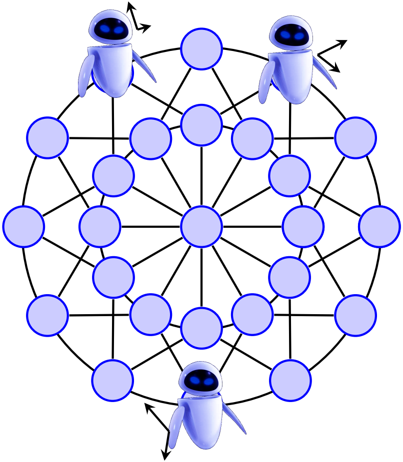
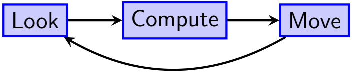
On the Synthesis of Mobile Robots Algorithms: the Case of Ring Gathering

Laure Millet, Maria Potop-Butucaru,
Nathalie Sznajder and Sébastien Tixeuil

Context [SY1993]

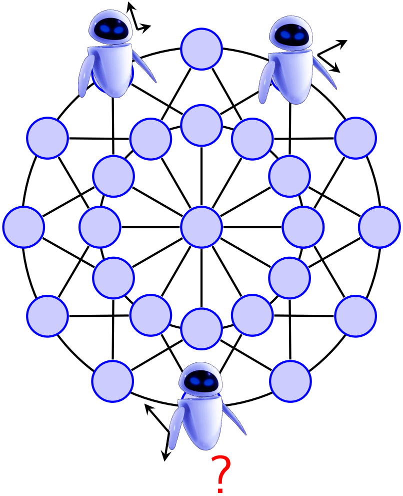


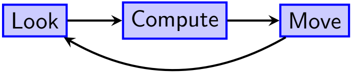


Restricted robots

- Identical and anonymous
- Oblivious
- No communication
- Only one sense: sight
- No common handedness

Context [SY1993]





Restricted robots

- Identical and anonymous
- Oblivious
- No communication
- Only one sense: sight
- No common handedness

Two execution models

Synchronous

Atomic and synchronous

Two variants:

- FSYNC
- SSYNC

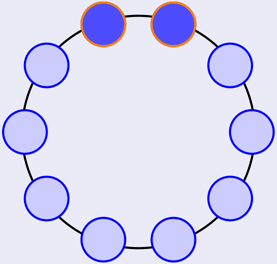
Asynchronous

- ASYNC

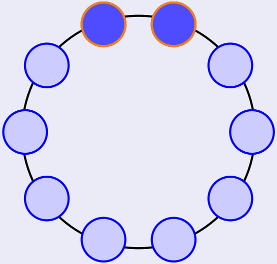
Two execution models

Robots synchronization level

Synchronous



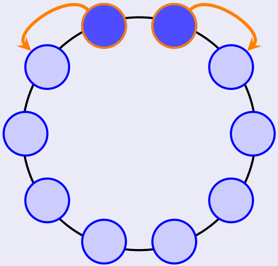
Asynchronous



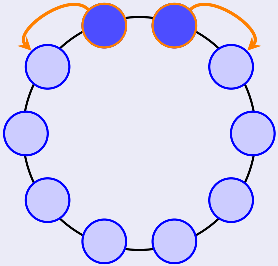
Two execution models

Robots synchronization level

Synchronous



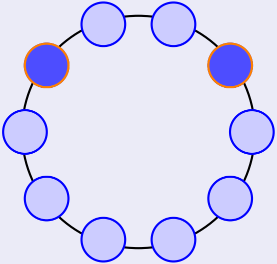
Asynchronous



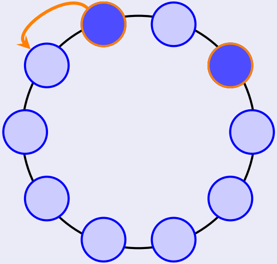
Two execution models

Robots synchronization level

Synchronous



Asynchronous



A common objective

The Gathering on a Ring

State of the art

- Multiple impossibility results [KMP08]
- Algorithms with less constraints on the robot model [BMPT10,ASN12]
- Sometimes incorrect ...

