

Stratégies d'encerclement réparties dans les graphes

Lélia Blin ¹ Pierre Fraigniaud ² Nicolas Nisse ²
Sandrine Vial ¹

IBISC, Université d'Evry, 91000 Evry, France

CNRS, Lab. de Recherche en Informatique, Université Paris-Sud,
91405 Orsay, France.

2006

Encerclement dans les graphes

But

Dans un graphe simple, non-orienté,

- un **fugitif** invisible, omniscient et arbitrairement rapide
- une équipe d'**agents**

Déterminer une **stratégie** qui capture le fugitif
en utilisant le moins d'**agents possible**.

Motivations

- sécurité dans les réseaux de type internet
- spéléologie
- ...

Encerclement dans les graphes

But

Dans un graphe simple, non-orienté,

- un **fugitif** invisible, omniscient et arbitrairement rapide
- une équipe d'**agents**

Déterminer une **stratégie** qui capture le fugitif
en utilisant le moins d'agents possible.

Motivations

- sécurité dans les réseaux de type internet
- spéléologie
- ...

Encerclement dans les graphes

But

Dans un graphe simple, non-orienté,

- un **fugitif** invisible, omniscient et arbitrairement rapide
- une équipe d'**agents**

Déterminer une **stratégie** qui capture le fugitif
en utilisant le moins d'agents possible.

Motivations

- sécurité dans les réseaux de type internet
- spéléologie
- ...

Stratégies d'Encerclement, Parson. [GTC,1976]

Séquence de trois opérations élémentaires, . . .

- 1 **Placer** un agent sur un sommet du graphe ;
- 2 **Déplacer** un agent le long d'une arête du graphe ;
- 3 **Supprimer** un agent d'un sommet du graphe.

. . . qui doit aboutir à la capture du fugitif

Un sommet occupé par un agent est propre. Le fugitif ne peut croiser un sommet (ni une arête) occupé par un agent.

Il faut minimiser le nombre d'agents.

Soit $s(G)$ le nombre minimum d'agents nécessaires pour capturer un fugitif dans le graphe G .

Stratégies d'Encerclement, Parson. [GTC,1976]

Séquence de trois opérations élémentaires,...

- 1 **Placer** un agent sur un sommet du graphe ;
- 2 **Déplacer** un agent le long d'une arête du graphe ;
- 3 **Supprimer** un agent d'un sommet du graphe.

...qui doit aboutir à la capture du fugitif

Un sommet occupé par un agent est propre. Le fugitif ne peut croiser un sommet (ni une arête) occupé par un agent.

Il faut minimiser le nombre d'agents.

Soit $s(G)$ le nombre minimum d'agents nécessaires pour capturer un fugitif dans le graphe G .

Stratégies d'Encerclement, Parson. [GTC,1976]

Séquence de trois opérations élémentaires,...

- 1 **Placer** un agent sur un sommet du graphe ;
- 2 **Déplacer** un agent le long d'une arête du graphe ;
- 3 **Supprimer** un agent d'un sommet du graphe.

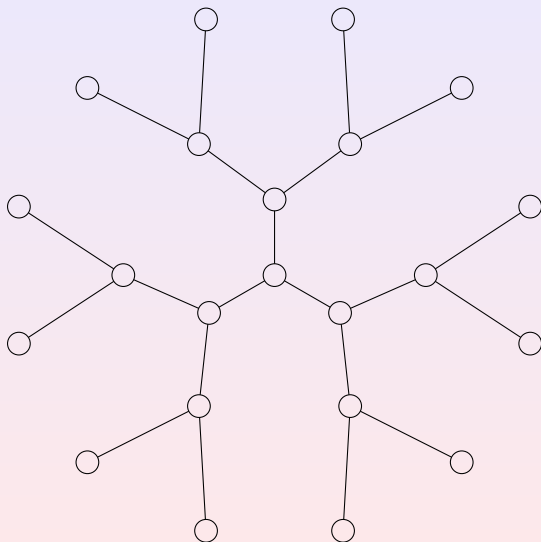
...qui doit aboutir à la capture du fugitif

Un sommet occupé par un agent est propre. Le fugitif ne peut croiser un sommet (ni une arête) occupé par un agent.

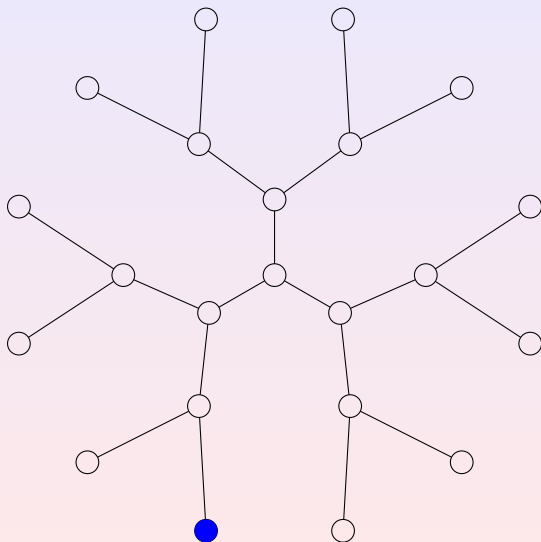
Il faut minimiser le nombre d'agents.

Soit $s(G)$ le nombre minimum d'agents nécessaires pour capturer un fugitif dans le graphe G .

