Propriété de monotonie des stratégies d'encerclement non-déterministes

Frédéric Mazoit[1] Nicolas Nisse[2]

LABRI, Université Bordeaux I, France.
LRI, Université Paris-Sud, France.

Journées Graphes et Algorithmes, Orléans, novembre 2006

Introduction

Encerclement dans les graphes

Dans un graphe simple, non-orienté,

- un fugitif ;
- une équipe d'agents ;

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

Motivation

- sécurité dans les réseaux de type internet, spéléologie...
- notion en étroite relation avec :
  les décompositions arborescentes et linéaires.

Introduction

Encerclement dans les graphes

Dans un graphe simple, non-orienté,

- un fugitif ;
- une équipe d'agents ;

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

Motivation

- sécurité dans les réseaux de type internet, spéléologie...
- notion en étroite relation avec :
  les décompositions arborescentes et linéaires.

Introduction

Encerclement dans les graphes

Dans un graphe simple, non-orienté,

- un fugitif ;
- une équipe d'agents ;

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

Problème

- intérêt des stratégies monotone
- **Résultat :** la variante non-déterministe des stratégies d'encerclement est monotone

Encerclement dans les graphes [Parson, 1978]

Les agents et le fugitif sont sur les *sommets* du graphe.

Le Fugitif

- est invisible, arbitrairement rapide et omnisicient.
- se déplace d'un sommet à un autre en suivant les arêtes du graphe, s'il ne croise pas d'agent.

Opérations possibles :

- placer un agent sur un sommet du graphe
- supprimer un agent d'un sommet du graphe

Le fugitif est capturé s'il occupe le même sommet qu'un agent.

Encerclement dans les graphes [Parson, 1978]

Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

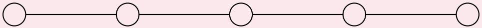
Encerclement dans les graphes [Parson, 1978]

Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin



Encerclement dans les graphes [Parson, 1978]

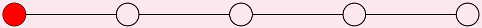
Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

placer rouge sur v_1



Encerclement dans les graphes [Parson, 1978]

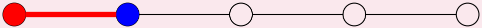
Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

placer rouge sur v_1 ,placer bleu sur v_2



Encerclement dans les graphes [Parson, 1978]

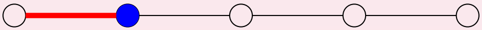
Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

placer rouge sur v_1 ,placer bleu sur v_2, supprimer rouge de v_1



Encerclement dans les graphes [Parson, 1978]

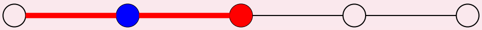
Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

placer rouge sur v_1 ,placer bleu sur v_2, supprimer rouge de v_1, placer rouge sur v_3...



Encerclement dans les graphes [Parson, 1978]

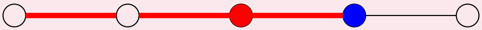
Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

placer rouge sur v_1 ,placer bleu sur v_2, supprimer rouge de v_1, placer rouge sur v_3...



Encerclement dans les graphes [Parson, 1978]

Stratégie d'encerclement

- séquence d'opérations élémentaires
- qui doit capturer le fugitif **quoiqu'il fasse**.

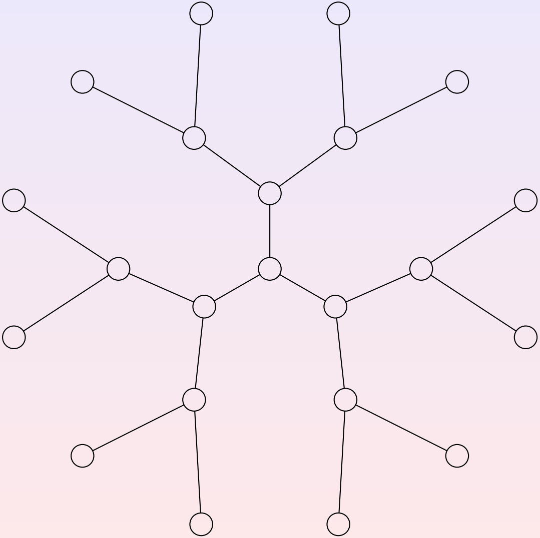
But : trouver le nombre minimum d'agents nécessaires $\mathbf{s}(G)$.

