

Exploration de Graphes par Automates

David Ilcinkas

LRI, Université Paris-Sud, France

Séminaire du LIP
7 décembre 2004

Exploration de graphes

But

Une entité mobile doit **traverser** chaque arête d'un graphe **inconnu** et **anonyme**.

Motivation

- exploration d'endroits inaccessibles à l'homme
- maintenance d'un réseau informatique
- cartographie

Exploration de graphes

But

Une entité mobile doit **traverser** chaque arête d'un graphe **inconnu** et **anonyme**.

Motivation

- exploration d'endroits inaccessibles à l'homme
- maintenance d'un réseau informatique
- cartographie

Inconnu, Anonyme

Inconnu

- topologie inconnue
- pas de bornes sur la taille

Anonyme

- pas d'étiquetage des sommets
- étiquetage local des arêtes

Inconnu, Anonyme

Inconnu

- topologie inconnue
- pas de bornes sur la taille

Anonyme

- pas d'étiquetage des sommets
- étiquetage local des arêtes

Inconnu, Anonyme

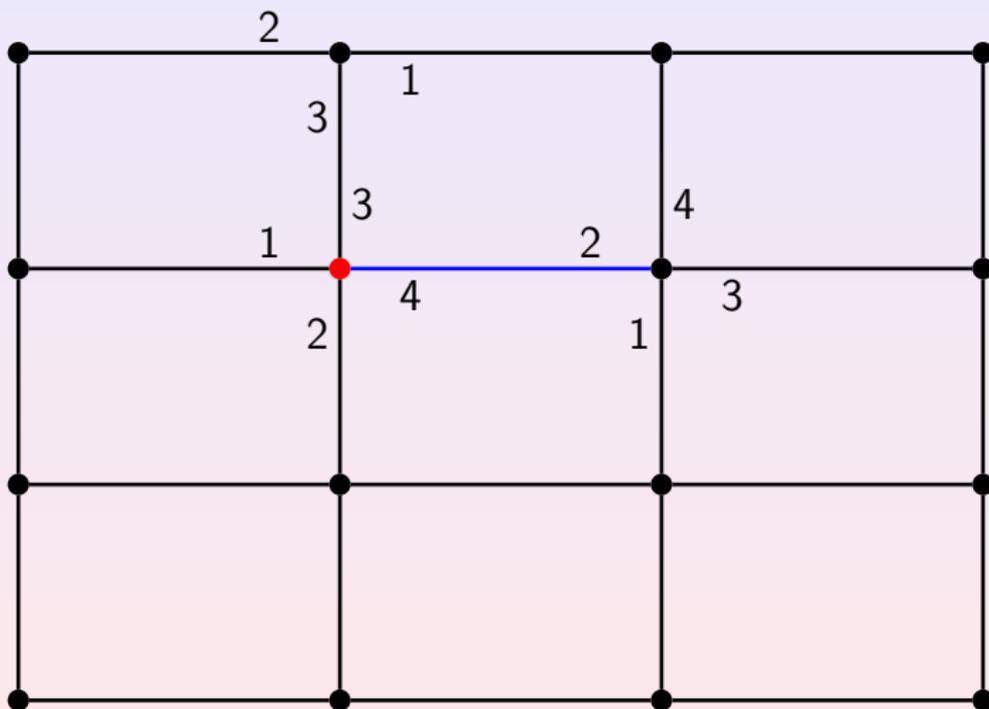
Inconnu

- topologie inconnue
- pas de bornes sur la taille

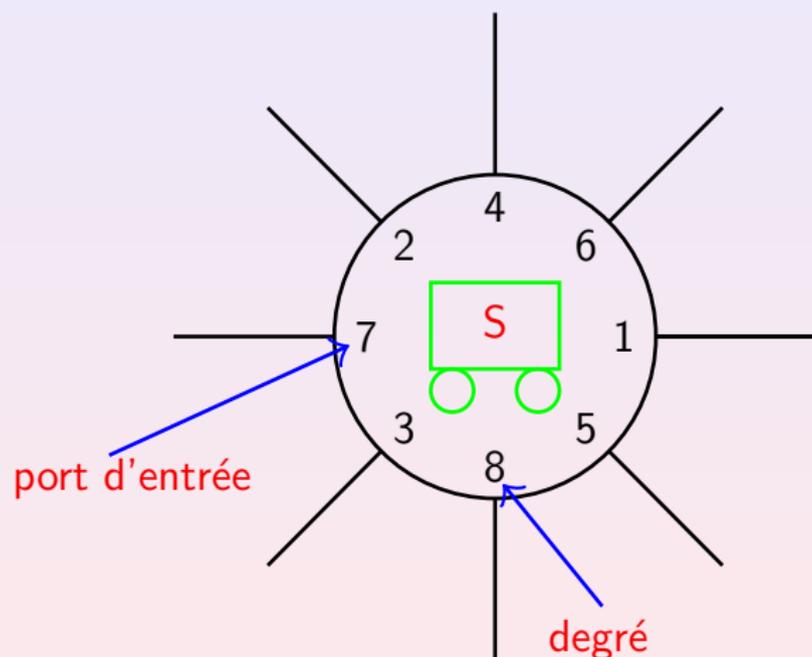
Anonyme

- pas d'étiquetage des sommets
- étiquetage local des arêtes

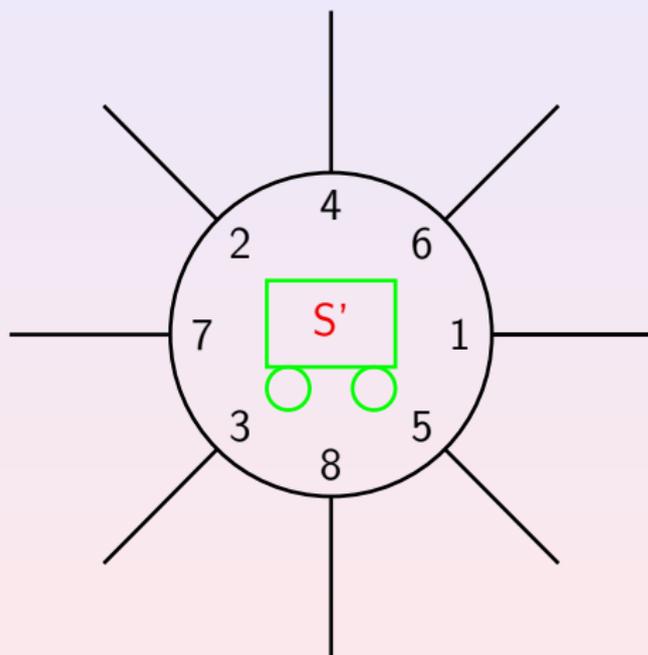
Exemple d'un graphe anonyme



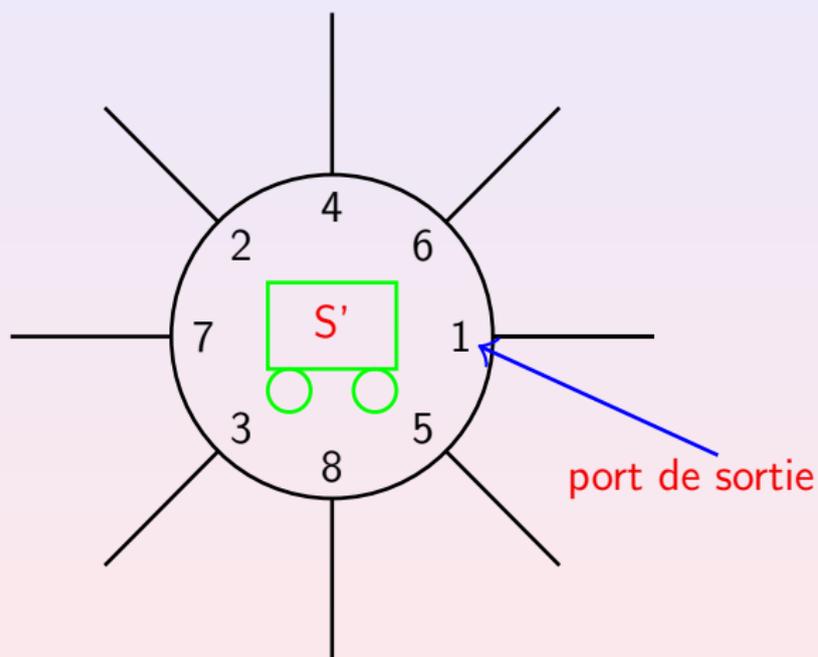
Entrée de l'automate



Entrée de l'automate



Entrée de l'automate



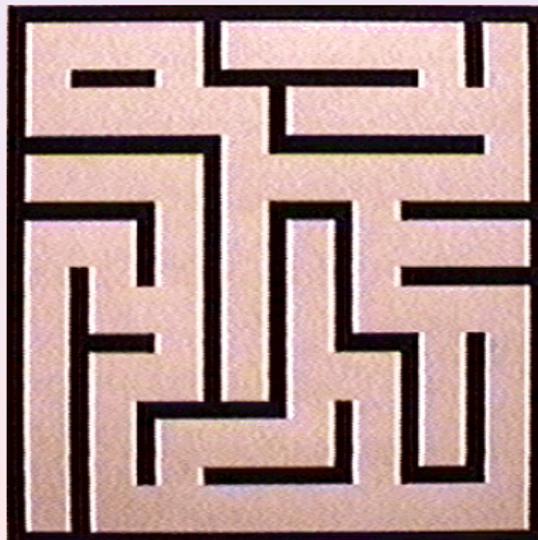
Plan

- 1 Introduction
- 2 Faisabilité**
 - Labyrinthes
 - Graphes
 - Hydre
- 3 Minimiser la mémoire
- 4 Autres modèles
- 5 Bonus