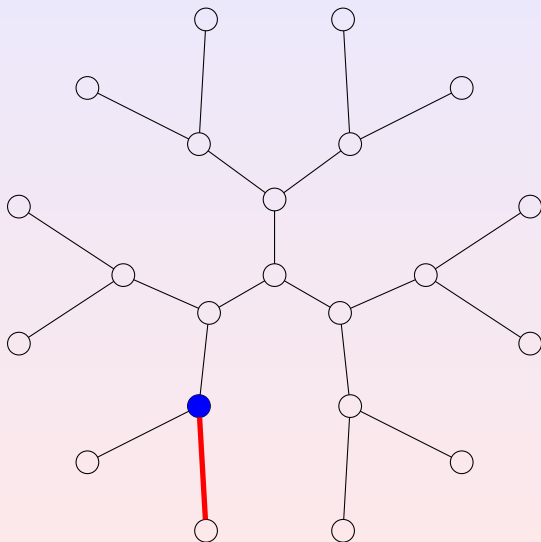
Exemple : encerclement dans un arbre



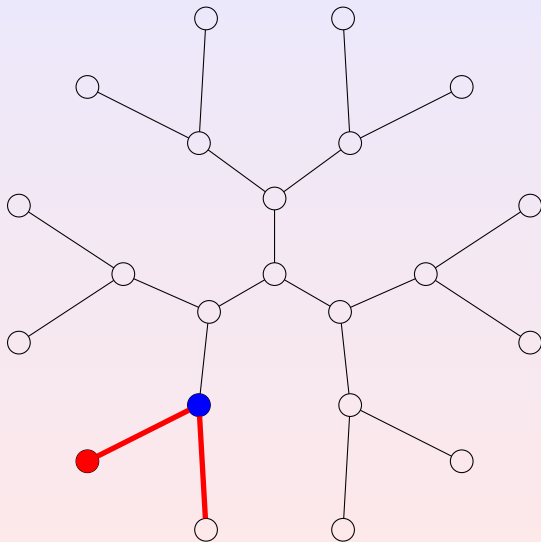
Exemple : encerclement dans un arbre



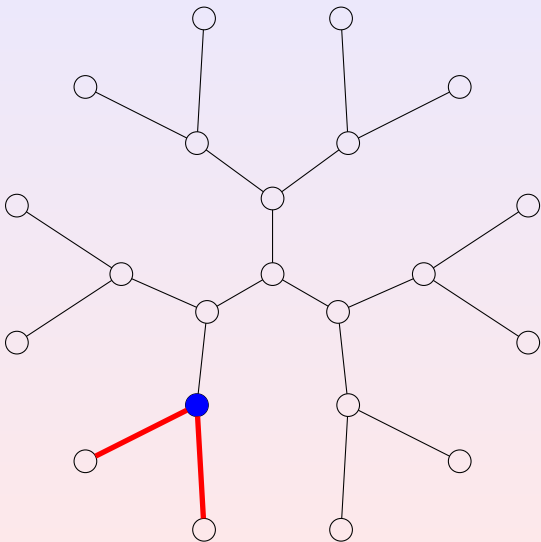
Exemple : encerclement dans un arbre



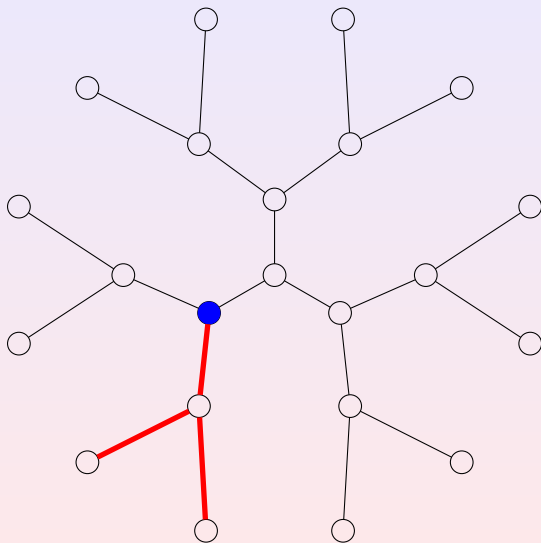
Exemple : encerclement dans un arbre



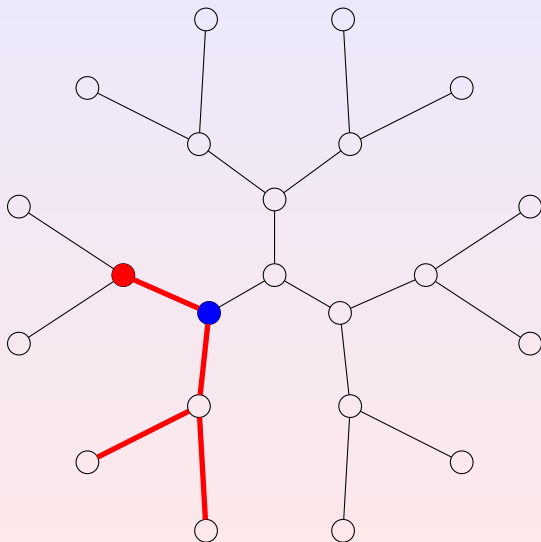
Exemple : encerclement dans un arbre



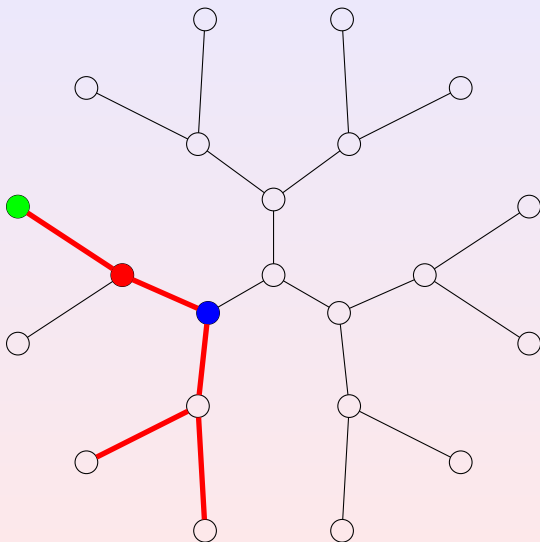
Exemple : encerclement dans un arbre



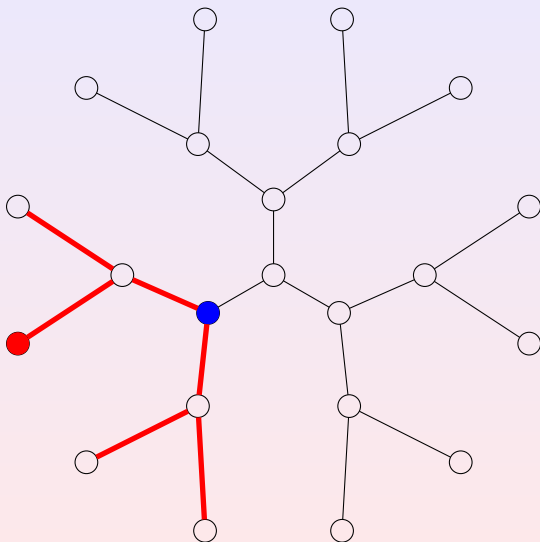
Exemple : encerclement dans un arbre



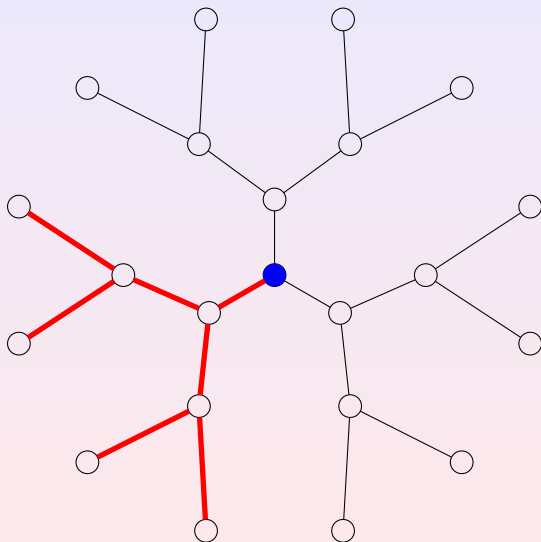
Exemple : encerclement dans un arbre



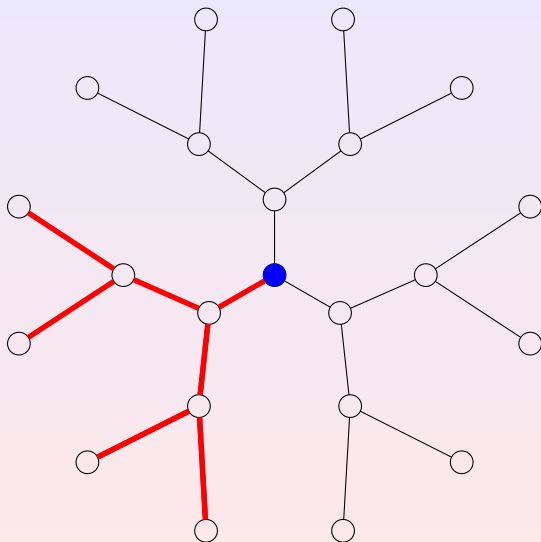
Exemple : encerclement dans un arbre



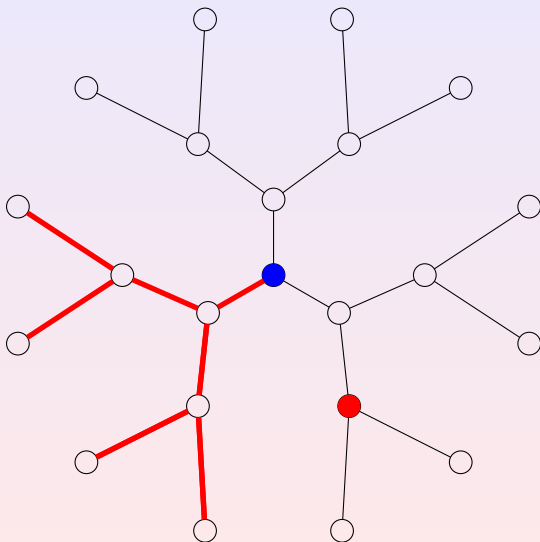
Exemple : encerclement dans un arbre



