Optimal Torus Exploration by Oblivious Robots

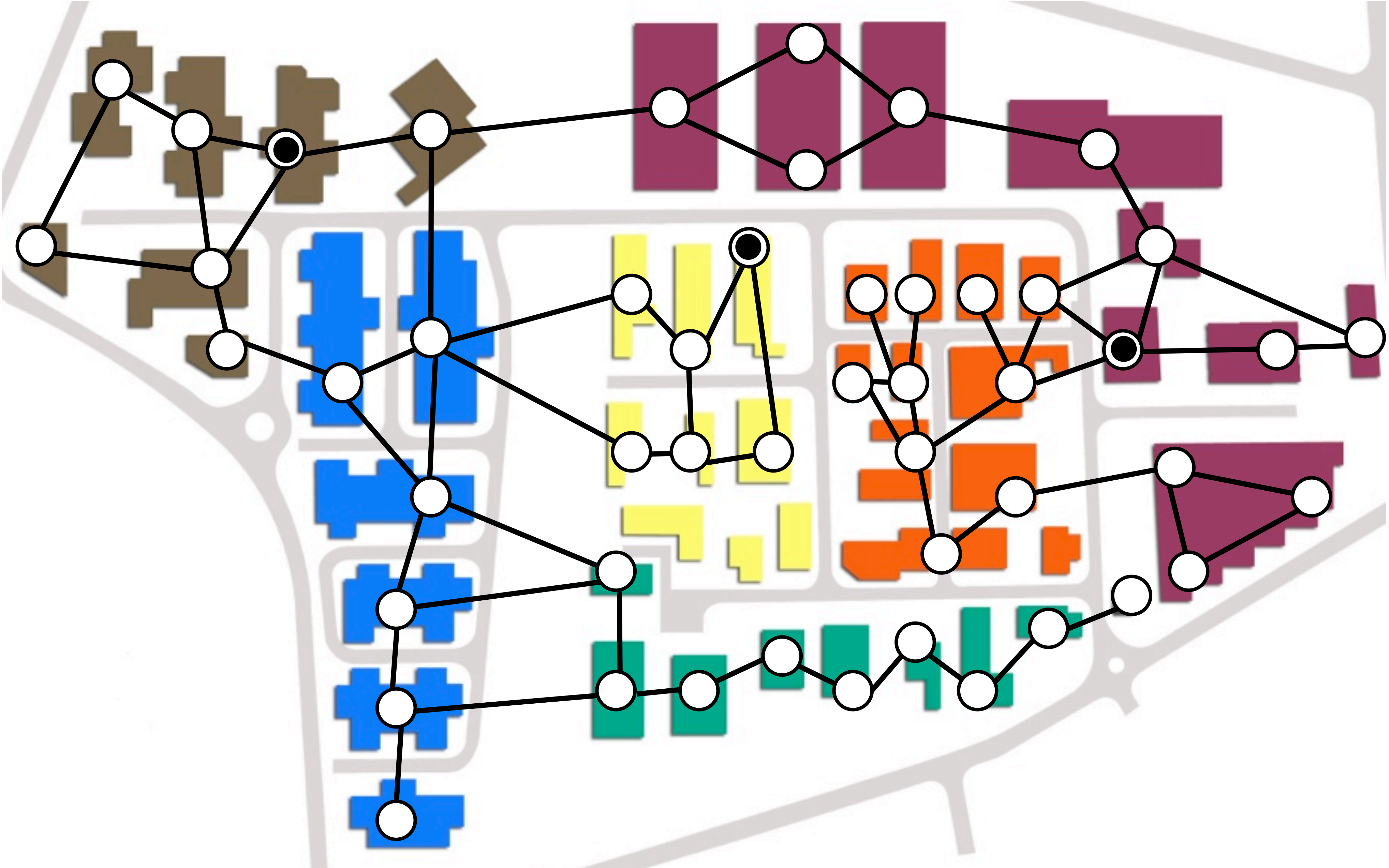
Franck Petit

University Pierre and Marie Curie, Sorbonne Univ.

LIP6 CNRS / INRIA

Joint work with Stéphane Devismes, Anissa Lamani, and Sébastien Tixeuil

Environment



Terminating Exploration

Starting from an arbitrary configuration where no pair of robots are located on the same node

✓ Exploration

Each node must be visited by at least one robot

✓ Termination

Eventually, every robot stays idle

Challenges

What are the minimal conditions to solve the exploration problem deterministically (probabilistically)?

What is the minimal number of robots?

Related Work

n: Number of nodes
k: Number of robots

✓ *Ring* [*Flocchini et al.*, OPODIS 2007] [*Devismes et al.*, SIROCCO 2009] [*Lamani et al.*, SIROCCO 2010] [*Datta et al.*, ICDCS 2013] [*Datta et al.*, APDCM 2015]

- Deterministic exploration impossible if k divides n (except if $k = n$)
- Asynchronous deterministic algorithm with $k > 16$
- Deterministic or probabilistic exploration impossible if $k < 4$
- Probabilistic algorithm impossible in asynchronous settings
- Optimal Semi-synchronous Probabilistic Algorithm
- Deterministic exploration impossible if $k < 5$ and n even
- Optimal asynchronous deterministic algorithm, k = 5 and n even
- Optimal semi-synchronous deterministic algorithm, k = 4 and n odd
- Vision limited to distance 1: possible iff synchronous
- Optimal deterministic synchronous algorithm with k = 5
- Vision limited to distance 2: semi-synchronous algorithms with k = 7
- Vision limited to distance 3: semi-synchronous algorithms with $k \in \{5,7\}$

Related Work

n: Number of nodes
k: Number of robots

✓ *Ring* [*Flocchini et al.*, OPODIS 2007] [*Devismes et al.*, SIROCCO 2009] [*Lamani et al.*, SIROCCO 2010] [*Datta et al.*, ICDCS 2013] [*Datta et al.*, APDCM 2015]

✓ *Tree* [*Flocchini et al.*, SIROCCO 2008]

- Asynchronous deterministic algorithm for trees with maximum degree equal to 3: $k \in \Theta(\log n / \log \log n)$
- Arbitrary tree: $k \in \Theta(\log n)$

✓ *Chain* [*Flocchini et al.*, IPL 2011]