Cas particulier : Labyrinthes

Définition

- grille avec des arêtes manquantes
- étiquetage cohérent des arêtes : Nord, Sud, Est, Ouest



Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978

Automata and Labyrinths

∄ automate fini universel

Blum, Kozen, FOCS 1978

On the power of the compass

∃ automate fini universel à deux cailloux

∃ deux automates coopératifs universels pour les labyrinthes

∄ deux automates coopératifs universels pour les graphes

Hoffmann, FCT 1981

One pebble does not suffice to search plane labyrinths

∄ automate fini universel à un caillou

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978

Automata and Labyrinths

\nexists automate fini universel

Blum, Kozen, FOCS 1978

On the power of the compass

\exists automate fini universel à deux cailloux

\exists deux automates coopératifs universels pour les labyrinthes

\nexists deux automates coopératifs universels pour les graphes

Hoffmann, FCT 1981

One pebble does not suffice to search plane labyrinths

\nexists automate fini universel à un caillou

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978

Automata and Labyrinths

∄ automate fini universel

Blum, Kozen, FOCS 1978

On the power of the compass

∃ automate fini universel à deux cailloux

∃ deux automates coopératifs universels pour les labyrinthes

∄ deux automates coopératifs universels pour les graphes

Hoffmann, FCT 1981

One pebble does not suffice to search plane labyrinths

∄ automate fini universel à un caillou

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978

Automata and Labyrinths

∄ automate fini universel

Blum, Kozen, FOCS 1978

On the power of the compass

∃ automate fini universel à deux cailloux

∃ deux automates coopératifs universels pour les labyrinthes

∄ deux automates coopératifs universels pour les graphes

Hoffmann, FCT 1981

One pebble does not suffice to search plane labyrinths

∄ automate fini universel à un caillou

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978

Automata and Labyrinths

\nexists automate fini universel

Blum, Kozen, FOCS 1978

On the power of the compass

\exists automate fini universel à deux cailloux

\exists deux automates coopératifs universels pour les labyrinthes

\nexists deux automates coopératifs universels pour les graphes

Hoffmann, FCT 1981

One pebble does not suffice to search plane labyrinths

\nexists automate fini universel à un caillou

Graphes arbitraires

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

Aucun automate fini n'explore tous les labyrinthes \Rightarrow graphes.

Roth, Acta Informatica, 1980

Automaten in planaren Graphen

Aucune équipe finie d'automates finis n'explore tous les graphes (mêmes planaires).

Graphes arbitraires

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

Aucun automate fini n'explore tous les labyrinthes \Rightarrow graphes.

Rollik, Acta Informatica, 1980

Automaten in planaren Graphen

Aucune équipe finie d'automates finis n'explore tous les graphes (même planaires).

Comment piéger un automate (Rollik)

Piège pour un automate spécifique

- automate fini \iff comportement devient périodique
- piéger l'automate dans un cycle

Piège pour k automates non coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête

Piège pour k automates coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête
- méta-structure plus complexe

Comment piéger un automate (Rollik)

Piège pour un automate spécifique

- automate fini \iff comportement devient périodique
- piéger l'automate dans un cycle

Piège pour k automates non coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête

Piège pour k automates coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête
- méta-structure plus complexe

Comment piéger un automate (Rollik)

Piège pour un automate spécifique

- automate fini \iff comportement devient périodique
- piéger l'automate dans un cycle

Piège pour k automates non coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête

Piège pour k automates coopératifs

- défini récursivement
- placer un piège pour $k - 1$ dans chaque arête
- méta-structure plus complexe

Modèles de coopération

Coopération locale (Rollik)

Deux robots ne peuvent communiquer que s'ils sont sur le **même nœud** au **même instant**.

Coopération globale

Les robots **communiquent en permanence**.
En particulier, ils se communiquent les rencontres.

