Network Exploration by Asynchronous Oblivious Robots

Paola FLOCCHINI[1] David ILCINKAS[2]
Andrzej PELC[3] Nicola SANTORO[4]

[1]University of Ottawa, Canada
[2]CNRS, Université de Bordeaux, France
[3]Université du Québec en Outaouais, Canada
[4]Carleton University, Canada

MAC '10
August 18, 2010

Problem

Focus of the talk: robots operating in Look-Compute-Move cycles in networks

Model/context

- Anonymous graphs
- Team of robots
  - sensing the environment by taking a snapshot of it
  - that do not communicate
  - that are anonymous and oblivious

Goal: exploration with stop

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

Problem

Focus of the talk: robots operating in Look-Compute-Move cycles in networks

Model/context

- Anonymous graphs
- Team of robots
  - sensing the environment by taking a snapshot of it
  - that do not communicate
  - that are anonymous and oblivious

Goal: exploration with stop

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

Problem

Focus of the talk: robots operating in Look-Compute-Move cycles in networks

Model/context

- Anonymous graphs
- Team of robots
  - sensing the environment by taking a snapshot of it
  - that do not communicate
  - that are anonymous and oblivious

Goal: exploration with stop

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

The Look-Compute-Move cycle

Look

The robot takes a rooted instantaneous snapshot of the network and its robots, with (weak) multiplicity detection.

Compute

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

Move

In the latter case it instantaneously moves towards its destination.

Identical oblivious asynchronous robots

Identical

Robots have no IDs. They execute the same program.

Oblivious

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

Asynchronous (CORDA with unbounded fair scheduler)

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

Reminder:

Non-communicating

No communication mechanisms between robots, even locally.

Precisions concerning the model

Initial configurations

Arbitrary but without multiplicity (at most 1 robot / node) (necessary for termination)

In case of symmetry

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

Multiplicity detection (global weak)

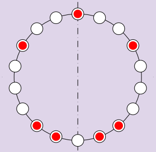
"zero", "one", or "more than one" robots

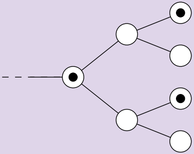
Precisions concerning the model

Initial configurations

Arbitrary but without multiplicity (at most 1 robot / node) (necessary for termination)

In case of symmetry





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

Multiplicity detection (global weak)

“zero”, “one”, or “more than one” robots

Precisions concerning the model

Initial configurations

Arbitrary but without multiplicity (at most 1 robot / node) (necessary for termination)

In case of symmetry

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

Multiplicity detection (global weak)

"zero", "one", or "more than one" robots

Smallest exploring team

Exploration

We say that exploration of a graph is possible with k robots, if there exists an algorithm enabling the robots to perform exploration with stop of this graph starting from any initial configuration of the k robots (thus, without multiplicity).

Smallest exploring team

Minimum number of robots that can explore any graph of a given family.

Smallest exploring team

Exploration

We say that exploration of a graph is possible with k robots, if there exists an algorithm enabling the robots to perform exploration with stop of this graph starting from any initial configuration of the k robots (thus, without multiplicity).

Smallest exploring team

Minimum number of robots that can explore any graph of a given family.

Related work

In the plane

Rich literature (gathering, pattern formation, etc.)

In graphs

- [Klasing, Markou, Pelc. ISAAC 2006 & TCS 2008]
  Feasibility of gathering in rings (except one case)
- [Klasing, Kosowski, Navarra. OPODIS 2008 & TCS 2010]
  Feasibility of gathering in rings in all cases (symmetry preserving algorithm)

Outline

Results (rings)

[Flocchini, I., Pelc, Santoro. OPODIS 2007]

Lemma

Exploration of a n-node ring by k robots is

- impossible if $k|n$ but $k \neq n$;
- possible if $gcd(n, k) = 1$, for $k \geq 17$.

Main result

Size of the smallest exploring team $\rho(n) \in \Theta(\log n)$

- There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.
- $\rho(n) \in O(\log n)$

Results (rings)

[Flocchini, I., Pelc, Santoro. OPODIS 2007]

Lemma

Exploration of a n-node ring by k robots is

- impossible if $k|n$ but $k \neq n$;
- possible if $gcd(n,k) = 1$, for $k \geq 17$.

Main result

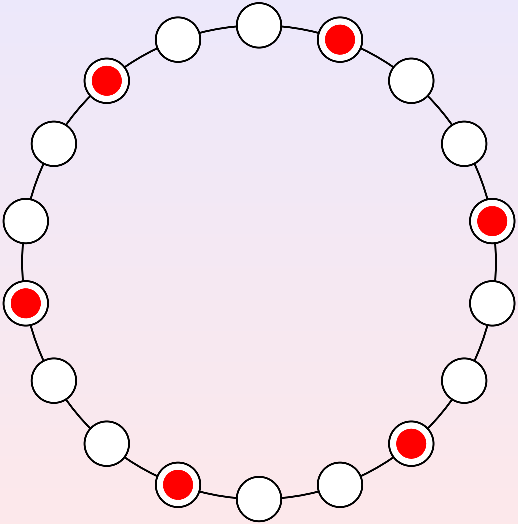
Size of the smallest exploring team $\rho(n) \in \Theta(\log n)$

- There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.
- $\rho(n) \in O(\log n)$

Lower bound (1/2)

Lemma

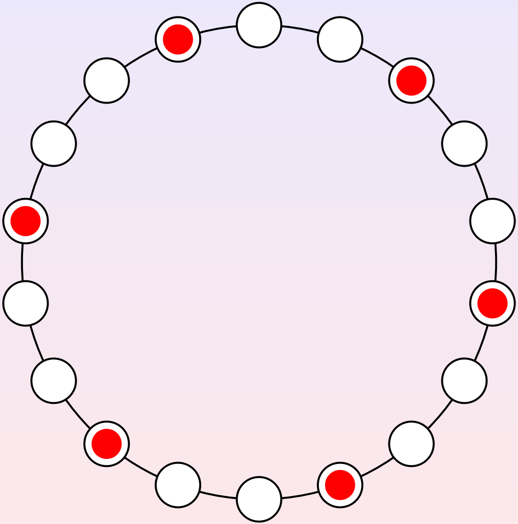
Impossible to stop (and sometimes to explore) when $k|n$.



Lower bound (1/2)

Lemma

Impossible to stop (and sometimes to explore) when $k|n$.



Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

- Let n be the least common multiple of integers $1, 2, \ldots, q$.
- From the previous slide, we have $\rho(n) \geq q + 1$.
- The Prime Number Theorem implies $q \sim \ln n$.
- This implies the existence of a constant c such that, for infinitely many n, $\rho(n) \geq c \log n$.

Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

- Let n be the least common multiple of integers $1, 2, \ldots, q$.
- From the previous slide, we have $\rho(n) \geq q + 1$.
- The Prime Number Theorem implies $q \sim \ln n$.
- This implies the existence of a constant c such that, for infinitely many n, $\rho(n) \geq c \log n$.

Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

- Let n be the least common multiple of integers $1, 2, \ldots, q$.
- From the previous slide, we have $\rho(n) \geq q + 1$.
- The Prime Number Theorem implies $q \sim \ln n$.
- This implies the existence of a constant c such that, for infinitely many n, $\rho(n) \geq c \log n$.

Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

- Let n be the least common multiple of integers $1, 2, \ldots, q$.
- From the previous slide, we have $\rho(n) \geq q + 1$.
- The Prime Number Theorem implies $q \sim \ln n$.
- This implies the existence of a constant c such that, for infinitely many n, $\rho(n) \geq c \log n$.

Lower bound (2/2)

Theorem

There exists a constant c such that, for infinitely many n, we have $\rho(n) \geq c \log n$.

Proof

- Let n be the least common multiple of integers $1, 2, \ldots, q$.
- From the previous slide, we have $\rho(n) \geq q + 1$.
- The Prime Number Theorem implies $q \sim \ln n$.
- This implies the existence of a constant c such that, for infinitely many n, $\rho(n) \geq c \log n$.

Upper bound

Lemma

Exploration is possible if $gcd(n,k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

- Take j such that $\frac{p_j\#}{13\#} \leq n < \frac{p_{j+1}\#}{13\#}$. We have $\rho(n) \leq p_{j+1}$. (all primes in $\{17, \ldots, p_{j+1}\}$ divide $n \implies n \geq \frac{p_{j+1}\#}{13\#}$)
- From [Ruiz, Math. Gaz. '97], we have $p_j \sim \ln(p_j\#)$.
- Hence $\rho(n) \leq p_{j+1} \in O(\log n)$.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

- Take j such that $\frac{p_j\#}{13\#} \leq n < \frac{p_{j+1}\#}{13\#}$. We have $\rho(n) \leq p_{j+1}$. (all primes in $\{17, \ldots, p_{j+1}\}$ divide $n \implies n \geq \frac{p_{j+1}\#}{13\#}$)
- From [Ruiz, Math. Gaz. '97], we have $p_j \sim \ln(p_j\#)$.
- Hence $\rho(n) \leq p_{j+1} \in O(\log n)$.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

- Take j such that $\frac{p_j\#}{13\#} \leq n < \frac{p_{j+1}\#}{13\#}$. We have $\rho(n) \leq p_{j+1}$. (all primes in $\{17, \ldots, p_{j+1}\}$ divide $n \implies n \geq \frac{p_{j+1}\#}{13\#}$)
- From [Ruiz, Math. Gaz. '97], we have $p_j \sim \ln(p_j\#)$.
- Hence $\rho(n) \leq p_{j+1} \in O(\log n)$.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

- Take j such that $\frac{p_j\#}{13\#} \leq n < \frac{p_{j+1}\#}{13\#}$. We have $\rho(n) \leq p_{j+1}$. (all primes in $\{17, \ldots, p_{j+1}\}$ divide $n \implies n \geq \frac{p_{j+1}\#}{13\#}$)
- From [Ruiz, Math. Gaz. '97], we have $p_j \sim \ln(p_j\#)$.
- Hence $\rho(n) \leq p_{j+1} \in O(\log n)$.

Upper bound

Lemma

Exploration is possible if $gcd(n, k) = 1$, for $k \geq 17$.

Theorem

The size $\rho(n)$ of the smallest exploring team is in $O(\log n)$.

Proof

Let p_j be the j-th prime, and $p_j\# = \prod_{i=1}^{j} p_i$ the p_j-primorial.

- Take j such that $\frac{p_j\#}{13\#} \leq n < \frac{p_{j+1}\#}{13\#}$. We have $\rho(n) \leq p_{j+1}$. (all primes in $\{17, \ldots, p_{j+1}\}$ divide $n \implies n \geq \frac{p_{j+1}\#}{13\#}$)
- From [Ruiz, Math. Gaz. '97], we have $p_j \sim \ln(p_j\#)$.
- Hence $\rho(n) \leq p_{j+1} \in O(\log n)$.

Some definitions

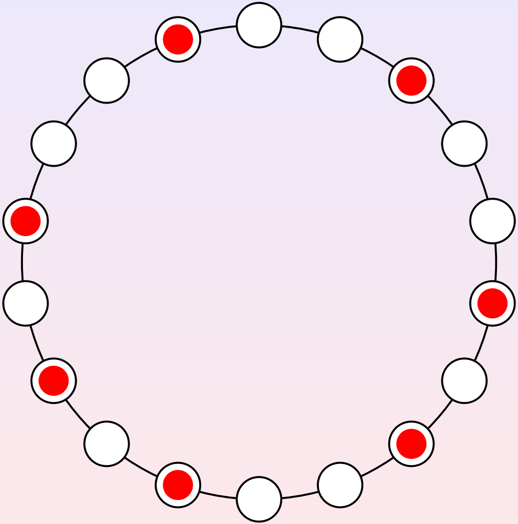
Interdistance

Minimum distance taken over all pairs of distinct robots.

Here interdistance=2.

Block

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



Some definitions

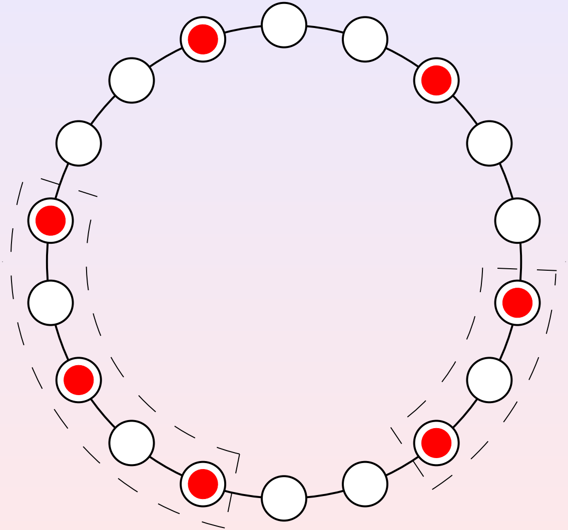
Interdistance

Minimum distance taken over all pairs of distinct robots.

Here interdistance=2.

Block

Maximal set of robots, of size at least 2, forming a line with a robot every *d* nodes. (*d*=interdistance)



Our 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.