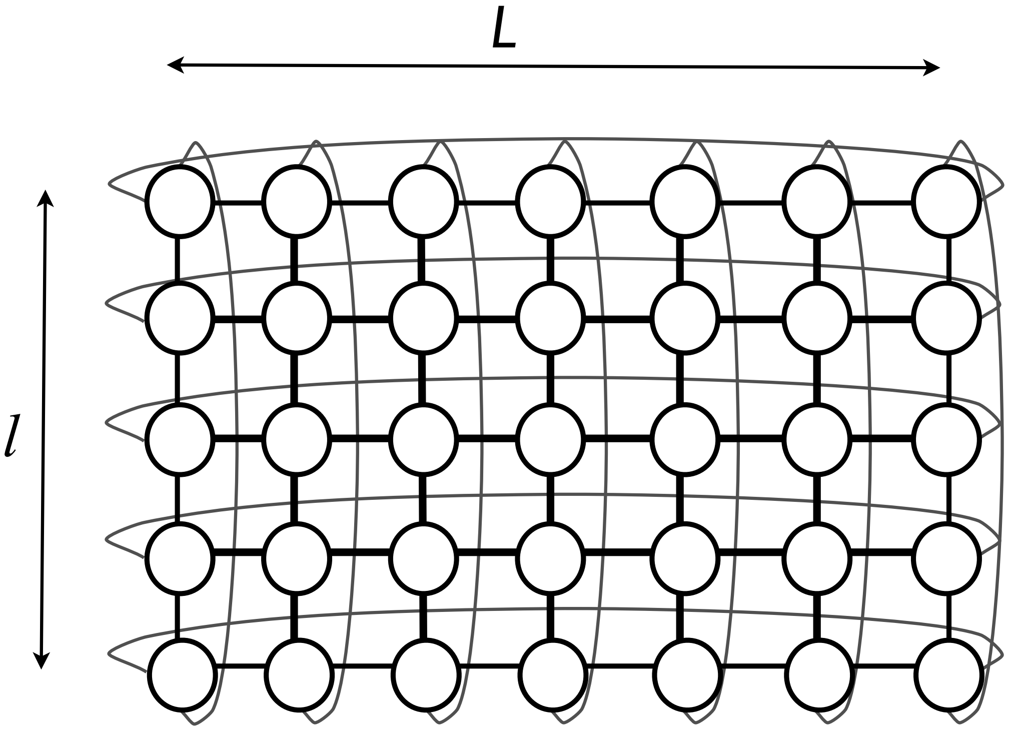
- Characterization of *k*: $k = 3$, $k > 4$, or $k = 4$ and *n* odd

✓ *Grid* [*Devismes et al.*, SSS 2012]

- Deterministic or probabilistic exploration impossible if $k < 3$
- Optimal Semi-synchronous Deterministic Algorithm, $k = 3$

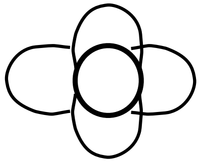
Torus

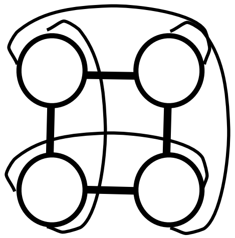
- *Graph (G) with n nodes*
    - ✓Anonymous simple (l,L)-torus ($l \leq L$)

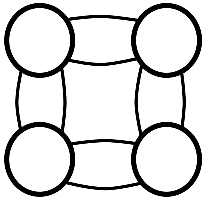


Torus

- ❑ *Graph (G) with n nodes*
  - ✓Anonymous simple (l,L)-torus ($l \le L$)



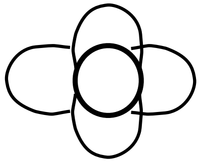


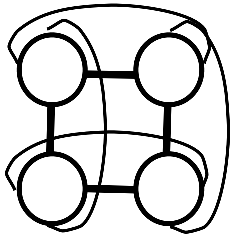


Torus

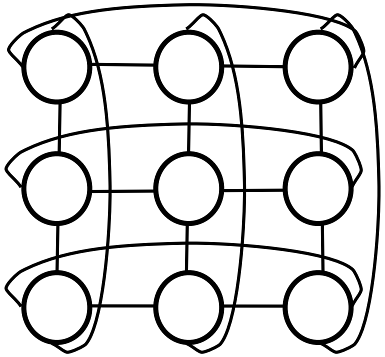
❑ *Graph (G) with n nodes*

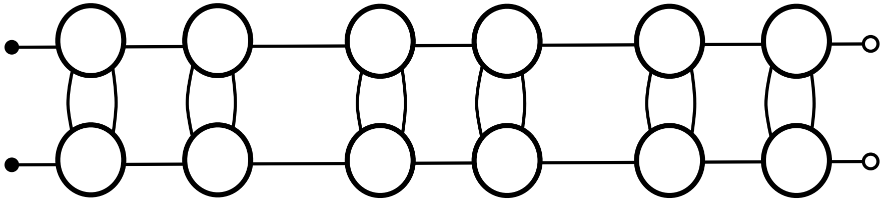
- ✓Anonymous simple (l,L)-torus $(l \leq L)$





$3 \leq l \leq L$





Torus

❑ *Graph (G) with n nodes*

- ✓Anonymous simple (l,L)-torus ($l \leq L$)

Why addressing such an odd and abstract topology ?

✓ *Ring* [*Flocchini et al.*, OPODIS 2007] [*Devismes et al.*, SIROCCO 2009] [*Lamani et al.*, SIROCCO 2010] [*Datta et al.*, ICDCS 2013] [*Datta et al.*, APDCM 2015] ← Regular topology

Does the increase of the degree of symmetry make the problem harder to solve ?

Model

□ *Graph (G) with n nodes*

- ✓Anonymous simple (l,L)-torus $(l \leq L)$

□ *k robots*

- ✓ *Autonomous*
- ✓ *Uniform and anonymous*
- ✓*Mobile*
- ✓*Oblivious*
- ✓*Cannot communicate directly*
- ✓ *Vision*

Model

❑ *Graph (G) with n nodes*

- ✓Anonymous simple (l,L)-torus ($l \leq L$)

❑ *k robots*

- ✓ *Look*
  - ➢ *Take a snapshot to see the position of the other robots on the torus*

- ✓ *Compute*
  - ➢*Compute a neighboring destination*

Destination

●

- ✓ *Move*
  - ➢ *Move towards the computed neighboring destination*

Model

❑ *Graph (G) with n nodes*

✓Anonymous simple (l,L)-torus ($l \leq L$)

❑ *k robots*

❑ *Semi-Synchronous Model (SSM)*

✓ In each configuration, *k’* robots are activated ($0 < k' \leq k$)

✓ The *k’* robots execute their cycle L-C-M synchronously

Contribution

- ✓ Negative Results *(Also valid in the asynchronous model)*

  - *No (probabilistic or deterministic) algorithm exists to explore (with termination) any torus with less than 4 semi-synchronous robots*

  - *No deterministic algorithm exists to explore (with termination) any torus with less than 5 semi-synchronous robots*

