Uniform Scattering of Robots on Alternate Nodes of a Grid

Moumita Mondal¹, Sruti Gan Chaudhuri¹, Punyasha Chatterjee¹

¹ Jadavpur University, Kolkata, India

Abstract. In this paper, we propose a distributed algorithm to uniformly scatter the robots along a grid, with robots on alternate nodes of this grid distribution. These homogeneous, autonomous mobile robots place themselves equidistant apart on the grid, which can be required for guarding or covering a geographical area by the robots. The robots operate by executing cycles of the states "look-compute-move". In the look phase, it looks to see the position of the other robots; in the compute phase, it computes a destination to move to; and then in the move phase, it moves to that computed destination. They do not interact by message passing and can recollect neither the past actions nor the looked data from the previous cycle, i.e., oblivious. The robots are semi-synchronous, anonymous and have unlimited visibility. Eventually, the robots uniformly distribute themselves on alternate nodes of a grid, leaving the adjacent nodes of the grid vacant. The algorithm presented also assures no collision or deadlock among the robots.

Keywords: Scattering in Grid, Autonomous, Mobile Robots, Oblivious

1 Introduction

A costly big complex robot may be replaced by a group of tiny autonomous robots also known as swarm robots working collaboratively. To study on the collective behaviour of a group of tiny autonomous robots is an important area of research in the field of robotics. The swarm robots are basically mobile autonomous programmable particles. The goals of this autonomous mobile robot system have been patrolling, sensing, and exploring in a harsh environment like disaster-prone area, under the deep ocean and in the space with minimal human intervention [20]. The interest on distributed or decentralized control of multiple robots with constrained capabilities is a popular field of research for a long time. Theoretical representation of mobile robots in the Euclidean space have attracted researchers. The fundamental task for executing a job in collaboration is to form geometric shapes on the plane by the robots’ positions. This pattern formation job has significantly supported in various fields, such as operations in hazardous environments, space missions, military operations, tumour excision, etc [9]. In this paper we address one of such pattern formation tasks, scattering in alternate nodes of a grid. The robots position themselves equidistant apart
forming a grid in finite time. Each robot can sense its immediate surrounding or nearby robots, operates on the sensed data to compute the location to move to, and move to the computed destination. A significant application of the algorithm on scattering in grid, during Covid-19 pandemic, maybe in covering a region with autonomous robots, having UV ray emitting capabilities to disinfect the area [1][2]. The robots freely move on a 2D plane. They are anonymous, homogeneous, oblivious and can not communicate but interact by observing others positions. Based on this model, we study the problem of scattering on alternate nodes of a grid by oblivious robots.

1.1 Framework

We consider a group of robots moving on a 2D plane. The robots are endowed with (a) Motorial capabilities i.e., they move independently in the Euclidean space and (b) Sensorial capabilities i.e., they look at the locations of the other robots. The robots have no means of communication through explicit message passing. However, the robots coordinate among themselves by observing the positions of the other robots on the plane. A robot is always able to look another robot within its visibility or sensing range (limited or unlimited). The robots are homogeneous (all executes the same algorithm) and anonymous (no unique identifiers). The autonomy of the robot system enables the robots to work without centralized control. The robots are assumed to be free of any kind computational and structural fault. The robots considered are point robots and are oblivious, i.e. they do not save any information for the future. The robots execute a cycle of the following phases:

- **look** - Robots collect the positions of other robots within its sensing range.
- **Compute** - In the compute state, the robots execute our algorithm for computing destinations. This algorithm is the same for all robots.
- **Move** - The robots move to the computed destinations in this state.

In general, the robots may or may not be synchronized.

- The **asynchronous (ASYNC)** model is a pragmatic model, where the actions of the robots are totally independent. When a robot completes its computation, many of the other robots may have moved from their positions, based on which the computation is done. The asynchronous (ASYNC) scheduler activates the robots independently, and the duration of each Compute, Move and the time between successive activities is finite and unpredictable. As a result, a robot can be seen while moving and the snapshot and its actual configuration are not the same, and so its computation may be done with the old configuration.
- The **semi-synchronous (SSYNC)** model, confirms that when a robot is moving no other robot is sensing. In this scheduling an arbitrary group of robots execute their cycles together and there is a common clock but, at each cycle, all robots may or may not be active. The semi-synchronous (SSYNC)
scheduler activates a subset of all robots synchronously and their Look-
Compute-Move cycles are performed at the same time. Here, we can assume
that activated robots at the same time obtain the same snapshot, and their
Compute and Move are executed instantaneously. In SSYNC, we can assume
that each activation defines a discrete time called round and Look-Compute-
Move is performed instantaneously in one round. A subset of activated robots
in each round is determined by an adversary and robots do not know about
this subset. \[19\]

– In **fully-synchronous (FSYNC)** model, all the robots operate their cycles
at the same time. Thus all robots get the same view and they compute on
the same data. It can be considered as a special case of SSYNC, when all
robots are activated in each round.

