

Graph Exploration by Oblivious Agents

David ILCINKAS

CNRS, Bordeaux, France

Moving and Computing, La Maddalena, Italy

June 6, 2017

Problem

Model/context

- Team of mobile entities
 - **sensing** the environment by **taking a snapshot** of it
 - that **do not communicate**
 - that are **anonymous and oblivious**

• In a discrete environment: **anonymous graphs**

In this talk: **exploration**

- **Each node must be visited**

(Next talk: **gathering**)

Problem

Model/context

- Team of mobile entities
 - **sensing** the environment by **taking a snapshot** of it
 - that **do not communicate**
 - that are **anonymous and oblivious**
- In a discrete environment: **anonymous graphs**

In this talk: exploration

- Each node must be visited

(Next talk: gathering)

Problem

Model/context

- Team of mobile entities
 - **sensing** the environment by **taking a snapshot** of it
 - that **do not communicate**
 - that are **anonymous and oblivious**
- In a discrete environment: **anonymous graphs**

In this talk: exploration

- **Each node must be visited.**

(Next talk: gathering)

The Look-Compute-Move cycle

Look

The agent takes an **egocentric instantaneous snapshot** of the network and its agents

Compute

Based on this observation, it **decides to stay idle or to move to some neighboring node.**

Move

In the latter case it **instantaneously moves** towards its destination.

Identical oblivious asynchronous agents

Identical

Agents have **no IDs**. They execute the **same program**.

Oblivious

The agents have **no memory** of observations, computations and moves made in previous cycles.

Possibly asynchronous

The time between Look, Compute, and Move operations is **finite but unbounded**.

Reminder:

Non-communicating

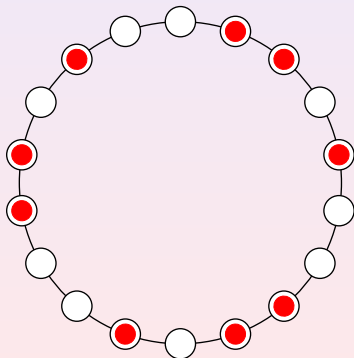
No communication mechanisms between agents, even locally.

Precisions concerning the model

Anonymous setting

- Agents and nodes have no IDs

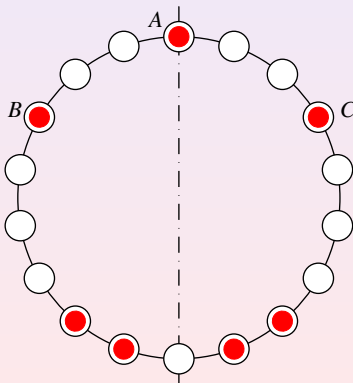
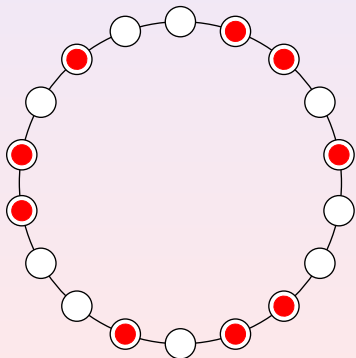
• There are no port numbers, and no chirality



Precisions concerning the model

Anonymous setting

- Agents and nodes have no IDs
- There are no port numbers, and **no chirality**



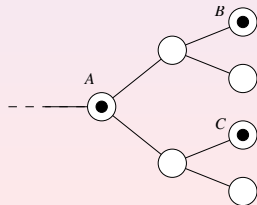
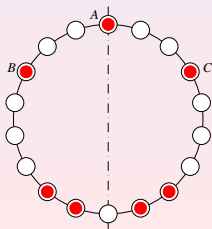
Precisions concerning the model

Anonymous setting

- Agents and nodes have no IDs
- There are no port numbers, and **no chirality**

In case of symmetry

- **Compute** : choice of an **equivalence class of neighbors**
- Actual choice: made by the adversary (i.e. worst case)



Precisions concerning the model

Anonymous setting

- Agents and nodes have no IDs
- There are no port numbers, and **no chirality**

In case of symmetry

- **Compute** : choice of an **equivalence class of neighbors**
- Actual choice: made by the adversary (i.e. worst case)

Weak multiplicity detection

“zero”, “one”, or “**more than one**” agents

Two types of exploration

Exploration with stop

- Each node must be visited by at least one agent.
- All agents must stop after finite time.

Exclusive perpetual exploration

- Each node must be visited by every agent infinitely often.
- Agents must not collide (meet at a node or cross each other on an edge).

In both cases, the initial configurations are the configurations without multiplicities.

Two types of exploration

Exploration with stop

- Each node must be visited by at least one agent.
- All agents must stop after finite time.

Exclusive perpetual exploration

- Each node must be visited by every agent infinitely often.
- Agents must not collide (meet at a node or cross each other on an edge).

In both cases, the initial configurations are the configurations without multiplicities.

Two types of exploration

Exploration with stop

- Each node must be visited by at least one agent.
- All agents must stop after finite time.

Exclusive perpetual exploration

- Each node must be visited by every agent infinitely often.
- Agents must not collide (meet at a node or cross each other on an edge).

In both cases, the initial configurations are the configurations without multiplicities.

Usual focus

n : number of nodes

Smallest exploring team

$\rho^-(n) = \text{Minimum}$ number of agents that can explore any n -node graph of a given family.

Largest exploring team

$\rho^+(n) = \text{Maximum}$ number of agents that can explore any n -node graph of a given family.

Families: Rings, (Partial) Grids, Trees, Tori, etc.

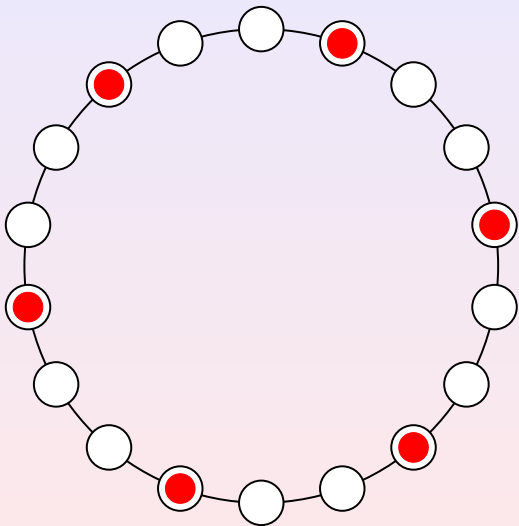
A simple impossibility result

Observation

Impossible for both
types of exploration
when k divides n .

n : number of nodes

k : number of agents



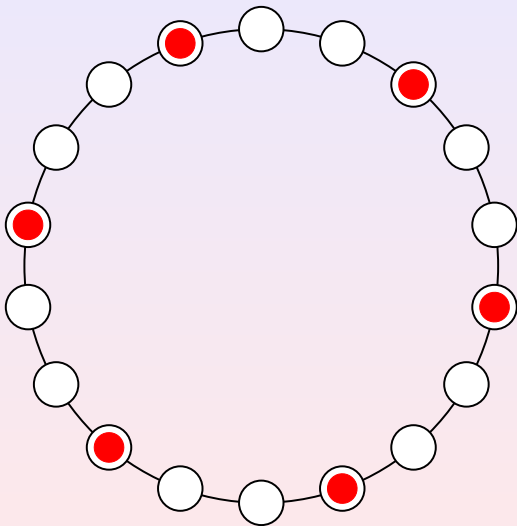
A simple impossibility result

Observation

Impossible for both types of exploration when k divides n .

n : number of nodes

k : number of agents



Focus on Exploration with stop

Exploration with stop (reminder)

- Each node must be visited by at least one agent.
- All agents must stop after finite time.
- Initial configurations are the configurations without multiplicities.

Family: Rings

Additional assumption

n and k are coprime.