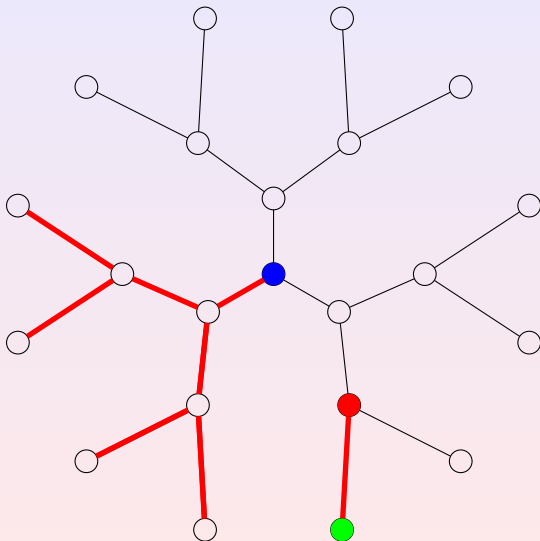
Exemple : encerclement dans un arbre



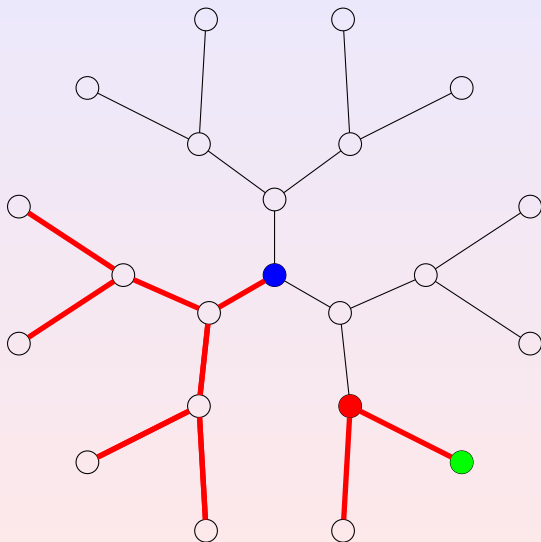
Exemple : encerclement dans un arbre



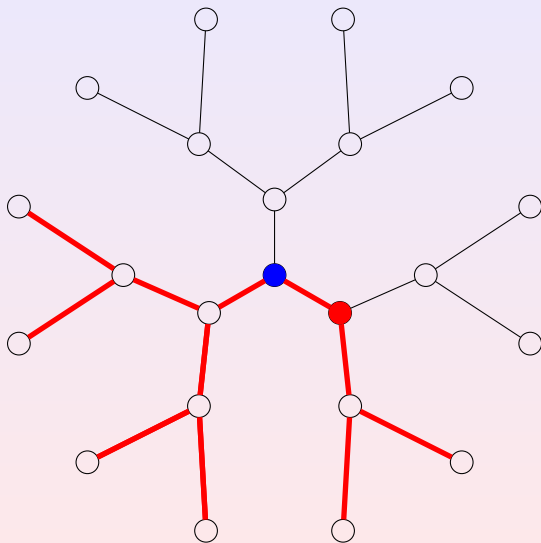
Exemple : encerclement dans un arbre



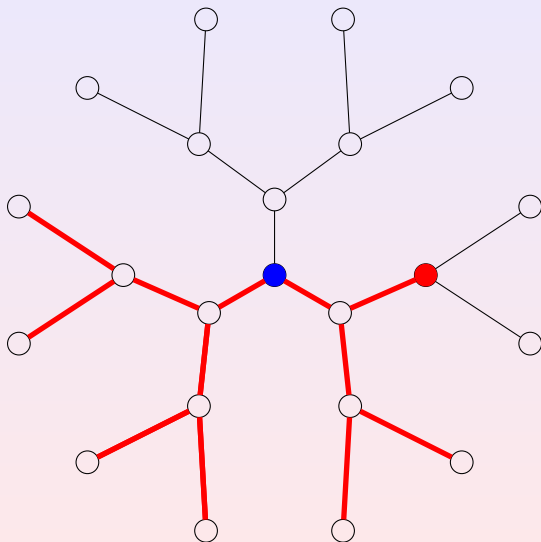
Exemple : encerclement dans un arbre



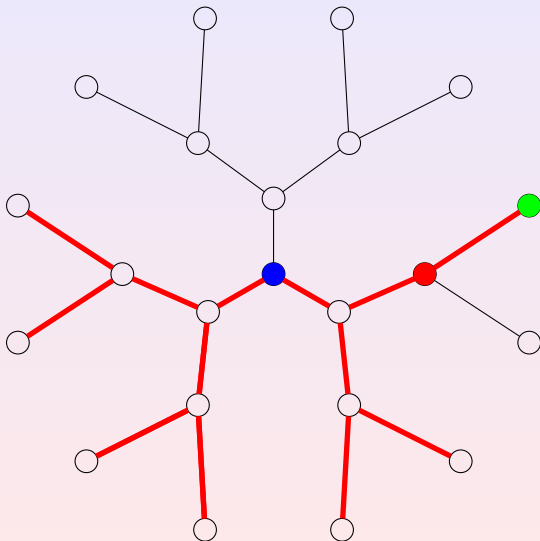
Exemple : encerclement dans un arbre



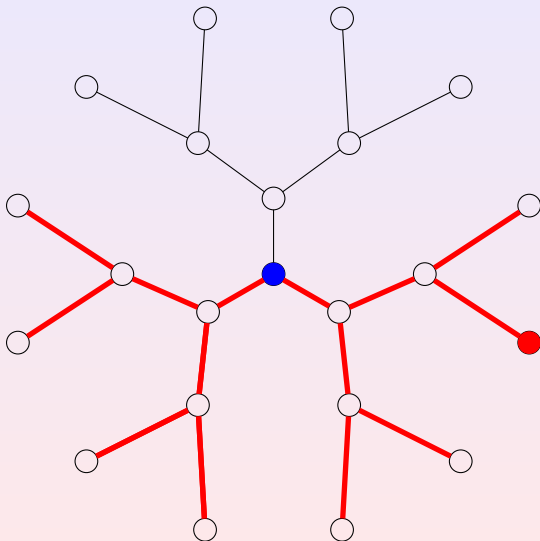
Exemple : encerclement dans un arbre



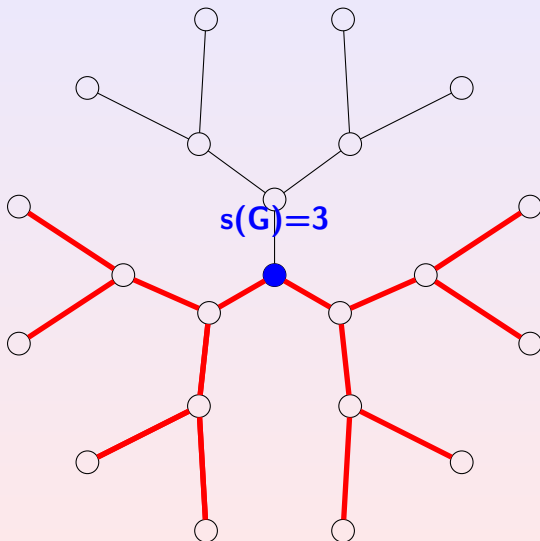
Exemple : encerclement dans un arbre



Exemple : encerclement dans un arbre



Exemple : encerclement dans un arbre



Principaux inconvénients du modèle

Dans le modèle de Parson :

- 1 le graphe est *connu*