Exemple : un chemin

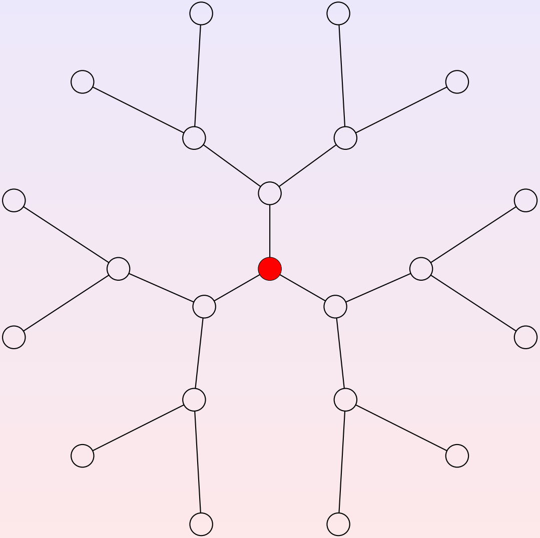
placer rouge sur v_1 ,placer bleu sur v_2, supprimer rouge de v_1, placer rouge sur v_3...

s(Chemin)=2

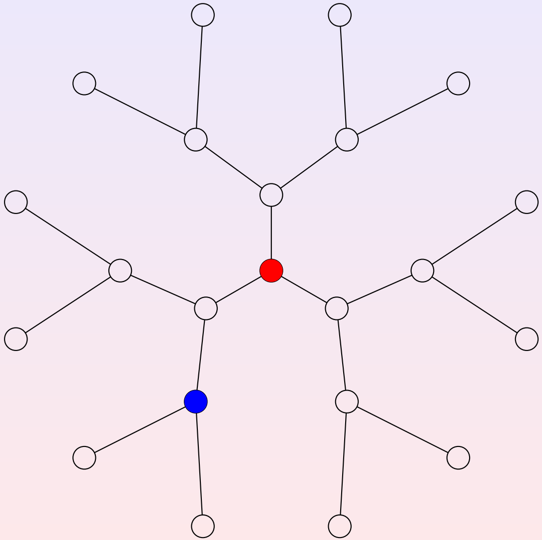
Encerclement dans un arbre



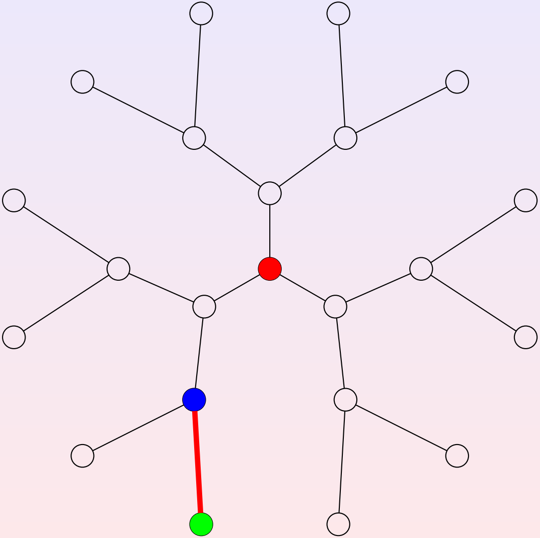
Encerclement dans un arbre



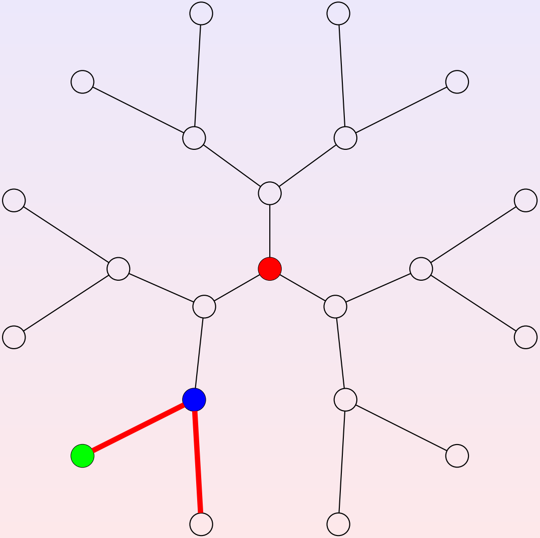