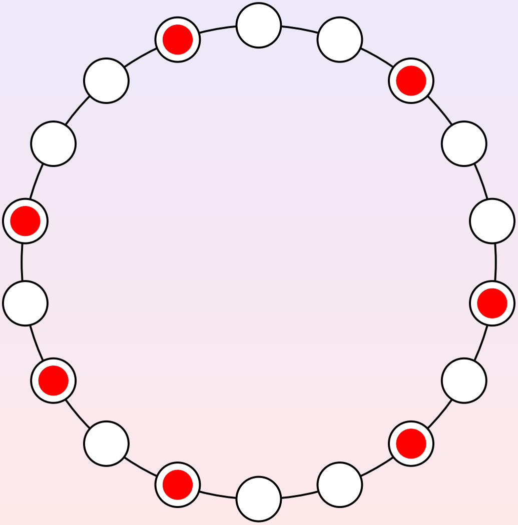
Tower-Creation Phase

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

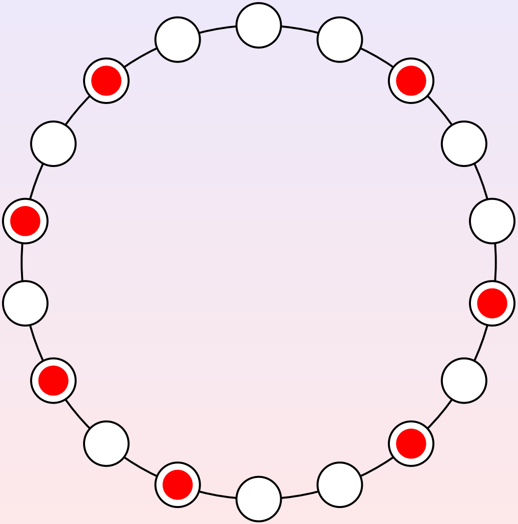
Exploration Phase

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

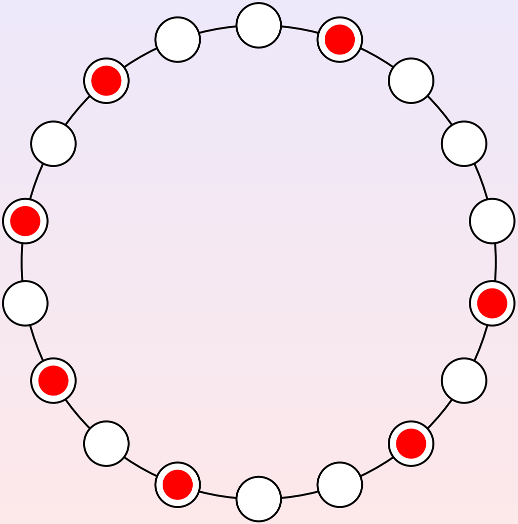
An example



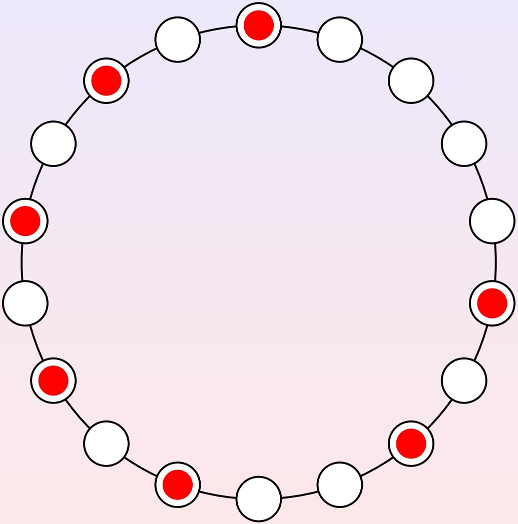
An example



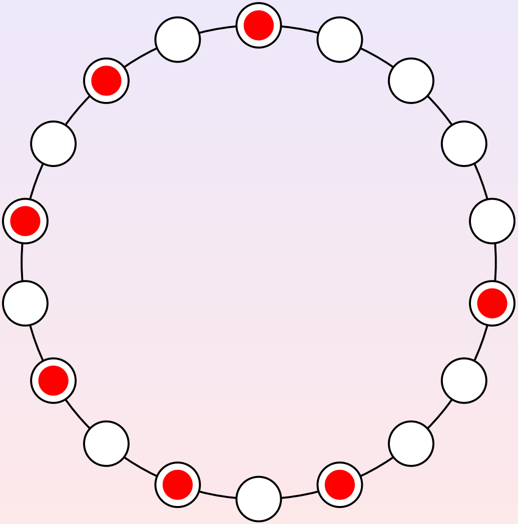
An example



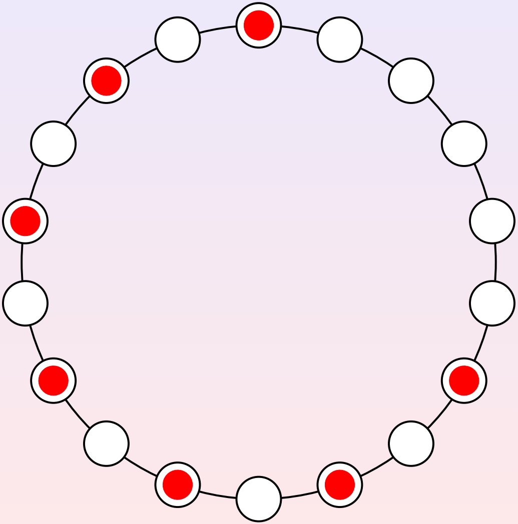
An example



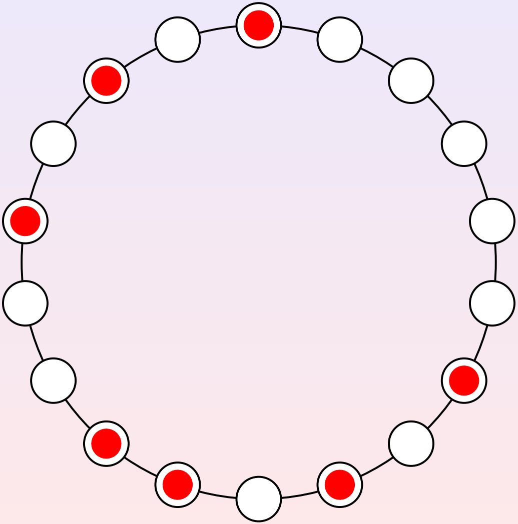
An example



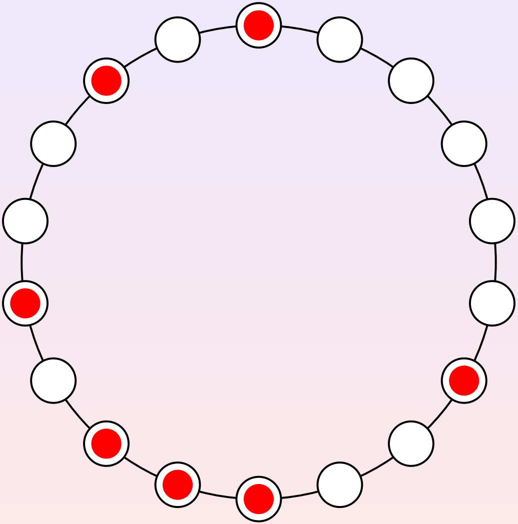
An example



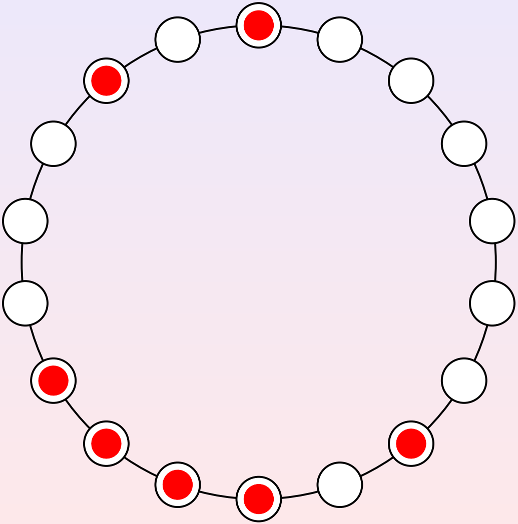
An example



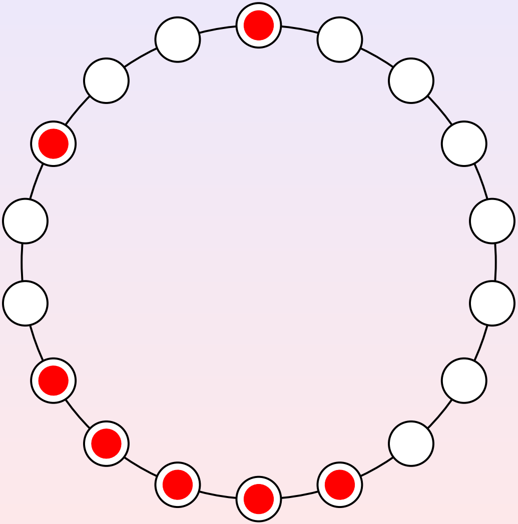
An example



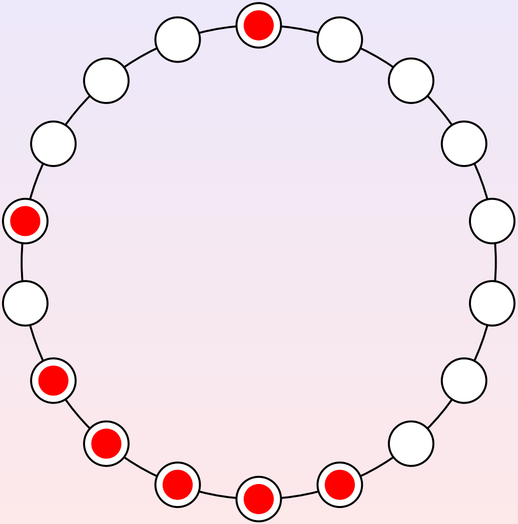
An example



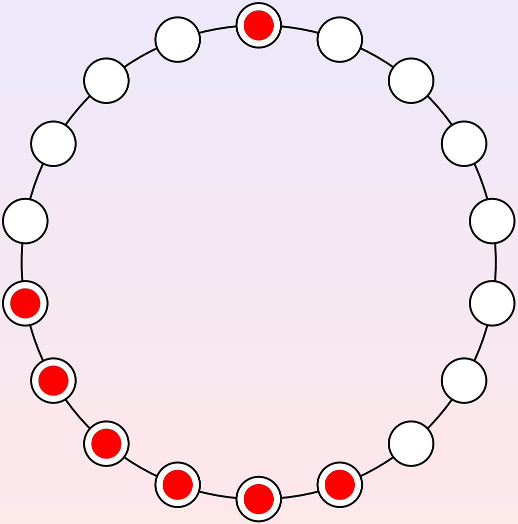
An example



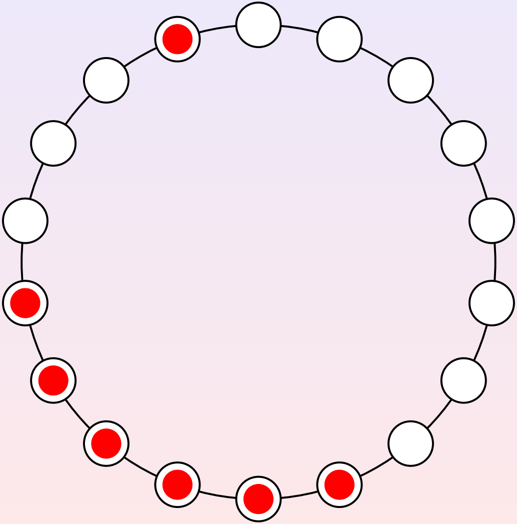
An example



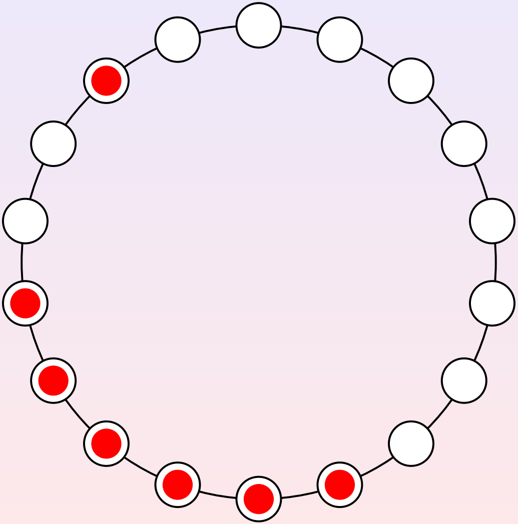
An example



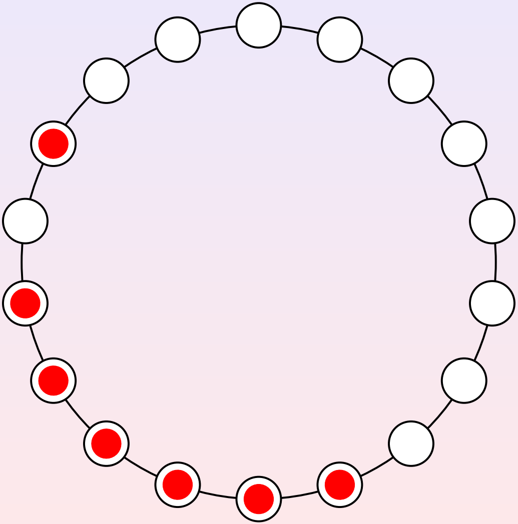
An example



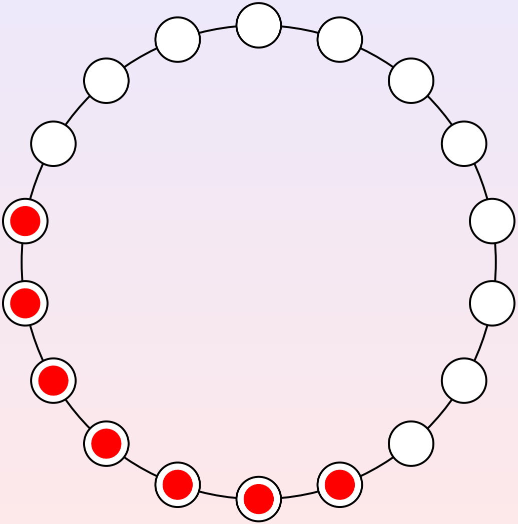
An example



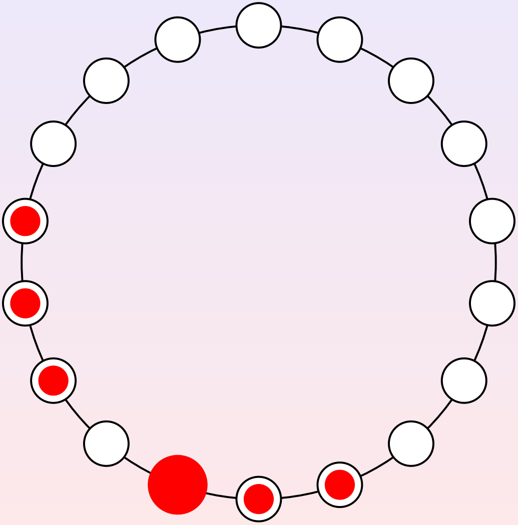
An example



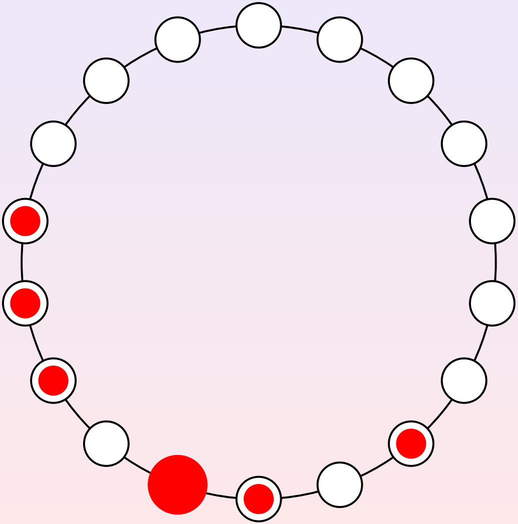
An example



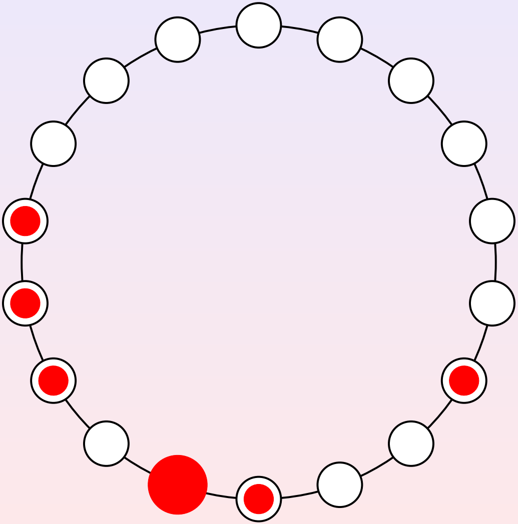
An example



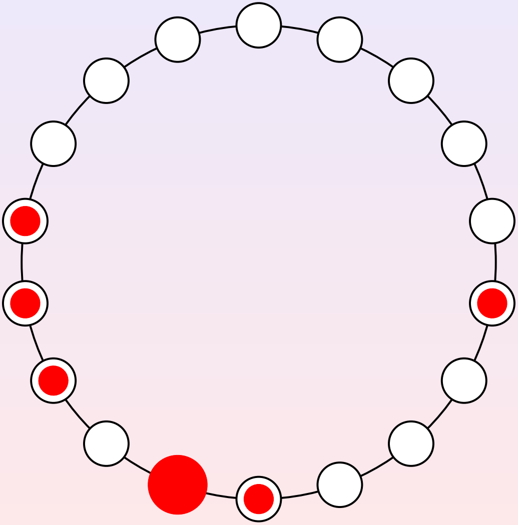
An example



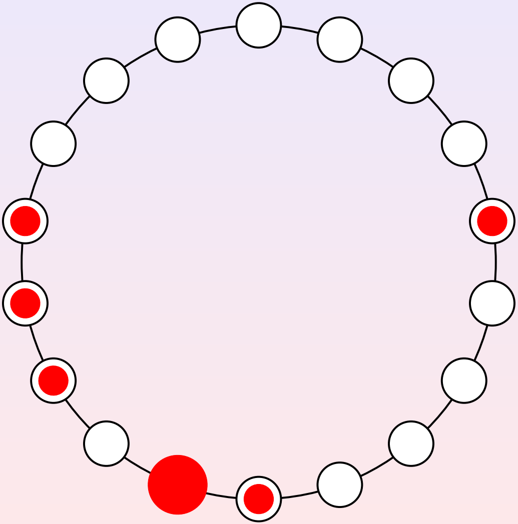
An example



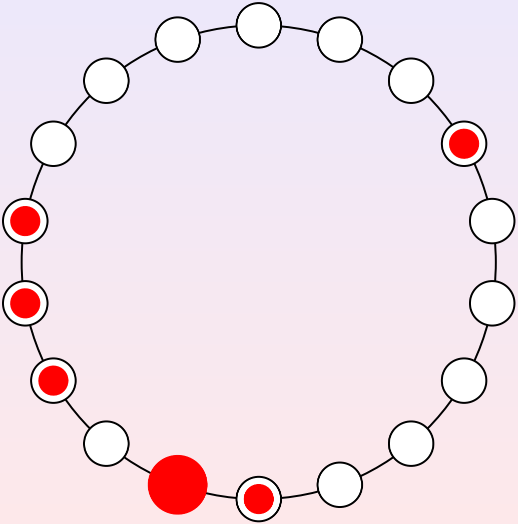
An example



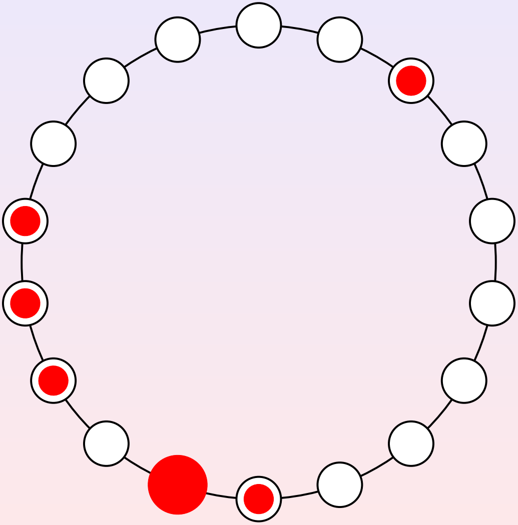
An example



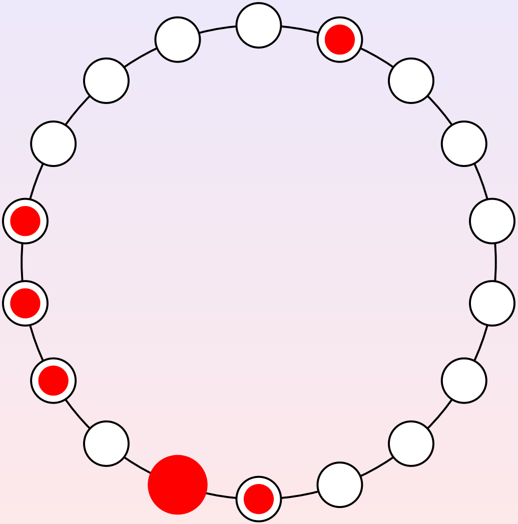
An example



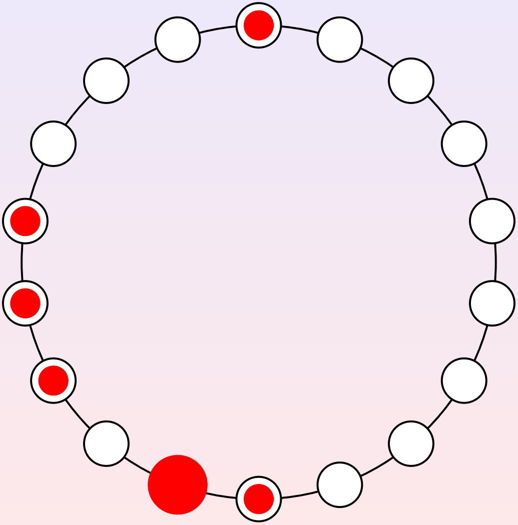
An example



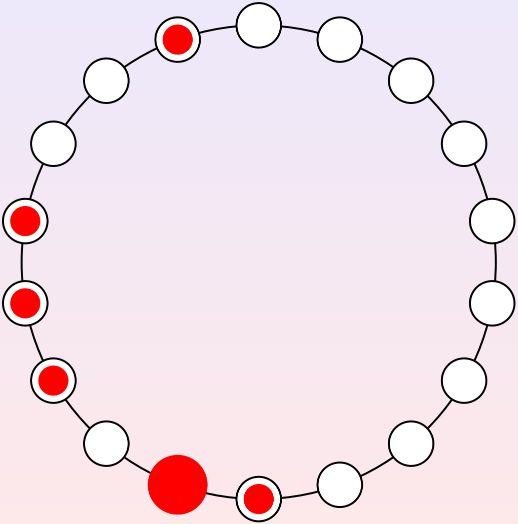
An example