- 2 à chaque étape, un agent peut être placé n'importe où, la partie propre du graphe n'est pas nécessairement *connexe*.
- 3 les opérations réalisées par les agents ont lieu de façon séquentielle et donc *synchrone*.

Dans la réalité :

- 1 les agents n'ont *a priori* aucune connaissance du réseau
- 2 se "téléporter" est impossible pour les agents, les communications doivent être sûres i.e. la zone propre doit être *connexe*.
- 3 le réseau est *a priori* *asynchrone*.

Graphe inconnu, anonyme

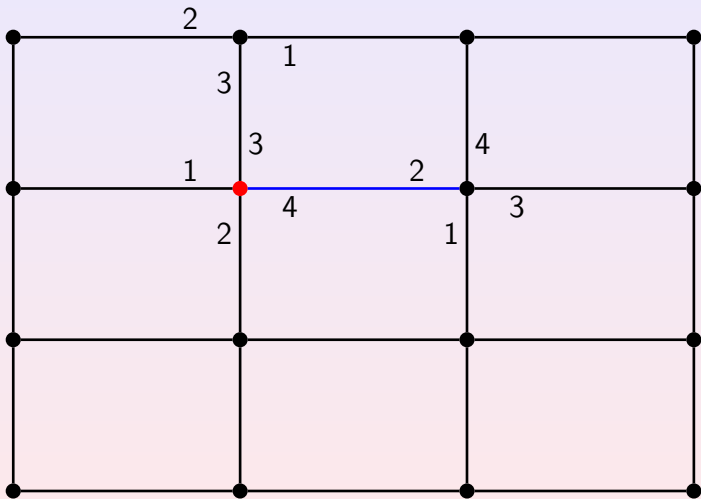
Inconnu

- topologie inconnue
- pas de bornes sur la taille

Anonyme

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

Exemple d'un graphe anonyme



Encerclement monotone connexe (1)

Stratégie monotone connexe

- **connectivité** : à chaque étape, la partie propre du graphe induit un sous-graphe connexe.