Problem

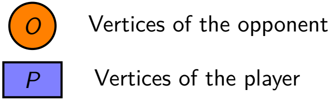
How to Automatically Generate an Efficient and Correct algorithm?

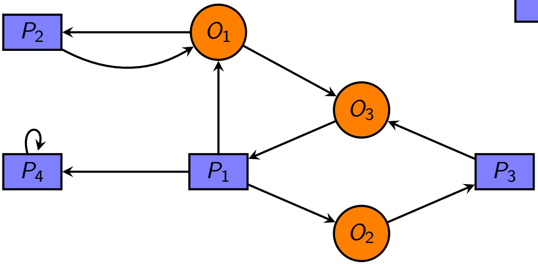
How this problem can be resolved thanks to games?

A distributed system made of synchronous robots can be seen as two-player game

- Player: coalition of robots
- Opponent: scheduler

Two-player game

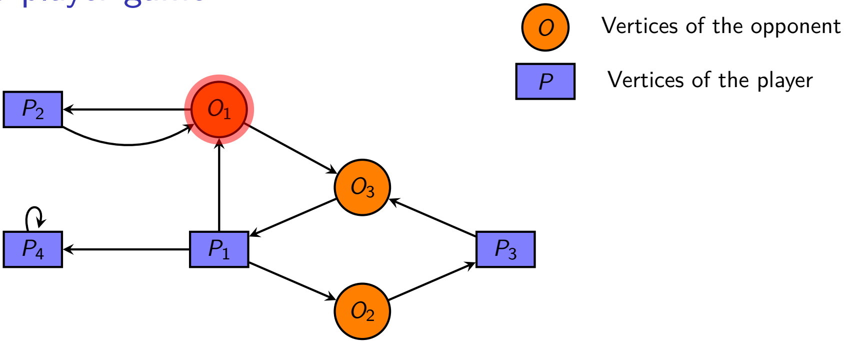




An Arena $\mathcal{A} = (V, E)$

$V = V_P \uplus V_O$
$E \subseteq V \times V$

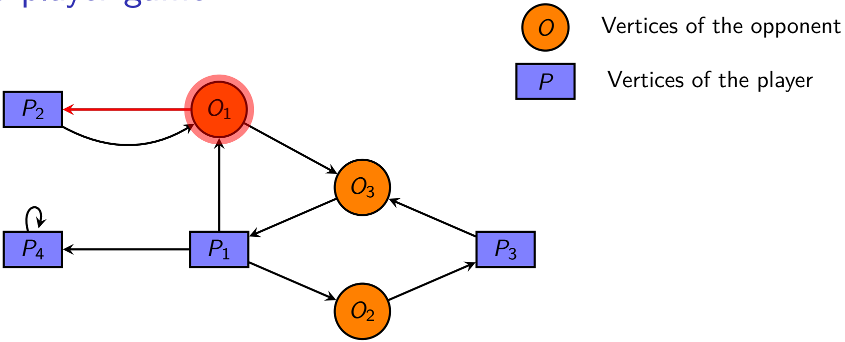
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

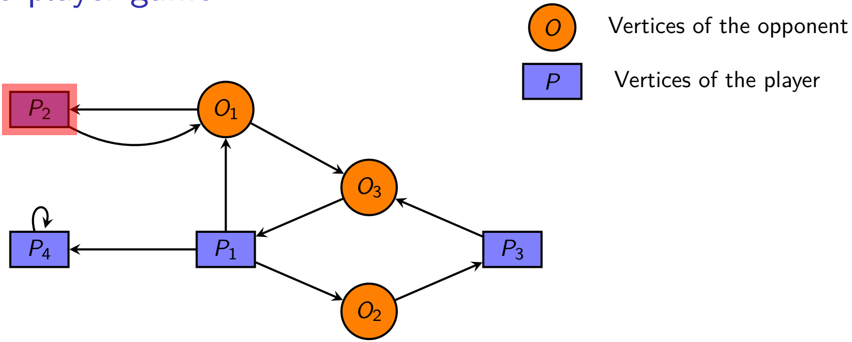
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