- ✓ Positive Results

  - *Probabilistic semi-synchronous algorithm with $k = 4$ robots*

Negative Results

Definitions

❑ Node Multiplicity

➢ *Node contains 0, 1, or more robots*

➢ *Tower : More than one robots*

❑ (Weak) Multiplicity Detection

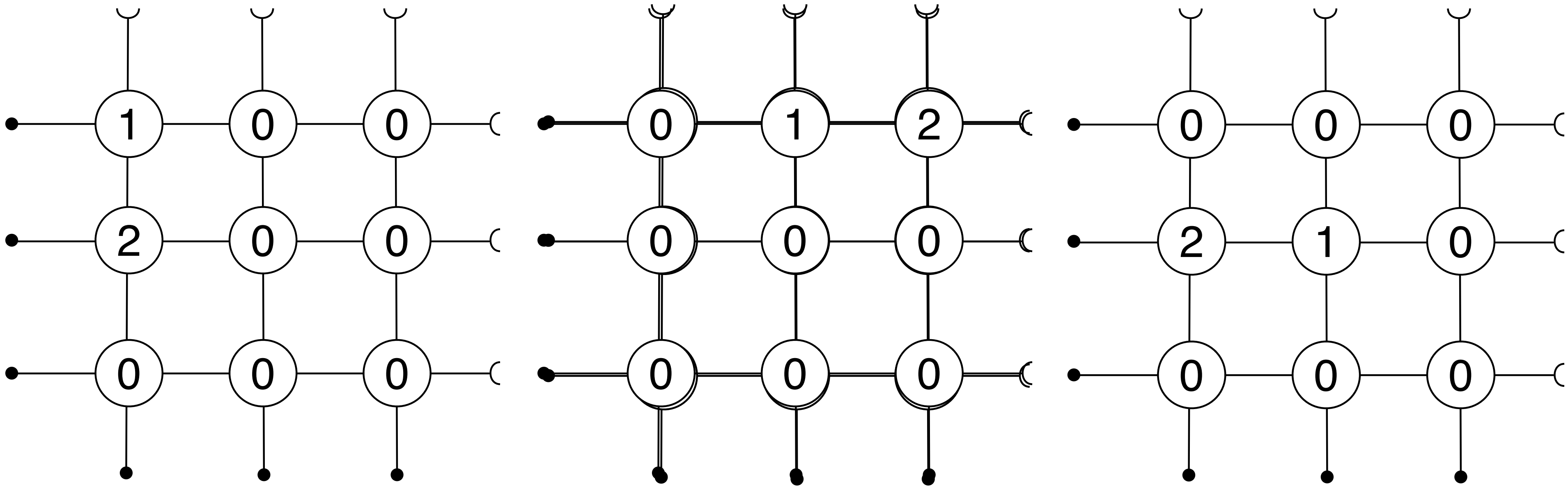
➢ *Ability to detect node multiplicity {0,1,T}*

❑ View

➢ *A labelled graph isomorphic to G, where each node is labelled with its multiplicity*

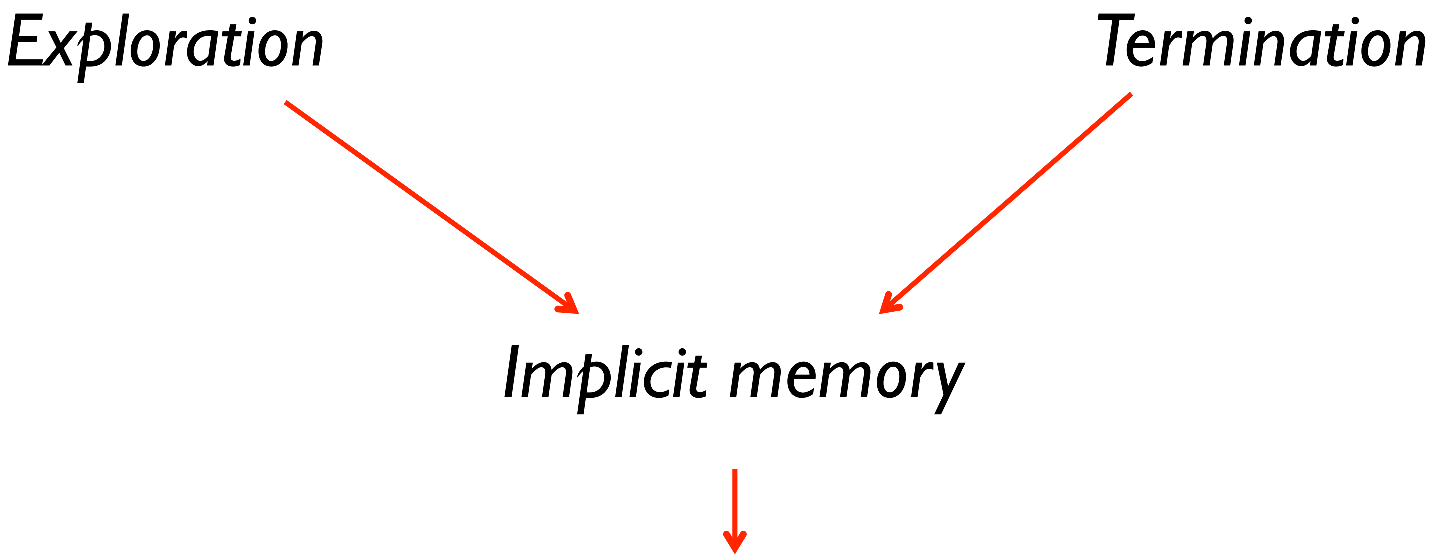
Definitions

- ❑ (Un)distinguishable configuration



(3-3)-torus

Oblivious robots



Exploration

Termination

Implicit memory

At least one configuration that is not an initial configuration

If $n > k$, any terminal configuration of any protocol contains at least one tower.

$k \geq 3$

If $n > k$, then there exists a set S of at least $n - k + 1$ configurations such that:

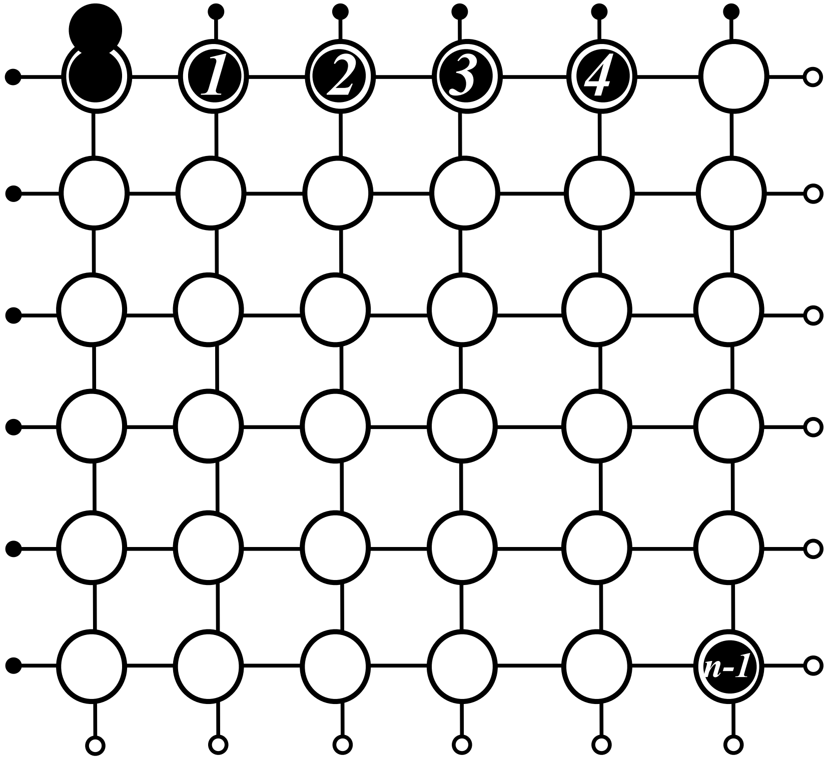
- $\forall\, c_1, c_2 \in S$: c_1 and c_2 are distinguishable
- $\forall\, c \in S$: there is a tower of less than k robots

[*Devismes et al.*, SIROCCO 2009] extended to arbitrary topologies

✓ Exploration requirement

- Distinction between visited and non visited nodes
- Memory of explored nodes encoded with configurations that contain at least one tower of less than k robots
- Fair sequential exploration implies at least $n - k + 1$ pairwise distinct configurations

$k \geq 3$



$k \geq 3$

If $n > k$, then there exists a set S of at least $n - k + 1$ configurations such that:

- $\forall$ $c_1, c_2 \in$ S: c_1 and c_2 are distinguishable
- $\forall$ $c \in$ S: there is a tower of less than k robots