In this paper, we consider the semi-synchronous (SSYNC) model. We propose a
robot’s movement strategy such that, after a finite time they are placed equidis-
tant apart on alternate nodes of a grid, leaving the adjacent nodes empty on the
grid.

2 Earlier Works and Our Contribution

A large body of research work exist in the context of multiple autonomous mobile
robots, exhibiting cooperative behaviour. The primary objective of such research
work is to study the issues of group architecture, resource conflict, origin of
cooperation, learning and geometric problems [5]. Traditionally the research on
mobile robots involves artificial intelligence in which most of the results are based
on experimental study or simulations. Recently, an emerging field of autonomous
mobile robots looks at the robots as distributed mobile entities or programmable
particle and investigate several coordination problems for them. They proceed
to solve them deterministically providing proof of correctness of the algorithms.

The computational model for robots popular in the literature under this field
is called a **weak model** [8]. Here, the robots execute repeated cycles consisting
of phases, look-compute-move. The robots do not communicate through any
wired or wireless medium. The robots may execute the cycle synchronously or
semi-synchronously or asynchronously.

Scattering of mobile robots on a plane is one of the most popular problems.
In this paper, the robot disperse on alternate nodes of a given infinite grid. The
robots are initially on distinct arbitrary nodes of the grid and they can move
along the edges of the grid to reach another node. The goal is to reach a state
of static equilibrium in which they are uniformly scattered on alternate nodes
of the grid. This uniform scattering (or covering, or self-deployment) problem
occurs in practice, when robots are randomly deployed in a region but the re-
quirement is for the region to be covered uniformly with maximizing coverage.
A scattering algorithm specifies which operation must be performed by a robot
whenever it is active, to achieve the given goal. All the robots perform the same
self-deployment algorithm. Finally, the robots will reach a state of static equilib-
rium, and scattering will be completed within finite time. The subject of efficient
uniform scattering is of extensive research in several fields (e.g. [6], [7], [4], [13], [23], [9], [18], [11], [10]). Depending on the assumptions they make, the existing protocols differ greatly from each other. Some of the major differences are based on the following:

- **Nature of the environment** – The robots can move on a Euclidean plane, called continuous (e.g., [4], [23]), or in a network or a graph, usually called discrete or graph world (e.g., [2], [11], [13]).
- **The robots may** (e.g., [6], [13], [23]) or may not be synchronized (e.g., [4])
- **Memory of the robots** - The robots can have a persistent memory (e.g., [6], [9], [23]) or oblivious (e.g., [4], [10]);
- **Visibility or sensing range of the robots** may be limited (e.g., [9], [13]) or unlimited, i.e., extends to the entire region (e.g., [4]);
- **Computational power** - the robots have the computational power of Turing machines (e.g., [14], [23]) or are simple Finite State machines (e.g., [2], [13]);
- **Nature of termination of protocol** - exact or approximate uniform covering is reached within finite time (e.g., [4], [13]) or the protocol converges without ever terminating (e.g., [6], [7]);
- **Type of protocol** (generic/specific) - the protocol is generic, i.e., it operates in any space/network (e.g., [4], [14], [23]) or only in specific for classes of regions/graphs (e.g., [6], [7]).

Barriere et al., [3] discussed the uniform scattering problem for a set of autonomous mobile robots deployed in a grid network. In this paper, robots with a constant memory (non-oblivious) and limited visibility are considered. The dispersion problem of mobile robots on graphs is also discussed by Ajay D. Kshemkalyani et al., [17] where the robots are initially placed arbitrarily on the nodes of an n-node anonymous graph and they autonomously reposition themselves to reach a configuration in which each robot is on a distinct node of the graph. Also in [21] [16] [15] [12] the dispersion problem is discussed.

### 3 Our Contribution

In this paper, we prove that starting from any arbitrary initial configuration of robots on a given infinite grid, the robots can uniformly distribute themselves along a grid, with robots on the alternate nodes of this grid distribution. This problem is of significantly important due to its relationship to many other fundamental robot coordination problems, such as exploration, scattering, load balancing, relocation of self-driven electric cars (robots) to recharge stations (nodes), etc. The protocol is fully localized and decentralized, and it makes minimal assumptions like; it does not require any direct or explicit communication between robots; the robots have no past memory; the robots are anonymous, semi-synchronous and identical. We propose a distributed algorithm to disperse unlimited visibility robots in alternate nodes of a grid. We show that if the robots agree only on the direction of both axes (i.e., X-axis and Y-axis), then they can form a uniform distribution along a grid, with alternate nodes empty, without encountering any collision or deadlock.
4 Algorithm

A set of $n$ stationary points on an infinite grid on a 2D plane is given. The set of robots, $R$ is scattered on various nodes of the grid. The robots move along the edges of the grid, from one grid node to another in such a way that after a finite number of cycle execution they are placed uniformly, equidistantly apart on alternate nodes of a grid.

