

# Exploration d'arbres avec oracle

Pierre Fraigniaud<sup>1</sup> David Ilcinkas<sup>2</sup> Andrzej Pelc<sup>3</sup>

<sup>1</sup>CNRS, LRI, Université Paris-Sud, France

<sup>2</sup>LRI, Université Paris-Sud, France

<sup>3</sup>Département d'informatique, Université du Québec en Outaouais, Canada

AlgoTel '06, 9 mai 2006

## Cadre

- Calcul distribué
- Agents mobiles

## Observation

L'efficacité des solutions à un problème dépend souvent de la connaissance donnée a priori sur la topologie.

## Oracle

Modélise les **connaissances** sur la topologie données aux nœuds et/ou aux agents.

# Exploration des graphes orientés anonymes

Exploration par un robot muni d'un caillou  
(caillou = marqueur de sommets que le robot peut déposer et reprendre à sa guise lors de ses déplacements)

[Bender et al., Information and computation 2002]

Aucune information

Impossible en temps polynomial  
(possible avec  $\Theta(\log \log n)$  cailloux)

Connaissance d'une borne sup.  $n$  sur le nombre de sommets  
Possible en temps polynomial en  $\hat{n}$

# Exploration des graphes orientés anonymes

Exploration par un robot muni d'un caillou  
(caillou = marqueur de sommets que le robot peut déposer et reprendre à sa guise lors de ses déplacements)

[Bender et al., Information and computation 2002]

Aucune information

Impossible en temps polynomial  
(possible avec  $\Theta(\log \log n)$  cailloux)

Connaissance d'une borne sup.  $n$  sur le nombre de sommets  
Possible en temps polynomial en  $\hat{n}$

# Exploration des graphes orientés anonymes

Exploration par un robot muni d'un caillou  
(caillou = marqueur de sommets que le robot peut déposer et reprendre à sa guise lors de ses déplacements)

[Bender et al., Information and computation 2002]

## Aucune information

**Impossible** en temps polynomial  
(possible avec  $\Theta(\log \log n)$  cailloux)

Connaissance d'une borne sup.  $\hat{n}$  sur le nombre de sommets  
Possible en temps polynomial en  $\hat{n}$

# Exploration des graphes orientés anonymes

Exploration par un robot muni d'un caillou  
(caillou = marqueur de sommets que le robot peut déposer et reprendre à sa guise lors de ses déplacements)

[Bender et al., Information and computation 2002]

## Aucune information

**Impossible** en temps polynomial  
(possible avec  $\Theta(\log \log n)$  cailloux)

## Connaissance d'une borne sup. $\hat{n}$ sur le nombre de sommets

**Possible** en temps polynomial en  $\hat{n}$

# Diffusion (broadcast) dans les réseaux radio

Diffusion déterministe synchrone dans les réseaux de taille  $n$  et de diamètre  $D$

(seulement l'identifiant)

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

du réseau

[Kowalski et Pelc, to appear] : temps  $O(D + \log^2 n)$

# Diffusion (broadcast) dans les réseaux radio

Diffusion déterministe synchrone dans les réseaux de taille  $n$  et de diamètre  $D$

**Aucune information** (seulement l'identifiant)

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

diff. réseau

[Kowalski et Pelc, to appear] : temps  $O(D + \log^2 n)$



# Diffusion (broadcast) dans les réseaux radio

Diffusion déterministe synchrone dans les réseaux de taille  $n$  et de diamètre  $D$

**Aucune information** (seulement l'identifiant)

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

**Connaissance complète** du réseau

[Kowalski et Pelc, to appear] : temps  $O(D + \log^2 n)$

# Connaissance du voisinage à distance $\rho$

Réveil (wakeup) dans les réseaux filaires

[Awerbuch et al., J. of ACM, 1990] :  $\Theta(\min\{m, n^{1+\Theta(1)/\rho}\})$   
messages de taille bornée

Routage dans le modèle de Kleinberg

[Fraigniaud, Gavoille et Paul, PODC 2004] : Longueur du  
chemin =  $f(\rho)$

# Connaissance du voisinage à distance $\rho$

## Réveil (wakeup) dans les réseaux filaires

[Awerbuch et al., J. of ACM, 1990] :  $\Theta(\min\{m, n^{1+\Theta(1)/\rho}\})$   
messages de taille bornée

Routage dans le modèle de Kleinberg

[Fraigniaud, Gavoille et Paul, PODC 2004] : Longueur du  
chemin =  $f(\rho)$

# Connaissance du voisinage à distance $\rho$

## Réveil (wakeup) dans les réseaux filaires

[Awerbuch et al., J. of ACM, 1990] :  $\Theta(\min\{m, n^{1+\Theta(1)/\rho}\})$   
messages de taille bornée

## Routage dans le modèle de Kleinberg

[Fraigniaud, Gavoille et Paul, PODC 2004] : Longueur du  
chemin =  $f(\rho)$

# Définition de l'oracle

## Définition

- Oracle fournit une chaîne binaire  $\mathcal{O}(G)$  aux nœuds/agents mobiles
- Taille de l'oracle :  $|\mathcal{O}(G)|$

## Question intéressante

Quelle est la taille minimale d'un oracle permettant de résoudre le problème  $\mathcal{P}$  (en un temps donné) ?

