Algorithmique Distribuée

Cours 5

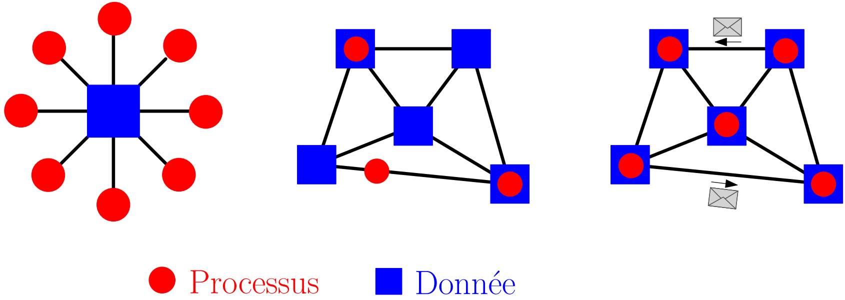
Shantanu Das

http://pageperso.lif.univ-mrs.fr/~shantanu.das/M1algodist/

Aix-Marseille Université

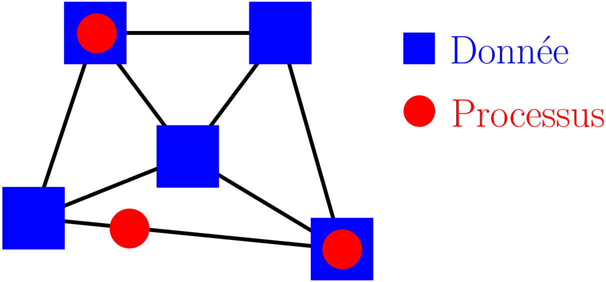
27/28 mars 2018

Trois types de modéles pour l'algorithmique distribuée



(1)Mémoire partagée (2)Agents Mobiles (3)Passage de message

Agents mobiles



- n données
- k processus qui exécutent le même algorithme et peuvent se déplacer dans le réseau
- Complexité : nombre de déplacements

Agent mobile

Agent mobile

Un logiciel mobile sur un réseau :

- Il peut se déplacer d'un site à un autre pour accéder à des données ou à des ressources.
- Il se déplace avec son code et ses données propres, mais aussi avec son état d'exécution.

Attention : L'agent décide lui-même de manière autonome de ses mouvements.

Pourquoi utiliser les agents mobiles ?

Les agents mobiles permet de

- Réduire le nombre et le volume des interactions distantes (e.g. entre clients et serveurs)
- Améliorer la performance ou satisfaire la tolérance aux pannes, ou réduire le trafic sur le réseau
- La mobilité du code offre un niveau de flexibilité aux applications.

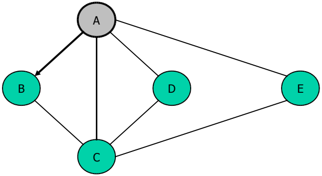
Algorithmes pour les robots mobiles

Le modèle d'agents mobile est également utilisé pour conceptualiser les algorithmes pour un équipe de robots autonomes.

Problèmes fundamentaux

- **Exploration :** construire la carte d'un réseau inconnu
- **Mise à jour :** parcourir le réseau de manière périodique pour mettre à jour les nœuds
- **Rendez-vous :** faire se rencontrer 2 ou plus agents
- **Recherche de trous noirs :** trouver dans le réseaux des nœuds "crashés" qui tue tous les agents les atteignant
- **Capture :** trouver des intrus dans un réseau

1. Exploration d'un réseau



- L'agent peut sortir d'un sommet par une de liens incidents.
  Les liens incidentes à un sommet sont numérotées localement.
- Explorer le graphe par l'algorithme DFS (Parcours en profondeur)
  L'agent doit mémorisé la partie du graphe déjà visité

2. Problème de Rendez-vous

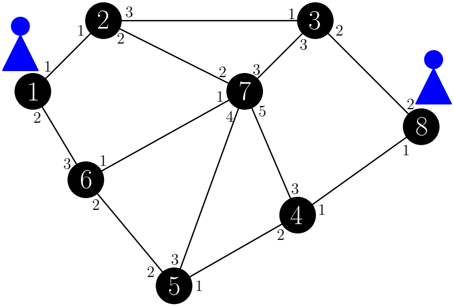
Le problème

Deux (ou plus) agents mobiles doivent se rencontrer dans un réseau (graphe).