[*Devismes et al.* , SIROCCO 2009] extended to arbitrary topologies

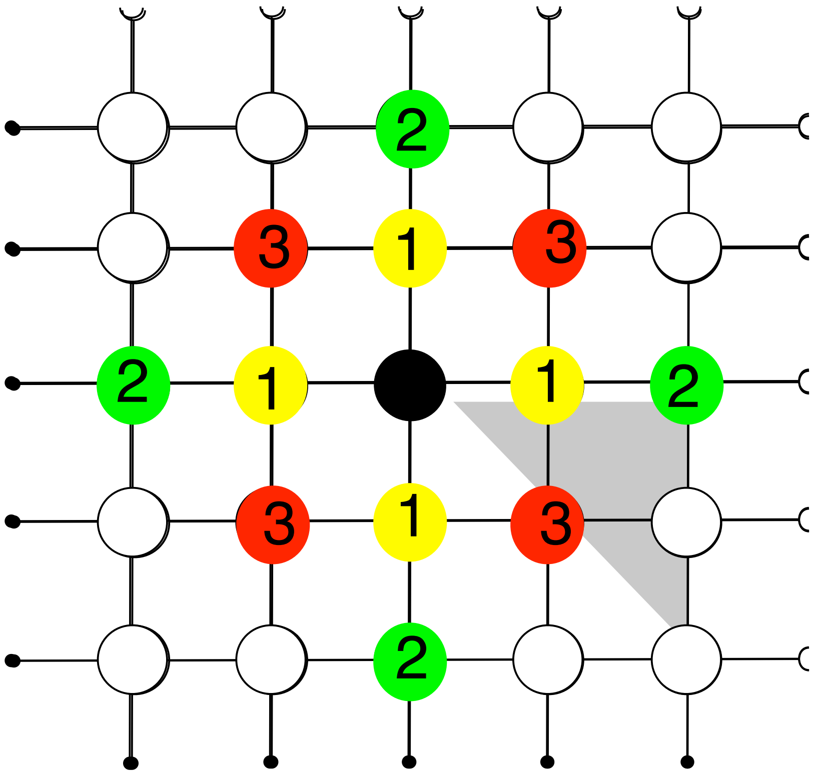
If $n > k$, then $k \geq 3$

k = 3

$3 \leq \text{њ} \leq L \Rightarrow n \geq 9$.

So, there must exist a set S of at least n - 2 configurations such that:

- ∀ $c_1, c_2 \in$ S: c_1 and c_2 are distinguishable
- ∀ c ∈ S: there is a tower of less than k robots



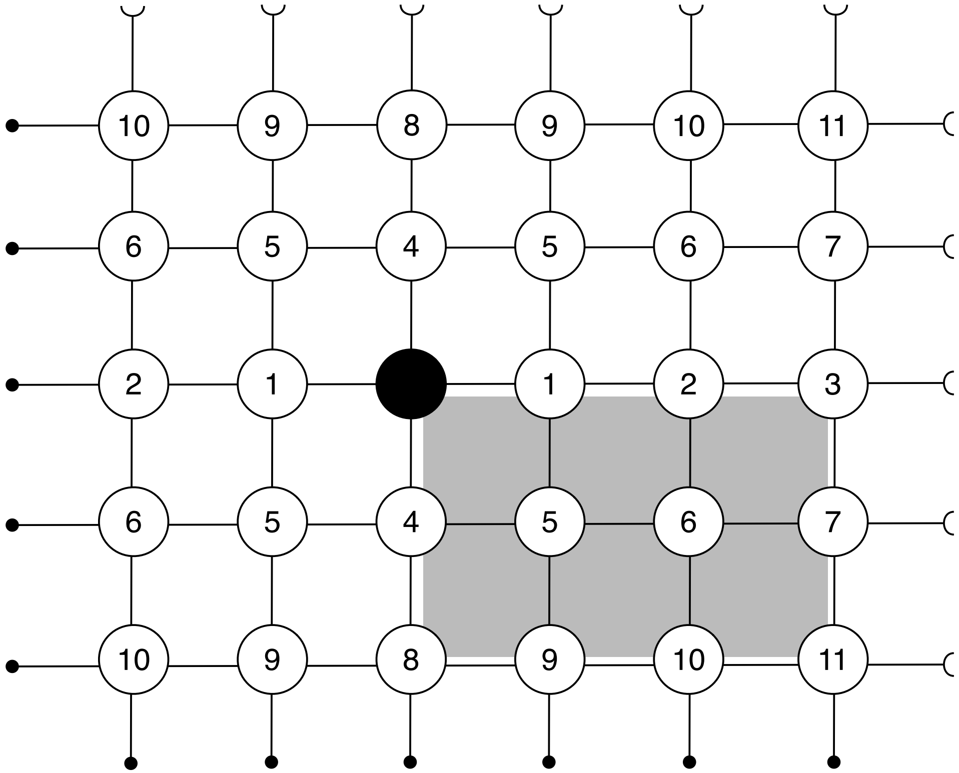
(5,5)-torus (n=25)

$k = 3$

$3 \leq \text{њ} \leq L \Rightarrow n \geq 9$.

So, there must exist a set S of at least $n - 2$ configurations such that:

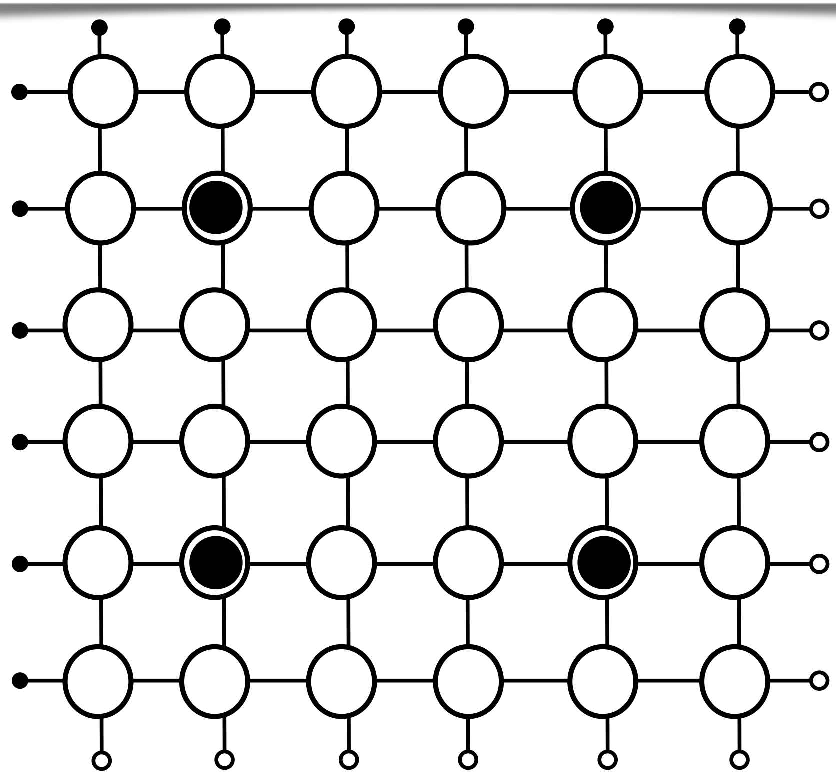
- $\forall$ $c_1, c_2 \in S$: c_1 and c_2 are distinguishable
- $\forall$ $c \in S$: there is a tower of less than k robots



(5,6)-torus (n=30)

$k \geq 4$

No (probabilistic or deterministic) algorithm exists to explore with termination any torus with less than 4 semi-synchronous robots.