Modèles de coopération

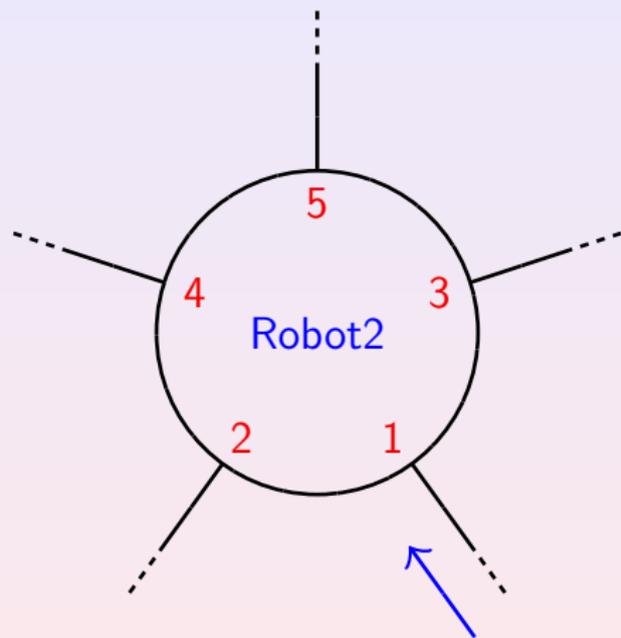
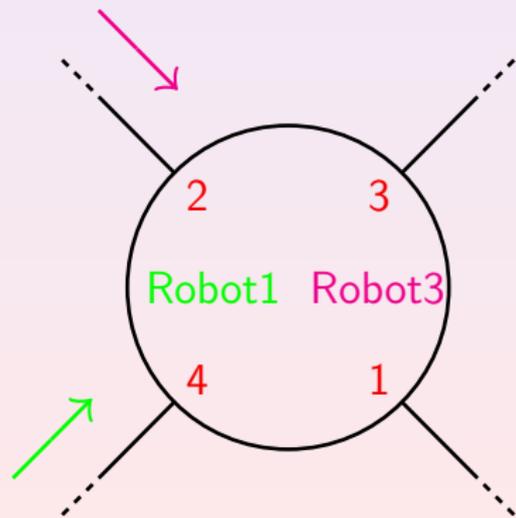
Coopération locale (Rollik)

Deux robots ne peuvent communiquer que s'ils sont sur le **même nœud** au **même instant**.

Coopération globale

Les robots **communiquent en permanence**.
En particulier, ils se communiquent les rencontres.

Exemple



Hydre

connaissance totale \iff état unique

Définition: Hydre

Automate fini à têtes multiples

k automates coopérant globalement \iff une hydre à k têtes

Hydre

connaissance totale \iff état unique

Définition: Hydre

Automate fini à têtes multiples

k automates coopérant globalement \iff une hydre à k têtes

Hydre

connaissance totale \iff état unique

Définition: Hyde

Automate fini à têtes multiples

k automates coopérant globalement \iff une hydre à k têtes

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

-
-
-
-
-

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Puissance de l'hydre

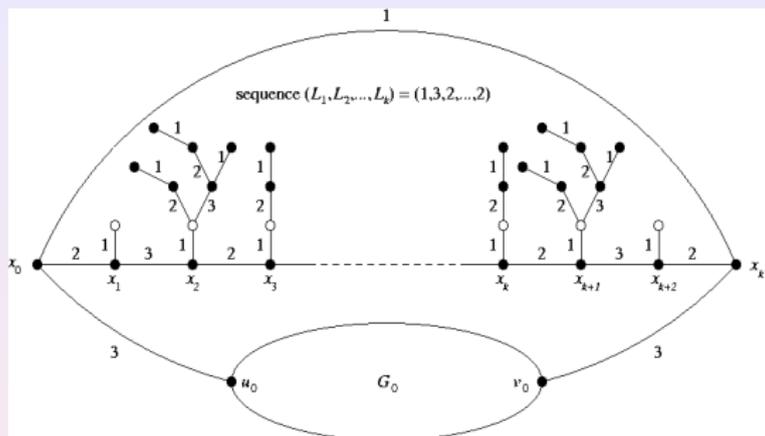
Fraigniaud, Ilcinkas, Markou, Pelc

Power of communication in cooperative exploration of graphs

- une hydre à 2 têtes $\not\asymp$ une équipe finie d'automates finis
- une hydre à k têtes \succ k automates coopérant localement
- une hydre à 2 têtes n'est pas universelle
- une hydre à 3 têtes \succ une hydre à 2 têtes
- une hydre à 3 têtes n'est pas universelle

Conjecture : Aucune hydre n'est universelle

Hydre vs local



Conclusion

une hydre à 2 têtes $\not\approx$ une équipe finie d'automates finis

Corollaire

une hydre à k têtes \succ k automates coopérant localement

Piège pour une hydre à 2 têtes

Graphe d -homogène

- d -régulier
- à arêtes colorées (même étiquette aux deux extrémités)

Dans les graphes d -homogènes :
hydre à deux têtes \equiv équipe de deux automates