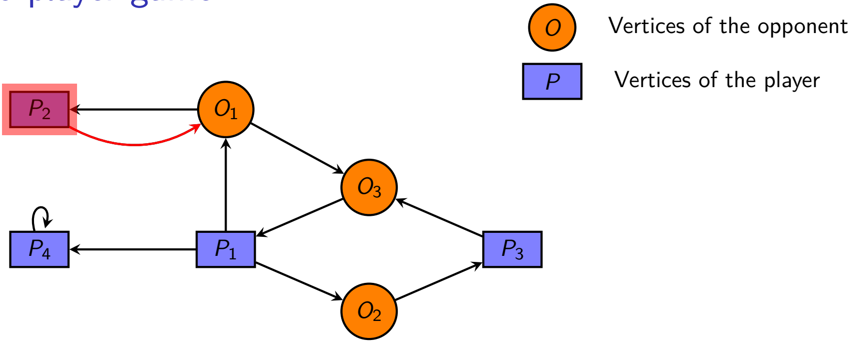
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

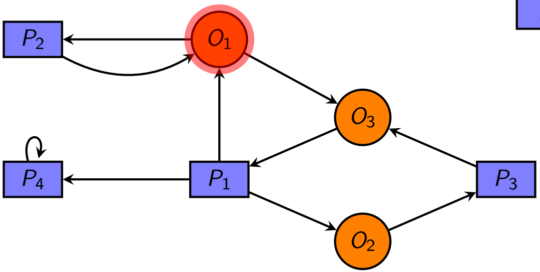
Two-player game

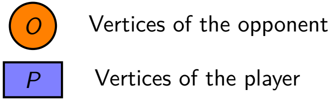


A Play in $\mathcal{A}$, $\pi \in V^{\infty}$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

Two-player game

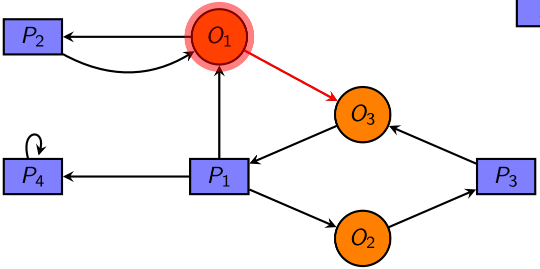


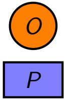


A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

Two-player game

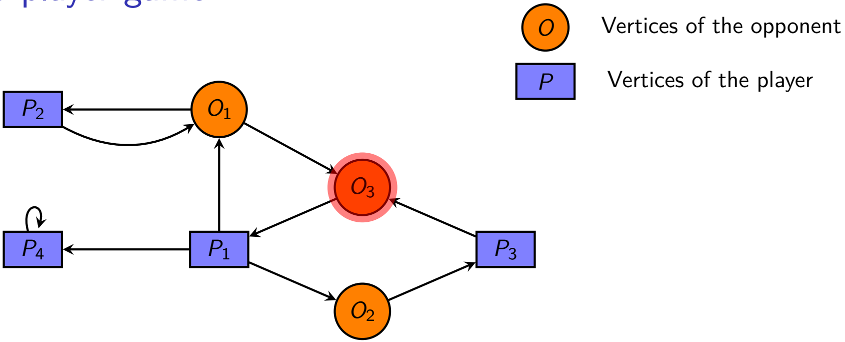




A Play in $\mathcal{A}$, $\pi \in V^{\infty}$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