4.1 Underlying Model

Let $R = \{r_1, r_2, ..., r_n\}$ be a set of autonomous mobile robots. The set of robots $R$ deployed on the given grid is described as follows:

- The robots are autonomous.
- Robots are anonymous and homogeneous.
- The robots are oblivious in the sense that they can not recollect any data from the past cycle.
- Robots can not communicate explicitly. Each robot is allowed to have a camera that can take pictures over 360 degrees. The robots communicate only by means of observing other robots with the camera.
- The robots are point robots.
- The robots have unlimited visibility.
- The robots execute look-compute-move under semi-synchronous schedules. In this scheduling, a set of robots execute the cycles synchronously. However, this set is chosen randomly. This scheduling gives the assurance that a robot will not capture the locations of moving robots.
- A robot considers its position as origin (i.e its local co-ordinate system). The robots do not have any global origin. However, they agree on the direction and orientation of $X$ and $Y$ axes.
- The robots reside in nodes and do not stop in edges. During the movement to the computed destination node, they do not stop in between nodes. This motion is also know is rigid motion.

4.2 Overview of the Problem

Our objective is to uniformly distribute the autonomous mobile robots, along a grid, such that the robots are on alternate nodes of this grid distribution. Initially all the robots are on distinct nodes. Following are the steps to be executed by each robot in the compute phase:

- Determine the maximum number of rows and columns required to uniformly distribute the robots along a grid such that the robots are on alternate nodes of this grid distribution, using the $FindDimension$ routine;
- Compute the north-most bound, (using the $FindYMAX$ routine) and west-most bound (using the $FindXMIN$ routine) of the uniform robot distribution along the grid;
- The robots uniformly distribute themselves along the grid, leaving the alternate nodes empty, using the $FormGrid$ routine.
4.3 Description of Algorithm FormGrid

Let the number of the robots in R be \( n \) and a robot, \( r_i \in R \). A grid node, \( T_{x,y} \) has coordinate \((x, y)\). Let \( grid_{final} \) be the uniform distribution of robots to be formed along the grid, with robots on alternate nodes of this grid distribution and \( row_j \) be the \( j^{th} \) row of this \( grid_{final} \). So, the number of robots, in the alternate rows or columns of \( grid_{final} \), will be \( \lceil \sqrt{n} \rceil \) (say, \( rc \)), such that the last row(s) of the grid may be completely or partially empty. The maximum dimensions of the final uniform distribution, \( grid_{final} \), with robots on alternate nodes, is determined as \(((rc * 2) - 1) \) (say, \( d \)), since the alternate rows and columns will be empty (Algorithm ).

The Y-axis value of the extreme north robot (\( Y_{MAX} \)) is the northwards bound of \( grid_{final} \) and this row is also considered the first row, \( row_1 \) (i.e. \( j = 1 \)) of the uniform distribution of robots. To determine the \( Y_{MAX} \), a robot \( r_i \) compares the Y-axis values of all the robots. So, the robot with the maximum Y-axis value is the north-most robot and its Y-axis value is considered as the \( Y_{MAX} \) (Algorithm ). Similarly, the X-axis value of the extreme west robot (\( X_{MIN} \)) is the westwards bound of the \( grid_{final} \). To determine the \( X_{MIN} \) also, a robot \( r_i \) compares the X-axis values of all the robots. The robot with the minimum X-axis value is the west-most robot and its X-axis value is considered as the \( X_{MIN} \) (Algorithm ). Although the robots locally compute these values, they all agree to the same robot as north-most and thus have the same \( Y_{MAX} \), as they agree on the direction and orientation of the axes. Similarly, the robots also agree to the same robot as west-most and thus have the same \( X_{MIN} \).

The robots, \( r_i \) compute their movement from their source node, \( T_{x,y} \) to a destination node on the grid. In case of any tie, the robot trying to move westwards to its destination node will get highest priority, followed by eastwards movement and then southwards and northwards respectively. Depending on the position of \( r_i \) on the grid, there are 5 configurations, denoted as \( \Psi_1, \Psi_2, \Psi_3, \Psi_4 \) and \( \Psi_5 \).

- CASE \( \Psi_1 \): Robot, \( r_i \) is in a row, \( row_j \) such that it is among the first \( d \) rows from \( Y_{MAX} \) (i.e. \( j \leq d \)) and it is an even row, (i.e. \( j \) is even). In this case, \( r_i \) tries to move northwards to an odd row. \( r_i \) checks if \( T_{x,y+1} \) is vacant and then moves there. Otherwise, if \( T_{x,y+1} \) is not empty, \( r_i \) moves a node eastwards \( T_{x+1,y} \) and then checks if the immediate north node is vacant. If \( T_{x+1,y} \) is also not vacant, \( r_i \) waits for it to be vacant. This process continues until \( r_i \) reaches the odd row northwards to it, i.e \( row_{j-1} \). Thus the robots, on the even rows, \( row_j \) (where, \( j \) is even), of the first \( d \) rows from the \( Y_{MAX} \), move to the odd rows \( (row_{j-1}) \) above it. This continues till all \( r_i \) of the first \( d \) rows from the \( Y_{MAX} \), are placed on a odd row. (Figure 1(i))