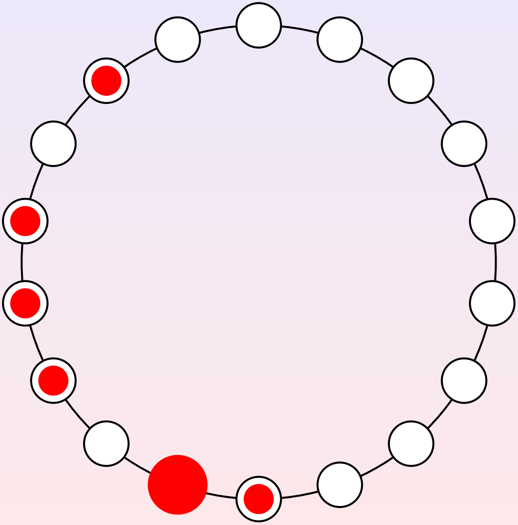
An example



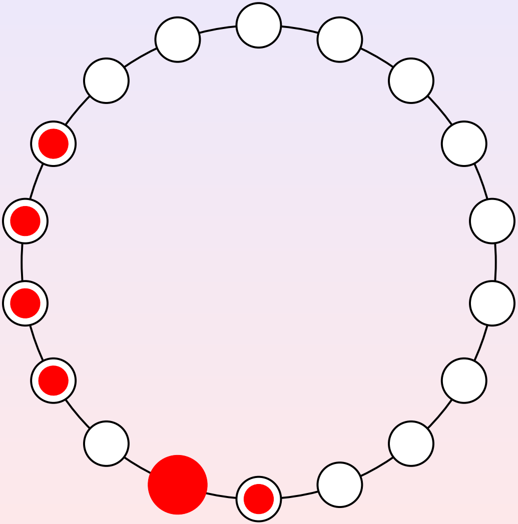
An example



An example



An example



Other results on the ring (1)

[Devismes, Petit, Tixeuil. SIROCCO 2009]

Main result

Four probabilistic robots are always necessary and sufficient (ATOM model)

Ideas of the algorithm

Use randomization to break symmetries

- Create one block of interdistance 1 (deterministic/randomized)
- Create a multiplicity (randomized)
- Explore the ring (deterministic)

Other results on the ring (1)

[Devismes, Petit, Tixeuil. SIROCCO 2009]

Main result

Four probabilistic robots are always necessary and sufficient (ATOM model)

Ideas of the algorithm

Use randomization to break symmetries

- Create one block of interdistance 1 (deterministic/randomized)
- Create a multiplicity (randomized)
- Explore the ring (deterministic)

Other results on the ring (2)

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

Focus

Size of the smallest exploring team for "good" values of n

Main results

- Lower bound on deterministic algorithm:
  - Five robots (when n is even)
  - Four robots (when n is odd)
- Deterministic algorithm for 5 robots when n and 5 are co-prime

Other results on the ring (2)

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

Focus

Size of the smallest exploring team for "good" values of n

Main results

- Lower bound on deterministic algorithm:
  - Five robots (when n is even)
  - Four robots (when n is odd)
- Deterministic algorithm for 5 robots when n and 5 are co-prime

Outline

Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)

Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

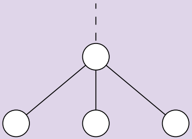
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

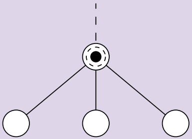
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log\log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

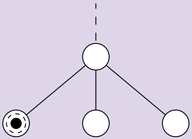
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

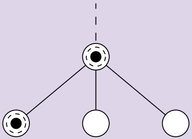
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

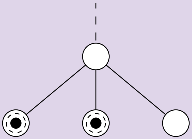
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

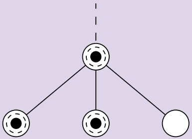
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log\log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

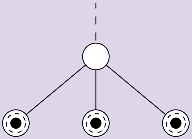
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

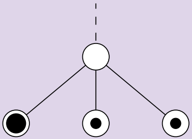
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log\log n)$ robots

Justification of the restrictions

$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Results (trees)

[Flocchini, I., Pelc, Santoro. SIROCCO 2008 & TCS 2010]

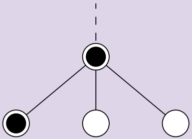
Main result

Trees of maximum degree 3:

- $\Theta(\log n / \log \log n)$ robots

Justification of the restrictions

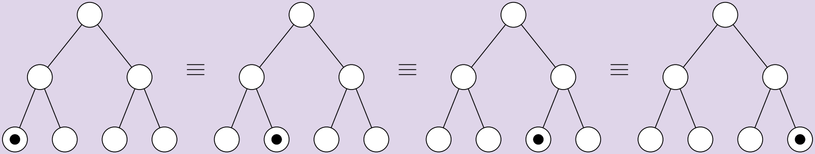
$\Theta(n)$ robots in some trees of maximum degree 4 (complete ternary trees)



Lower bound: $\Omega(\log n / \log\log n)$ robots

Observation

Many configurations are equivalent for the robots