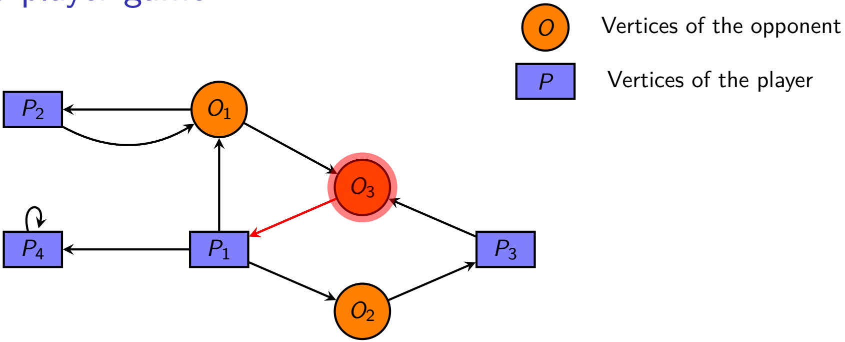
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

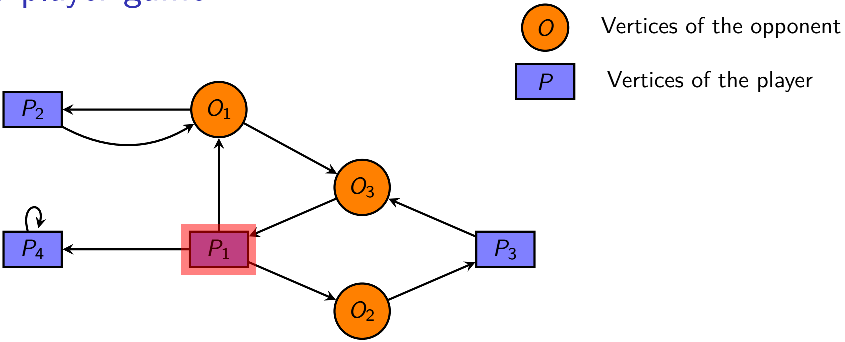
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^{\infty}$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

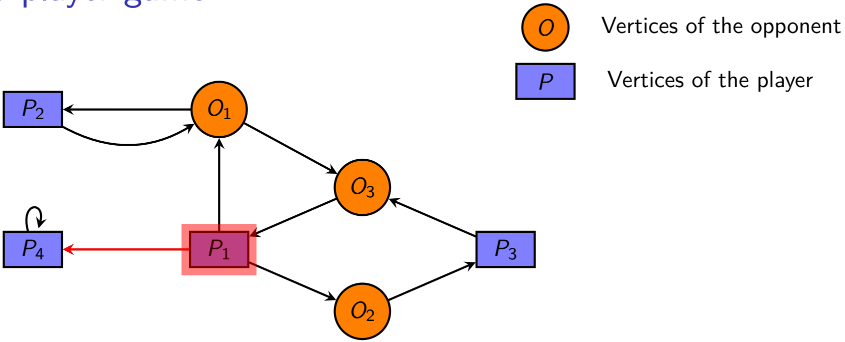
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

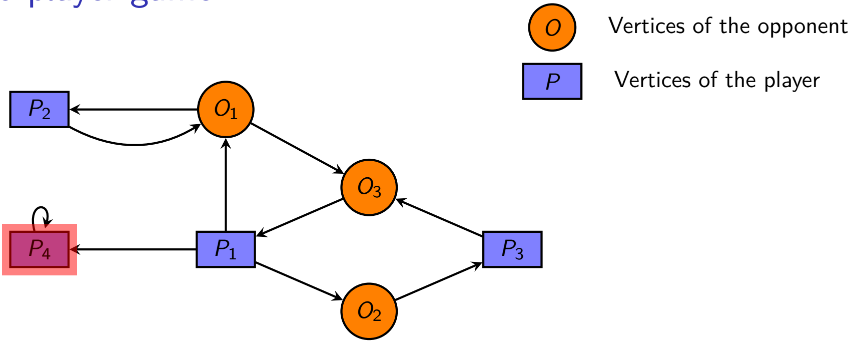
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$$