- **monotonie** : aucune recontamination n'est autorisée. Une fois qu'une arête est nettoyée, elle reste propre jusqu'à la fin..

mcs(G) le nombre minimum d'agents nécessaires pour capturer un fugitif de manière connexe et monotone.

Encerclement monotone connexe (2)

Définition formelle

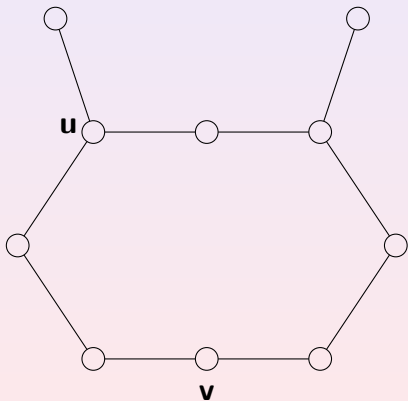
Soit un sommet particulier $v_0 \in V(G)$: la **base** des agents (par exemple, l'entrée de la grotte).

- Initialement, tous les agents sont placés en v_0 .
- Une seule opération permise : déplacer un agent le long d'une arête.

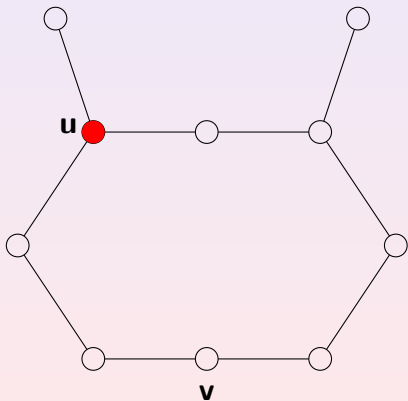
Remarques

- La base reste propre durant toute la stratégie.
- Soit **mcs**(G, v_0) le nombre minimum d'agents nécessaires pour capturer un fugitif de manière connexe et monotone, à partir de v_0 .