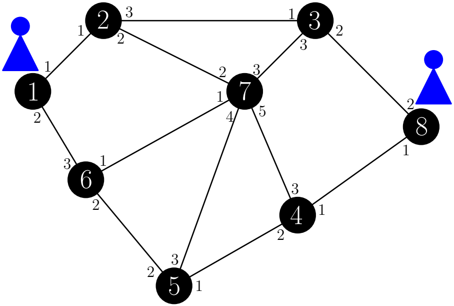
Les agents commence au sommets distincts.

- Computability : Existe-il toujours une solution ?
- Complexité :
  1. Longueur des routes des agents jusqu'au rendez-vous
  2. Nombre de déplacements en totale (*move complexity*)

Rendez-vous dans un graphe non-anonyme



Rendez-vous dans un graphe non-anonyme



- Faire un exploration DFS du graphe.
- Rendez-vous au sommet avec le minimum UID.
- Complexité : $O(m)$ dans un graphe de m arêtes.
(Mémoire : $O(n \log(n))$ bits pour chaque agent)

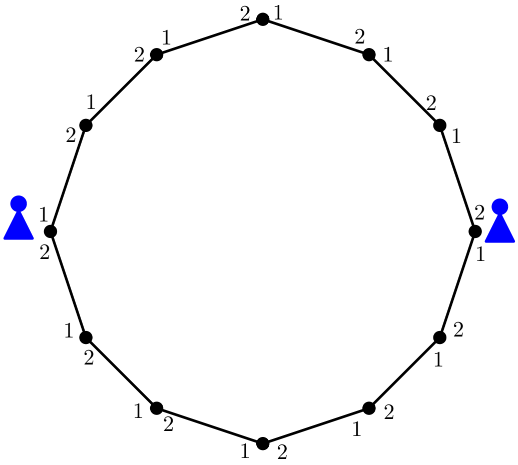
Rendez-vous dans un graphe anonyme

Graphe Anonyme => les sommets ont pas d'identifiants.

- Si les agents sont anonymes aussi ?
  Pas toujours possible de rendez-vous.
- Si les agents sont non-anonymes ?

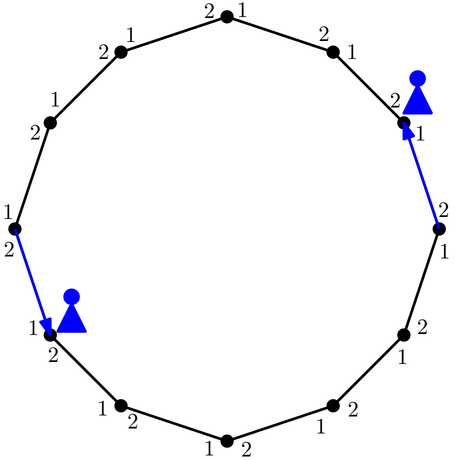
Rendez-vous dans un anneau anonyme

Les agents sont anonymes...



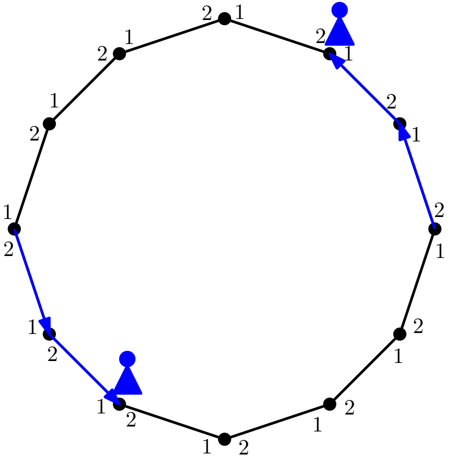
Rendez-vous dans un anneau anonyme

Les agents sont anonymes...