$(2p,2p)$-torus ($p \geq 2$)

$k \geq 4$

No (probabilistic or deterministic) algorithm exists to explore with termination any torus with less than 4 semi-synchronous robots.

No deterministic algorithm exists to explore with termination any torus with less than 5 semi-synchronous robots.

Deterministic algorithm with specific initial configurations [D'Angelo *et al.*, ICDCN 2014]

Probabilistic Algorithm with 4 semi-synchronous robots

Algorithm

(l, L)-torus $(7 \leq l \leq L)$

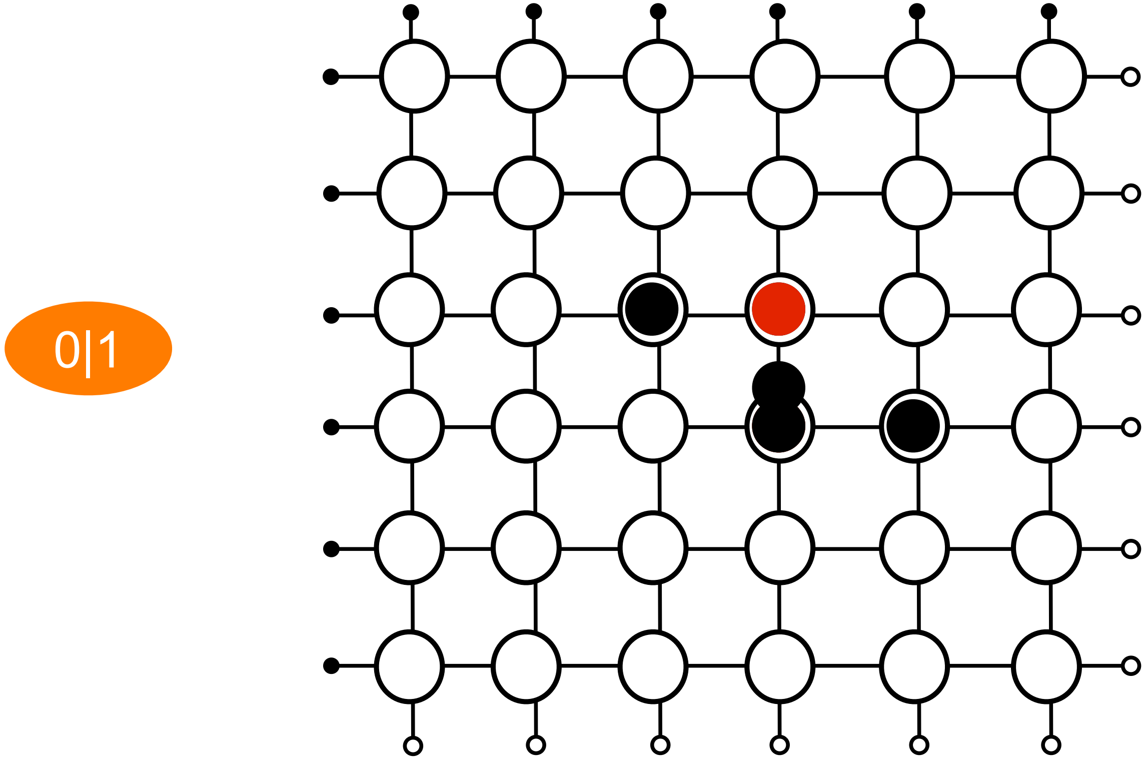
❏Phase 1: Set-Up

❏Phase 2: Tower

❏Phase 3: Exploration

Algorithm

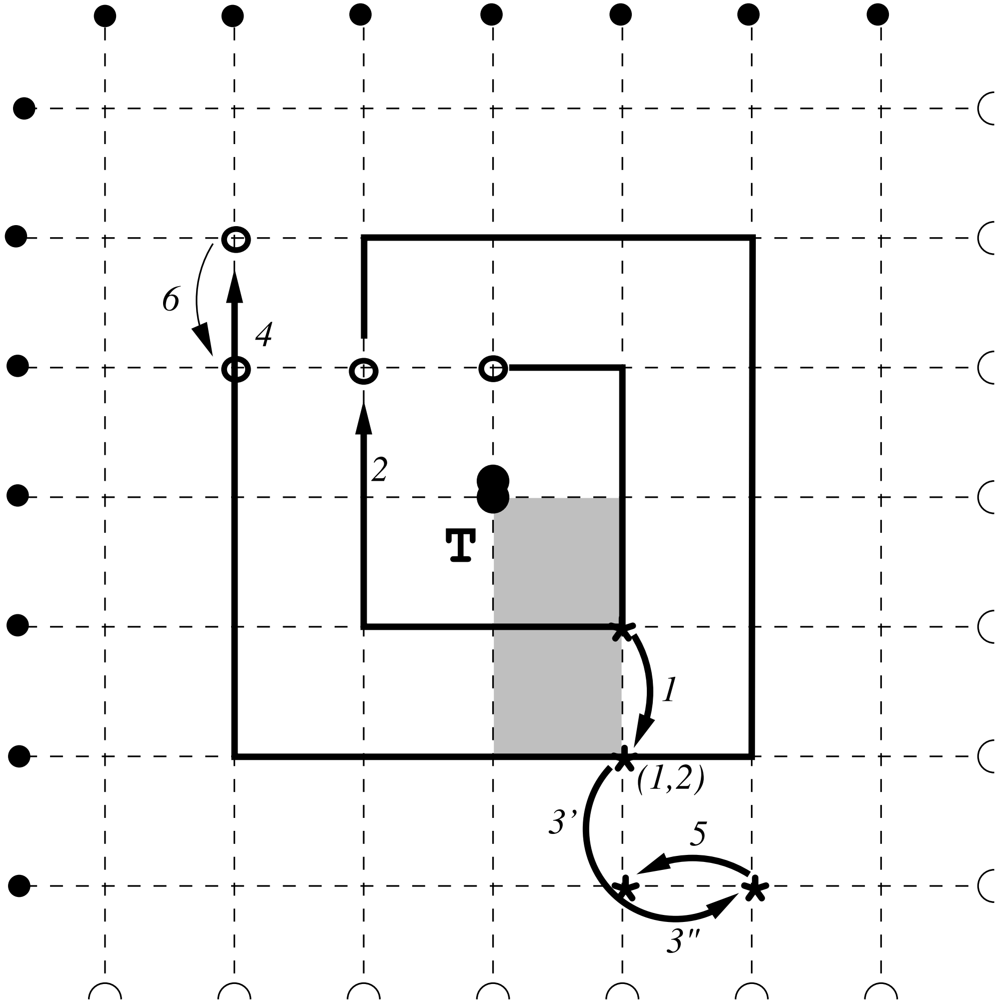
- ❑ Phase 2: Tower



◇ -configuration

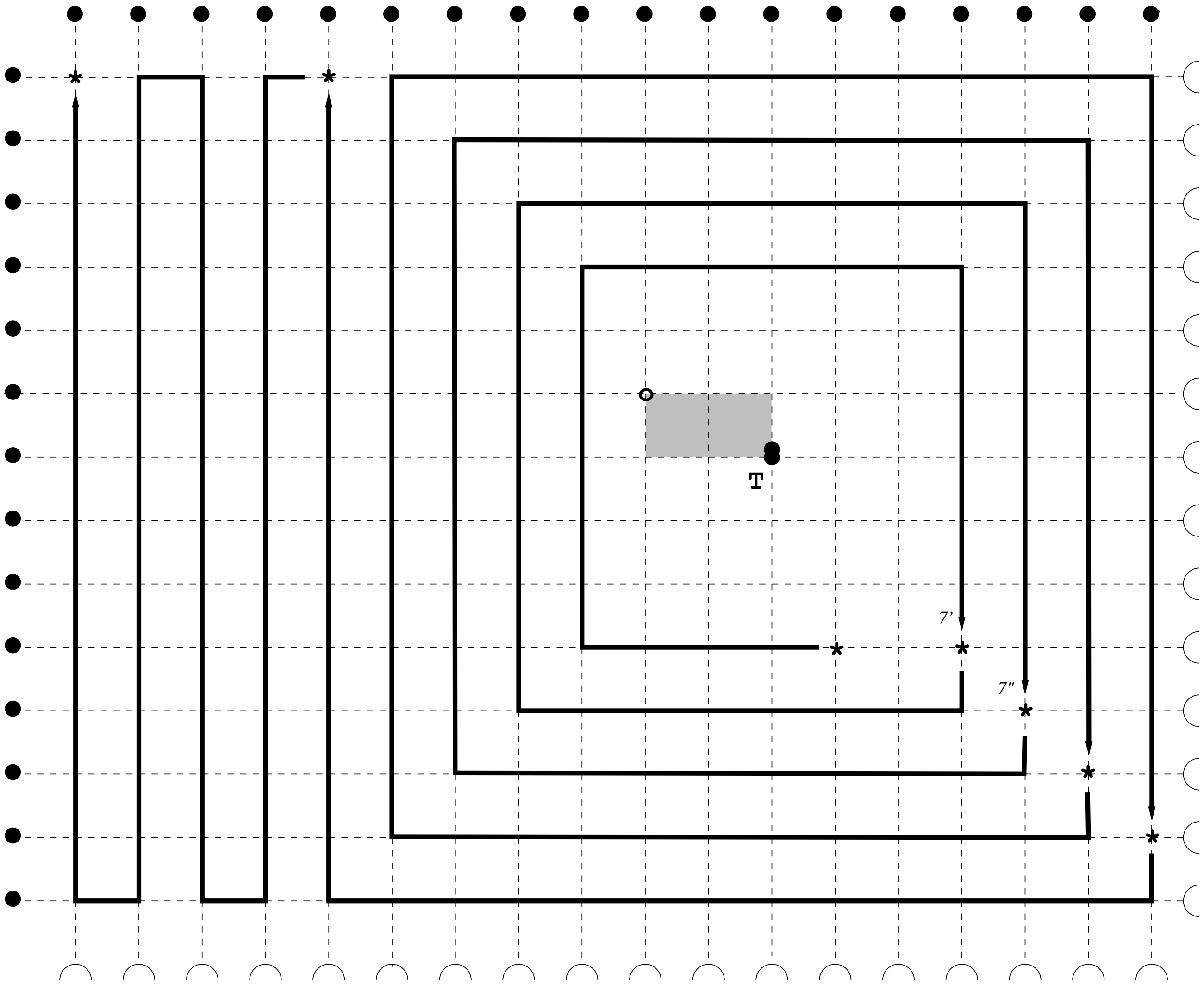
Algorithm

- ❑Phase 3: Exploration



Algorithm

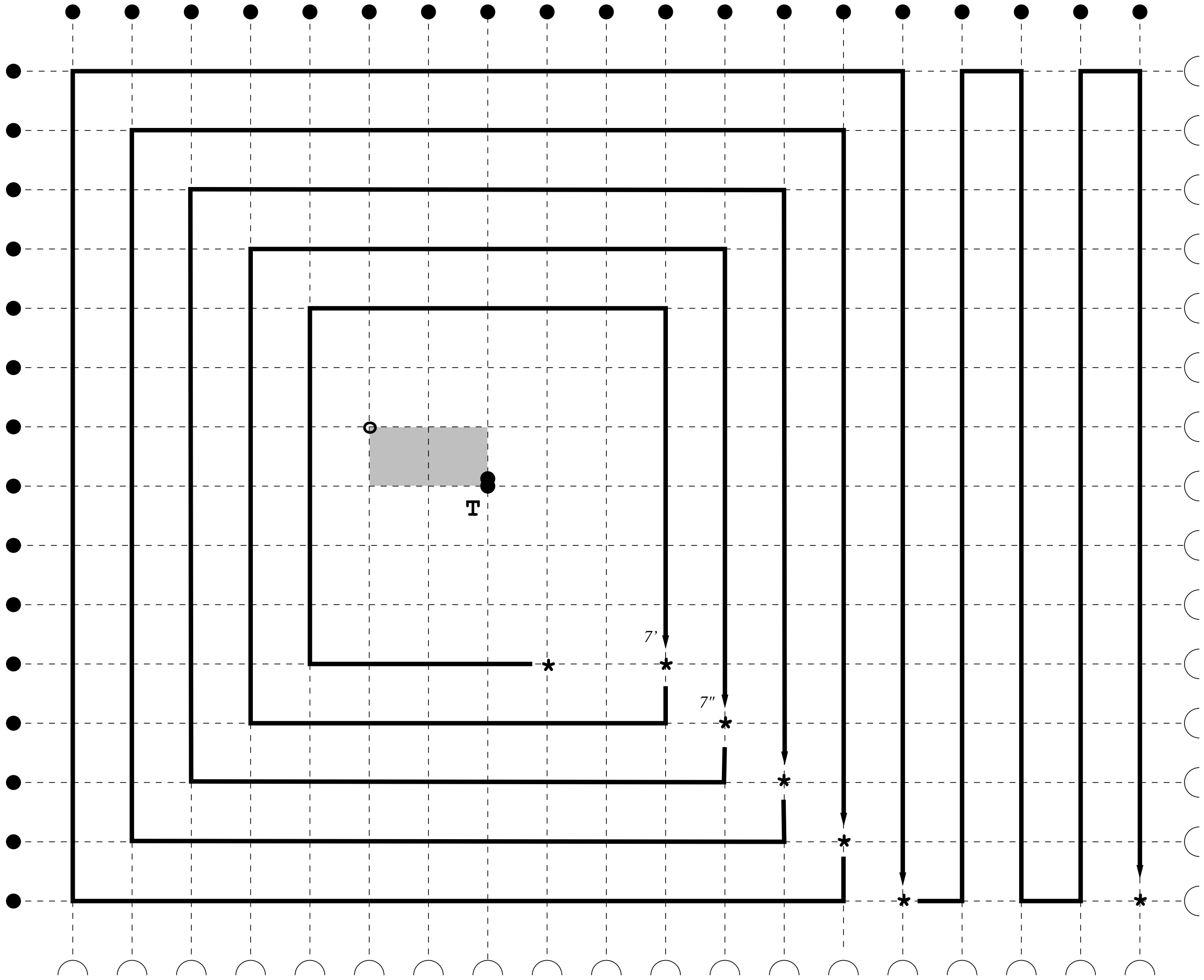
❑ Phase 3: Exploration



l is even

Algorithm

- ❑Phase 3: Exploration



l is odd

Algorithm

(l,L)-torus ($7 \leq l \leq L$)