Questions quantitatives indépendantes du type d'informations

# Définition de l'oracle

## Définition

- Oracle fournit une **chaîne binaire**  $\mathcal{O}(G)$  aux nœuds/agents mobiles
- **Taille** de l'oracle :  $|\mathcal{O}(G)|$

Question intéressante

Quelle est la taille minimale d'un oracle permettant de résoudre le problème  $\mathcal{P}$  (en un temps donné) ?

Questions quantitatives indépendantes du type d'informations

# Définition de l'oracle

## Définition

- Oracle fournit une **chaîne binaire**  $\mathcal{O}(G)$  aux nœuds/agents mobiles
- **Taille** de l'oracle :  $|\mathcal{O}(G)|$

## Question intéressante

Quelle est la taille minimale d'un oracle permettant de résoudre le problème  $\mathcal{P}$  (en un temps donné) ?

Questions **quantitatives** indépendantes du **type** d'informations.

# Notre exemple d'application

## Exploration d'arbres par un robot

### But

Visiter tous les sommets le plus rapidement possible

### Mesure

Rapport compétitif d'un algorithme  $\mathcal{A}$  :

$$\frac{\text{longueur du chemin suivi par } \mathcal{A}}{\text{plus court chemin couvrant l'arbre}}$$

(maximisé sur tous les graphes et tous les sommets de départ)



# Notre exemple d'application

Exploration d'arbres par un robot

But

Visiter tous les sommets le plus rapidement possible

Mesure

Rapport compétitif d'un algorithme  $\mathcal{A}$  :

$$\frac{\text{longueur du chemin suivi par } \mathcal{A}}{\text{plus court chemin couvrant l'arbre}}$$

(maximisé sur tous les graphes et tous les sommets de départ)

# Notre exemple d'application

Exploration d'arbres par un robot

But

Visiter tous les sommets le plus rapidement possible

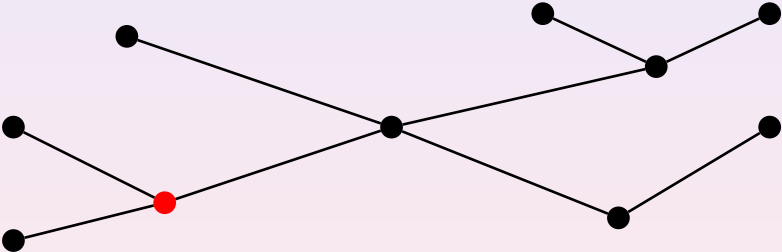
Mesure

**Rapport compétitif** d'un algorithme  $\mathcal{A}$  :

$$\frac{\text{longueur du chemin suivi par } \mathcal{A}}{\text{plus court chemin couvrant l'arbre}}$$

(maximisé sur tous les graphes et tous les sommets de départ)

# Exemple



# Résultats connus

[Dessmark et Pelc, TCS, 2004]

**DFS** a un rapport compétitif égal à 2

Aucune information

Aucun algorithme ne peut battre le DFS

Connaissance du graphe (non étiqueté)

Rapport compétitif strictement inférieur à 2 atteignable

Énoncé de notre problème

Quelle est la taille minimale d'un oracle permettant d'obtenir un rapport compétitif strictement inférieur à 2 ?

# Résultats connus

[Dessmark et Pelc, TCS, 2004]

**DFS** a un rapport compétitif égal à 2

Aucune information

Aucun algorithme ne peut battre le DFS

Connaissance du graphe (non étiqueté)

Rapport compétitif strictement inférieur à 2 atteignable

Énoncé de notre problème

Quelle est la taille minimale d'un oracle permettant d'obtenir un rapport compétitif strictement inférieur à 2 ?

# Résultats connus

[Dessmark et Pelc, TCS, 2004]

**DFS** a un rapport compétitif égal à 2

Aucune information

Aucun algorithme ne peut battre le DFS

Connaissance du graphe (non étiqueté)

Rapport compétitif strictement inférieur à 2 atteignable

Énoncé de notre problème

Quelle est la taille minimale d'un oracle permettant d'obtenir un rapport compétitif strictement inférieur à 2 ?

# Résultats connus

[Dessmark et Pelc, TCS, 2004]

**DFS** a un rapport compétitif égal à 2

Aucune information

Aucun algorithme ne peut battre le DFS

Connaissance du graphe (non étiqueté)

Rapport compétitif strictement inférieur à 2 atteignable

Énoncé de notre problème

Quelle est la **taille minimale d'un oracle** permettant d'obtenir un **rapport compétitif strictement inférieur à 2** ?