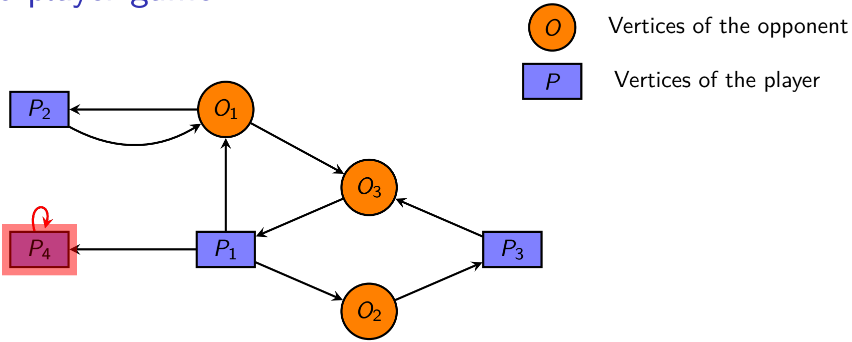
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

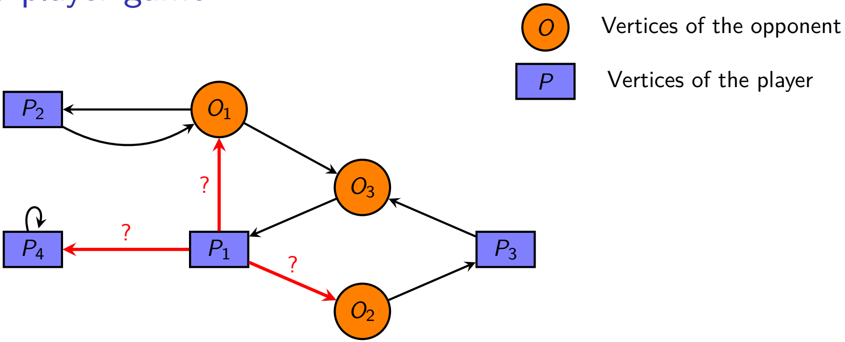
Two-player game



A Play in $\mathcal{A}$, $\pi \in V^\infty$

$\pi = O_1 P_2 O_1 O_3 P_1 P_4 P_4 \cdots$

Two-player game

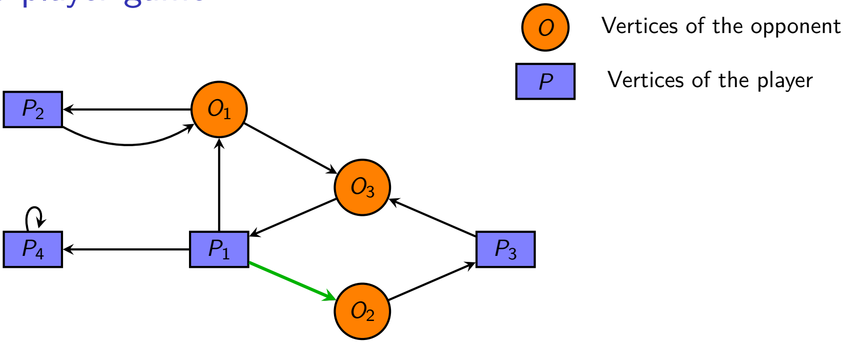


A Strategy for the player

A memoryless strategy:

P_1 leads to O_2, P_2 leads to O_1, P_3 leads to O_3, P_4 leads to P_4.

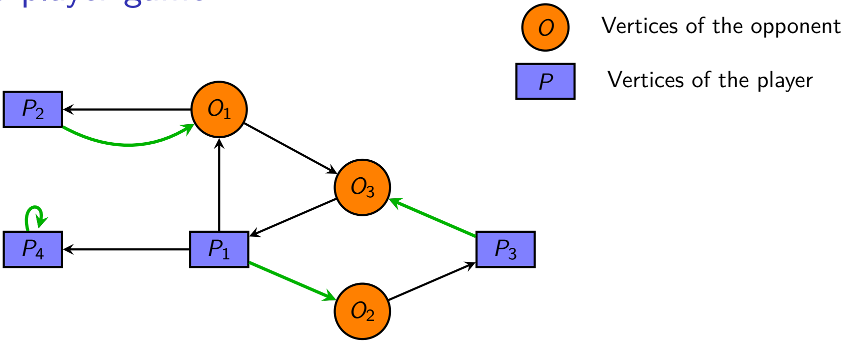
Two-player game



A Strategy for the player

A memoryless strategy:
P_1 leads to O_2, P_2 leads to O_1, P_3 leads to O_3, P_4 leads to P_4.