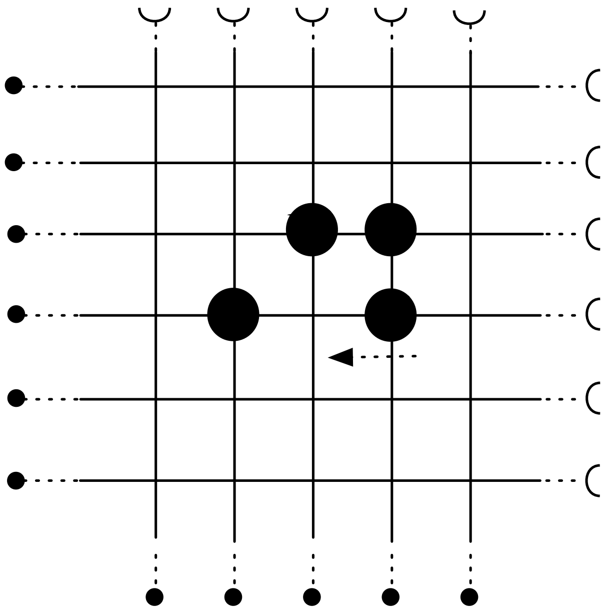
❑ Phase 1: Set-Up

❑ Phase 2: Tower

❑ Phase 3: Exploration

Algorithm

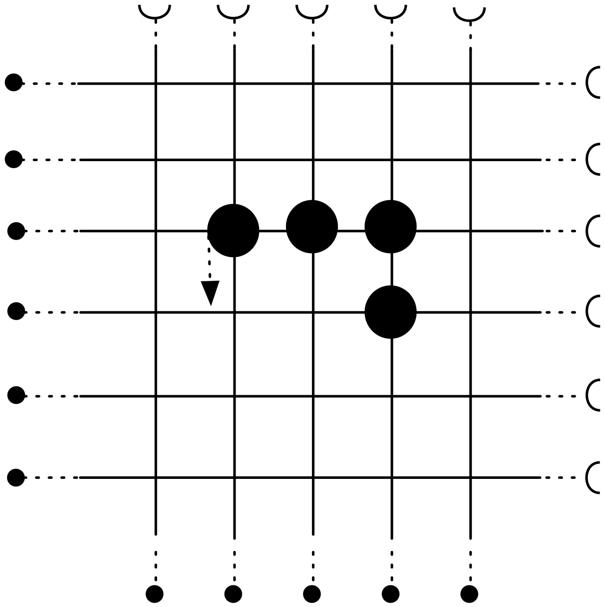
- ❑ Phase I: Set-Up → ◇-configuration
  - Double-Trap 2



Algorithm

❑ Phase 1: Set-Up → ◇-configuration

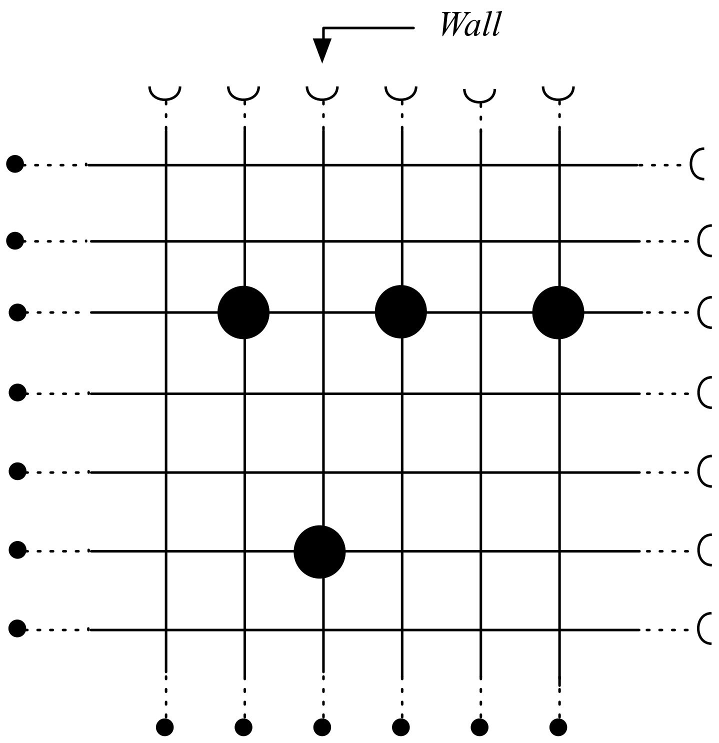
- Double-Trap 1



Algorithm

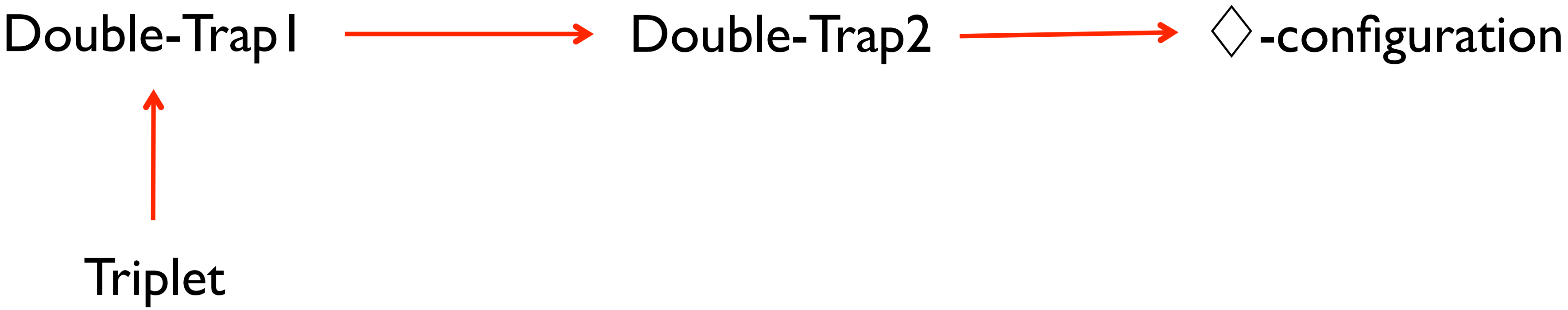
❑ Phase1: Set-Up → ◇-configuration

- Triplet: 3 robots belong to the same ring (≠ D-T 1)



Algorithm

❑Phase1: Set-Up → ◇-configuration



Double-Trap1

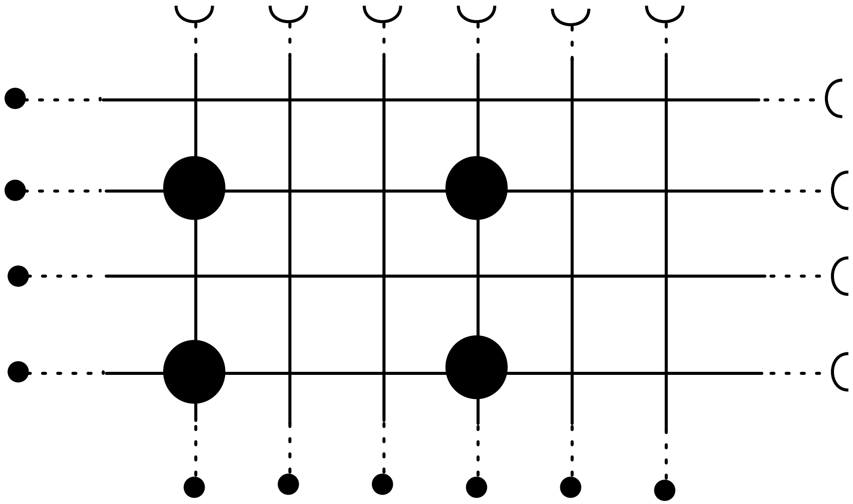
Double-Trap2

◇-configuration

Triplet

Algorithm

- ❑Phase 1: Set-Up → ◇-configuration
  - Regular: $\{r_1,r_2\}$ and $\{r_3,r_4\}$ *s.t.* r_1 (r_3) identical view as r_2 (resp. r_4) ($\neq$ ◇-configuration)