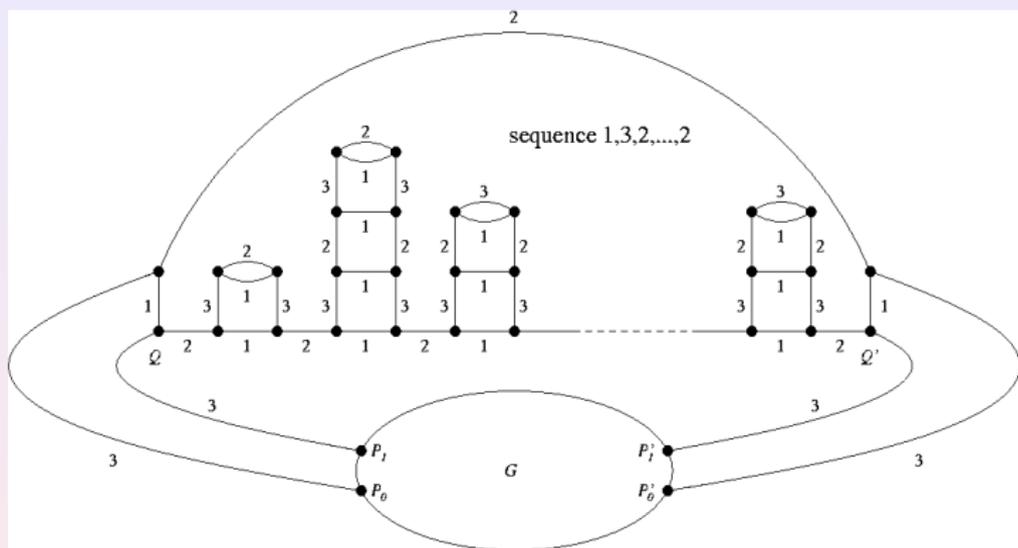
Piège pour une hydre à 2 têtes

Graphe d -homogène

- d -régulier
- à arêtes colorées (même étiquette aux deux extrémités)

Dans les graphes d -homogènes :
hydre à deux têtes \equiv équipe de deux automates

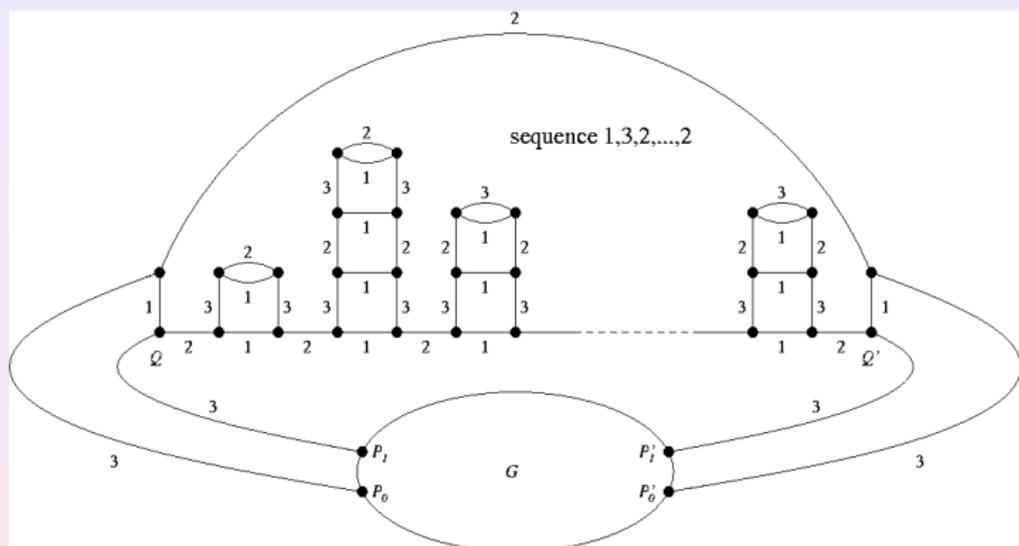
2-hydre vs 3-hydre



Conclusion

une hydre à 3 têtes \succ une hydre à 2 têtes

2-hydre vs 3-hydre



Conclusion

une hydre à 3 têtes \succ une hydre à 2 têtes

Piège pour une hydre à 3 têtes

- Têtes séparées
 - Têtes piégées dans une méta-structure
 - Simulation par un gros automate
 - Têtes piégées dans une méta-méta-structure
- Têtes non séparées (distance bornée les unes des autres)
 - Simulable par un gros automate
 - Têtes piégées dans une méta-méta-structure

Piège pour une hydre à 3 têtes

- **Têtes séparées**
 - Deux têtes (ensemble ou pas)
 - Aucune information utile de la troisième tête
 - Piège pour une hydre à deux têtes
 - Une tête séparée
 - Reçoit des informations périodiques des deux autres
 - Simulable par un gros automate
 - Tête piégée dans une méta-structure
- Têtes non séparées (distance bornée les unes des autres)
 - Simulable par un gros automate
 - Têtes piégées dans une méta-méta-structure

Piège pour une hydre à 3 têtes

- **Têtes séparées**
 - **Deux têtes** (ensemble ou pas)
 - Aucune information utile de la troisième tête
 - Piège pour une hydre à deux têtes
 - **Une tête séparée**
 - Reçoit des informations périodiques des deux autres
 - Simulable par un gros automate
 - Tête piégée dans une méta-structure
- **Têtes non séparées** (distance bornée les unes des autres)
 - Simulable par un gros automate
 - Têtes piégées dans une méta-méta-structure