- CASE \( \Psi_2 \): Robot, \( r_i \) is in a row, \( row_j \) such that it is beyond the first \( d \) rows from \( Y_{MAX} \) (i.e. \( j > d \)). In this case, \( r_i \) tries to move northwards, aiming to reach the \( d^{th} \) row of the \( grid_{final} \). \( r_i \) checks if immediate above node, \( T_{x,y+1} \) is vacant and then moves there. Otherwise, if \( T_{x,y+1} \) is not empty, \( r_i \) moves eastwards to a vacant \( T_{x+1,y} \) and then to the row northwards, \( T_{x+1,y+1} \). If
$T_{x+1,y}$ is also not vacant, $r_i$ waits for it to be vacant. This process continues until $r_i$ reaches the $d$th row from $Y_{MAX}$, i.e rowd. (Figure 1(ii))

- CASE $\Psi_3$: All the $n$ robots are in alternate rows (i.e. odd rows) of the first $d$ rows starting from $Y_{MAX}$. Robot, $r_i$ placed in row$_j$, is not on alternate nodes of the row$_j$. Also, $r_i$ is not on the west bound of grid$_{final}$, $X_{MIN}$.

Here, there can be two possibilities:

• Any robot on the west of $r_i$ in row$_j$, have a vacant target node to move westwards. $r_i$ tries to move westwards in its row, till the westwards bound, $X_{min}$ is meet or there is no further vacant target node to move westwards. $r_i$ checks if westwards nodes, $T_{x-1,y}$ and $T_{x-2,y}$ are vacant, if so then $r_i$ moves westwards to $T_{x-1,y}$.

• All robots on the west of $r_i$ in row$_j$, have no vacant target node to move westwards. Now, $r_i$ tries to move eastwards, to position itself in alternate nodes of the row. If checks if the eastwards nodes, $T_{x+1,y}$ and $T_{x+2,y}$ are vacant, if so then $r_i$ moves eastwards to $T_{x+1,y}$, else it waits.

This process continues until all the robots in row$_j$ are on alternate nodes.

Following this, all the robots are on alternate rows (i.e., odd rows only) and also on alternate nodes of the same row. (Figure 2, Figure 3, Figure 4(ii), Figure 5). However, after all the robots are placed on the alternate nodes, the rows may have greater or less than the $rc$ number of robots.

- CASE $\Psi_4$: All robots are placed in alternate nodes of alternate rows. Robot, $r_i$ in row$_j$ is such that starting from the west bound, $X_{MIN}$, $r_i$ is in the first $rc$ number of robots in row$_j$, and the northwards alternate row (odd row) node, $T_{x,y+2}$ is empty. Also, row$_j$ is not the north-most row (i.e., $j \neq 1$). In this case, $r_i$ moves to the northwards alternate row, $T_{x,y+2}$ (Figure 4(i))

- CASE $\Psi_5$: Robot, $r_i$ has greater than or equal to $rc$ number of robots on its west in the same row. Here, there can be two possibilities:

• $r_i$ is in the north-most row, row$_1$. Here, $r_i$ tries to move to a southwards odd row. $r_i$ checks if its immediate alternate south row node, $T_{x,y-2}$ is vacant, if so, then moves southwards to node $T_{x,y-2}$, else waits. After $r_i$ moves to node $T_{x,y-2}$ in row$_j+2$, it then tries to move westwards, in case row$_j+2$ has less than $rc$ robots, as per, $\Psi_3$. Otherwise, $r_i$ continues to move to a further south odd row and when it encounters a row$_j$ with lesser then $rc$ robots, $r_i$ moves westwards to place itself in its target node as per, $\Psi_3$.

• $r_i$ is not the north-most row (i.e., $j > 1$). Here, the robots try to move to a northward or southward odd row to form the uniform distribution, grid$_{final}$. The robots that has $rc$ number of robots on their west on the same row, are the excess robots on that row. So, they move to the other odd rows that has lesser than $rc$ number of robots. In this case, if odd rows above row$_j$ have less than $rc$ robots, and the northwards alternate row node, $T_{x,y+2}$ is vacant, then $r_i$ moves northwards to the alternate row node, $T_{x,y+2}$. Then if $r_i$ has less than $rc$ number of robots on its west
in row \( j - 2 \), it tries to place itself in that row, as per \( \Psi_3 \). If \( T_{x,y+2} \) is not vacant, \( r_i \) waits till \( T_{x,y+2} \) is empty. Only when all odd rows above \( row_j \) have greater than or equal to \( rc \) robots and the southwards alternate row node, \( T_{x,y-2} \) is vacant, then \( r_i \) moves southwards to the alternate row node, \( T_{x,y-2} \). Then if \( r_i \) has less than \( rc \) number of robots on its west in \( row_j+2 \), it tries to place itself in that row, as per \( \Psi_3 \). If \( T_{x,y-2} \) is not vacant, \( r_i \) waits till \( T_{x,y-2} \) is empty.

This process continues until no row has greater than \( rc \) robots. (Figure 6(i)).

Thus the robots uniformly distribute themselves along a grid with robots on alternate nodes of this grid distribution, by computing their target nodes on the grid (Algorithm ) (Figure 6(ii)). An example for the algorithm, with 8 arbitrarily placed robots on the nodes of an infinite grid is shown in Figures 1,2,3,4,5,6 respectively.