Two-player game

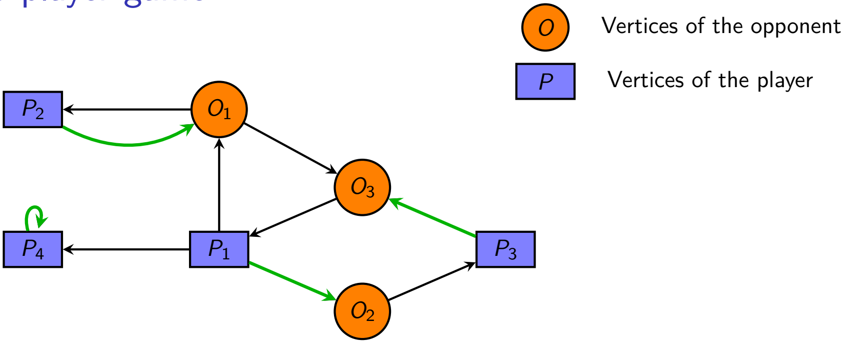


A Strategy for the player

A memoryless strategy:

P_1 leads to O_2, P_2 leads to O_1, P_3 leads to O_3, P_4 leads to P_4.

Two-player game



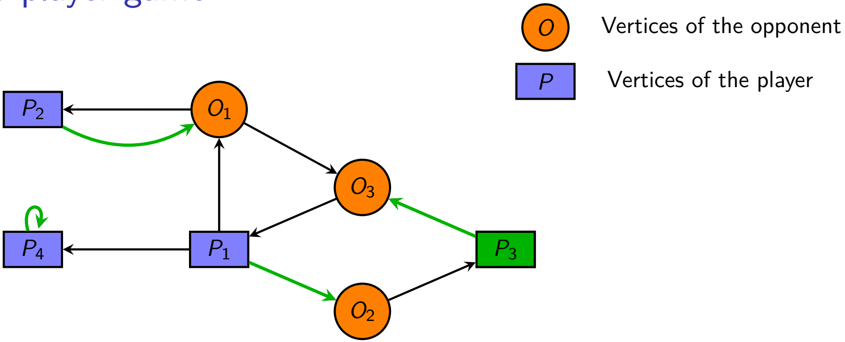
Winning strategy

Is this strategy winning?

Winning condition $\subseteq V^\infty$

$Reach(P_3)$

Two-player game



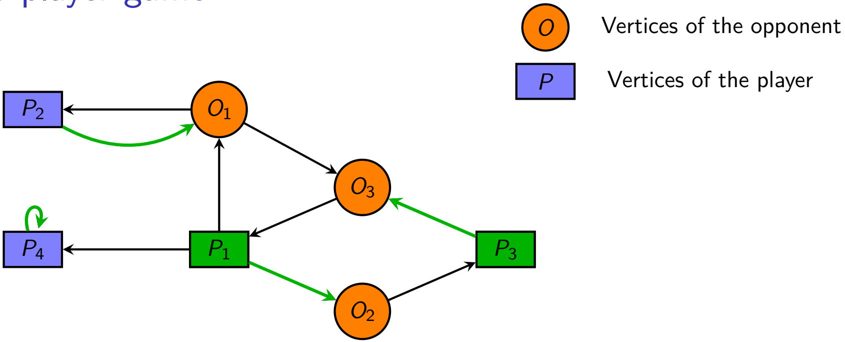
Winning strategy

Is this strategy winning?

Winning condition $\subseteq V^\infty$

$Reach(P_3)$

Two-player game



Winning strategy

Is this strategy winning?

