Arbitrary Pattern Formation on Infinite Grid by Asynchronous Oblivious Robots

Kaustav Bose, Ranendu Adhikary, Manash Kumar Kundu, Buddhadeb Sau
Arbitrary Pattern Formation Problem

Design a *distributed algorithm* that allows a set of *autonomous* mobile robots to form *any* pattern given as input, *without any collisions*.
Mobile Robot Model

The robots are

• **Autonomous** (there is no central control)
• **Homogeneous** (the robots execute the same distributed algorithm)
• **Anonymous** (the robots have no unique identifiers)
• **Identical** (the robots are indistinguishable by their appearance)
• **Silent** (no explicit means of communication)
• **Disoriented** (they do not have access to any global coordinate system)
• **Oblivious** (no memory of past actions)
Look-Compute-Move Cycle
Look-Compute-Move Cycle

Look

Move

Compute

Takes a snapshot of the positions of all the robots
Look-Compute-Move Cycle

Look

Compute

Move

Performs computations according to a deterministic algorithm to decide a destination.
Look-Compute-Move Cycle

Moves to the computed destination
Look-Compute-Move Cycle

- Each robot repeats the cycle infinite number of times
- Since the robots are *oblivious*, after the completion of each cycle, the contents of the local memory of a robot are erased
Fully Asynchronous Scheduler (ASYNC)

- There is no common notion of time
- The amount of times spent in LOOK, COMPUTE, MOVE and inactive states are different and unpredictable for different robots.
Continuous vs Discrete Environment

The robots are deployed in either continuous or discrete domain.

**Continuous Domain**
- 2-dimensional plane
- 3-dimensional space

**Discrete Domain**
- Graph
In continuous domains,
- **robots are usually modelled as dimensionless points**
- robots can move in any arbitrary direction with infinite precision
- robots can move by any amount, even by arbitrarily small amounts, with infinite precision
- sometimes robots are allowed to move along curved trajectories
- it becomes easy to avoid collisions
In continuous domains,
- robots are usually modelled as dimensionless points
- **robots can move in any arbitrary direction with infinite precision**
- robots can move by any amount, even by arbitrarily small amounts, with infinite precision
- sometimes robots are allowed to move along curved trajectories
- it becomes easy to avoid collisions
In continuous domains,
- robots are usually modelled as dimensionless points
- robots can move in any arbitrary direction with infinite precision
- robots can move by any amount, even by arbitrarily small amounts, with infinite precision
- sometimes robots are allowed to move along curved trajectories
- it becomes easy to avoid collisions
In continuous domains,
- robots are usually modelled as dimensionless points
- robots can move in any arbitrary direction with infinite precision
- robots can move by any amount, even by arbitrarily small amounts, with infinite precision
- sometimes robots are allowed to move along curved trajectories
- it becomes easy to avoid collisions
Continuous vs Discrete Environment

In continuous domains,

- robots are usually modelled as dimensionless points
- robots can move in any arbitrary direction with infinite precision
- robots can move by any amount, even by arbitrarily small amounts, with infinite precision
- robots are allowed to move along curved trajectories
- it becomes easy to avoid collisions
Robots in Discrete Environment

Many real-life robot navigation systems are based on discretized terrains using magnetic or visible guidance.
Previous works studied the problem in continuous domains. We have considered the problem in an infinite grid.

- The movements of the robots are restricted along grid lines.
- In one step, a robot can move to one of its four neighboring grid points.
Our Result

Any arbitrary pattern can be formed by a set of asynchronous and oblivious robots on infinite grid if the initial configuration is asymmetric.
How to Embed the Pattern?

Since the robots do not have access to any global coordinate system, there is no agreement regarding where and how the given pattern is to be embedded.
Exploit the **asymmetric** nature of the configuration to reach an agreement on a common global coordinate system.
Reaching Agreement on Coordinate System

- Take the smallest enclosing rectangle ABCD of the configuration
- Let $m \times n$ be the size of ABCD
- Associate binary strings of length $mn$ to corner
Reaching Agreement on Coordinate System

For the corner A, if $|AD| \leq |AB|$, define the binary string $\lambda_{AD}$.
For the corner A, if $|AD| \leq |AB|$, define the binary string $\lambda_{AD}$:

\[
\lambda_{AD} = 00110100000010101 \\
01100000000001000 \\
10100110000000001 \\
001000001000
\]
Reaching Agreement on Coordinate System

- 8 strings for square and 4 strings for non-square rectangle
- Asymmetric configuration $\Rightarrow$ no two strings are equal
- Take the unique lexicographically largest string
Reaching Agreement on Coordinate System

If $\lambda_{AD}$ is lexicographically largest,

- A is the origin
- AD is the (positive) Y-axis
- AB is the (positive) X-axis
Try to keep the coordinate system invariant during the movements of the robots.
Reaching Agreement on Coordinate System