Piège pour une hydre à 3 têtes

- **Têtes séparées**
 - **Deux têtes** (ensemble ou pas)
 - Aucune information utile de la troisième tête
 - Piège pour une hydre à deux têtes
 - **Une tête séparée**
 - Reçoit des informations périodiques des deux autres
 - Simulable par un gros automate
 - Tête piégée dans une méta-structure
- **Têtes non séparées** (distance bornée les unes des autres)
 - Simulable par un gros automate
 - Têtes piégées dans une méta-méta-structure

Piège pour une hydre à 3 têtes

- **Têtes séparées**
 - **Deux têtes** (ensemble ou pas)
 - Aucune information utile de la troisième tête
 - Piège pour une hydre à deux têtes
 - **Une tête séparée**
 - Reçoit des informations périodiques des deux autres
 - Simulable par un gros automate
 - Tête piégée dans une méta-structure
- **Têtes non séparées** (distance bornée les unes des autres)
 - Simulable par un gros automate
 - Têtes piégées dans une méta-méta-structure

Plan

- 1 Introduction
- 2 Faisabilité
- 3 Minimiser la mémoire**
 - Différentes tâches
 - Arbres
 - Graphes arbitraires
- 4 Autres modèles
- 5 Bonus

Différentes tâches

- Exploration perpétuelle
- Exploration avec arrêt
le robot doit s'arrêter en étant sûr d'avoir fini l'exploration
- Exploration avec retour
le robot doit revenir s'arrêter sur le sommet de départ
- Cartographie
le robot doit fournir une carte étiquetée isomorphe au graphe

Arbres

Diks, Fraigniaud, Kranakis, Pelc, SODA 2002

Tree exploration with little memory

- Perpétuelle : $\Theta(\log \Delta)$ bits
- Avec arrêt : $\Omega(\log \log \log n)$ bits
- Avec retour : $\Omega(\log n)$, $O(\log^2 n)$ bits

Arbres

Diks, Fraigniaud, Kranakis, Pelc, SODA 2002

Tree exploration with little memory

- Perpétuelle : $\Theta(\log \Delta)$ bits
- Avec arrêt : $\Omega(\log \log \log n)$ bits
- Avec retour : $\Omega(\log n)$, $O(\log^2 n)$ bits

Arbres

Diks, Fraigniaud, Kranakis, Pelc, SODA 2002

Tree exploration with little memory

- Perpétuelle : $\Theta(\log \Delta)$ bits
- Avec arrêt : $\Omega(\log \log \log n)$ bits
- Avec retour : $\Omega(\log n)$, $O(\log^2 n)$ bits

Graphes arbitraires

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

Exploration perpétuelle :

- Pour tout automate à K états, il existe un piège d'au plus $K + 1$ nœuds.
- Donc un automate qui explore tous les graphes d'au plus K sommets a au moins K états.
- DFS est optimal en mémoire : $\Theta(D \log \Delta)$ bits

Graphes arbitraires

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

Exploration perpétuelle :

- Pour tout automate à K états, il existe un piège d'au plus $K + 1$ nœuds.
- Donc un automate qui explore tous les graphes d'au plus K sommets a au moins K états.
- DFS est optimal en mémoire : $\Theta(D \log \Delta)$ bits

Graphes arbitraires

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

Exploration perpétuelle :

- Pour tout automate à K états, il existe un piège d'au plus $K + 1$ nœuds.
- Donc un automate qui explore tous les graphes d'au plus K sommets a au moins K états.
- **DFS est optimal en mémoire** : $\Theta(D \log \Delta)$ bits

DFS est optimal

Algorithme (borne supérieure)

- Depth-first search (**DFS**) de profondeur croissante
- Mémoire : pile des numéros de port menant à la racine
→ $O(D \log \Delta)$ bits

Borne inférieure

Attacher un arbre au piège pour réduire le diamètre.

DFS est optimal

Algorithme (borne supérieure)

- Depth-first search (**DFS**) de profondeur croissante
- Mémoire : pile des numéros de port menant à la racine
→ $O(D \log \Delta)$ bits

Borne inférieure

Attacher un arbre au piège pour réduire le diamètre.

Plan

- 1 Introduction
- 2 Faisabilité
- 3 Minimiser la mémoire
- 4 Autres modèles**
 - Minimiser le temps
 - Autres cadres
 - Problème continu
 - UTS, UXS
- 5 Bonus

Minimiser le temps

Temps = nombre de traversées d'arêtes

Dudek, Jenkin, Milos, Wilkes, IEEE TRA 1991

Robotic exploration as graph construction

- Cartographie avec cailloux en temps $O(mn)$

Deng, Mirzajan, IEEE 1996

Competitive robot mapping with homogeneous markers

- Competitive ratio: cartographie / vérification de carte
- Avec un seul caillou, l'algorithme de DJMW est optimal (relaxed depth-one strategies)