Winning condition $\subseteq V^\infty$

$Reach(P_3)$

Specificities of the encoding

A global strategy that can be distributed

Robots are identical

A memoryless strategy

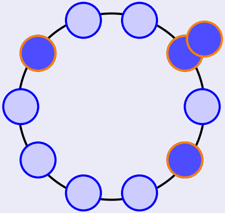
Robots are oblivious

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$

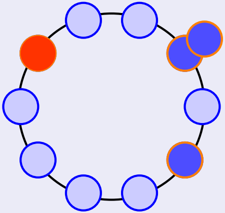


Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



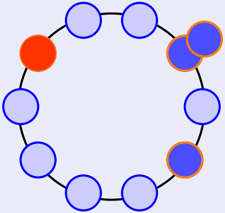
- $(2, -1, 1, 4)$

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



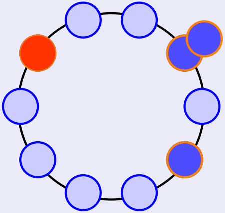
- $(2, -1, 1, 4)$
- $(4, 1, -1, 2)$

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



- $(2, -1, 1, 4)$
- $(4, 1, -1, 2)$

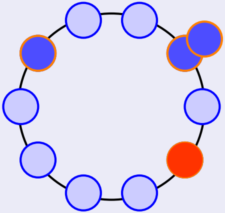
$\sim$

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



- $(2, -1, 1, 4)$
- $(4, 1, -1, 2)$
- $(1, -1, 2, 4)$

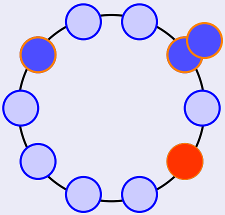
$\sim$

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



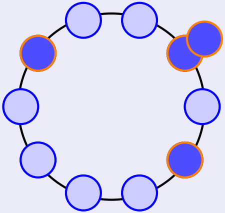
- $(2, -1, 1, 4)$
- $(4, 1, -1, 2)$ $\sim$
- $(1, -1, 2, 4)$ $\circlearrowleft$

Configurations:

Positions of k robots in a ring of n nodes

Definition

$(d_1, \cdots, d_k)$, such that $\Sigma_{i=1}^{k} d_i = n - k$, and $d_i \in \mathbb{N} \cup \{-1\}$



- $(2, -1, 1, 4)$
- $(4, 1, -1, 2)$ $\sim$
- $(1, -1, 2, 4)$ $\circlearrowleft$

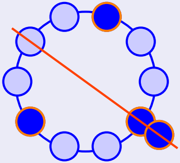
Equivalence class $\equiv$

$\equiv \stackrel{\text{def}}{=} (\circlearrowleft \cup \sim)^*$

Actions: Depend on the current configuration

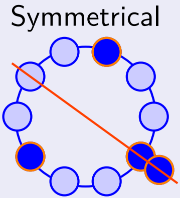
Different types of configurations

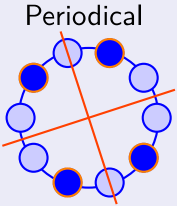
Symmetrical



Actions: Depend on the current configuration

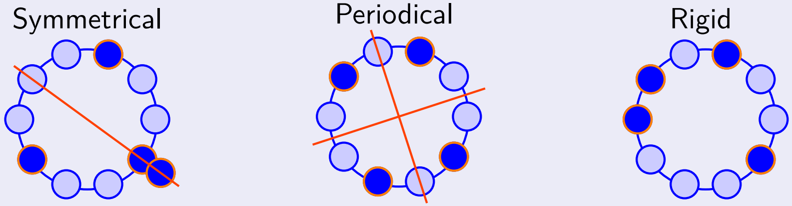
Different types of configurations





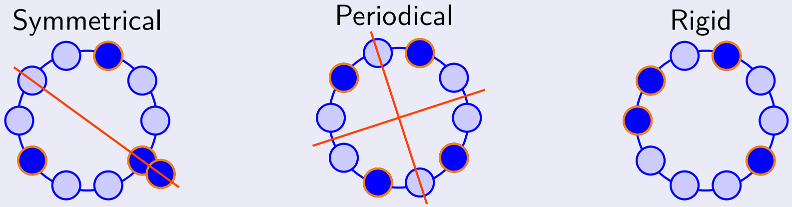
Actions: Depend on the current configuration