Sketch of the proof

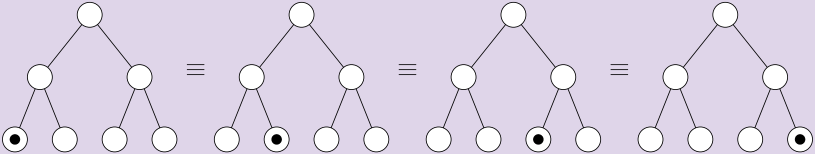
Complete binary tree, synchronous case

- few robots $\Rightarrow$ few different snapshots, say x
- at most x different snapshots $\Rightarrow$ at most $x \cdot k$ explored nodes before stopping

Lower bound: $\Omega(\log n / \log \log n)$ robots

Observation

Many configurations are equivalent for the robots



Sketch of the proof

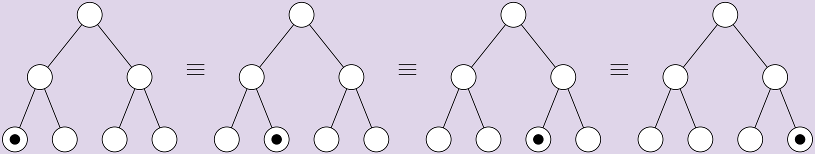
Complete binary tree, synchronous case

- few robots $\Rightarrow$ few different snapshots, say x
- at most x different snapshots $\Rightarrow$ at most $x \cdot k$ explored nodes before stopping

Lower bound: $\Omega(\log n / \log\log n)$ robots

Observation

Many configurations are equivalent for the robots



Sketch of the proof

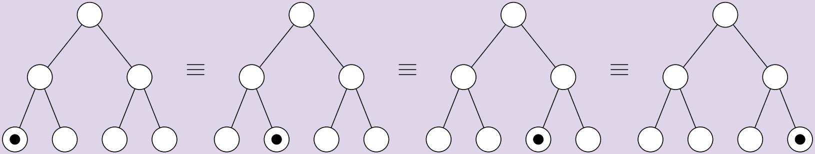
Complete binary tree, synchronous case

- few robots $\Rightarrow$ few different snapshots, say x
- at most x different snapshots $\Rightarrow$ at most $x \cdot k$ explored nodes before stopping

Lower bound: $\Omega(\log n / \log \log n)$ robots

Observation

Many configurations are equivalent for the robots



Sketch of the proof

Complete binary tree, synchronous case

- few robots ⇒ few different snapshots, say x
- at most x different snapshots ⇒ at most $x \cdot k$ explored nodes before stopping

Upper bound: $O(\log n/\log\log n)$ robots

Theorem

For any n, there exists a team of $k \in \Theta(\log n/\log\log n)$ robots, with $k \equiv 5 \pmod 6$ that can explore all n-node trees of maximum degree 3, starting from any initial configuration.