Minimiser le temps

Temps = nombre de traversées d'arêtes

Dudek, Jenkin, Milios, Wilkes, IEEE TRA 1991

Robotic exploration as graph construction

- Cartographie avec cailloux en temps $O(mn)$

Deng, Mirzavari, IEEE 1996

Competitive robot mapping with homogeneous markers

- Competitive ratio: cartographie / vérification de carte
- Avec un seul caillou, l'algorithme de DJMW est optimal (relaxed depth-one strategies)

Minimiser le temps

Temps = nombre de traversées d'arêtes

Dudek, Jenkin, Milios, Wilkes, IEEE TRA 1991

Robotic exploration as graph construction

- Cartographie avec cailloux en temps $O(mn)$

Deng, Mirzaian, IEEE 1996

Competitive robot mapping with homogeneous markers

- Competitive ratio: cartographie / vérification de carte
- Avec un seul caillou, l'algorithme de DJMW est optimal (relaxed depth-one strategies)

Autres cadres

- Graphes orientés
- Graphes pondérés
- Graphes orientés
- Exploration par morceaux

Autres cadres

- Graphes étiquetés
- Graphes géométriques
- Graphes orientés
- Exploration par morceaux

Autres cadres

- Graphes étiquetés
- Graphes géométriques
- Graphes orientés
- Exploration par morceaux

Autres cadres

- Graphes étiquetés
- Graphes géométriques
- Graphes orientés
- Exploration par morceaux

Autres cadres

- Graphes étiquetés
- Graphes géométriques
- Graphes orientés
- Exploration par morceaux

Pièces avec obstacles

Deng, Kameda, Papadimitriou, JACM 1998

How to learn an unknown environment I: the rectilinear case



Pièces avec obstacles

Deng, Kameda, Papadimitriou, JACM 1998

How to learn an unknown environment I: the rectilinear case



Pièces avec obstacles

Deng, Kameda, Papadimitriou, JACM 1998

How to learn an unknown environment I: the rectilinear case

- GT : **Gallery Tour**
Visiter tout le musée, depuis l'entrée jusqu'à la sortie
- WR : **Watchman's Route**
Passer par tous les recoins du musée et revenir au point de départ

Pièces avec obstacles

Deng, Kameda, Papadimitriou, JACM 1998

How to learn an unknown environment I: the rectilinear case

- GT : **Gallery Tour**
Visiter tout le musée, depuis l'entrée jusqu'à la sortie
- WR : **Watchman's Route**
Passer par tous les recoins du musée et revenir au point de départ

UTS, UXS

UTS = Universal Traversal Sequence

Séquence de numéros de port, i.e., d'arêtes à prendre

UXS = Universal Exploration Sequence

Séquence d'incréments de numéros de port

"0" signifie "revenir en arrière"

"1" signifie "prendre la première à droite" par exemple

But

- minimiser la taille de la séquence
- minimiser la quantité de mémoire pour la construire

UTS, UXS

UTS = Universal Traversal Sequence

Séquence de numéros de port, i.e., d'arêtes à prendre

UXS = Universal Exploration Sequence

Séquence d'incréments de numéros de port

"0" signifie "revenir en arrière"

"1" signifie "prendre la première à droite" par exemple

But

- minimiser la taille de la séquence
- minimiser la quantité de mémoire pour la construire

UTS, UXS

UTS = Universal Traversal Sequence

Séquence de numéros de port, i.e., d'arêtes à prendre

UXS = Universal Exploration Sequence

Séquence d'incréments de numéros de port

"0" signifie "revenir en arrière"

"1" signifie "prendre la première à droite" par exemple

But

- minimiser la taille de la séquence
- minimiser la quantité de mémoire pour la construire

Plan

- 1 Introduction
- 2 Faisabilité
- 3 Minimiser la mémoire
- 4 Autres modèles
- 5 Bonus**
 - Base
 - Trois couleurs
 - Seulement deux couleurs?

Colorier les nœuds

Un oracle colorie (étiquette) les nœuds pour aider les automates.

Idée de base

arbre couvrant : l'étiquette indique quelles arêtes appartiennent à l'arbre

Etiquetage amélioré

seulement indiquer l'arête menant au père \longrightarrow Δ couleurs

Colorier les nœuds

Un oracle colorie (étiquette) les nœuds pour aider les automates.

Idée de base

arbre couvrant : l'étiquette indique quelles arêtes appartiennent à l'arbre

Etiquetage amélioré

seulement indiquer l'arête menant au père $\longrightarrow \triangle$ couleurs

Colorier les nœuds

Un oracle colorie (étiquette) les nœuds pour aider les automates.