Rendez-vous dans un anneau anonyme

Les agents sont anonymes...



Rendez-vous n'est pas possible !

Rendez-vous dans un anneau anonyme

Agents avec identifiants...

- Si les agents connais les identifiant de l'autre agent :

Rendez-vous dans un anneau anonyme

Agents avec identifiants...

- Si les agents connais les identifiant de l'autre agent :

  Stratégie attendre maman (*Wait for Mommy*) :

  - L'agent A avec la plus petit identifiant attend
  - L'autre agent B explore le graphe jusqu'á qu'il trouve l'agent A.

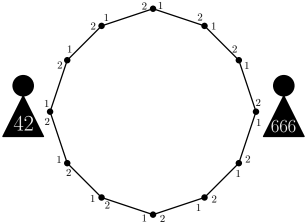
- Si les agents connais seulement son propre ID (pas celui d'autres) ?

Anneau anonyme, Agents ont UIDs

Les agents ont chacun un identifiant distinct choisi parmi une infinité de possibilités (entiers).

Exercice

Est-ce que le rendez-vous d'agents ayant des identifiants distincts dans un anneau anonyme est toujours possible si on suppose que les agents connaissent n ?



Solution de l'exercice

Chaque agent a identifiant $id \in \mathbb{N}$

Algorithme :
Faire $id \times n$ déplacements dans une direction.

Preuve de correction :
Les deux agents se rencontrent car un des deux a fait un tour de plus que l'autre.

Complexité : $O(n \times \min(id_1, id_2))$
avec id_1 et id_2 les identifiants des deux agents.

Rendez-vous dans un arbre anonyme

Réseau Synchrone, Topologie d'Arbre

Algorithm RV-arbre(ℓ)

Explorer de l'arbre ;

Si il y a un noeud central **alors**

rendez-vous ce noeud

Sinon

Pour chaque bit b de son ID

Si $b = 0$, traverser l'arête centrale ;

Si $b = 1$, rester sur place ;

(Et si le réseau est asynchrone ?)

Rendez-vous dans un graphe arbitraire

- Réseau Synchrone
- Chaque agent a identifiant $id \in \mathbb{N}$
- Chaque agent connais un carte du graphe

Algorithme :

Pour chaque bit b de son ID

Si $b = 0$, traverser toute le graphe ;

Si $b = 1$, rester sur place pour $2n$ rondes ;

Preuve de correction :

Les deux agents se rencontrent car un des deux a fait un tour pendant que l'autre attend.

Complexité : $O(n \times \log(ID))$

Rendez-vous dans un graphe inconnu

- Si le graphe est inconnu, au début.
- Est-ce que c'est possible d'explorer le graphe ?

Rendez-vous dans un graphe inconnu

- Si le graphe est inconnu, au début.
- Est-ce que c'est possible d'explorer le graphe ?
- Sans aucun connaissance (topologie, taille), exploration est impossible !

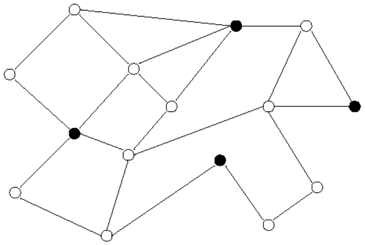
Exploration avec marquage

Modèle de tableau blanc

On suppose qu'il y a un tableau blanc dans chaque sommet. Un agent peut écrire de l'information sur le tableau de son sommet courant.

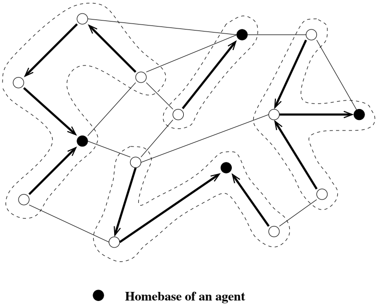
- Un seule agent
- Marquer chaque sommet au première visite, mémoriser une carte du partie visité
- L'agent peut visiter chaque sommet du graphe en utilisant l'algorithme DFS.