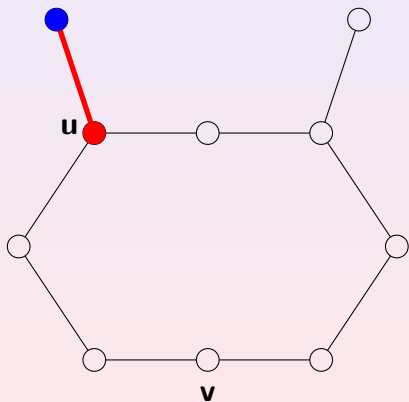
Un exemple simple



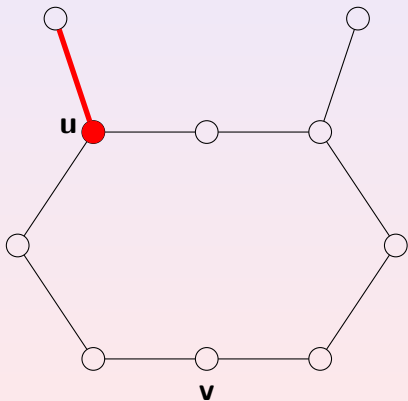
Un exemple simple



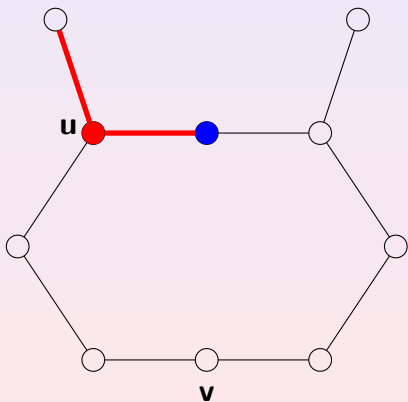
Un exemple simple



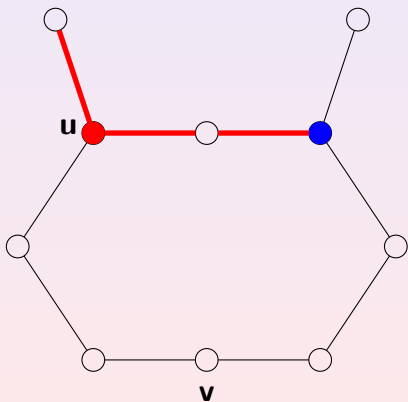
Un exemple simple



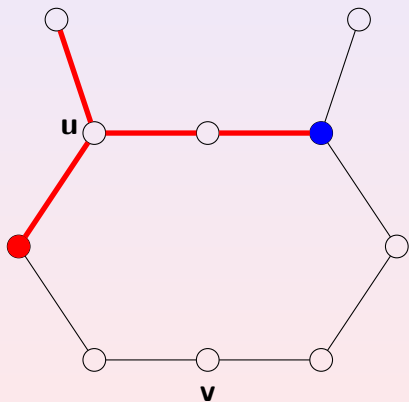
Un exemple simple



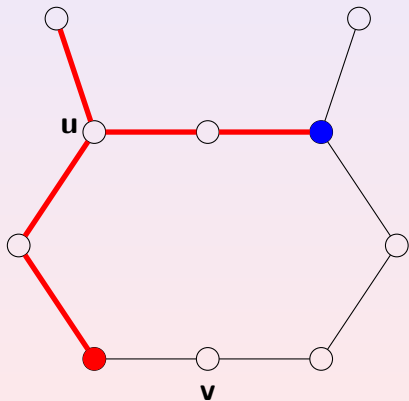
Un exemple simple



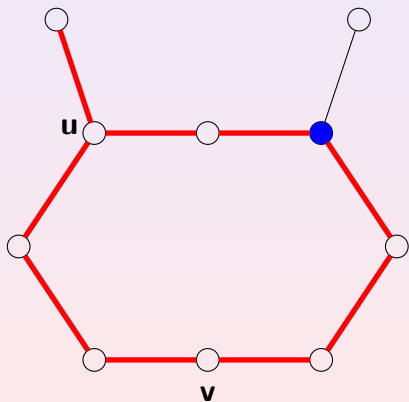
Un exemple simple



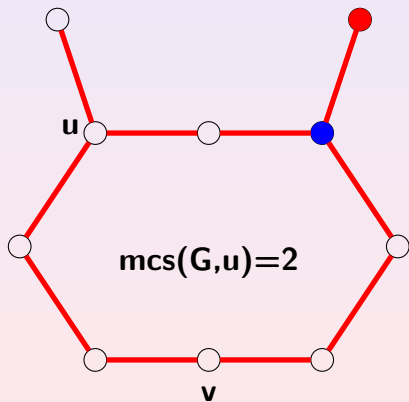
Un exemple simple



Un exemple simple

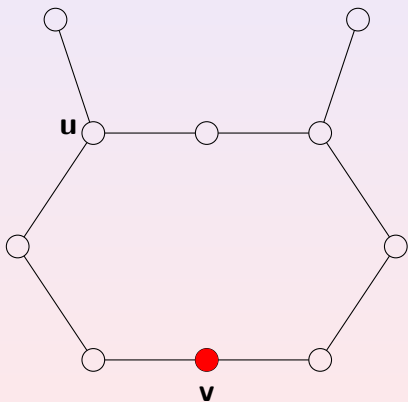


Un exemple simple



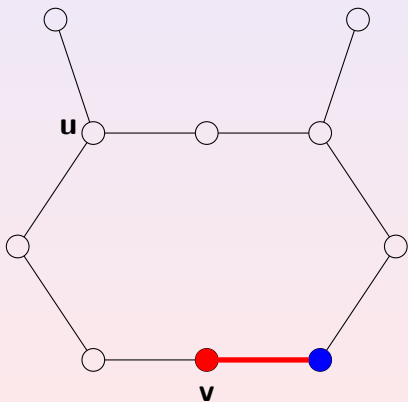
Un exemple simple