Encerclement dans un arbre



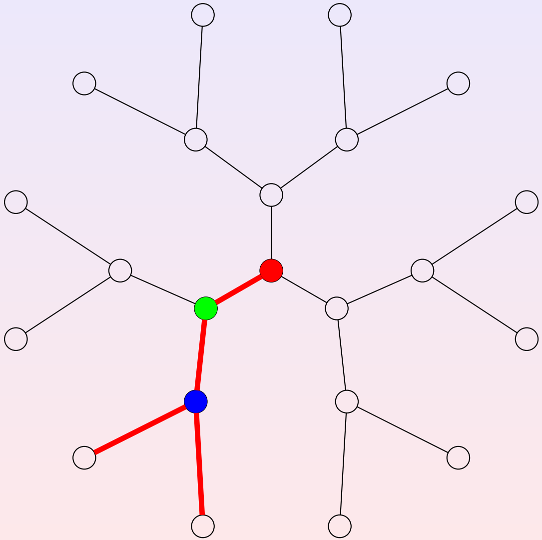
Encerclement dans un arbre



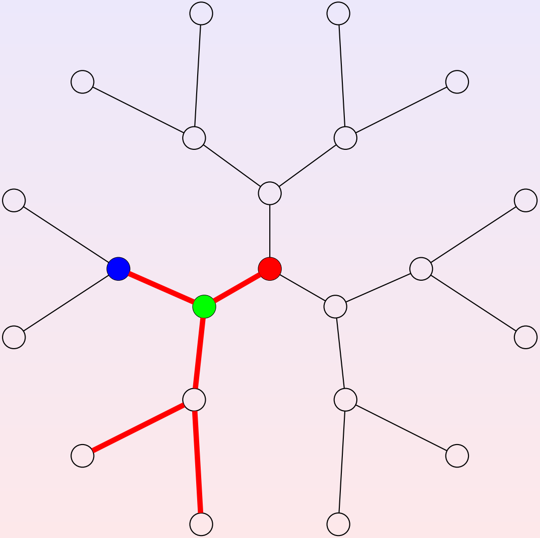
Encerclement dans un arbre



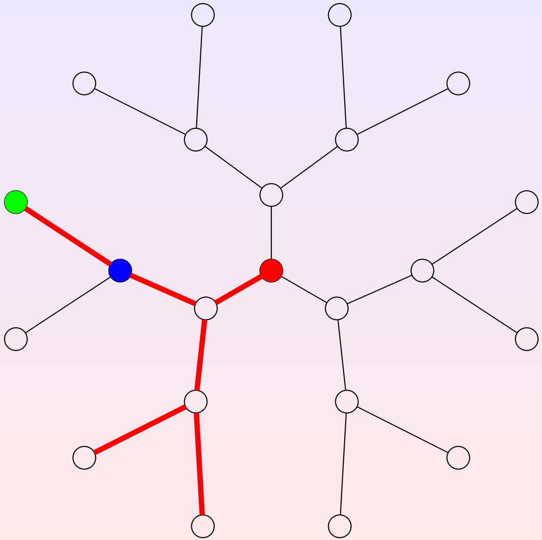
Encerclement dans un arbre



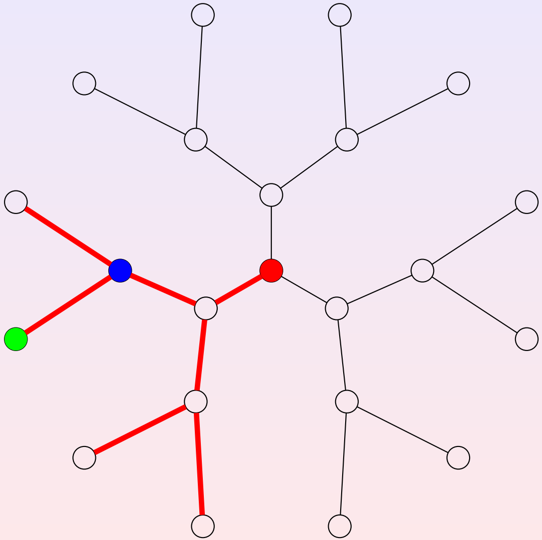
Encerclement dans un arbre