Exploration avec marquage



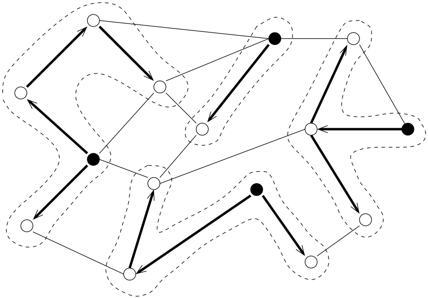
- S'il y $k > 1$ agents (qui commence au sommets distincts)
- Chaque agent exécute l'algorithme DFS.
- Les agents vont ensemble visiter toute les sommets (chaque sommet va être visité par un agent)

Exploration avec marquage



- S'il y $k > 1$ agents (qui commence au sommets distincts)
- Chaque agent exécute l'algorithme DFS.
- Les agents vont ensemble visiter toute les sommets (chaque sommet va être visité par un agent)

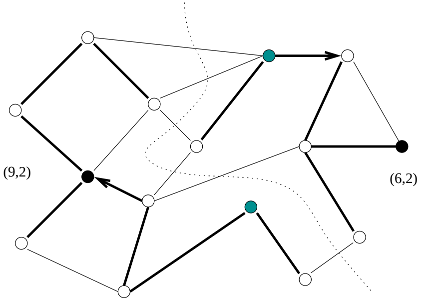
Exploration avec marquage



Comment construire le carte du graphe ?

- Compétition entre les agents.
- Le grand arbre mange le petit arbre.
- Si n et k premiers entre eux, on va réussir de construire un arbre couvrant (et élire un agent).

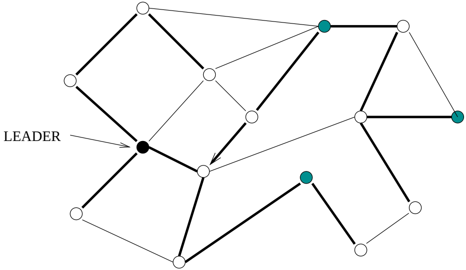
Exploration avec marquage



Comment construire le carte du graphe ?

- Compétition entre les agents.
- Le grand arbre mange le petit arbre.
- Si n et k premiers entre eux, on va réussir de construire un arbre couvrant (et élire un agent).

Exploration avec marquage



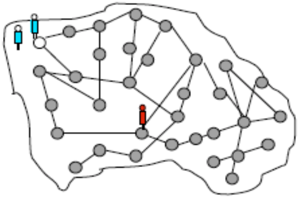
Comment construire le carte du graphe ?

- Compétition entre les agents.
- Le grand arbre mange le petit arbre.
- Si n et k premiers entre eux, on va réussir de construire un arbre couvrant (et élire un agent).

3. Capture des intrus (Network Decontamination)

Capture des Intrus

- Il y a un mauvaise agent (intrus ou virus) dans le réseau.
- Un équipe de bons agents (policiers).
- Le but : Capturer le intrus and décontaminer le réseau



Capture des Intrus

Le Intrus

- Le intrus est beaucoup plus rapide que les policiers.
- Les sommets visités par le intrus sont contaminés.

Les Policiers

- Tout les policiers commence au meme sommet.
- Un sommet deviens propre s'il y au moins une policier sur cette sommet.

Le intrus est capturé quand il est dans un sommet qui contient un policier.
(Supposition : Le intrus peut se cacher dans une arête.)

Capture des Intrus

- Combien de policiers sont suffisant pour capturer un intrus ?
- Quels est la stratégie de policiers ?

Capture des Intrus

- Combien de policiers sont suffisant pour capturer un intrus ?
- Quels est la stratégie de policiers ?

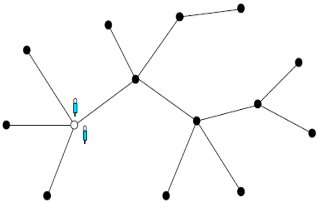
Exemple : Dans un Anneau

- 1 policier : pas suffisant
- 2 policiers : suffisant

Dans un Arbre, combien de policiers sont nécessaire ?