Idée de base

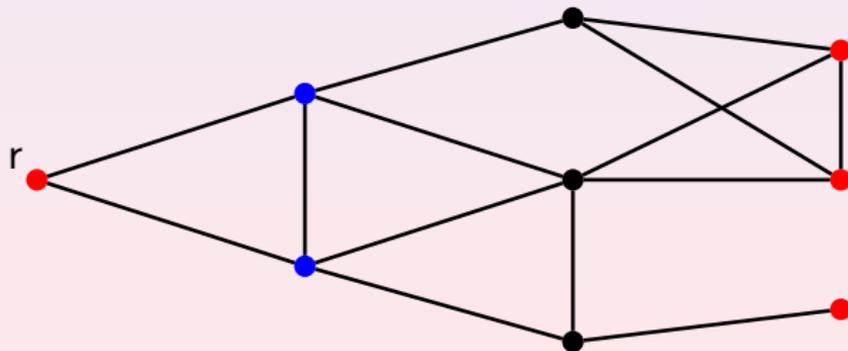
arbre couvrant : l'étiquette indique quelles arêtes appartiennent à l'arbre

Etiquetage amélioré

seulement indiquer l'arête menant au père \longrightarrow Δ couleurs

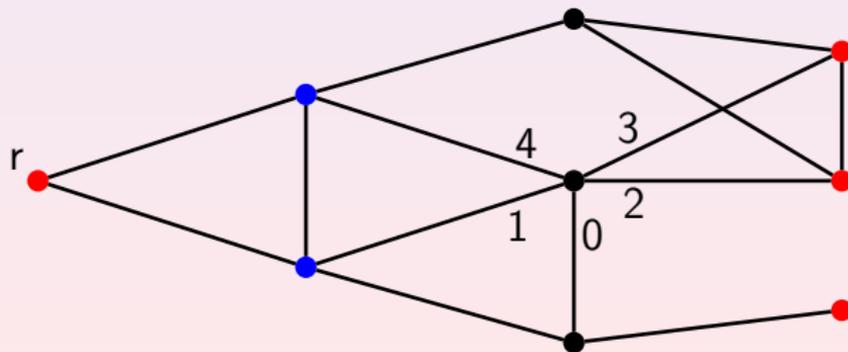
Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
 - distance $d \cong 0[n]$ rouge
 - distance $d \cong 1[n]$ bleu
 - distance $d \cong 2[n]$ noir



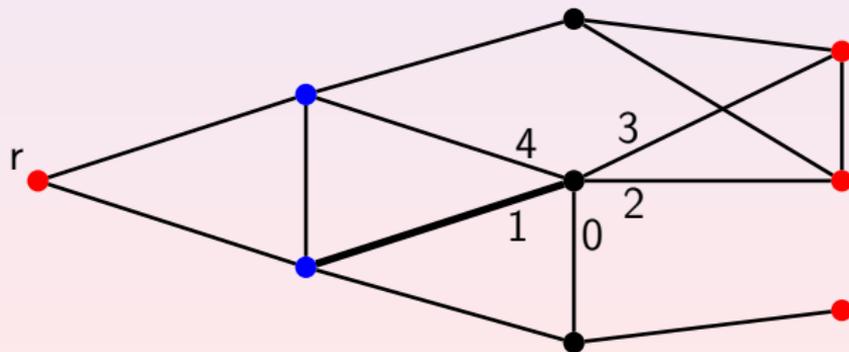
Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
 - distance $d \cong 0[n]$ rouge
 - distance $d \cong 1[n]$ bleu
 - distance $d \cong 2[n]$ noir



Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
 - distance $d \cong 0[n]$ rouge
 - distance $d \cong 1[n]$ bleu
 - distance $d \cong 2[n]$ noir



Seulement deux couleurs ?

- Layer 1: rouge
- Layer 2: bleu
- Layer 3: rouge
- Layer 4: rouge
- Layer 5: rouge
- Layer 6: bleu
- Layer 7: bleu
- Layer 8: bleu

Plan

- 1 Introduction
- 2 Faisabilité
- 3 Minimiser la mémoire
- 4 Autres modèles
- 5 Bonus
- 6 Conclusion**

Problèmes ouverts

Hydre

- $\forall k$ une hydre à $(k + 1)$ têtes \succ une hydre à k têtes ?
- Existe-t-il k tel que l'hydre à k têtes soit universelle ?

USTCON

- USTCON est-il dans log-space ($SL = L$) ? **OUI**
- Meilleure compréhension des UTS, UXS, automates, hydres.

Problèmes ouverts

Hydre

- $\forall k$ une hydre à $(k + 1)$ têtes \succ une hydre à k têtes ?
- Existe-t-il k tel que l'hydre à k têtes soit universelle ?

USTCON

- USTCON est-il dans log-space ($SL = L$) ? OUI
- Meilleure compréhension des UTS, UXS, automates, hydres.

Problèmes ouverts

Hydre

- $\forall k$ une hydre à $(k + 1)$ têtes \succ une hydre à k têtes ?
- Existe-t-il k tel que l'hydre à k têtes soit universelle ?

USTCON

- USTCON est-il dans log-space ($SL = L$) ? **OUI**
- Meilleure compréhension des UTS, UXS, automates, hydres.