Different types of configurations



Actions: Depend on the current configuration

Different types of configurations



A decision function, $f : View \rightarrow \Delta$

$\Delta = \{\curvearrowright, \curvearrowleft, \uparrow, ?\}$

- non-disoriented robots: $\curvearrowright$ or $\curvearrowleft$ or $\uparrow$
- disoriented robots: ? or $\uparrow$

Hence symmetrical\tower robots act the same way

Encoding of the Arena

The Arena, $\mathcal{A}_{\text{gather}} = (V_P \uplus V_O, E)$

Encoding of the Arena

The Arena, $\mathcal{A}_{\text{gather}} = (V_P \uplus V_O, E)$

- $V_P = (\mathcal{C}/\equiv)$

An example for 3 robots in a 6 nodes ring

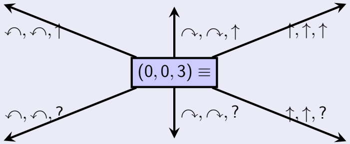
$(0,0,3)\equiv$

Encoding of the Arena

The Arena, $\mathcal{A}_{\text{gather}} = (V_P \uplus V_O, E)$

- $V_P = (\mathcal{C}/\equiv)$
- $E \subseteq (V_P \times V_O) \cup \ldots$

An example for 3 robots in a 6 nodes ring

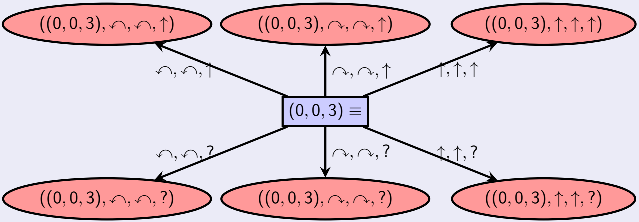


Encoding of the Arena

The Arena, $\mathcal{A}_{\text{gather}} = (V_P \uplus V_O, E)$

- $V_P = (\mathcal{C}/\equiv)$
- $V_0 = \mathcal{C} \times \Delta^k$
- $E \subseteq (V_P \times V_O) \cup$

An example for 3 robots in a 6 nodes ring

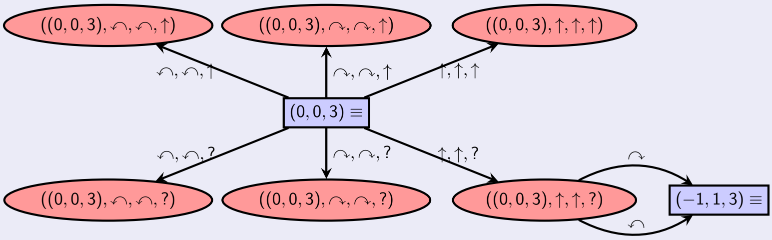


Encoding of the Arena

The Arena, $\mathcal{A}_{\text{gather}} = (V_P \uplus V_O, E)$

- $V_P = (\mathcal{C}/\equiv)$
- $V_0 = \mathcal{C} \times \Delta^k$
- $E \subseteq (V_P \times V_O) \cup (V_O \times V_P)$

An example for 3 robots in a 6 nodes ring



Results

The best algorithm

For 3 robots and on a ring of size $n \in [4, 100]$

A parameterized algorithm

An original approach:

- base-case: algorithm synthesis
- induction: hand made

Future works

Synthesis for synchronous robots

For 4 robots:

- : only ad-hoc algorithm
- : no impossibility results

For any number of robots

Distributed synthesis for asynchronous robots

Find sub-classes where the problem is decidable