# Nos résultats (1)

## Résultat principal

seuil  $\approx \log \log D$  bits ( $D = \text{diamètre}$ )

## Notations

- $f(D)$  = taille de l'oracle pour les arbres de diamètre au plus  $D$
- $f(D) = \log \log D - g(D)$

## Énoncé précis

rapport compétitif  $< 2$  possible  $\Leftrightarrow g$  bornée supérieurement



# Nos résultats (1)

## Résultat principal

seuil  $\approx \log \log D$  bits ( $D = \text{diamètre}$ )

## Notations

- $f(D)$  = taille de l'oracle pour les arbres de diamètre au plus  $D$
- $f(D) = \log \log D - g(D)$

Énoncé précis

rapport compétitif  $< 2$  possible  $\Leftrightarrow g$  bornée supérieurement

# Nos résultats (1)

## Résultat principal

seuil  $\approx \log \log D$  bits ( $D = \text{diamètre}$ )

## Notations

- $f(D)$  = taille de l'oracle pour les arbres de diamètre au plus  $D$
- $f(D) = \log \log D - g(D)$

## Énoncé précis

rapport compétitif  $< 2$  possible  $\Leftrightarrow g$  bornée supérieurement

# Nos résultats(2)

## Interprétation

$$\log \log D - c \text{ bits} \iff D_0 \leq D \leq c' D_0$$

Résultat complémentaire

Seulement  $D$  connu, même exactement  $\Rightarrow$  rapport  $\geq 2$

# Nos résultats(2)

## Interprétation

$$\log \log D - c \text{ bits} \iff D_0 \leq D \leq c'D_0$$

## Résultat complémentaire

Seulement  $D$  connu, **même exactement**  $\Rightarrow$  rapport  $\geq 2$

# Borne supérieure

## Remarque

DFS se comporte bien lorsque

- diamètre  $D$  petit
- nombre  $n$  de sommets petit
- $D$  pas trop proche de  $n$

## Oracle

- bit  $b$  :
  - 0  $\rightarrow$  DFS (si  $D$  pas trop proche de  $n$ )
  - 1  $\rightarrow$  algorithme bcp plus subtil (sinon)
- entier  $k$  : approximation de  $D$  sur  $\log \log D - c$  bits

# Borne supérieure

## Remarque

DFS se comporte bien lorsque

- diamètre  $D$  petit
- nombre  $n$  de sommets petit
- $D$  pas trop proche de  $n$

## Oracle

- bit  $b$  :
  - 0  $\rightarrow$  DFS (si  $D$  pas trop proche de  $n$ )
  - 1  $\rightarrow$  algorithme bcp plus subtil (sinon)
- entier  $k$  : approximation de  $D$  sur  $\log \log D - c$  bits

# Borne supérieure

## Remarque

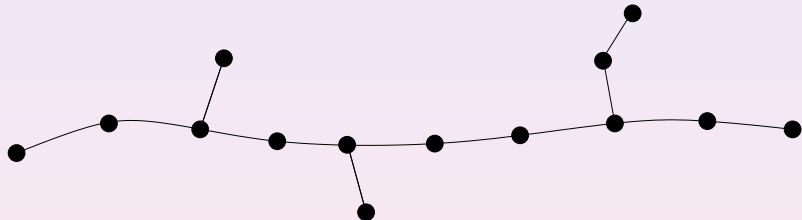
DFS se comporte bien lorsque

- diamètre  $D$  petit
- nombre  $n$  de sommets petit
- $D$  pas trop proche de  $n$

## Oracle

- bit  $b$  :
  - 0  $\rightarrow$  DFS (si  $D$  pas trop proche de  $n$ )
  - 1  $\rightarrow$  algorithme bcp plus subtil (sinon)
- entier  $k$  : approximation de  $D$  sur  $\log \log D - c$  bits

# Idées de l'algorithme





# Autre utilisation de l'oracle

Fraigniaud, Ilcinkas et Pelc, PODC 2006

Application de l'oracle à la **dissémination d'informations**

Contrainte : #messages  $O(n)$

- **réveil** :  $\Omega(n \log n)$  bits
- **diffusion** :  $O(n)$  bits

⇒ différence clairement établie entre les deux problèmes

Perspectives

Concept d'oracle applicable à de très nombreux problèmes

# Autre utilisation de l'oracle

Fraigniaud, Ilcinkas et Pelc, PODC 2006

Application de l'oracle à la **dissémination d'informations**

Contrainte : #messages  $O(n)$

- **réveil** :  $\Omega(n \log n)$  bits
- **diffusion** :  $O(n)$  bits

⇒ différence clairement établie entre les deux problèmes

## Perspectives

Concept d'oracle applicable à de très nombreux problèmes