- n : number of nodes
- k : number of agents

Focus on Exploration with stop

Exploration with stop (reminder)

- Each node must be visited by at least one agent.
- All agents must stop after finite time.
- Initial configurations are the configurations without multiplicities.

Family: Rings

Additional assumption

n and k are coprime.

- n : number of nodes
- k : number of agents

Some additional definitions

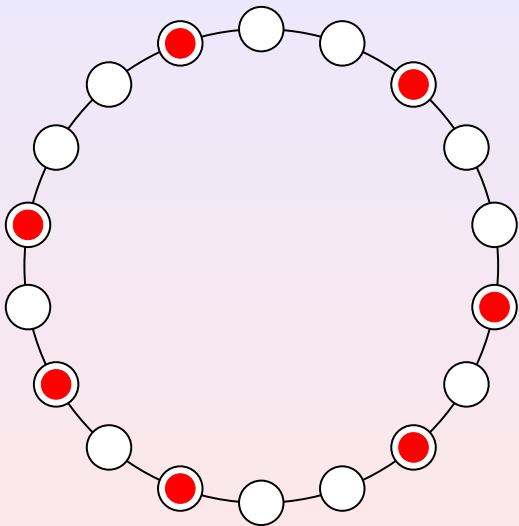
Interdistance

Minimum distance taken over all pairs of distinct agents.

Here interdistance=2.

Block

Maximal set of agents, of size at least 2, forming a line with an agent every d nodes. (d =interdistance)



Some additional definitions

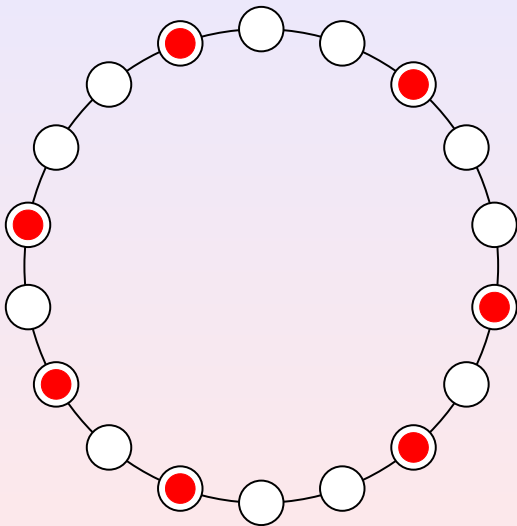
Interdistance

Minimum distance taken over all pairs of distinct agents.

Here interdistance=2.

Block

Maximal set of agents, of size at least 2, forming a line with an agent every d nodes. (d =interdistance)



Some additional definitions

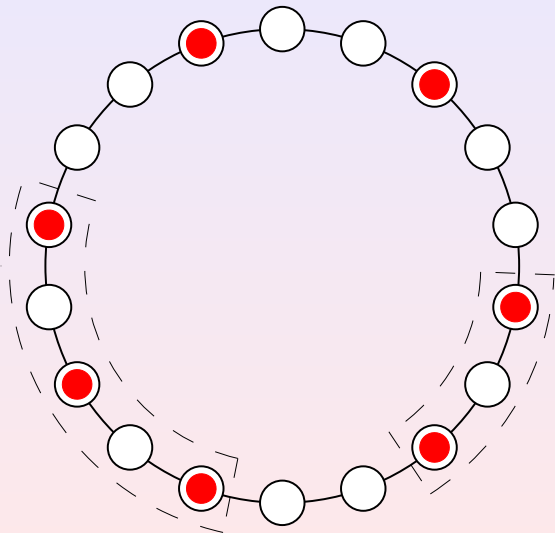
Interdistance

Minimum distance taken over all pairs of distinct agents.

Here interdistance=2.

Block

Maximal set of agents, of size at least 2, forming a line with an agent every d nodes. (d =interdistance)



The algorithm

Set-Up Phase

Goal: to transform the (arbitrary) initial configuration into a configuration of interdistance 1 where there is a single block or two blocks of the same size.

Method: decrease the number of blocks whenever possible. Otherwise, decrease the interdistance.

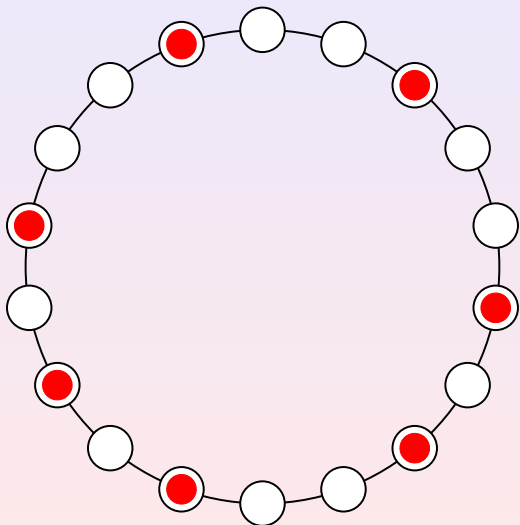
Tower-Creation Phase

Goal: to create one or two multiplicities (towers) inside each block; furthermore a number of agents become uniquely identified as explorers.

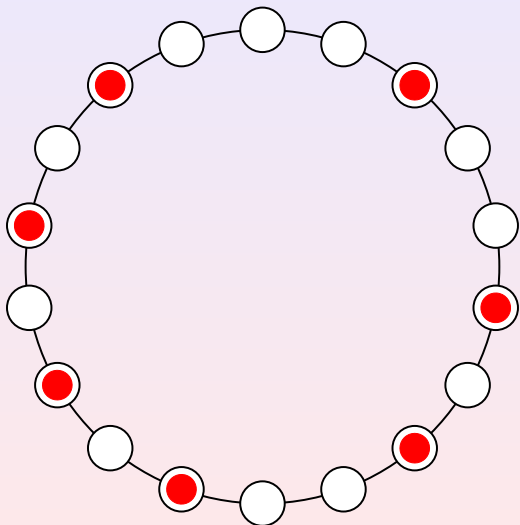
Exploration Phase

Goal: to perform exploration thanks to the explorers until reaching an identified final configuration.