Main ideas

- A team of three robots aims at exploring the tree
- All other robots are used to keep track of progress
- A visual pattern, called the "brain", formed by the robots counts the number of explored leaves
- The tree is divided into few pieces and is explored piece by piece.

Upper bound: $O(\log n / \log \log n)$ robots

Theorem

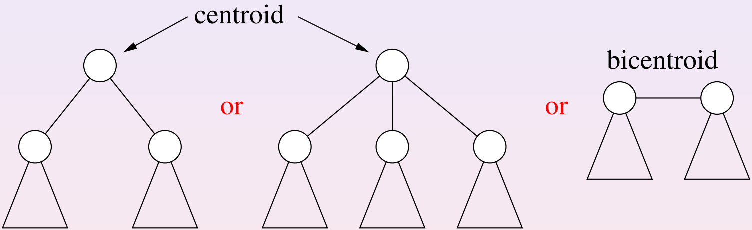
For any n, there exists a team of $k \in \Theta(\log n / \log \log n)$ robots, with $k \equiv 5 \pmod 6$ that can explore all n-node trees of maximum degree 3, starting from any initial configuration.

Main ideas

- A team of three robots aims at exploring the tree
- All other robots are used to keep track of progress
- A visual pattern, called the "brain", formed by the robots counts the number of explored leaves
- The tree is divided into few pieces and is explored piece by piece.

Pieces

The centroid defines pieces in the tree.

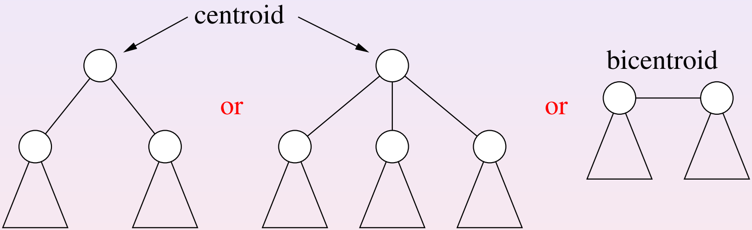


Property

The two largest pieces have size at least $n/4$.

Pieces

The centroid defines pieces in the tree.



Property

The two largest pieces have size at least $n/4$.

Phase 1 (1)

Goal: Make room in the pieces and create one multiplicity

Steps

- Any robot goes down if it does not create a multiplicity
- A leader is elected in the heaviest piece P (i.e. the one with the largest number of robots)
- The leader helps in creating a single multiplicity in P

Property

The core zone is connected and is formed by at least $\frac{n}{\log n}$ nodes.

Phase 1 (1)

Goal: Make room in the pieces and create one multiplicity

Steps

- Any robot goes down if it does not create a multiplicity
- A leader is elected in the heaviest piece P (i.e. the one with the largest number of robots)
- The leader helps in creating a single multiplicity in P

Property

The core zone is connected and is formed by at least $\frac{n}{\log n}$ nodes.

Phase 1 (1)

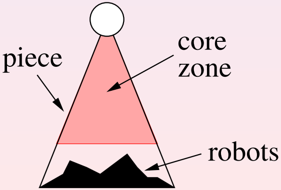
Goal: Make room in the pieces and create one multiplicity

Steps

- Any robot goes down if it does not create a multiplicity
- A leader is elected in the heaviest piece P (i.e. the one with the largest number of robots)
- The leader helps in creating a single multiplicity in P

Property

The core zone is connected and is formed by at least $\frac{n}{\log n}$ nodes.



Phase 1 (2)

Observation

In a piece, the number of robots having the same view is always a power of two and thus either even or one (solitaire).

Corollary

- A piece of odd weight has a (local) leader
- Since $k \equiv 5 \pmod{6}$, there always exists a global leader
- It is possible to have a single heaviest piece P, having a leader

Phase 1 (2)

Observation

In a piece, the number of robots having the same view is always a power of two and thus either even or one (solitaire).

Corollary

- A piece of odd weight has a (local) leader
- Since $k \equiv 5 \pmod 6$, there always exists a global leader
- It is possible to have a single heaviest piece P, having a leader

Phase 2

The brain

It synchronizes the actions of the robots and counts the number of explored leaves.

Goal of Phase 2

- Construct and initialize the brain in the core zone of the largest piece Q (different from P) by moving robots from the heavy piece P, using the leader to break symmetries.
- Form the exploring team of three robots in P.
- Remove (move in Q) all other robots in Q.

Phase 2

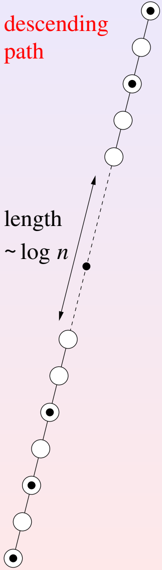
The brain

It synchronizes the actions of the robots and counts the number of explored leaves.

Goal of Phase 2

- Construct and initialize the brain in the core zone of the largest piece Q (different from P) by moving robots from the heavy piece P, using the leader to break symmetries.
- Form the exploring team of three robots in P.
- Remove (move in Q) all other robots in Q.

A counter



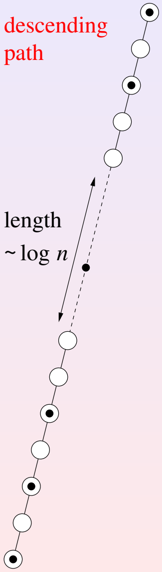
Lemma

In a core zone of size m, one can construct $\log^2 m$ disjoint descending paths of length $\frac{1}{4}\log m$.

Counter

One can construct a counter with range n by using $\Theta(\log n/\log\log n)$ descending paths and thus $\Theta(\log n/\log\log n)$ robots.

A counter



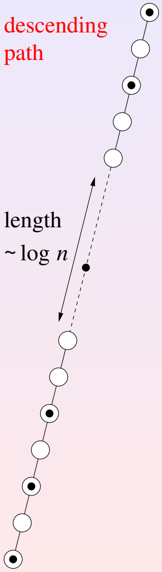
Lemma

In a core zone of size m, one can construct $\log^2 m$ disjoint descending paths of length $\frac{1}{4}\log m$.

Counter

One can construct a counter with range n by using $\Theta(\log n / \log\log n)$ descending paths and thus $\Theta(\log n / \log\log n)$ robots.

A counter



Lemma

In a core zone of size m, one can construct $\log^2 m$ disjoint descending paths of length $\frac{1}{4}\log m$.

Counter

One can construct a counter with range n by using $\Theta(\log n / \log\log n)$ descending paths and thus $\Theta(\log n / \log\log n)$ robots.