---

**Fig. 1.** (i) The robots on the even rows move northwards to the odd rows of grid\( grid_{final} \). (CASE \( \Psi_1 \)) (ii) \( r_8 \) is beyond the first \( d \) (i.e \((3*2)-1=5\)) rows from \( Y_{MAX} \), so it moves northwards to the 5\(^{th} \) row. (CASE \( \Psi_2 \))

**Fig. 2.** (i) and (ii) The robots move to their immediate west node on the same row when the two consecutive west nodes are vacant. Robots try to move eastwards only when there is no vacant target node to move west. (CASE \( \Psi_3 \))
Fig. 3. (i) and (ii) The robots move further to place themselves on alternate nodes of the same row. (CASE \( \Psi_3 \))

Fig. 4. (i) Since all robots on the alternate rows are placed on the alternate nodes, \( r_4 \in \text{first } rc \text{ robots of row}_3 \), moves northwards to the alternate row, \( \text{row}_1 \), as it finds a vacant node. (CASE \( \Psi_4 \))
(ii) Robot \( r_7 \) in \( \text{row}_3 \) again continue to hop west, since it again finds consecutive vacant nodes on its west. (CASE \( \Psi_3 \))

Fig. 5. (i) Robot \( r_7 \) and then \( r_5 \) in \( \text{row}_3 \) continue to hop west, as they find two consecutive vacant nodes on their west. (CASE \( \Psi_3 \))
(ii) Robot \( r_5 \) continue to move west to place itself on an alternate node of that row. (CASE \( \Psi_3 \))
Fig. 6. (i) Robot $r_5$ in row $3$ has $rc$ robots on its west in row $3$ and the above odd row row1 has $rc$ robots (CASE $\Psi_5$). So, $r_5$ moves southwards to alternate row, row5 and then move westwards to place itself on an alternate node of the grid distribution. (ii) An uniform distribution along the grid ($grid_{final}$) is formed, with robots on alternate nodes of this grid distribution.

Algorithm FindDimension($n$)

Input: $n$
Output: Maximum dimensions of the uniform distribution of robots, to be formed

Let $grid_{final}$ be uniform distribution of robots to be formed, with robots on alternate nodes of this grid distribution;
$rc$ be the maximum number of robots, in a row or column of $grid_{final}$; and
$d$ be the maximum number of rows or columns of $grid_{final}$;
$rc \leftarrow \lceil \sqrt{n} \rceil$;
$d \leftarrow (rc \times 2) - 1$;
return $rc, d$

Algorithm FindYMAX($n$)

Input: $n$
Output: The north-most bound of the distribution, $Y_{MAX}$

$r_i \in R$ considers its position as origin (0,0).
Let $Y_{MAX}$ be the $Y$-axis value of the north-most robot in the grid and Initially let, $Y_{MAX} \leftarrow Y$-axis value of $r_1$;
Let $c$ be the robot counter to compare all robots and $c \leftarrow 2$;
while $c <= n$ do
    if $Y$-axis value of $r_c > Y_{MAX}$ then
        $Y_{MAX} \leftarrow Y$-axis value of $r_c$;
    end
    $c \leftarrow c + 1$;
end
return $Y_{MAX}$
Algorithm FindXMIN(n)

Input: n
Output: The west-most bound of the distribution, XMIN

ri ∈ R considers its position as origin (0,0).
Let XMIN be the X-axis value of the west-most robot and
Initially let, XMIN ← X-axis value of r1;
Let c be the robot counter to compare all robots and c ← 2;
while c <= n do
  if X-axis value of rc < XMIN then
    XMIN ← X-axis value of rc;
  end
  c ← c + 1;
end
return XMIN

Algorithm FormGrid(n)

Input: n
Output: Robot R reaches its target point on the grid

Let ri ∈ R and T(x,y) be a grid node with coordinate (x,y);
Initially, consider r1 is at T(x,y) (where x,y=0). Let the final target point of ri be Ti;
Dimensions of the uniform distribution to be formed (grid_final) ← FindDimension;
YMAX ← FindYMAX; and XMIN ← FindXMIN;
j is the row number of rowj starting from the north bound, YMAX
Priority of movement in case of any tie in ri movement ← ri trying to move westwards will get highest priority, followed by eastwards, southwards and northwards respectively;
if ri ∈ rowj such that (j < d and j (mod 2) = 0) then
  while ri ∈ rowj do
    if Tx,y+1 is empty then
      ri moves to Tx,y+1;
      j=j-1;
    end
    else if Tx+1,y is empty then
      ri moves to Tx+1,y;
    end
  end
while ri ∈ rowj such that j > d do
  if Tx,y+1 is empty then
    ri moves to Tx,y+1;
    j=j-1;
  end
  else if Tx+1,y is empty then
    ri moves to Tx+1,y;
  end
end
Algorithm FormGrid(n) continued

