

Oracle size: a new measure of difficulty for communication tasks

Pierre Fraigniaud¹ David Ilcinkas² Andrzej Pelc³

¹CNRS, LRI, Univ. Paris-Sud, France

²LRI, Univ. Paris-Sud, France

³Dép. d'informatique, Univ. du Québec en Outaouais (UQO), Canada

PODC
July 25, 2006

Oracle

Framework

- Distributed computing
- Mobile computing

Observation

The **quality** of the algorithmic solutions often **depends** on the amount of **knowledge** that is given about the network.

Oracle

Models the **amount** of knowledge about the network that is given to the nodes and/or to the mobile agents.

Example 1: Exploration of anonymous digraphs

Problem

Digraph exploration by a robot

[Bender, Fernández, Ron, Sahai, Vadhan, Inf. & Comp. 2002]

No information

Impossible in polynomial time

Knowledge of an upper bound \hat{n} on the number of nodes

Possible in time polynomial in \hat{n}

Example 1: Exploration of anonymous digraphs

Problem

Digraph exploration by a robot

[Bender, Fernández, Ron, Sahai, Vadhan, Inf. & Comp. 2002]

No information

Impossible in polynomial time

Knowledge of an upper bound \tilde{n} on the number of nodes

Possible in time polynomial in \tilde{n}

Example 1: Exploration of anonymous digraphs

Problem

Digraph exploration by a robot

[Bender, Fernández, Ron, Sahai, Vadhan, Inf. & Comp. 2002]

No information

Impossible in polynomial time

Knowledge of an upper bound \tilde{n} on the number of nodes

Possible in time polynomial in \tilde{n}

Example 1: Exploration of anonymous digraphs

Problem

Digraph exploration by a robot

[Bender, Fernández, Ron, Sahai, Vadhan, Inf. & Comp. 2002]

No information

Impossible in polynomial time

Knowledge of an upper bound \hat{n} on the number of nodes

Possible in time polynomial in \hat{n}

Example 2: Broadcast in radio networks

Problem

Synchronous deterministic broadcast in n -node networks of diameter D

No information (only its own identity)

[Clementi, Monti, Silvestri, SODA 2001]: time $\Omega(n \log D)$

Complete knowledge of the network

[Kowalski, Pelc, Distributed Computing]: time $O(D + \log^2 n)$

Example 2: Broadcast in radio networks

Problem

Synchronous deterministic broadcast in n -node networks of diameter D

No information (only its own identity)

[Clementi, Monti, Silvestri, SODA 2001]: time $\Omega(n \log D)$

Complete knowledge of the network

[Kowalski, Pelc, Distributed Computing]: time $O(D + \log^2 n)$

Example 2: Broadcast in radio networks

Problem

Synchronous deterministic broadcast in n -node networks of diameter D

No information (only its own identity)

[Clementi, Monti, Silvestri, SODA 2001]: time $\Omega(n \log D)$

Complete knowledge of the network

[Kowalski, Pelc, Distributed Computing]: time $O(D + \log^2 n)$

Example 3: Small worlds navigability

Problem

- Expected time of routing in augmented graphs
- (non-oblivious) greedy algorithm

No information (only the direct neighbors)

[Kleinberg, STOC 2000]: time $O(\log^2 n)$

Knowledge of the contacts of the $\log n$ nodes closest to the current node in the grid

[Martel, Nguyen, PODC 2004]: time $O(\log^{3/2} n)$

Example 3: Small worlds navigability

Problem

- Expected time of routing in augmented graphs
- (non-oblivious) greedy algorithm

No information (only the direct neighbors)

[Kleinberg, STOC 2000]: time $O(\log^2 n)$

Knowledge of the contacts of the $\log n$ nodes closest to the current node in the grid

[Martel, Nguyen, PODC 2004]: time $O(\log^{3/2} n)$

Example 3: Small worlds navigability

Problem

- Expected time of routing in augmented graphs
- (non-oblivious) greedy algorithm