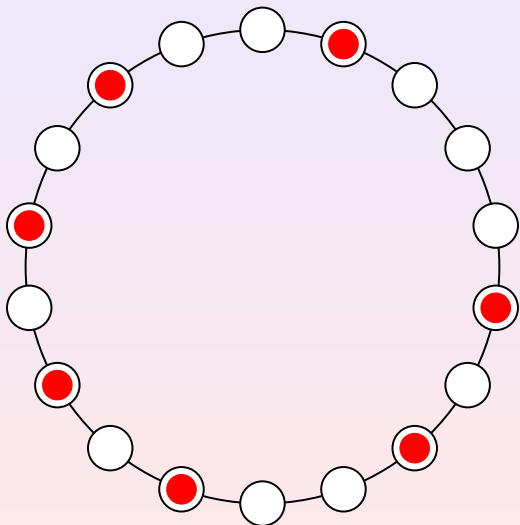
An example of execution



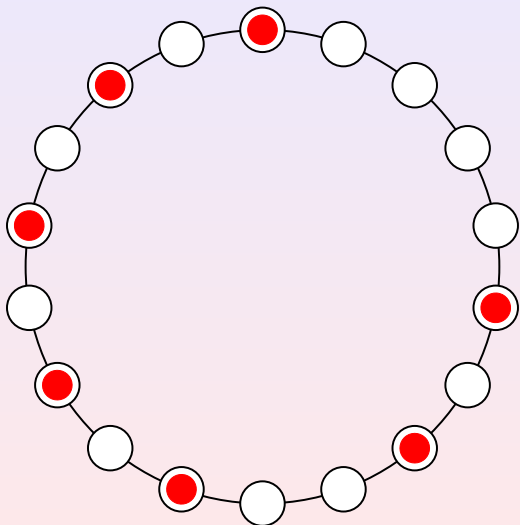
An example of execution



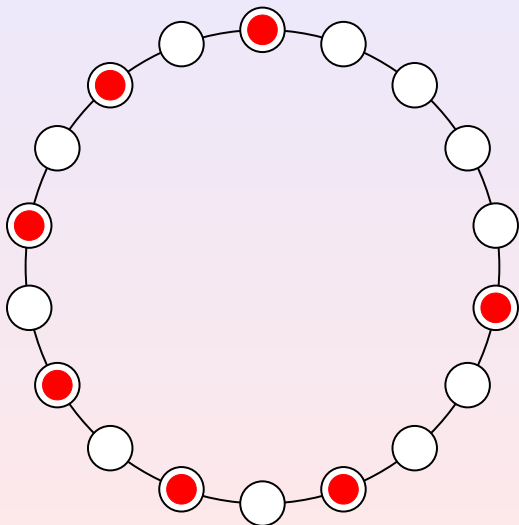
An example of execution



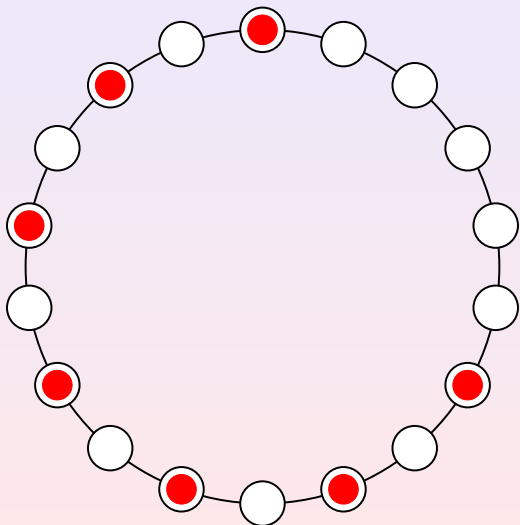
An example of execution



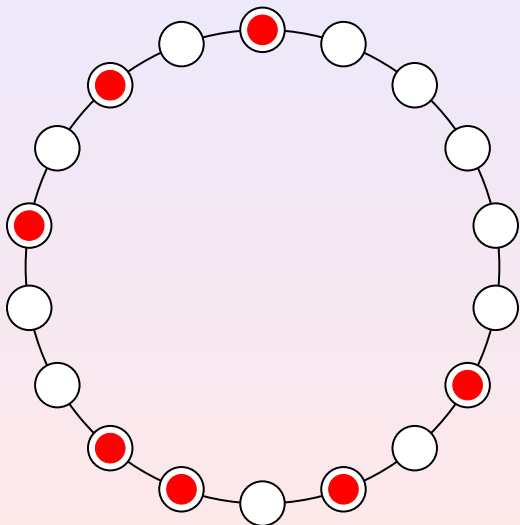
An example of execution



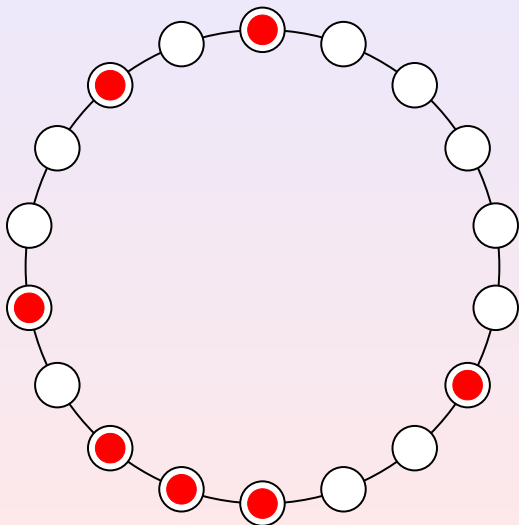
An example of execution



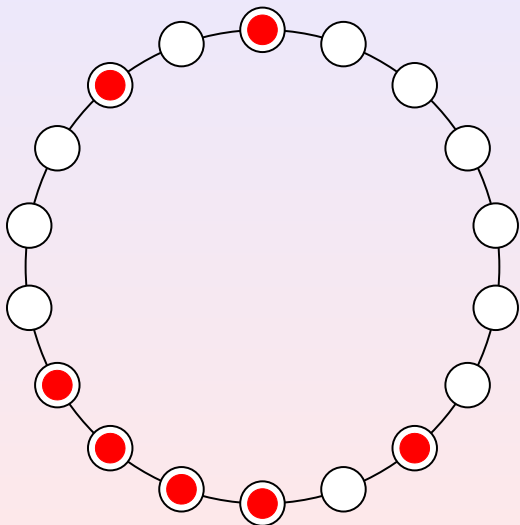
An example of execution



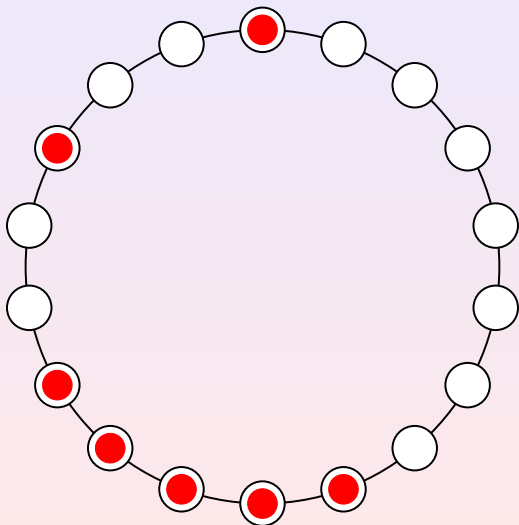
An example of execution



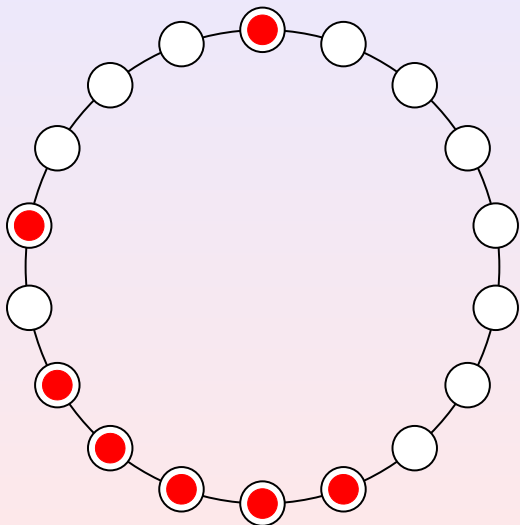
An example of execution



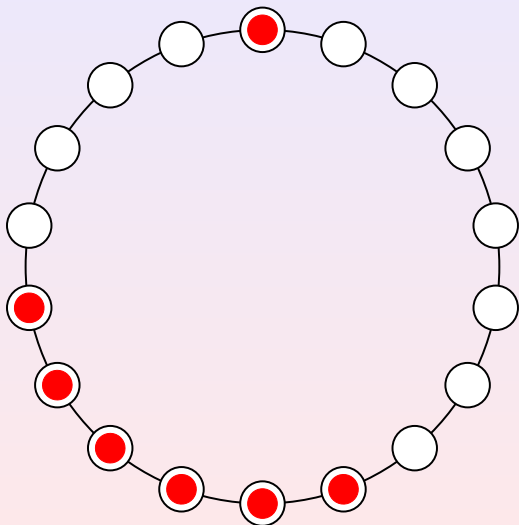
An example of execution



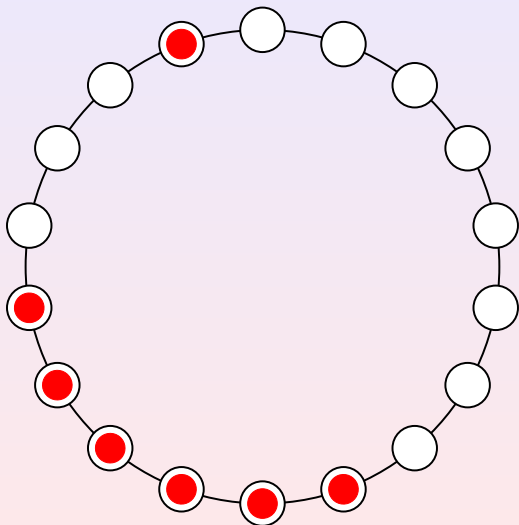
An example of execution



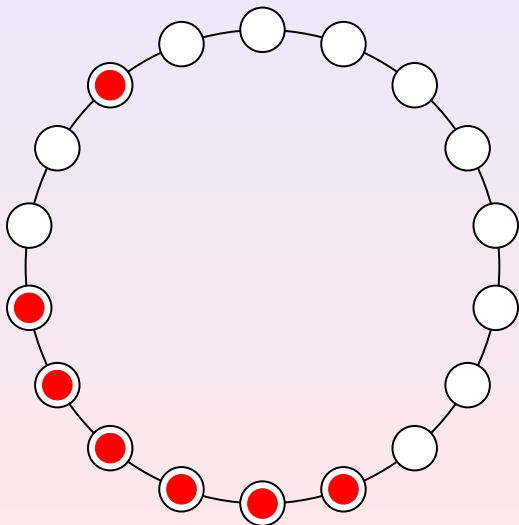
An example of execution



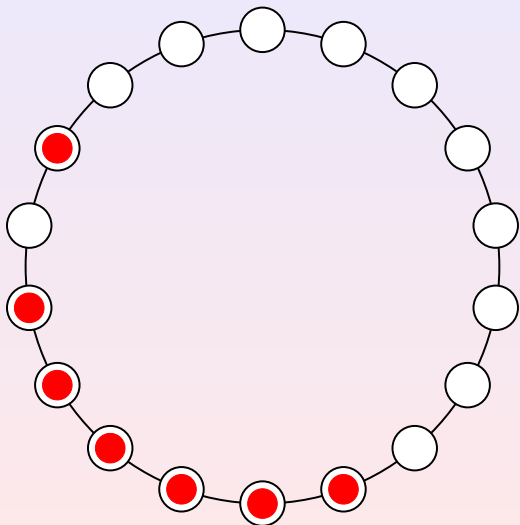
An example of execution



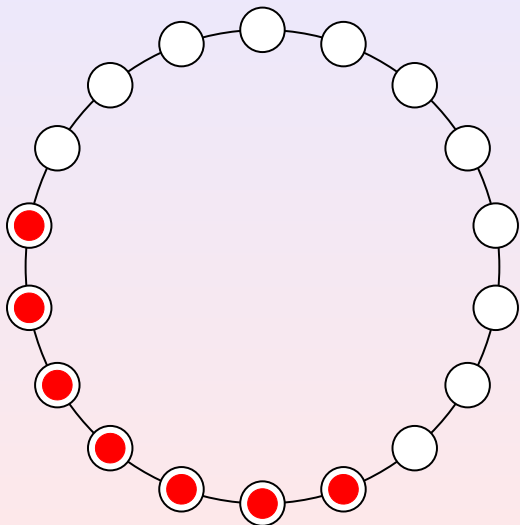
An example of execution