while $r_i \notin X_{MIN}$ and $\forall r_i \in row_j$ not on alternate nodes do
  if $T_{x-1,y}$ and $T_{x-2,y}$ are empty then
    $r_i$ moves to $T_{x-1,y}$;
  end
  else if all robots on the west of $r_i$ in row $j$, have no vacant target node to move westwards then
    if $T_{x+1,y}$ and $T_{x+2,y}$ are empty then
      $r_i$ moves to $T_{x+1,y}$;
    end
  end
end

while $\forall r_i$ placed in alternate nodes of alternate rows and $r_i \in row_j$ such that $r_i \in$ first $rc$ robots of row $j$, and $j \neq 1$ and $T_{x,y+2}$ is empty do
  $r_i$ moves to $T_{x,y+2}$;
  $j=j-2$;
end
if $r_i \in row_j$ such that, $j = 1$ and $r_i$ has $\geq rc$ number of robots on its west in row $j$ then
  while $T_{x,y-2}$ is empty and $r_i \notin$ its target point $T_i$ do
    $r_i$ moves to $T_{x,y-2}$;
    $j=j+2$;
    if $r_i$ has $\leq rc$ number of robots on its west in row $j$ then
      while $(T_{x-1,y}$ and $T_{x-2,y}$ are empty) and $r_i$ not on $X_{MIN}$ do
        $r_i$ moves to $T_{x-1,y}$;
      end
      $r_i$ placed in target point $T_i$;
    end
  end
if $r_i \in row_j$ such that, $j > 1$ and $r_i$ has $\geq rc$ robots on its west in row $j$ then
  while $T_{x,y+2}$ is empty and $r_i \notin$ its target point $T_i$ do
    $r_i$ moves to $T_{x,y+2}$;
    $j=j+2$;
    if $r_i$ has $\leq rc$ number of robots on its west in row $j$ then
      while $(T_{x-1,y}$ and $T_{x-2,y}$ are empty) and $r_i$ not on $X_{MIN}$ do
        $r_i$ moves to $T_{x-1,y}$;
      end
      $r_i$ placed in its target point $T_i$;
    end
  end
else if odd rows above row $j$ have $< rc$ robots then
  while $T_{x,y-2}$ is empty and $r_i \notin$ its target point $T_i$ do
    $r_i$ moves to $T_{x,y-2}$;
    $j=j+2$;
    if $r_i$ has $< rc$ robots on its west in row $j$ then
      while $(T_{x-1,y}$ and $T_{x-2,y}$ are empty) and $r_i$ not on $X_{MIN}$ do
        $r_i$ moves to $T_{x-1,y}$;
      end
      $r_i$ placed in its target point $T_i$;
    end
  end
else if odd rows above row $j$ have $\geq rc$ robots then
  while $T_{x,y-2}$ is empty and $r_i \notin$ its target point $T_i$ do
    $r_i$ moves to $T_{x,y-2}$;
    $j=j+2$;
    if $r_i$ has $< rc$ robots on its west in row $j$ then
      while $(T_{x-1,y}$ and $T_{x-2,y}$ are empty) and $r_i$ not on $X_{MIN}$ do
        $r_i$ moves to $T_{x-1,y}$;
      end
      $r_i$ placed in its target point $T_i$;
    end
  end
5 Correctness

Lemma 1. The robots do not move back to their previous positions.

Proof. The robots reside in nodes and do not stop in edges. Also, during the movement to the computed destination node, they do not stop in any in-between nodes, i.e. they execute rigid motion. So, in configurations Ψ_4 and Ψ_5, a robot \( r_i \) in row\( j \) can compute and move to the next odd row, northwards or southwards (i.e. row\( j+2 \) or row\( j-2 \)). Thus, in Figure 4(i), \( r_i \) computes and moves to \( T_{x,y+2} \) and in Figure 6(i), \( r_i \) moves to \( T_{x,y-2} \). As \( r_i \) do not compute and move to its immediate adjacent row (i.e. even row, row\( j+1 \) or row\( j-1 \)) and thus not reaching the configuration Ψ_1, so by the algorithm, \( r_i \) does not move back to its previous position in row\( j \). Consequently, the robots move forward in the algorithm, to achieve the required configuration, without repeating the same positions.

Lemma 2. The algorithm is free of deadlock. Hence progress is assured.

Proof. Let \( r_i \) be any arbitrary robot. By contradiction, let us assume \( r_i \) is in deadlock.

- Case 1: If \( r_i \) is in \( \Psi_1 \), it checks if northward node \( T_{x,y+1} \) is vacant and moves there. Otherwise, \( r_i \) first moves eastwards to \( T_{x+1,y} \) and then northwards. Thus, we can see, \( r_7 \) in Figure 1(i), that in \( \Psi_1 \), \( r_i \) follows distinct movement either eastward or northward as per the condition in Algorithm and finally moves to an odd row. \( r_i \) do not repeat the actions of \( \Psi_1 \) further, as it no longer remain in \( \Psi_1 \). This assures that \( r_i \) is not in deadlock.