No information (only the direct neighbors)

[Kleinberg, STOC 2000]: time $O(\log^2 n)$

Knowledge of the contacts of the $\log n$ nodes closest to the current node in the grid

[Martel, Nguyen, PODC 2004]: time $O(\log^{3/2} n)$

Example 4: Wakeup

Problem

- Wakeup in arbitrary networks
- Knowledge of the topology within radius ρ

Results

[Awerbuch, Goldreich, Peleg, Vainish, J. of ACM, 1990]:
 $\Theta(\min\{m, n^{1+\Theta(1)/\rho}\})$ messages of bounded length

Example 4: Wakeup

Problem

- Wakeup in arbitrary networks
- Knowledge of the topology within radius ρ

Results

[Awerbuch, Goldreich, Peleg, Vainish, J. of ACM, 1990]:
 $\Theta(\min\{m, n^{1+\Theta(1)/\rho}\})$ messages of bounded length

Drawback of the qualitative approach

Difficult to compare

- Algorithms knowing n
- Algorithms knowing D
- Algorithms knowing the neighborhood

A need for a quantitative approach

Drawback of the qualitative approach

Difficult to compare

- Algorithms knowing n
- Algorithms knowing D
- Algorithms knowing the neighborhood

A need for a **quantitative** approach

Outline

- 1 Introduction
- 2 Our results**
- 3 Sketch of the proofs
- 4 Conclusion and perspectives

Definition of a quantitative measure

Definition

- An oracle distributes a binary string $\mathcal{O}(G)$ to the nodes (and/or to the mobile agents)
- Size of an oracle: $|\mathcal{O}(G)|$

Related questions

What is the minimum size of an oracle permitting to solve problem \mathcal{P} ?

Quantitative questions about the required knowledge, regardless of what kind of knowledge is supplied.

Definition of a quantitative measure

Definition

- An oracle distributes a **binary string** $\mathcal{O}(G)$ to the nodes (and/or to the mobile agents)
- **Size** of an oracle: $|\mathcal{O}(G)|$

Related questions

What is the minimum size of an oracle permitting to solve problem \mathcal{P} ?

Quantitative questions about the required knowledge, regardless of what kind of knowledge is supplied.

Definition of a quantitative measure

Definition

- An oracle distributes a **binary string** $\mathcal{O}(G)$ to the nodes (and/or to the mobile agents)
- **Size** of an oracle: $|\mathcal{O}(G)|$

Related questions

What is the **minimum size of an oracle** permitting to solve problem \mathcal{P} ?

Quantitative questions about the required knowledge, regardless of what **kind** of knowledge is supplied.

Application to distributed computing

Two fundamental tasks

Disseminating a message M from a source to all the nodes of a network

- **Wakeup:** a node cannot communicate before it has received the message M
- **Broadcast:** a node can communicate at any time

Problem

Achieving these tasks using a number of messages linear in n

Application to distributed computing

Two fundamental tasks

Disseminating a message M from a source to all the nodes of a network

- **Wakeup**: a node cannot communicate before it has received the message M
- **Broadcast**: a node can communicate at any time

Problem

Achieving these tasks using a **number of messages linear in n**

Our results

Wakeup

Minimum oracle size is $\Theta(n \log n)$ bits.

Broadcast

Minimum oracle size is $\Theta(n)$ bits.

Our results

Wakeup

Minimum oracle size is $\Theta(n \log n)$ bits.

Broadcast

Minimum oracle size is $\Theta(n)$ bits.

Our results

Wakeup

Minimum oracle size is $\Theta(n \log n)$ bits.

Broadcast

Minimum oracle size is $\Theta(n)$ bits.