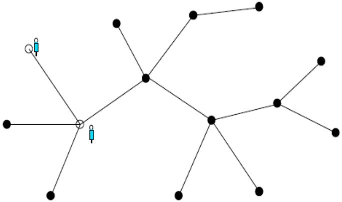
Capture des Intrus dans les arbres

Dans un Arbre, combien de policiers sont nécessaire ?



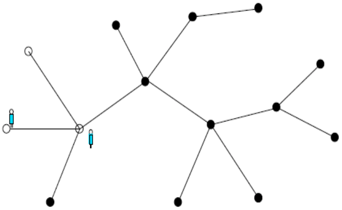
Capture des Intrus dans les arbres

Dans un Arbre, combien de policiers sont nécessaire ?



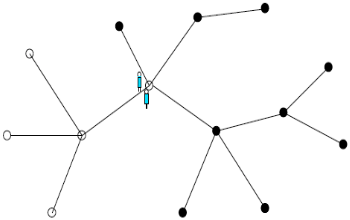
Capture des Intrus dans les arbres

Dans un Arbre, combien de policiers sont nécessaire ?



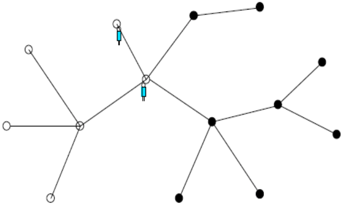
Capture des Intrus dans les arbres

Dans un Arbre, combien de policiers sont nécessaire ?



Capture des Intrus dans les arbres

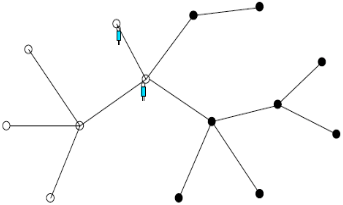
Dans un Arbre, combien de policiers sont nécessaire ?



Dans cette arbre 2 policiers sont suffisant.

Capture des Intrus dans les arbres

Dans un Arbre, combien de policiers sont nécessaire ?



Dans cette arbre 2 policiers sont suffisant.
Complexité : $O(n)$ mouvements en totale.

Capture des Intrus dans un graphe

Mais 2 policiers suffise pas dans toutes les arbres !

Dans un graphe quelconque, combien de policiers sont nécessaire ?

Capture des Intrus dans un graphe

Mais 2 policiers suffise pas dans toutes les arbres !

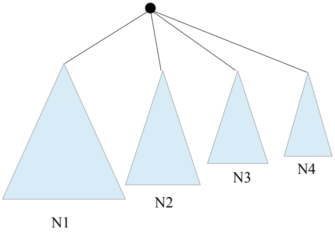
Dans un graphe quelconque, combien de policiers sont nécessaire ?

- *Search-Number*(G) : le minimum nombre de policiers nécessaire pour décontaminer un graphe G
- C'est NP-dur de trouver le *search-number* de un graphe G.
- Si G est un arbre, c'est facile de calculer le nombre de policiers nécessaire pour décontaminer G a partir de un sommet v.

Exercice : Quel est le Search-number d'un graphe complet ?

Capture des Intrus dans un arbre

Calculer le nombre d'agents nécessaire pour décontaminer un arbre T.



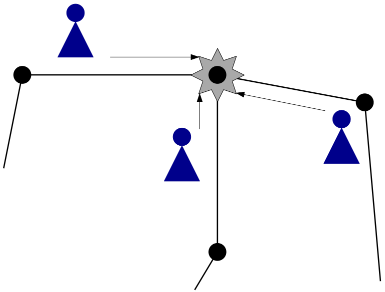
Nombre de policiers nécessaire = Max (N1, 1+N2)
Si $N1 \geq N2 \geq N3 \dots$

4. Recherche de Trou Noir (Black-Hole Search)

Recherche de trou noir

Trou Noir

Un Trou Noir est un sommet dangereux qui détruit n'importe quel agent qui y rentre.