Internal Robots

Tail

Head

A B C D
The algorithm is divided into **seven phases**
Since the robots are **oblivious**, each time a robot wakes up it has to decide the current phase from certain characteristics of the configuration
Sketch of The Algorithm

The head and tail will move to

- expand the smallest enclosing rectangle to make enough space for collision less movements of the internal robots
- ensure that the coordinate system remains unchanged when the internal robots move
Sketch of The Algorithm

The internal robots will partially form the pattern
Sketch of The Algorithm

The head and tail reposition themselves to complete the pattern.
Phase 1

The tail moves right until
- $n \geq \max\{M, m\} + 2$
- $n \geq 2 \max\{N, H\}$

Size of current config $\mathcal{C}$

$= m \times n$

Size of the given pattern $\mathcal{P}$

$= M \times N$

Size of $\mathcal{C} \setminus \{\text{tail}\} = V \times H$
Phase 1

The tail moves right until
- $n \geq \max\{M, m\} + 2$
- $n \geq 2 \max\{N, H\}$

Size of current config $C = m \times n$

Size of the given pattern $P = M \times N$

Size of $C \setminus \{\text{tail}\} = V \times H$
Phase 2

The head moves to the origin
Phase 2

The head moves to the origin

A

D

C

B

The head moves to the origin
Further extend the smallest enclosing rectangle so that

- \( m \geq \max\{M, V\} + 1 \)

Size of current config \( \mathcal{C} \)

\[ = m \times n \]

Size of the given pattern \( \mathcal{P} \)

\[ = M \times N \]

Size of \( \mathcal{C} \setminus \{\text{tail}\} = V \times H \)
Phase 3

Case 1: \( \mathcal{C} \setminus \{ \text{tail} \} \) has no horizontal reflectional symmetry

Further extend the smallest enclosing rectangle so that

- \( m \geq \max\{M, V\} + 1 \)

Size of current config \( \mathcal{C} \)

\[ = m \times n \]

Size of the given pattern \( \mathcal{P} \)

\[ = M \times N \]

Size of \( \mathcal{C} \setminus \{ \text{tail} \} \)

\[ = V \times H \]
Phase 3

Case 1: $\mathcal{C} \setminus \{\text{tail}\}$ has no horizontal reflectional symmetry

Further extend the smallest enclosing rectangle so that
- $m \geq \max\{M, V\} + 1$

Size of current config $\mathcal{C}$
$= m \times n$

Size of the given pattern $\mathcal{P}$
$= M \times N$

Size of $\mathcal{C} \setminus \{\text{tail}\} = V \times H$
Phase 3

Case 2: $\mathcal{G} \setminus \{\text{tail}\}$ has a horizontal reflectional symmetry

Further extend the smallest enclosing rectangle so that
- $m \geq \max\{M, V\} + 1$

Size of current config $\mathcal{G}$
$= m \times n$

Size of the given pattern $\mathcal{P}$
$= M \times N$

Size of $\mathcal{G} \setminus \{\text{tail}\} = V \times H$
 Phase 3

**Case 2:** $\mathcal{C} \setminus \{\text{tail}\}$ has a horizontal reflectional symmetry

Further extend the smallest enclosing rectangle so that

- $m \geq \max\{M, V\} + 1$

Size of current config $\mathcal{C}$

$= m \times n$

Size of the given pattern $\mathcal{P}$

$= M \times N$

Size of $\mathcal{C} \setminus \{\text{tail}\} = V \times H$
Case 2: $\mathcal{C} \setminus \{\text{tail}\}$ has a horizontal reflectional symmetry

The coordinate system flips

Further extend the smallest enclosing rectangle so that
- $m \geq \max\{M, V\} + 1$

Size of current config $\mathcal{C}$
- $= m \times n$

Size of the given pattern $\mathcal{P}$
- $= M \times N$

Size of $\mathcal{C} \setminus \{\text{tail}\} = V \times H$
Phase 4

- The interior robots are inside the finite subgrid of size \((m - 1) \times \lfloor n/2 \rfloor\).
- If the interior robots move while remaining inside the finite grid, the coordinate system will not change.
- The interior robots will partially form the given pattern.
Phase 4

The finite subgrid can be seen as a coiled up path.
Phase 4

The problem reduces to pattern formation on a finite path with

**Full agreement in coordinate system**

⇒ The pattern is given as **fixed points** on the path
Phase 4

$r_i$ will move towards $t_i$ if it is not obstructed by another robot.
Phase 5, 6 and 7

Case 2: τ \{tail\} has no horizontal reflectional symmetry
Summing Up

If the initial configuration is asymmetric, then any arbitrary pattern can be formed by a set of $k$ oblivious robots in ASYNC in $O(D^2 \cdot k)$ moves, where

\[
\begin{align*}
m \times n &= \text{size of the initial configuration} \\
M \times N &= \text{size of the given pattern} \\
D &= \max\{M, N, m, n\}
\end{align*}
\]
Thank You!!