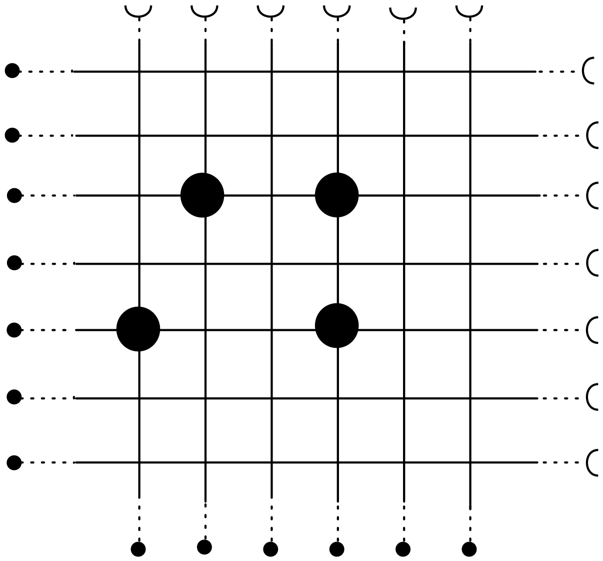


Particular case

Algorithm

❑ Phase 1: Set-Up → ◇-configuration

- Regular: $\{r_1,r_2\}$ and $\{r_3,r_4\}$ *s.t.* r_1 (r_3) identical view as r_2 (resp. r_4) ($\neq$ ◇-configuration)
- Twin: 2 robots belong to the same ring ($\neq$ D-T 1, D-T 2, Regular, ◇-configuration)



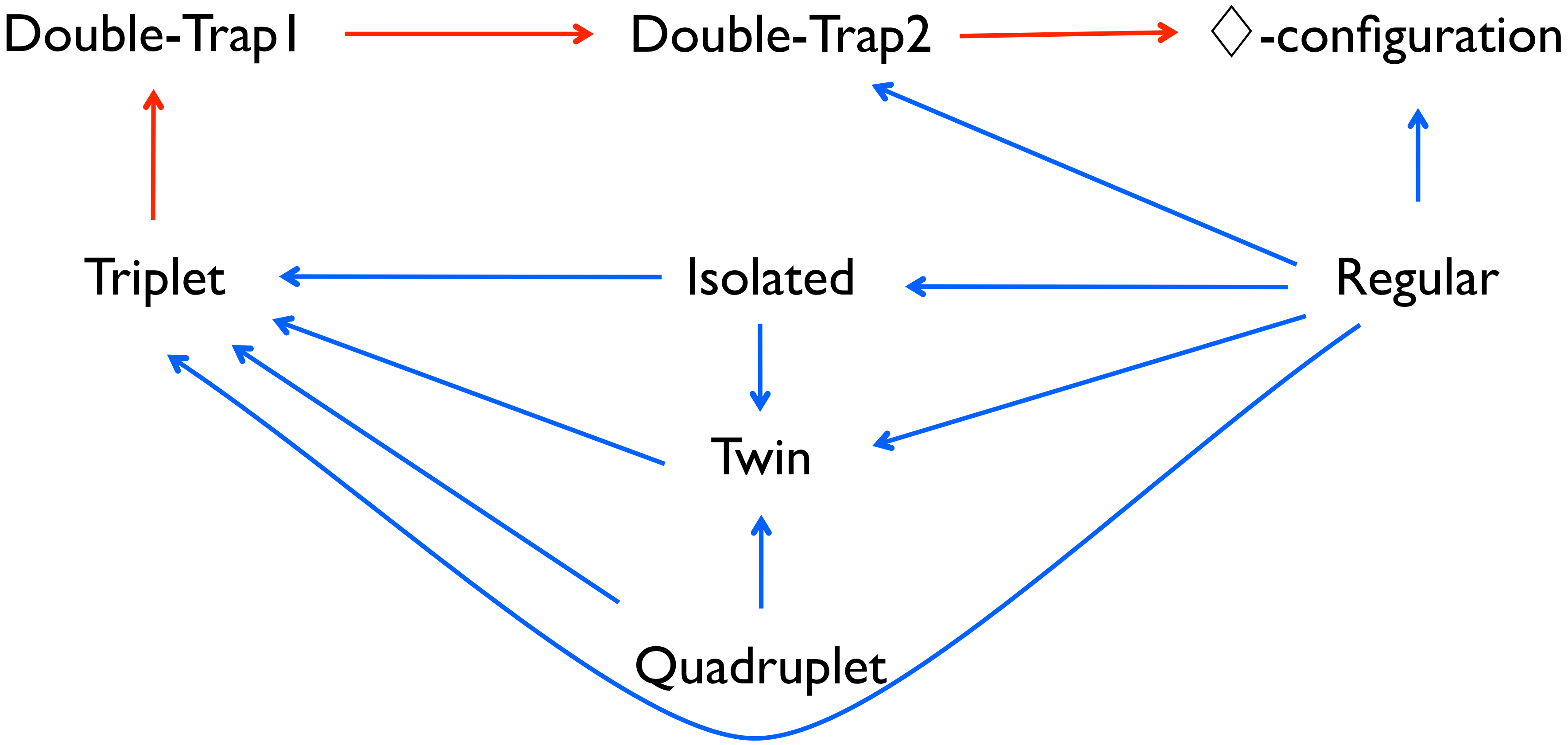
Algorithm

❑Phase1: Set-Up → $\Diamond$-configuration

- Regular: $\{r_1,r_2\}$ and $\{r_3,r_4\}$ *s.t.* r_1 (r_3) identical view as r_2 (resp. r_4) ($\neq$ $\Diamond$-configuration)
- Twin: 2 robots belong to the same ring ($\neq$ D-T 1, D-T 2, Regular, $\Diamond$-configuration)
- Quadruplet: 4 robots belong to the same ring ($\neq$ Regular)
- Isolated: 4 robots belong to different rings ($\neq$ Regular)

Algorithm

❑Phase1: Set-Up → ◇-configuration



Double-Trap1

Double-Trap2

◇-configuration

Triplet

Isolated

Regular

Twin

Quadruplet

Summary

- ✓ 4 probabilistic robots are necessary and sufficient for any (*l*,*L*)-torus ($7 \leq l \leq L$)
- ✓ No (probabilistic or deterministic) algorithm exists to explore (with termination) any torus with less than 4 semi-synchronous robots
- ✓ No deterministic algorithm exists to explore (with termination) any torus with less than 4 semi-synchronous robots

Extensions

- (*l*,*L*)-Tori *s.t.* 3 ≤ *l* ≤7 ?
- Deterministic solution ?
- Weaker Assumption (limited vision, *l*, L...) ?
- *Other regular topologies ?*
- *Fault-tolerance ?*