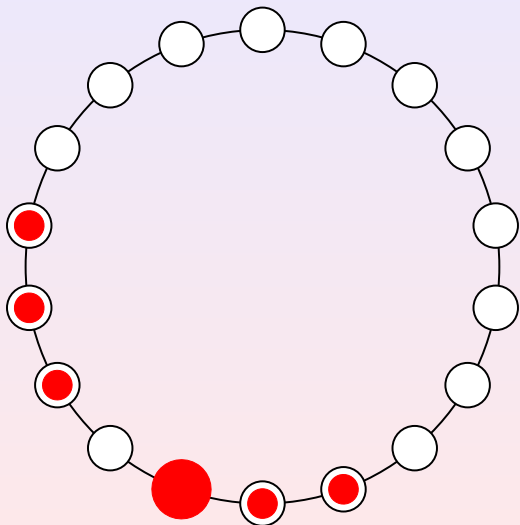
An example of execution



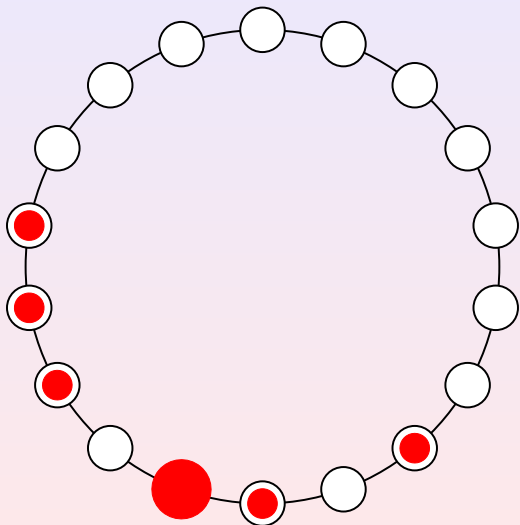
An example of execution



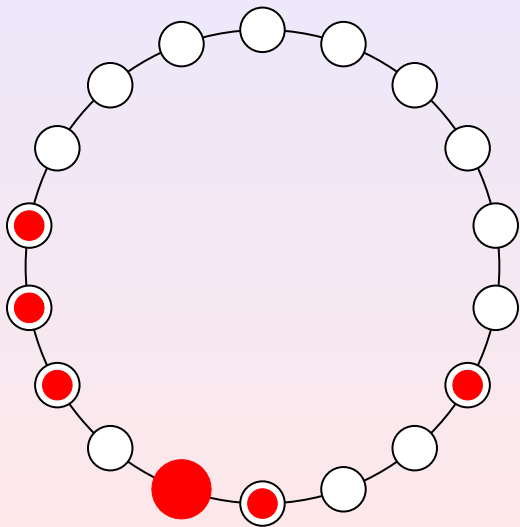
An example of execution



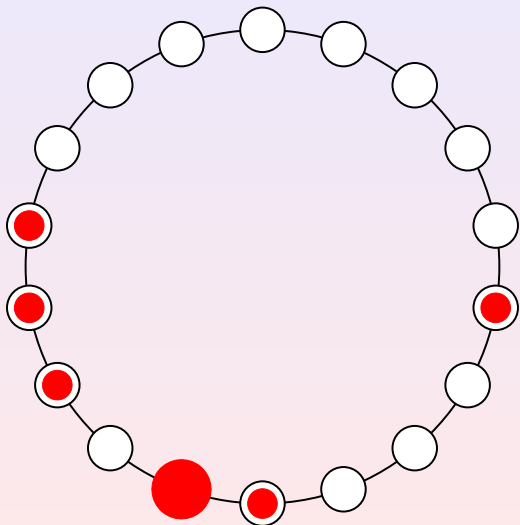
An example of execution



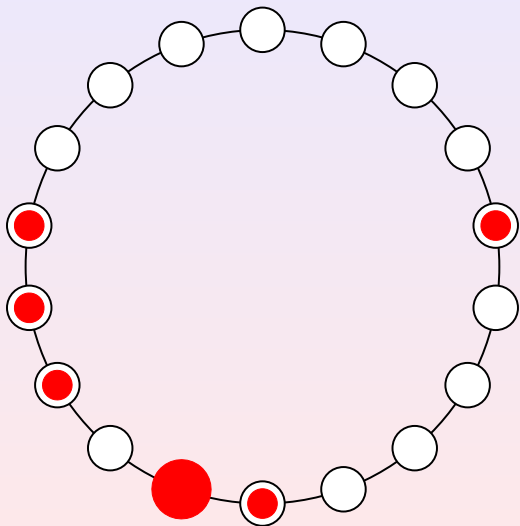
An example of execution



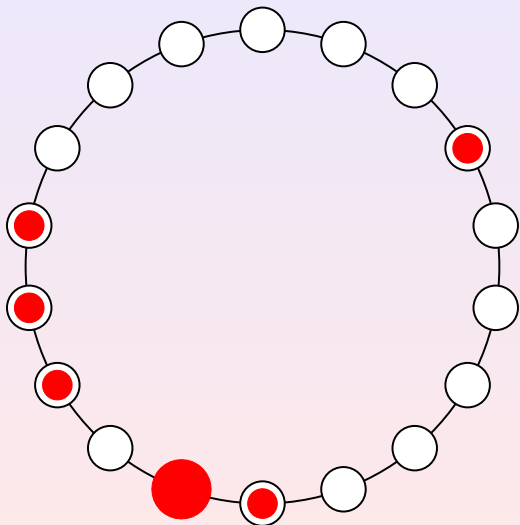
An example of execution



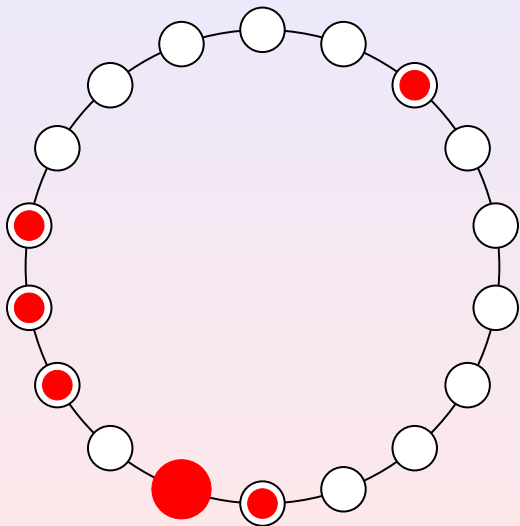
An example of execution



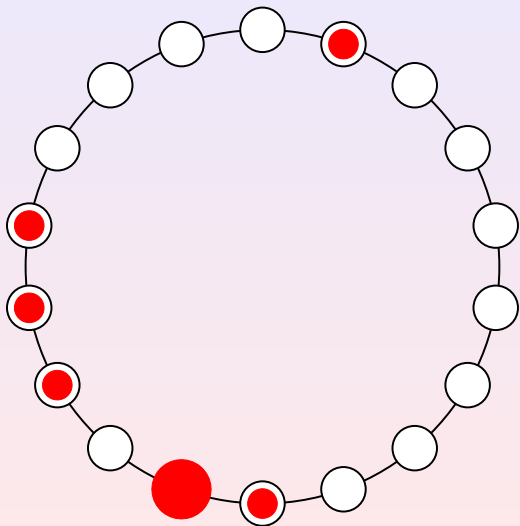
An example of execution



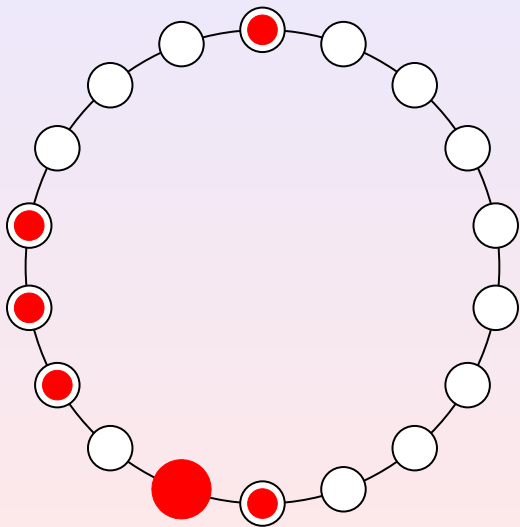
An example of execution



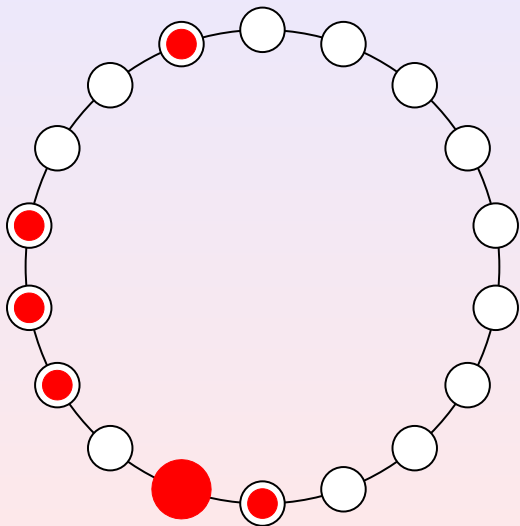
An example of execution



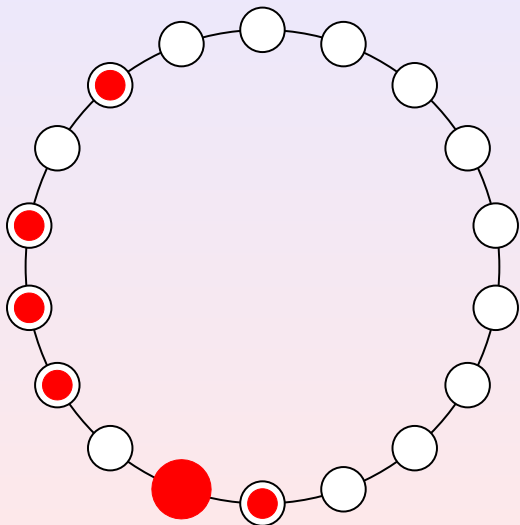
An example of execution



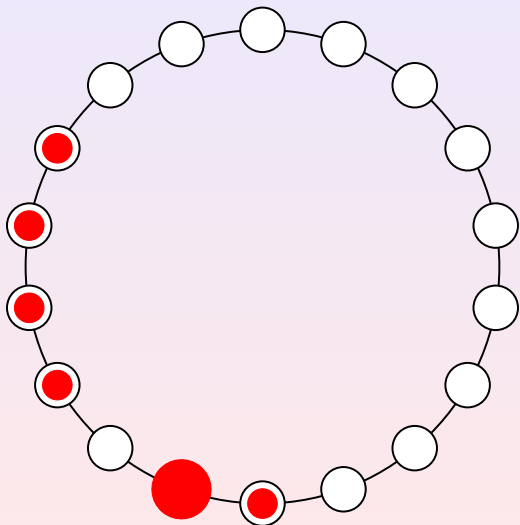
An example of execution



An example of execution



An example of execution



Algorithmic techniques (exploration with stop)

Phase 1

Form a **special configuration without multiplicities**.

- Uses (co-)primality of n and k .

Phase 2

Create a multiplicity to mark beginning of exploration.

Phase 3

Perform exploration by few (1–3) agents.

- Uses multiplicity and other agents to
 - break symmetry
 - keep track of the progress

But symmetry and asynchrony

Algorithmic techniques (exploration with stop)

Phase 1

Form a **special configuration without multiplicities**.

- Uses (co-)primality of n and k .

Phase 2

Create a **multiplicity** to mark **beginning of exploration**.

Phase 3

Perform exploration by few (1–3) agents.

- Uses multiplicity and other agents to
 - break symmetry
 - keep track of the progress

But symmetry and asynchrony

Algorithmic techniques (exploration with stop)

Phase 1

Form a **special configuration without multiplicities**.

- Uses (co-)primality of n and k .

Phase 2

Create a **multiplicity** to mark **beginning of exploration**.

