international conference, CALDAM 2020, Hyderabad, February 13–15, 2020. Springer; 2020. p. 347–359. (Lecture Notes in Computer Science; Vol. 12016). https://doi.org/10.1007/978-3-030-39219-2_28.

[2] Bose K, Kundu MK, Adhikary R, et al. Arbitrary pattern formation by asynchronous opaque robots with lights. Theor Comput Sci. 2021;849(4):138–158. https://doi.org/10.1016/j.tcs.2020.10.015.

[3] Bose K, Adhikary R, Kundu MK, et al. Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. Theor Comput Sci. 2020;815(4):213–227. https://doi.org/10.1016/j.tcs.2020.02.016.

[4] Flocchini P, Prencipe G, Santoro N, et al. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. Theor Comput Sci. 2008;407(1–3):412–447. https://doi.org/10.1016/j.tcs.2008.07.026.

[5] Adhikary R, Kundu MK, Sau B. Circle formation by asynchronous opaque robots on infinite grid. Comput Sci. 2021;22(1):81–100. https://doi.org/10.7494/csci.2021.22.1.3840.

[6] Kundu MK, Goswami P, Ghosh S, et al. Arbitrary pattern formation by asynchronous opaque robots on infinite grid. CoRR; 2022. https://doi.org/10.48550/arXiv.2205.03053.

[7] Bramas Q, Tixeuil S. Probabilistic asynchronous arbitrary pattern formation (short paper). In: Bonakdarpour B, Petit F, editors. Stabilization, safety, and security of distributed systems – proceedings of the 18th international symposium, SSS 2016, Lyon, November 7–10. 2016. p. 88–93. (Lecture Notes in Computer Science; Vol. 10083). https://doi.org/10.1007/978-3-319-49259-9_7.

[8] Das S, Flocchini P, Santoro N, et al. Forming sequences of geometric patterns with oblivious mobile robots. Distrib Comput. 2015;28(2):131–145. https://doi.org/10.1007/s00446-014-0220-9.

[9] Bhagat S, Chaudhuri SG, Mukhopadhyaya K. Gathering of opaque robots in 3d space. In: Bellavista P, Garg VK, editors. Proceedings of the 19th international conference on distributed computing and networking, ICDCN 2018, Varanasi, January 4–7, 2018. ACM; 2018. p. 2:1–2:10. https://doi.org/10.1145/3154273.3154322.

[10] Feletti C, Mereghetti C, Palano B. Uniform circle formation for swarms of opaque robots with lights. In: Izumi T, Kuznetsov P, editors. Stabilization, safety, and security of distributed systems – proceedings of the 20th international symposium, SSS 2018, Tokyo, November 4–7, 2018. Springer; 2018. p. 317–332. (Lecture Notes in Computer Science; Vol. 11201). https://doi.org/10.1007/978-3-030-03232-6_21.

[11] Goswami P, Ghosh S, Sharma A, et al. Gathering on an infinite triangular grid with limited visibility under asynchronous scheduler. CoRR; 2022. https://doi.org/10.48550/arXiv.2204.14042.

[12] Cohen R, Peleg D. Convergence of autonomous mobile robots with inaccurate sensors and movements. SIAM J Comput. 2008;38(1):276–302. https://doi.org/10.1137/060665257.

[13] Lukovszki T. Fast collisionless pattern formation by anonymous, position-aware robots. In: Aguilera MK, Querzoni L, Shapiro M, editors. Principles of distributed systems – proceedings of the 18th international conference, OPODIS 2014, Cortina d'Ampezzo, December 16–19, 2014. Springer; 2014. p. 248–262. (Lecture Notes in Computer Science; Vol. 8878). https://doi.org/10.1007/978-3-319-14472-6_17.

[14] 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); 1996. p. 1347–1363.

[15] Bramas Q, Tixeuil S. Arbitrary pattern formation with four robots. In: Izumi T, Kuznetsov P, editors. Stabilization, safety, and security of distributed systems – proceedings of the 20th international symposium, SSS 2018, Tokyo, November 4–7, 2018. Springer; 2018. p. 333–348. (Lecture Notes in Computer Science; Vol. 11201). https://doi.org/10.1007/978-3-030-03232-6_22.

[16] Cicerone S, Stefano GD, Navarra A. Asynchronous arbitrary pattern formation: the effects of a rigorous approach. Distrib Comput. 2019;32(2):91–132. https://doi.org/10.1007/s00446-018-0325-7.

[17] Cicerone S, Stefano GD, Navarra A. Embedded pattern formation by asynchronous robots without chirality. Distrib Comput. 2019;32(4):291–315. https://doi.org/10.1007/s00446-018-0333-7.

[18] Dieudonné Y, Petit F, Villain V. Leader election problem versus pattern formation problem. In: Lynch NA, Shvartsman AA, editors. Distributed computing, proceedings of the 24th international symposium, DISC 2010, Cambridge, MA, September 13–15, 2010. Springer; 2010. p. 267–281. (Lecture Notes in Computer Science; Vol. 6343). https://doi.org/10.1007/978-3-642-15763-9_26.

[19] Peleg D. Distributed coordination algorithms for mobile robot swarms: new directions and challenges. In: Pal A, Kshemkalyani AD, Kumar R, et al., editors. Distributed computing – IWDC 2005, proceedings of the 7th international workshop, Kharagpur, December 27–30, 2005. Springer; 2005. p. 1–12. (Lecture Notes in Computer Science; Vol. 3741). https://doi.org/10.1007/11603771_1.