- Broadcast protocol using an oracle of size $O(n)$ bits
- No oracle of size $o(n)$ bits can solve linear broadcasting

Strength of our results

Lower bounds

- Synchronous environment
- Node identifiers between 1 and n
- Arbitrary long messages
- A unique fixed source

Upper bounds

- Asynchronous environment
- No identifiers
- Bounded-length messages
- An arbitrary set of sources

All our upper bounds are constructive

Strength of our results

Lower bounds

- Synchronous environment
- Node identifiers between 1 and n
- Arbitrary long messages
- A unique fixed source

Upper bounds

- Asynchronous environment
- No identifiers
- Bounded-length messages
- An arbitrary set of sources

All our upper bounds are constructive

Strength of our results

Lower bounds

- Synchronous environment
- Node identifiers between 1 and n
- Arbitrary long messages
- A unique fixed source

Upper bounds

- Asynchronous environment
- No identifiers
- Bounded-length messages
- An arbitrary set of sources

All our upper bounds are constructive

Outline

- 1 Introduction
- 2 Our results
- 3 Sketch of the proofs**
- 4 Conclusion and perspectives

Wakeup: upper bound

Theorem

There exists an **oracle of size $O(n \log n)$** bits permitting the **wakeup** with a linear number of messages.

Proof

- Spanning tree of the network
- The oracle gives to each node the port numbers of the edges of the spanning tree that are incident to that node

Wakeup: upper bound

Theorem

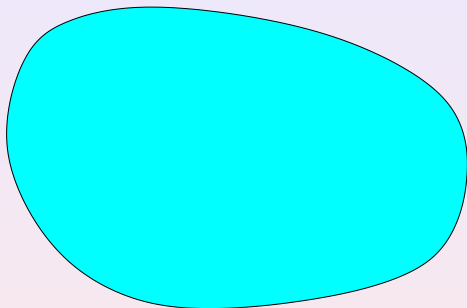
There exists an **oracle of size $O(n \log n)$** bits permitting the **wakeup** with a linear number of messages.

Proof

- Spanning tree of the network
- The oracle gives to each node the port numbers of the edges of the spanning tree that are incident to that node

Lower bounds: general ideas

Without oracle \implies all instances are possible

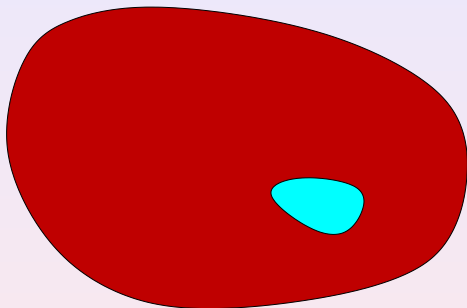


Tool

Design of an adversary that depends on the oracle

Lower bounds: general ideas

With oracle \implies only a subset of instances is possible

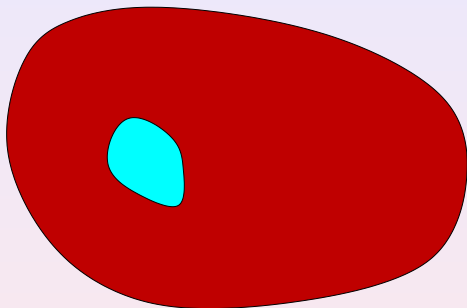


Tool

Design of an adversary that depends on the oracle

Lower bounds: general ideas

With oracle \implies only a subset of instances is possible

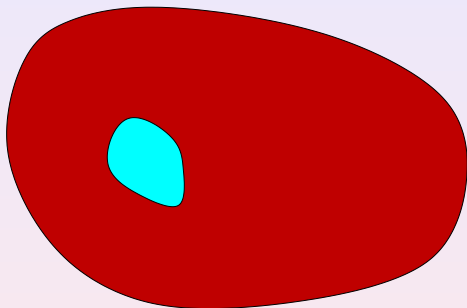


Tool

Design of an adversary that depends on the oracle

Lower bounds: general ideas

With oracle \implies only a subset of instances is possible



Tool

Design of an adversary that depends on the oracle

Wakeup: lower bound (1)

Theorem

The minimum oracle size permitting the **wakeup** with a linear number of messages is $\Omega(n \log n)$ bits.