Phase 3

Perform exploration by few (1–3) agents.

- Uses multiplicity and other agents to
 - break symmetry
 - keep track of the progress

But symmetry and asynchrony

Models of synchrony

FSYNC (fully synchronous)

All agents execute their Look-Compute-Move **simultaneously**.

SSYNC (semi-synchronous)

Look-Compute-Move is **atomic**.

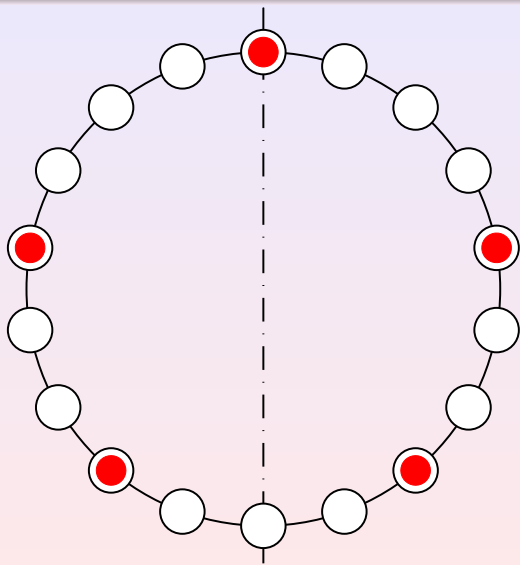
⇒ Difficulties to move multiplicities

ASYNC (asynchronous)

The time between Look, Compute, and Move operations is **finite but unbounded**.

⇒ Pending moves (moves based on outdated snapshots)

Symmetries and asynchrony



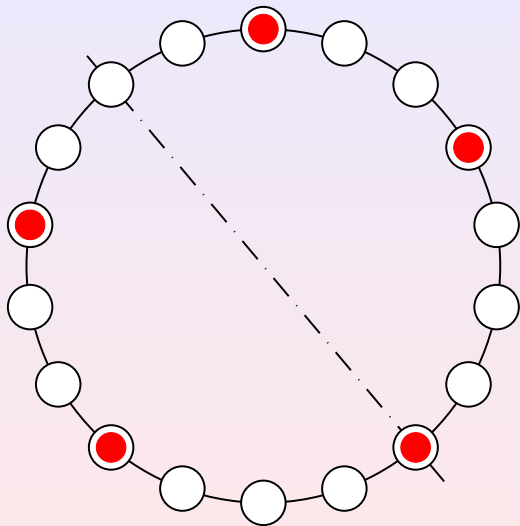
Guideline 1

Move only one agent at a time if possible.

Guideline 2

If the current configuration could come from a symmetric configuration, then move the "risky" agent first.

Symmetries and asynchrony



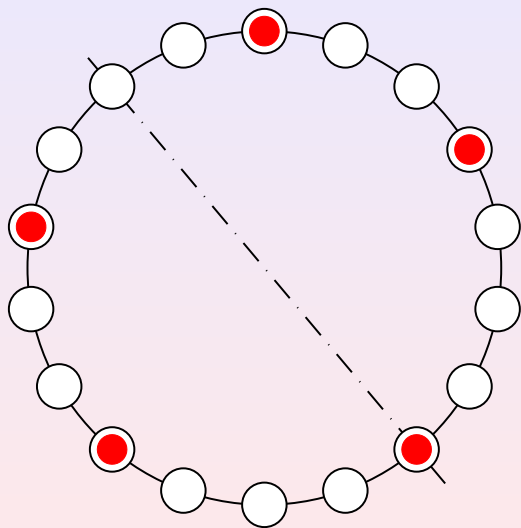
Guideline 1

Move only one agent at a time if possible.

Guideline 2

If the current configuration could come from a symmetric configuration, then move the "risky" agent first.

Symmetries and asynchrony



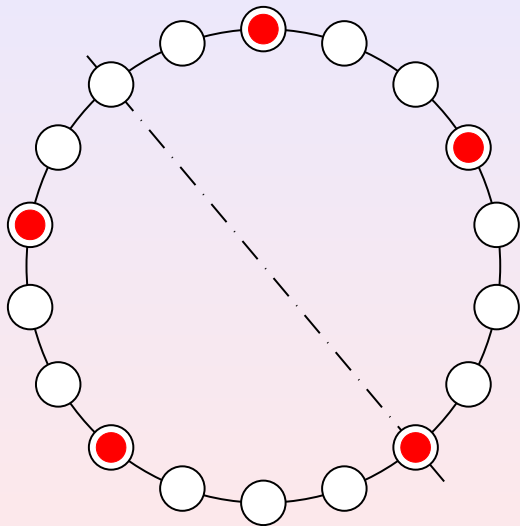
Guideline 1

Move only **one agent at a time** if possible.

Guideline 2

If the current configuration could come from a symmetric configuration, then move the 'risky' agent first.

Symmetries and asynchrony



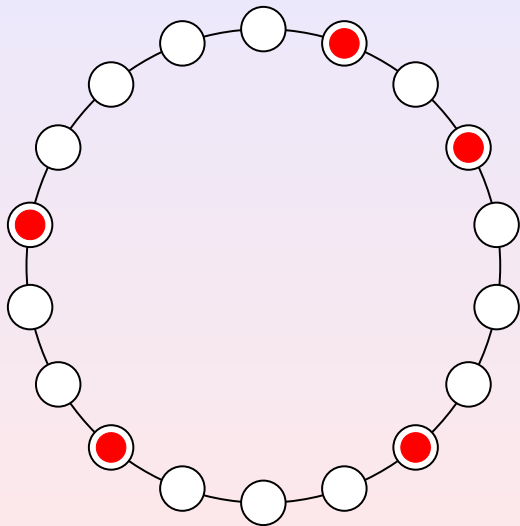
Guideline 1

Move only **one agent at a time** if possible.

Guideline 2

If the current configuration could come from a symmetric configuration, then **move the "risky" agent first**.

Symmetries and asynchrony



Guideline 1

Move only **one agent at a time** if possible.

Guideline 2

If the current configuration could come from a symmetric configuration, then **move the "risky" agent first**.

Known results (explo. with stop) 1/3

In the rings

- [1] ASYNC det.: $\rho^-(n)$ is in $O(\log n)$
- [1] SSYNC det.: $\rho^-(n)$ is in $\Omega(\log n)$ for inf. many n
- [2] SSYNC det.: $\rho^-(n) = 4$ when n is odd.
- [2] SSYNC det.: $\rho^-(n) \geq 5$ if n is even
- [2] ASYNC det.: $\rho^-(n) \leq 5$ when n is not a multiple of 5
- [3] ASYNC prob.: $\rho^-(n)$ is in $\Omega(\log n)$ for inf. many n
- [3] SSYNC prob.: $\rho^-(n) = 4$ for any $n > 4$

[1] Flocchini, Ilcinkas, Pelc, Santoro. Algorithmica 2013

[2] Lamani, Potop-Butucaru, Tixeuil. SIROCCO 2010

[3] Devismes, Petit, Tixeuil. TCS 2013

Known results (explo. with stop) 2/3

Limited visibility: snapshot up to distance φ only
[Datta, Lamani, Larmore, Petit]

In the rings, with $\varphi = 1$ (ICDCS 2013)

- SSYNC det.: impossible
- FSYNC det.: $\rho^-(n) \geq 5$
- FSYNC det.: algo. $k = 5$ when initial config. is special

In the rings, with $\varphi = 2$ and 3 (IPDPS workshop 2015)

For $\varphi = 3$:

- ASYNC det.: $\rho^-(n) \leq 7$ for any $n > 21$

For $\varphi = 2$:

- ASYNC det.: algo. $k = 7$ when initial config. is special

Known results (explo. with stop) 3/3

[Flocchini, Ilcinkas, Pelc, Santoro. TCS 2010]

- **Trees max. deg. 3:** $\rho^-(n) \in \Theta(\log n / \log \log n)$
- **Complete ternary trees:** $\rho^-(n) \in \Theta(n)$

[Flocchini, Ilcinkas, Pelc, Santoro. IPL 2011]

- **Line:** iff $k = 3$, $k \geq 5$, or $k = 4$ and n odd

[Devismes, Lamani, Petit, Raymond, Tixeuil. SSS 2012]

- **Grid:** $\rho^-(n) = 3$ (SSYNC/ASYNC)

[Devismes, Lamani, Petit, Tixeuil. NETYS 2015]

- **Torus:** SSYNC $\rho^-(n) = 4$ prob. (≥ 5 det.)