- Case 2: When \( r_i \) is in \( \Psi_2 \), it checks if northward node \( T_{x,y+1} \) is vacant and then moves there. Otherwise, \( r_i \) first moves eastwards to \( T_{x+1,y} \) and then northwards. \( r_i \) waits if both \( T_{x,y+1} \) and \( T_{x+1,y} \) are occupied. Thus similar to \( \Psi_1 \), \( r_i \) moves either either northward or eastward till \( r_i \) reaches the \( d^{th} \) row of the grid_{final}, as \( r_8 \) in Figure 1(ii). Further it does not repeat the \( \Psi_2 \) actions, as it no longer remains in \( \Psi_2 \), thus confirming there is no repetition of actions in loop. Hence, \( r_i \) cannot be in deadlock.

- Case 3: When \( r_i \) is in \( \Psi_3 \), it moves westwards to \( T_{x-1,y} \), when two adjacent west nodes, \( T_{x-1,y} \) and \( T_{x-2,y} \) are vacant as \( r_1, r_2, r_8 \) of Figure 2(i). Otherwise, \( r_i \) moves eastwards as \( r_5 \) of Figure 2(ii). If both west and eastwards nodes are unavailable for movement, it waits. Also, in Figures 2(ii), 3(i)(ii), 4(ii), 5(i)(ii) we see, there is always either westward or eastward movement prescribed for \( r_i \) in \( \Psi_3 \), till all the robots in that row are at alternate nodes of that row. Since there is no bound on the east, there will always be vacant node available to accommodate all the robots in that particular row at the alternate nodes in that row. Thus, there is no deadlock condition.

- Case 4: If \( r_i \) is in \( \Psi_4 \), it moves to the northward alternate row node, \( T_{x,y+2} \) like, \( r_4 \) in Figure 4(i). It is a rigid movement and \( r_i \) does not stop in any in-between node (i.e. \( T_{x,y+1} \)) before reaching its destination. This assures no possibility of deadlock.
– Case 5: When \( r_i \) is in \( \Psi_5 \), it either moves to the alternate northward row node \( T_{x,y+2} \) or the alternate southward node \( T_{x,y-2} \) as \( r_5 \) in Figure 6(i), depending on the position of \( r_i \). In either case, \( r_i \) moves to an alternate row, it will always be an odd row. Also as per the dimensions of the grid distribution calculated in Algorithm, there will always be a vacant node available to move the robots, in order to form the grid with robots on alternate nodes, thus assuring progress of the algorithm.

Thus we see, in all the cases, \( r_i \) is not in deadlock. Also by Lemma 1 we can assure that the robots do not move back to their previous positions and does not keep repeating the same configuration in a loop. Hence, there is no possibility of \( r_i \) to be in deadlock condition. Thus, by contradiction we proof that the algorithm is free of deadlock and consequently progress is assured.

**Lemma 3.** When a robot \( r_i \) is moving from one node to another node of the grid, along an edge of the grid, no other robot comes in its path, i.e., the movement of \( r_i \) is collision-free.

**Proof.** In this algorithm, the robots move along the edges of the grid to reach their destination node. For existence of any obstacle robot following two situations may arise.

– The obstacle robot is between the source and destination nodes. This is not possible, as the robots can be placed only on the nodes and not on the edges of the grid. So, as the robots move along the edges of the grid, between the source and destination node, no other robot can exist along the path of the robot.

– A destination node computed by one robot is also been computed by another robot. This is not possible, as the robots have a consistent ordering of their movement based on priority. So, at a time a vacant node can be computed as the target node for one robot only. Also, the semi-synchronous scheduler confirms that when a robot is moving no other robot is looking.

Thus the robots reach their destinations without collision.

**Theorem 1.** A group of autonomous, homogeneous, oblivious, semi-synchronous, mobile robots can deterministically scatter uniformly in alternate nodes of a grid under unlimited visibility and agreement on direction of axes.

**Proof.** The robots in this model are autonomous mobile robots as they can operate independently by executing cycles of the states "look-compute-move". They are homogeneous and cannot be distinguished from each other. Thus, in the look phase, the robots looks to identify the positions of the other robots at that point of time. In this algorithm, the robots do not need to recollect any past action or looked data from the previous cycle, i.e they are oblivious. Here the robots have semi-synchronous scheduler which ensures, that the activated robots at the same time obtain the same snapshot, and their compute and move are executed instantaneously. Since the activated robots obtain the same snapshot, their determination of the grid dimensions, north and west bounds and the computed
destination for a robot are consistent. The robots have full visibility as they need to determine the dimensions and bounds of the grid distribution to be formed. Finally, we have assumed that though the robots have local coordinate system, they have agreement on the direction of the axes. This assumption ensures that all the robots, compute and identify the same robots as the north-bound $Y_{MAX}$ robot and west-bound $X_{MIN}$ robot respectively. With this underlying model, from lemma 2 we have a deadlock free algorithm that assures progress and from lemma 3, we guarantee no collision of robots, thus deterministically reaching the goal of the algorithm, i.e., robots on alternate nodes of a grid distribution.