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

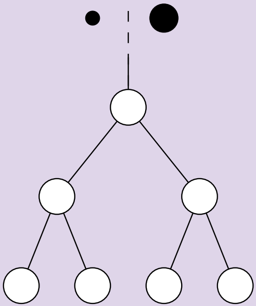
Exploration of a pair of leaves

Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

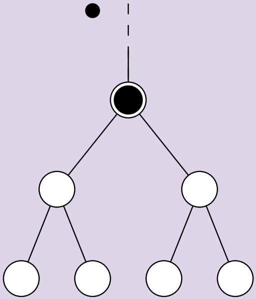


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

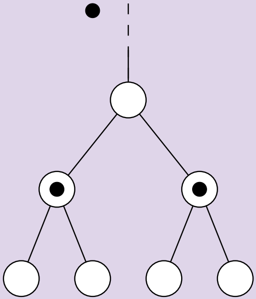


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

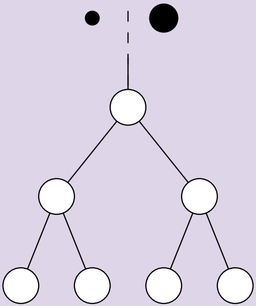


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

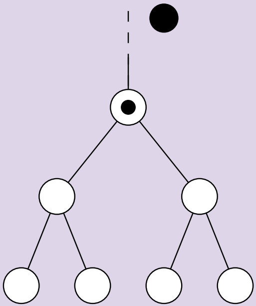


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

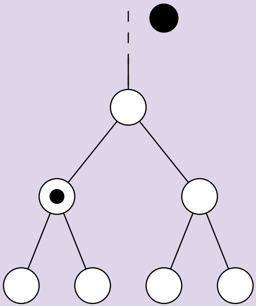


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

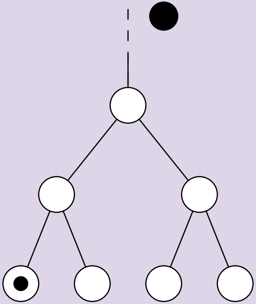


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

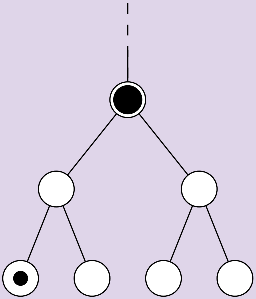


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

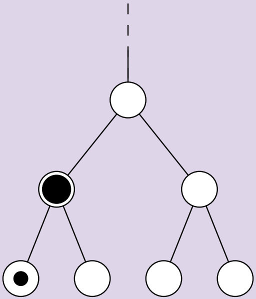


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

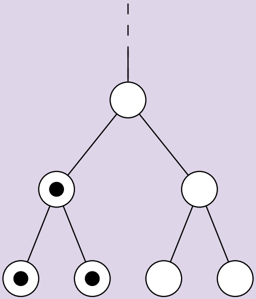


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

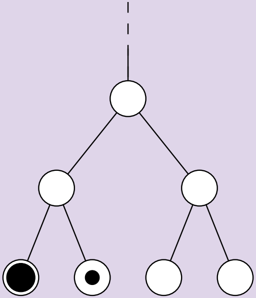


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

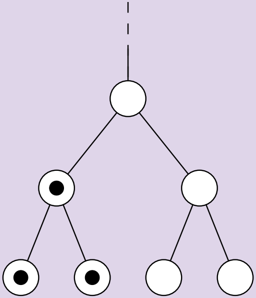


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

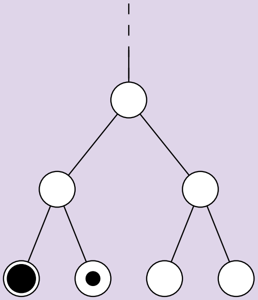


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves

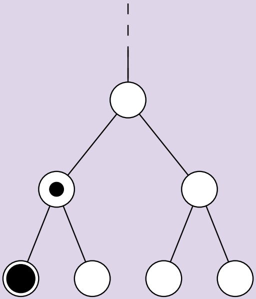


Use the counter value to determine the next leaf/pair of leaves to be explored

Phase 3

Goal: Explore P'', the largest of the pieces other than Q.

Exploration of a pair of leaves



Use the counter value to determine the next leaf/pair of leaves to be explored

Remaining phases

Phase 4

Relocate the brain from Q to P''

Phase 5

Explore piece Q and stop if there are only two pieces

Phase 6

Reinitialize the brain and relocate the exploring team in the unexplored piece

Phase 7

Explore the last piece and stop

Remaining phases

Phase 4

Relocate the brain from Q to P''

Phase 5

Explore piece Q and stop if there are only two pieces

Phase 6

Reinitialize the brain and relocate the exploring team in the unexplored piece

Phase 7

Explore the last piece and stop

Remaining phases

Phase 4

Relocate the brain from Q to P''

Phase 5

Explore piece Q and stop if there are only two pieces

Phase 6

Reinitialize the brain and relocate the exploring team in the unexplored piece

Phase 7

Explore the last piece and stop

Remaining phases

Phase 4

Relocate the brain from Q to P''

Phase 5

Explore piece Q and stop if there are only two pieces

Phase 6

Reinitialize the brain and relocate the exploring team in the unexplored piece

Phase 7

Explore the last piece and stop

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

A small sample of the problems to solve

- How to create a single multiplicity in Phase 1 without blocking the other robots?
- How to break symmetries using the leader? (problem of trapped solitaires)
- How to move multiplicities? How to move robots to their precise targets?
- How do the leader and the other robots cross each other in path-like trees?
- Is the counter up-to-date or currently updating?
- How to remember the phase number?
- ...

Outline

Conclusion (1)

My experience

- Very complicated algorithms
- Unreasonable complexity coming from the model

Potential "solutions"

- degree 3 vs > 3 → strong multiplicity detection
- complicated algorithm → ATOM?

Another (ideal?) solution: sense of direction (port numbers)

[Chalopin, Flocchini, Mans, Santoro. WG 2010]
Study in more general classes of graphs (CORDA model)

Conclusion (1)

My experience

- Very complicated algorithms
- Unreasonable complexity coming from the model

Potential "solutions"

- degree 3 vs $> 3 \rightarrow$ strong multiplicity detection
- complicated algorithm $\rightarrow$ ATOM?

Another (ideal?) solution: sense of direction (port numbers)

[Chalopin, Flocchini, Mans, Santoro. WG 2010]
Study in more general classes of graphs (CORDA model)

Conclusion (1)

My experience

- Very complicated algorithms
- Unreasonable complexity coming from the model

Potential "solutions"

- degree 3 vs $> 3 \rightarrow$ strong multiplicity detection
- complicated algorithm $\rightarrow$ ATOM?

Another (ideal?) solution: sense of direction (port numbers)

[Chalopin, Flocchini, Mans, Santoro. WG 2010]
Study in more general classes of graphs (CORDA model)

Conclusion and perspectives

Perspectives

- Limited visibility
- Fault tolerant protocols

Perpetual exploration without collision

- [Baldoni, Bonnet, Milani, Raynal, IPL 2008]
  Partial study (FSYNCH, unlimited vision)
- [Baldoni, Bonnet, Milani, Raynal, OPODIS 2008]
  Characterization in partial grids (FSYNCH, limited vision)
- [Blin, Milani, Gradinariu, Tixeuil, DISC 2010]
  Study in rings (ASYNCH, unlimited vision)

Conclusion and perspectives

Perspectives

- Limited visibility
- Fault tolerant protocols

Perpetual exploration without collision

- [Baldoni, Bonnet, Milani, Raynal. IPL 2008]
  Partial study (FSYNCH, unlimited vision)
- [Baldoni, Bonnet, Milani, Raynal. OPODIS 2008]
  Characterization in partial grids (FSYNCH, limited vision)
- [Blin, Milani, Gradinariu, Tixeuil. DISC 2010]
  Study in rings (ASYNCH, unlimited vision)

Thank You
for your attention