Sketch of the proof

- Complete graph K_n with n subdivided edges

Wakeup: lower bound (1)

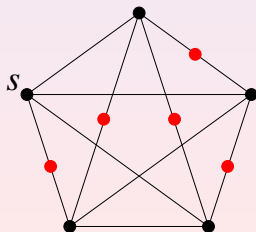
Theorem

The minimum oracle size permitting the **wakeup** with a linear number of messages is $\Omega(n \log n)$ bits.

Sketch of the proof

- Complete graph K_n with n subdivided edges

• $\Omega(n \log n)$ bits are necessary to reduce the $\binom{n}{2}$ choices



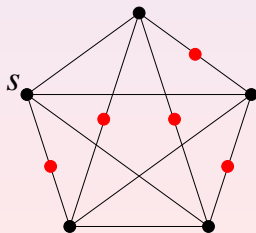
Wakeup: lower bound (1)

Theorem

The minimum oracle size permitting the **wakeup** with a linear number of messages is $\Omega(n \log n)$ bits.

Sketch of the proof

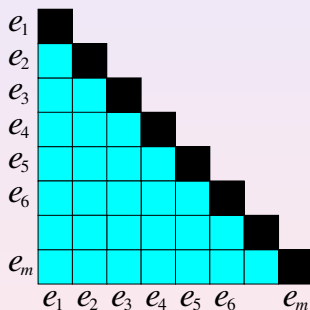
- Complete graph K_n with n subdivided edges
- $\Omega(n \log n)$ bits are necessary to reduce the $\binom{n^2}{n}$ choices



Wakeup: lower bound (2)

Example with two subdivided edges

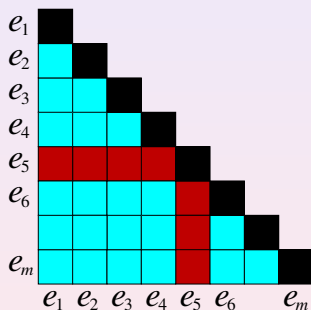
Without oracle \implies all instances are possible



Wakeup: lower bound (2)

Example with two subdivided edges

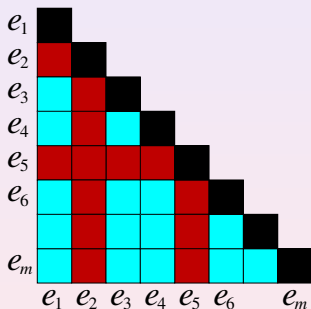
Without oracle \implies all instances are possible



Wakeup: lower bound (2)

Example with two subdivided edges

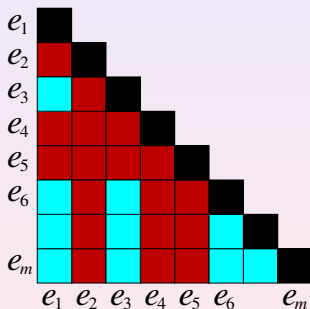
Without oracle \implies all instances are possible



Wakeup: lower bound (2)

Example with two subdivided edges

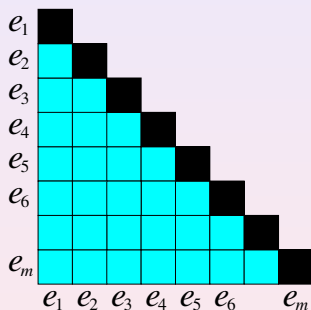
Without oracle \implies all instances are possible



Wakeup: lower bound (3)

Example with two subdivided edges

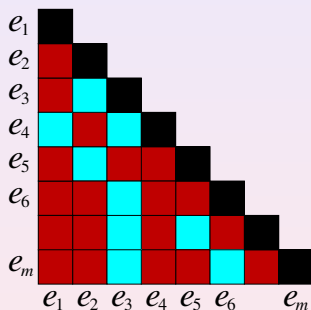
With oracle \implies only a subset of instances is possible