6 Conclusion

In this paper, we have addressed the scattering in a grid problem for autonomous oblivious mobile robots. The paper presents a distributed algorithm that assumes full visibility robots with semi-synchronous movement and agreement on both axis direction. Our proposed algorithm converges in finite time without collision. The future scope of this work would be to consider limited visibility of the robots. Also, we will try to minimize the overall distance between the robots and make the robots occupy consecutive rows but alternative columns.

References

1. E. Ackerman. Robots that can efficiently disinfect hospitals using uv light could slow coronavirus infections. https://spectrum.ieee.org/automaton/robotics/medical-robots/autonomous-robots-are-helping-kill-coronavirus-in-hospitals. Accessed: 2020.08.09.
2. E. M. Barrameda, S. Das, and N. Santoro. Deployment of asynchronous robotic sensors in unknown orthogonal environments. In S. P. Fekete, editor, Algorithmic Aspects of Wireless Sensor Networks, pages 125–140, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
3. L. Barriere, P. Flocchini, E. Mesa-Barrameda, and N. Santoro. Uniform scattering of autonomous mobile robots in a grid. In 2009 IEEE International Symposium on Parallel Distributed Processing, pages 1–8, May 2009.
4. K. Bose, R. Adhikary, M. K. Kundu, and B. Sau. Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. Theoretical Computer Science, 815:213 – 227, 2020.
5. Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng. Cooperative mobile robotics: antecedents and directions. In Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, volume 1, pages 226–234 vol.1, Aug 1995.
6. R. Cohen and D. Peleg. Local spreading algorithms for autonomous robot systems. Theoretical Computer Science, 399(1):71 – 82, 2008. Structural Information and Communication Complexity (SIROCCO 2006).
7. X. Défago and S. Souissi. Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. Theoretical Computer Science, 396(1):97 – 112, 2008.
8. A. Efrima and D. Peleg. Distributed algorithms for partitioning a swarm of autonomous mobile robots. In G. Prencipe and S. Zaks, editors, Structural Information and Communication Complexity, pages 180–194, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

9. N. Heo and P. K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003., volume 3, pages 1597–1602 vol.3, March 2003.

10. A. Howard, M. J. Matarić, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, editors, Distributed Autonomous Robotic Systems 5, pages 299–308, Tokyo, 2002. Springer Japan.

11. A. Howard, M. J. Matarić, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. Autonomous Robots 13 (2), 2:113–126, September 2002.

12. T.-R. Hsiang, E. M. Arkin, M. A. Bender, S. Fekete, and J. S. B. Mitchell. Online dispersion algorithms for swarms of robots. In Proceedings of the Nineteenth Annual Symposium on Computational Geometry, SCG ’03, page 382–383, New York, NY, USA, 2003. Association for Computing Machinery.

13. T.-R. Hsiang, E. M. Arkin, M. A. Bender, S. P. Fekete, and J. S. B. Mitchell. Algorithms for Rapidly Dispersing Robot Swarms in Unknown Environments, pages 77–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

14. M. Kasuya, N. Ito, N. Inuzuka, and K. Wada. A pattern formation algorithm for a set of autonomous distributed robots with agreement on orientation along one axis. Systems and Computers in Japan, 37(10):89–100, 2006.

15. A. D. Kshemkalyani, A. R. Molla, and G. Sharma. Fast dispersion of mobile robots on arbitrary graphs. In F. Dressler and C. Scheideler, editors, Algorithms for Sensor Systems, pages 23–40, Cham, 2019. Springer International Publishing.

16. A. D. Kshemkalyani, A. R. Molla, and G. Sharma. Dispersion of mobile robots in the global communication model. In Proceedings of the 21st International Conference on Distributed Computing and Networking, ICDCN 2020, New York, NY, USA, 2020. Association for Computing Machinery.

17. A. D. Kshemkalyani, A. R. Molla, and G. Sharma. Dispersion of mobile robots on grids. In M. S. Rahman, K. Sadakane, and W.-K. Sung, editors, WALCOM: Algorithms and Computation, pages 183–197, Cham, 2020. Springer International Publishing.

18. Nojeong Heo and P. K. Varshney. Energy-efficient deployment of intelligent mobile sensor networks. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 35(1):78–92, Jan 2005.

19. T. Okumura, K. Wada, and Y. Katayama. Rendezvous of Asynchronous Mobile Robots with Lights, pages 434–448. Springer International Publishing, Cham, 2018.

20. D. Payton, R. Estkowski, and M. Howard. Pheromone robotics and the logic of virtual pheromones. In E. Sahin and W. M. Spears, editors, Swarm Robotics, pages 45–57, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

21. P. Poudel and G. Sharma. Time-optimal uniform scattering in a grid. In Proceedings of the 20th International Conference on Distributed Computing and Networking, ICDCN ’19, page 228–237, New York, NY, USA, 2019. Association for Computing Machinery.

22. M. Simon. The covid-19 pandemic is a crisis that robots were built for. https://www.wired.com/story/covid-19-pandemic-robots/. Accessed:2020.08.09.

23. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. SIAM Journal on Computing, 28(4):1347–1363, 1999.