importance de la base



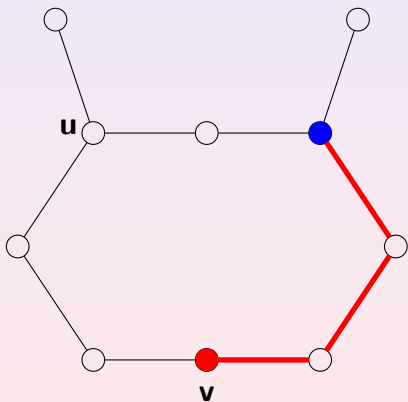
Un exemple simple

importance de la base



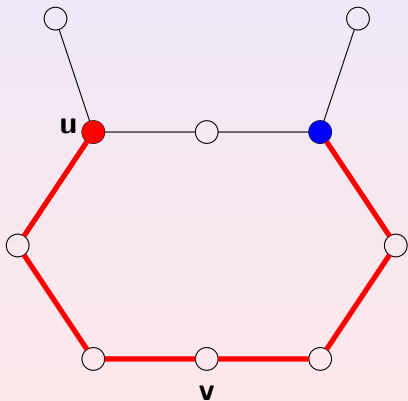
Un exemple simple

importance de la base



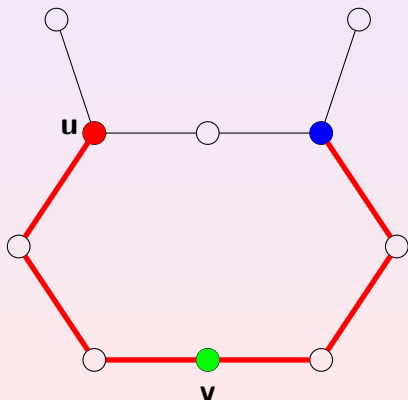
Un exemple simple

importance de la base



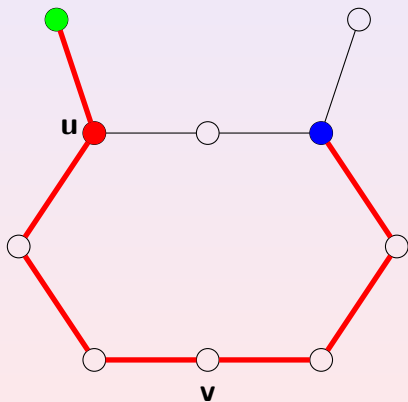
Un exemple simple

importance de la base



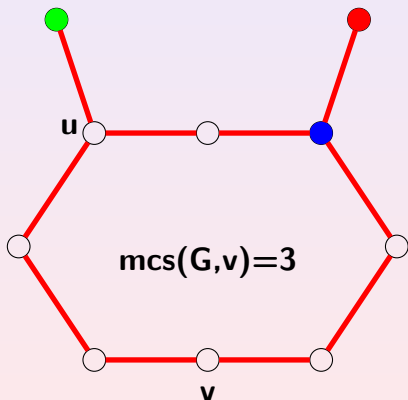
Un exemple simple

importance de la base

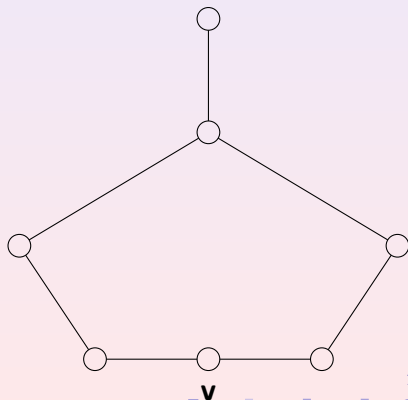
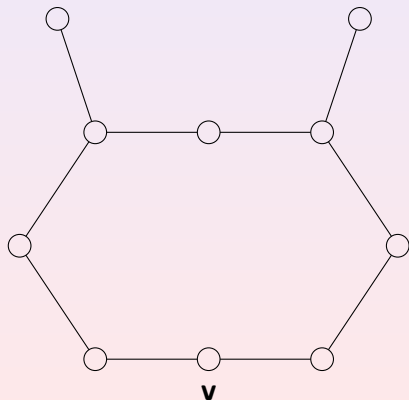


Un exemple simple

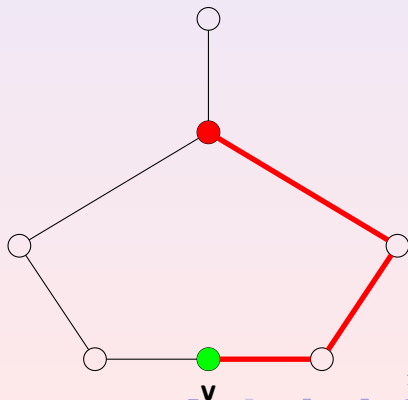
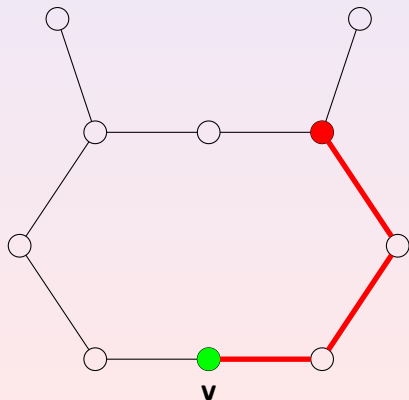
importance de la base