Wakeup: lower bound (3)

Example with two subdivided edges

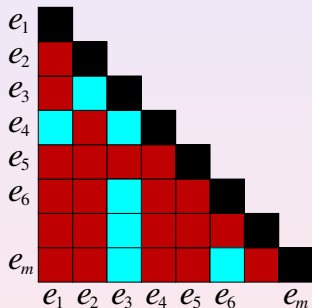
With oracle \implies only a subset of instances is possible



Wakeup: lower bound (3)

Example with two subdivided edges

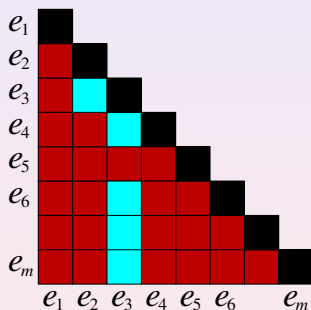
With oracle \implies only a subset of instances is possible



Wakeup: lower bound (3)

Example with two subdivided edges

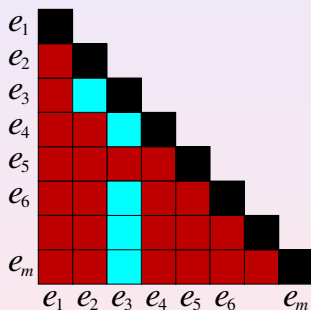
With oracle \implies only a subset of instances is possible



Wakeup: lower bound (3)

Example with two subdivided edges

With oracle \implies only a subset of instances is possible



Greedy adversary: maximize the number of remaining instances

Broadcast: upper bound

Theorem

There exists an **oracle of size $O(n)$** bits permitting the **broadcast** with a linear number of messages.

Sketch of the proof

- Spanning tree of the network
- Oracle: informs one of the two extremities of a tree edge

Technical arguments $\Rightarrow O(n)$ bits are sufficient

Broadcast: upper bound

Theorem

There exists an **oracle of size $O(n)$** bits permitting the **broadcast** with a linear number of messages.

Sketch of the proof

- Spanning tree of the network
- Oracle: informs one of the two extremities of a tree edge

Technical arguments $\Rightarrow O(n)$ bits are sufficient

Broadcast: lower bound

Theorem

No oracle of size $o(n)$ bits can permit to broadcast efficiently.

Sketch of the proof

- Complete graph K_n with n/k special edges
- A k -node complete graph in each special edge

Broadcast: lower bound

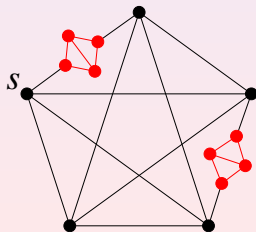
Theorem

No oracle of size $o(n)$ bits can permit to **broadcast** efficiently.

Sketch of the proof

- Complete graph K_n with n/k special edges
- A k -node complete graph in each special edge

• Size $O(n/k)$ bits implies $\Omega(nk)$ messages



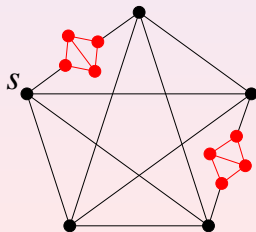
Broadcast: lower bound

Theorem

No oracle of size $o(n)$ bits can permit to **broadcast** efficiently.

Sketch of the proof

- Complete graph K_n with n/k special edges
- A k -node complete graph in each special edge
- Size $O(n/k)$ bits implies $\Omega(nk)$ messages



Outline

- 1 Introduction
- 2 Our results
- 3 Sketch of the proofs
- 4 Conclusion and perspectives**

Conclusion

Our results

Wakeup and broadcast

- Constraint: linear number of messages
 - Wakeup: $\Theta(n \log n)$ bits
 - Broadcast: $\approx \Theta(n)$ bits
- **Quantitative comparison** between similar tasks