[Chalopin, Flocchini, Mans, Santoro. WG 2010]

- **Arbitrary graphs** with port numbers: rigid initial config.

Focus on Exclusive Perpetual Exploration

Exclusive perpetual exploration (reminder)

- **Each** node must be visited by **every** agent **infinitely often**.
- Agents must **not collide** (meet at a node or cross each other on an edge).
- **Initial configurations** are the configurations without multiplicities.

First studied family: Partial grids with NSEW

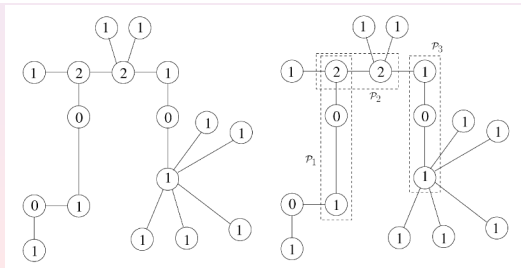
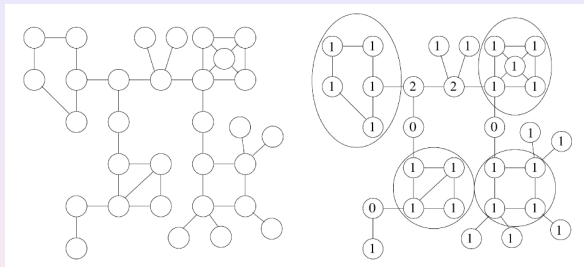
Focus on Exclusive Perpetual Exploration

Exclusive perpetual exploration (reminder)

- **Each** node must be visited by **every** agent **infinitely often**.
- Agents must **not collide** (meet at a node or cross each other on an edge).
- **Initial configurations** are the configurations without multiplicities.

First studied family: Partial grids with NSEW

Mobility tree and parameter q



Known results (exclusive perpetual explo.) 1/2

In the partial grids (FSYNC)

[Baldoni, Bonnet, Milani, Raynal. IPL 2008]

- Definition of the parameter q
- Solvable iff $k \leq n - q$

[Baldoni, Bonnet, Milani, Raynal. OPODIS 2008]

- Limited visibility $\varphi = 1$
- Solvable iff $k \leq n - q$, except for $q = 0 \implies n - 1$

In the (complete) grids

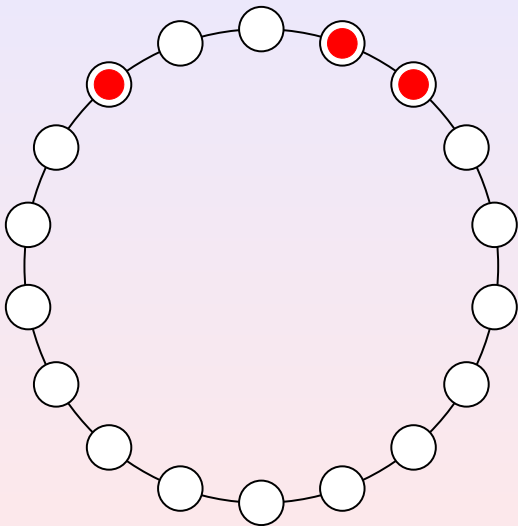
[Bonnet, Milani, Potop-Butucaru, Tixeuil. OPODIS 2011]

- ASYNC, no sense of direction
- Deterministically: $\rho^-(n) = 3$

Algorithmic techniques (exclusive perpetual explo.)

Two phases

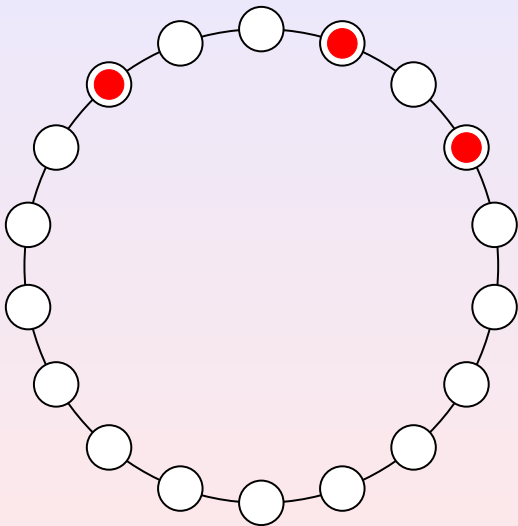
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

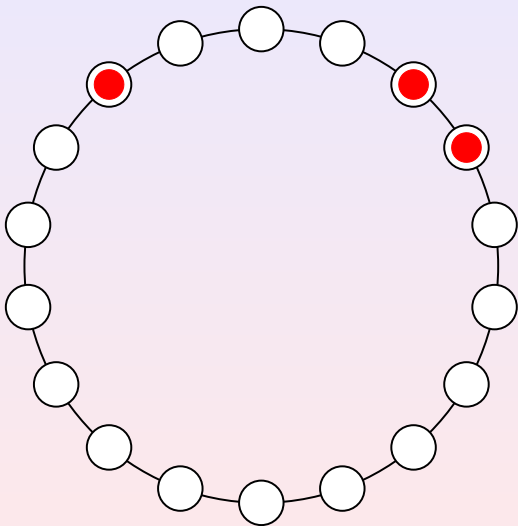
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

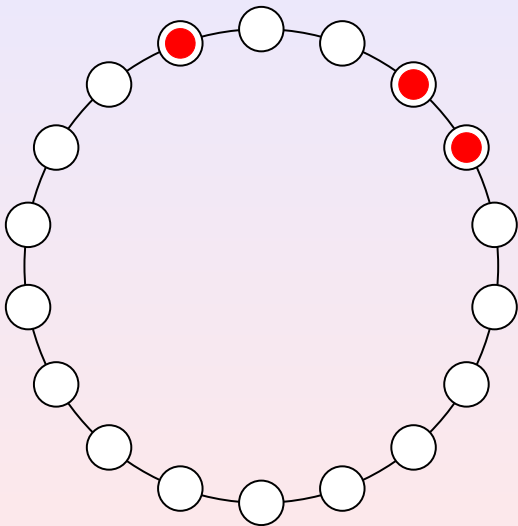
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

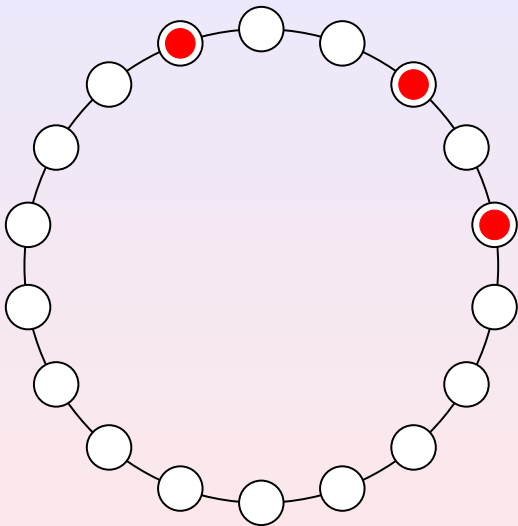
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