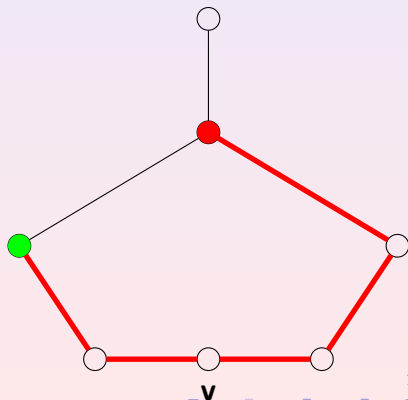
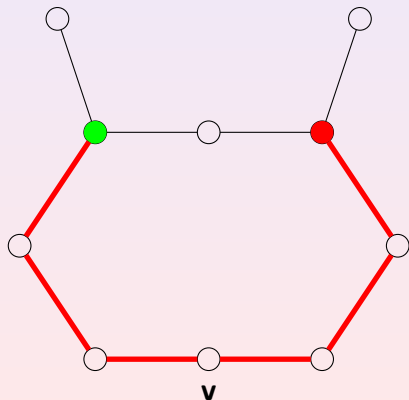
Le coût de l'asynchronisme



Le coût de l'asynchronisme

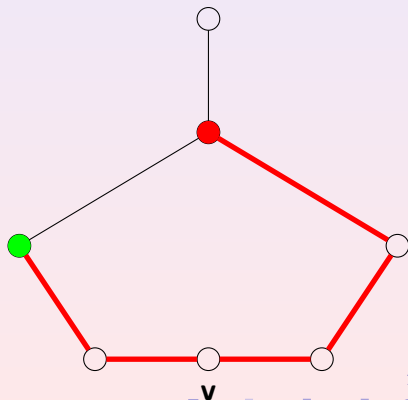
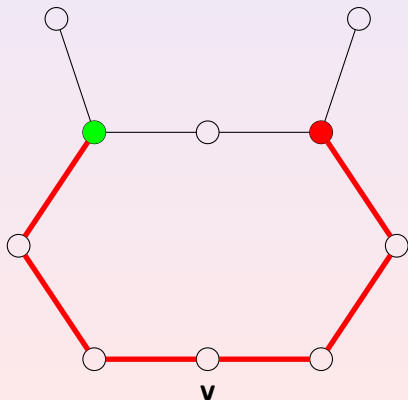


Le coût de l'asynchronisme



Le coût de l'asynchronisme

Les agents ne savent pas dans quel graphe ils se trouvent.
Les agents rouge ont la même vision locale.
Un agent supplémentaire est nécessaire dans les 2 graphes.



Description du modèle

Agents

- ① automates,
- ② peuvent communiquer lorsqu'ils sont sur le même sommet.

Mémoire répartie

- ① tableaux blancs = zone de mémoire disponible sur chaque sommet,
- ② accessible en exclusion mutuelle.

Théorème 1

Pour tout graphe G connexe, anonyme et asynchrone, et pour tout $v_0 \in V(G)$, nous proposons un algorithme distribué qui permet la capture d'un intrus, avec les caractéristiques suivantes :

- 1 au plus $k = \mathbf{mcs}(G, v_0) + 1$ agents sont utilisés,
- 2 chaque agent peut être modélisé par un automate fini à $O(\log k)$ états,
- 3 au cours de l'exécution de l'algorithme, au plus $O(m \log n)$ bits sont stockés sur les tableaux blancs.

Remarques

- notre algorithme est optimal en terme de nombre d'agents utilisés dans le cadre asynchrone,
- notre algorithme retourne une stratégie d'encercllement monotone connexe utilisant au plus $OPT + 1$ agents,
- dans le cas où le nombre d'agents nécessaires est borné par une constante, les agents peuvent être implémentés par des automates finis.

Principes de l'algorithme distribué (1)

Le coordinateur

L'agent supplémentaire sert à la synchronisation des agents. Dans la suite, on suppose que les agents communiquent.

L'algorithme

Initialement, un agent se trouve sur $v_0 \in V(G)$, on essaie de capturer l'intrus avec $k = 1$ agent.

Tant que l'intrus court toujours :

- On essaie toutes les stratégies possibles avec k agents
- On ajoute un agent.

Remarque

A la fin de chaque boucle, tous les agents se retrouvent sur v_0 et le fugitif peut se trouver n'importe où.

Principes de l'algorithme distribué (2)

Algorithme centralisé

Nous proposons un algorithme centralisé, dans un environnement synchrone, qui essaie toutes les stratégies d'encercllement monotone et connexe, utilisant k agents.

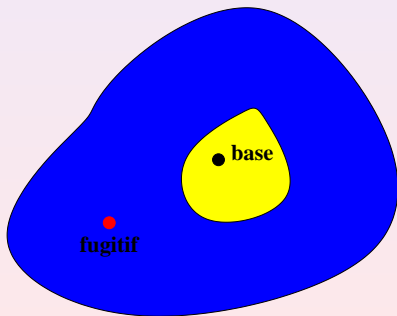
Preuve de l'algorithme distribué

- 1 Preuve de l'algorithme centralisé
- 2 Preuve de l'équivalence entre les 2 algorithmes. Pour cela, nous montrons que les stratégies sont testées dans le même **ordre**

k -Configuration

Une situation a une étape donnée de la stratégie :

- Tous les sommets de la frontière sont gardés par des agents
- A l'intérieur les arêtes sont propres.



Passage d'une configuration à une autre

Un seul agent bouge

- Un seul port incident est "sale" : il le nettoie.
- Aucun port est sale! **Il va aider un autre agent**

Déplacement dans le graphe des configurations

- On part de la configuration initiale : tous les agents sont sur la base
- On fait un DFS “guidé” sur le graphe des configurations
- On doit arriver à la configuration où tous les sommets sont propres.

Conclusion et Perspectives

Algorithme distribué

- La topologie du graphe est inconnue.
- Optimal en terme de nombre d'agents.

Questions ouvertes

- Existe-t-il un tel protocole utilisant des automates finis ?
- Comment diminuer la taille des tableaux blancs ?