Recherche de trou noir

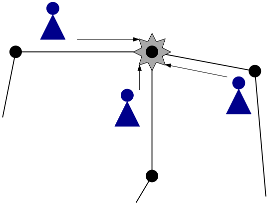
Explorer G et localiser le trou noir

Minimiser :

- # agents
- # déplacements

Recherche de trou noir

- Il y a un seule trou noir dans le graphe.
- Le graphe sans le trou noir est connexe.
- Le but : Trouver chaque lien qui est incident au trou noir.



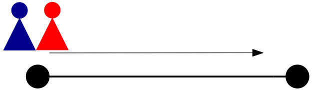
Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Mécanisme de time-out



Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Mécanisme de time-out

Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Mécanisme de time-out

Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Mécanisme de time-out

Trou noir dans le réseau synchrone

- S'il le réseau est synchrone et,
- les agents commence au même sommet

Mécanisme de time-out

Si l'agent reviens pas,
le prochaine sommet est le trou noir.

Trou noir dans le réseau asynchrone

Réseau Asynchrone => Time-Out marche pas !

Mécanisme de *Cautious-Walk*

- Utiliser les tableau blancs.
- Avant de traverser un arête e pour le premiére fois, Marquer dans le tableau "DANGER : (e)"
- Si la prochaine sommet est sûr, reviens tout de suite et effacer le marque.
- Un agent va jamais traverser un arête qui est marqué "DANGER".

Recherche de Trou noir avec tableau blanc

Theoreme

S'il existe un trou noir de degré Δ dans un graphe inconnu G, $\Delta + 1$ agents sont nécessaire et suffisant pour le recherche de trou noir.

Recherche de Trou noir avec tableau blanc

Theoreme

S'il existe un trou noir de degré Δ dans un graphe inconnu G, $\Delta + 1$ agents sont nécessaire et suffisant pour le recherche de trou noir.

Algorithme :

- Au début, le partie exploré est seulement le sommet de départ.
- S'il y a un arête inexploré e incident à la partie déjà exploré, la prochaine agent disponible traverse l'arête e, en utilisant le *Cautious-Walk*.
- Si l'agent a réussi, il reviens au sommet de départ.

Recherche de Trou noir avec tableau blanc

Theoreme

S'il existe un trou noir de degré Δ dans un graphe inconnu G, $\Delta + 1$ agents sont nécessaire et suffisant pour le recherche de trou noir.

Algorithme :

- Au début, le partie exploré est seulement le sommet de départ.
- S'il y a un arête inexploré e incident à la partie déjà exploré, la prochaine agent disponible traverse l'arête e, en utilisant le *Cautious-Walk*.
- Si l'agent a réussi, il reviens au sommet de départ.

Complexité : # déplacements $= O(mn)$

Recherche de Trou noir avec tableau blanc

Si le graphe G est connu au départ (on a un carte de G), combien d'agents sont nécessaire ?

Recherche de Trou noir avec tableau blanc

Si le graphe G est connu au départ (on a un carte de G), combien d'agents sont nécessaire ?
2 agents suffice !

1. Diviser le partie inexploré a deux sous-graphes disjoint.
2. Chaque agent traverse son partie du graphe en utilisant le *Cautious-Walk*.
3. Si l'agent a réussi, il va chercher l'autre (en utilisant les arrêts sûr).
4. Répéter étapes 1,2, et 3.

Recherche de Trou noir avec tableau blanc

Si le graphe G est connu au départ (on a un carte de G), combien d'agents sont nécessaire ?
2 agents suffice !

1. Diviser le partie inexploré a deux sous-graphes disjoint.
2. Chaque agent traverse son partie du graphe en utilisant le *Cautious-Walk*.
3. Si l'agent a réussi, il va chercher l'autre (en utilisant les arrêts sûr).
4. Répéter étapes 1,2, et 3.

Complexité : # déplacements $= O(n \log n)$