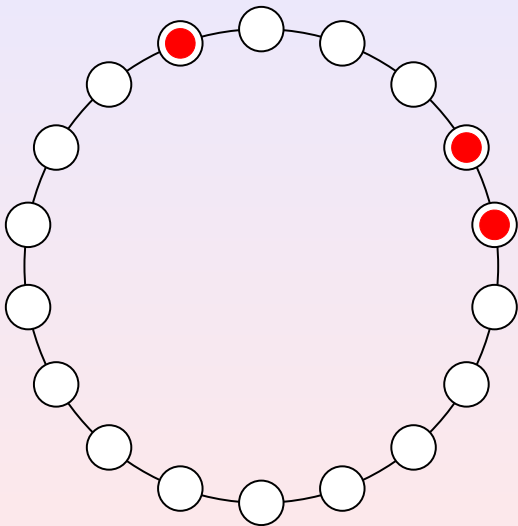
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

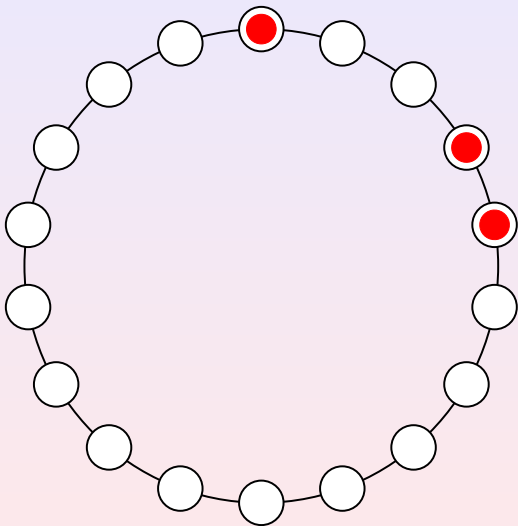
- Form a **special asymmetric pattern**
- Move the pattern through the graph



Algorithmic techniques (exclusive perpetual explo.)

Two phases

- Form a **special asymmetric pattern**
- Move the pattern through the graph



Known results (exclusive perpetual explo.) 2/2

Case of the rings

[Blin, Milani, Potop-Butucaru, Tixeuil. DISC 2010]

- FSYNC det.: Impossible when k is even
- ASYNC det.: $\rho^-(n) = 3$ for $n \geq 10$ and n and k coprime
- ASYNC det.: $\rho^+(n) = n - 5$ for k odd & n and k coprime

[Bonnet, Défago, Petit, Potop-Butucaru, Tixeuil. SRDS Workshop 2014]

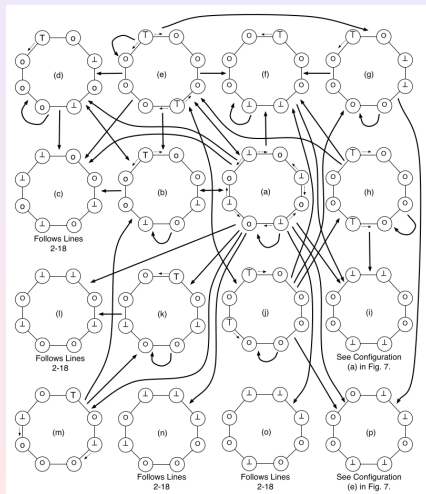
- Generic method to list all protocols (SSYNC)
- Application: impossible for $k = 5$ and $n = 10$

[D'Angelo, Di Stefano, Navarra, Nisse, Suchan. Algorithmica 2015]

- ASYNC but only for rigid initial configurations
- Det. algo for any k, n such that $n \geq 10$ and $5 \leq k \leq n - 3$, except the case $k = 5$ and $n = 10$

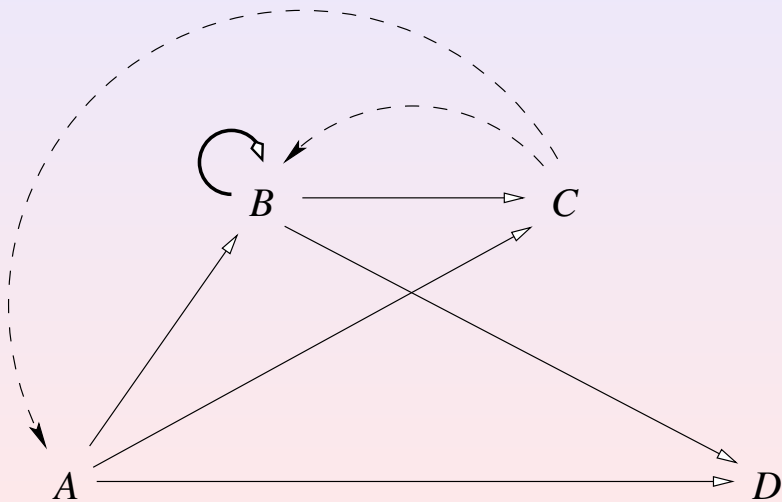
Proofs of correctness (1/2)

Case-by-case analysis



Proofs of correctness (2/2)

DAG with potential functions



Formal verification via model checking

[Bérard, Lafourcade, Millet, Potop-Butucaru, Thierry-Mieg, Tixeuil. Distributed Computing 2016]

Study of two algorithms

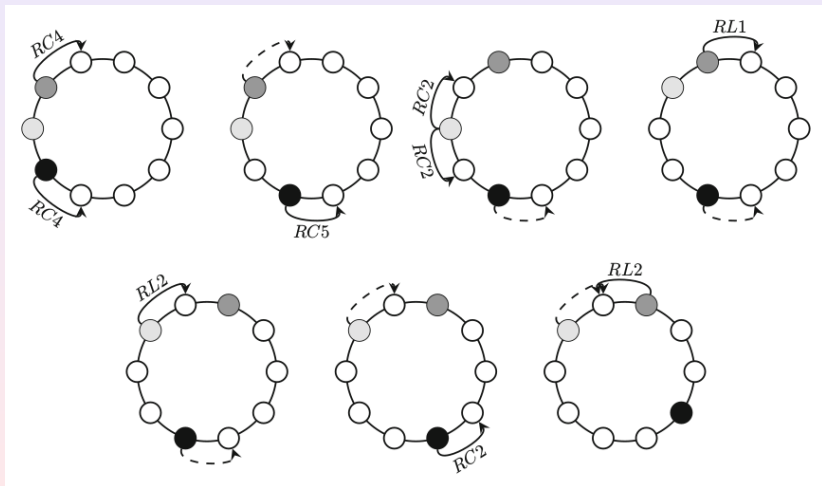
- **Exploration with stop** [FIPS Algorithmica 2013]
 - Correct for $17 \leq k \leq 21$ and $k < n \leq 22$
 - Also correct for smaller values of k (5, 7 and 10)
- **Exclusive perpetual exploration** [BMPT DISC 2010]
 - Counter-example for $k = 3$ and $n = 10$

[Doan, Bonnet, Ogata. SOFG+MSVL 2016]

[Bonnet, Milani, Potop-Butucaru, Tixeuil. OPODIS 2011]

- Also study of [BMPT DISC 2010]
- **Same counter-example**, with a different technique

The counter-example



Conclusion and perspectives

Conclusion

- Computability depends on the amount of symmetries
- **Tricky** to (correctly) handle **symmetries and asynchrony**

Interesting perspectives

- Limited visibility
- Formal verification
 - (Parametrized) Model checking
 - Proof assistant

Conclusion and perspectives

Conclusion

- Computability depends on the amount of symmetries
- **Tricky** to (correctly) handle **symmetries and asynchrony**

Interesting perspectives

- Limited visibility
- **Formal verification**
 - (Parametrized) Model checking
 - Proof assistant

Thank you
for your attention