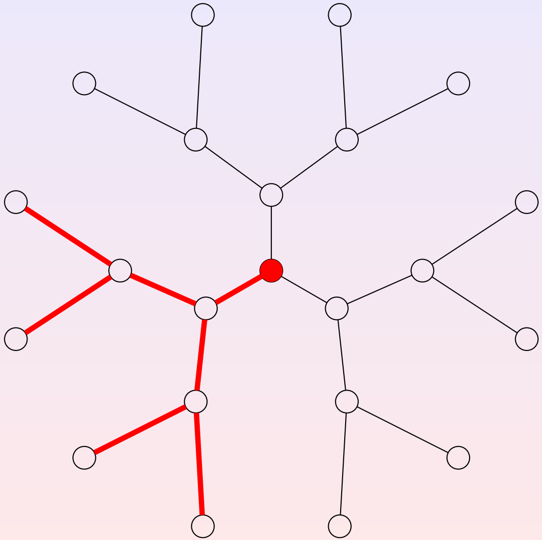
Encerclement dans un arbre



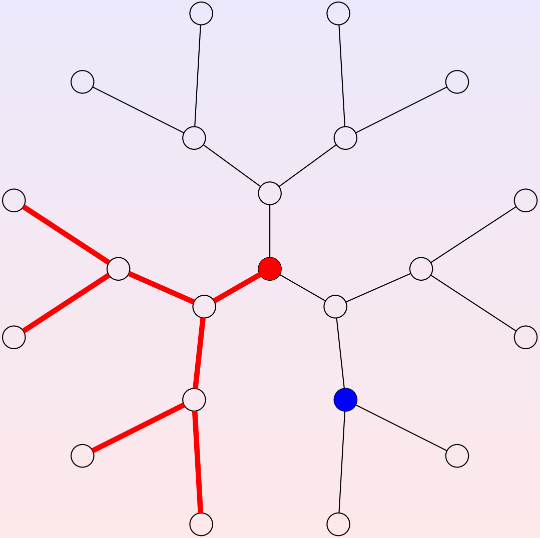
Encerclement dans un arbre



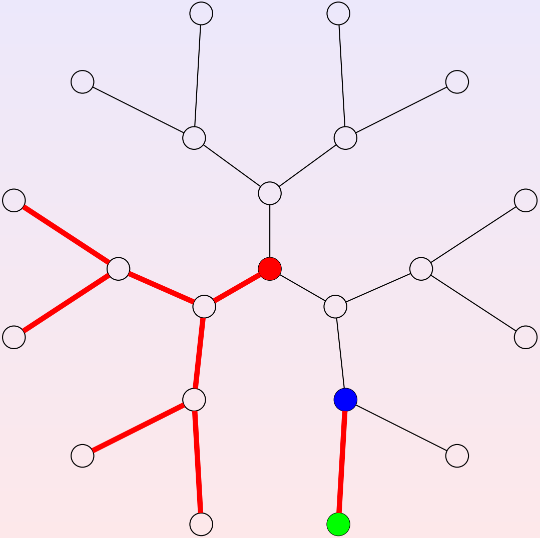
Encerclement dans un arbre



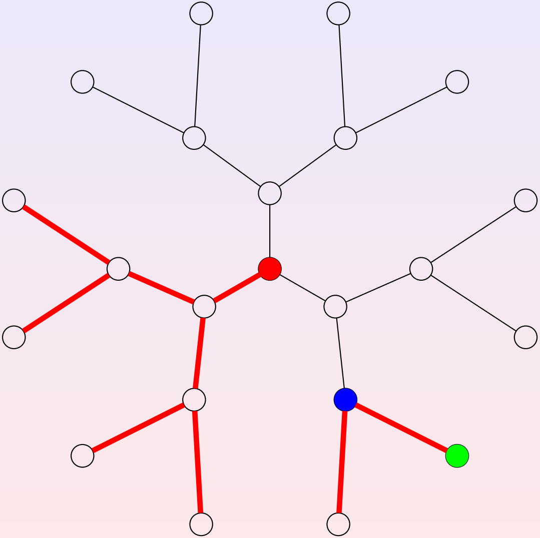
Encerclement dans un arbre



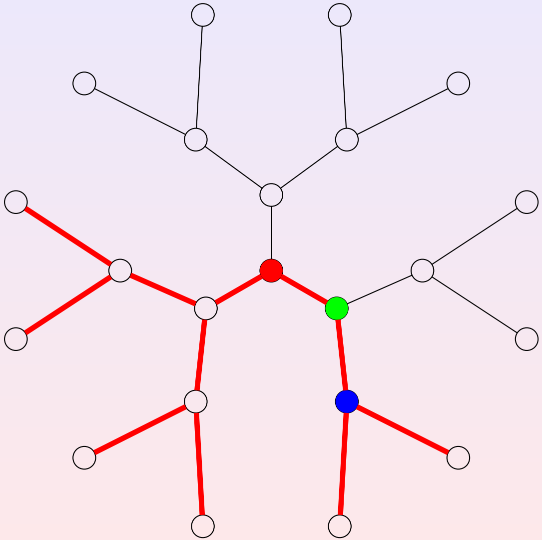
Encerclement dans un arbre



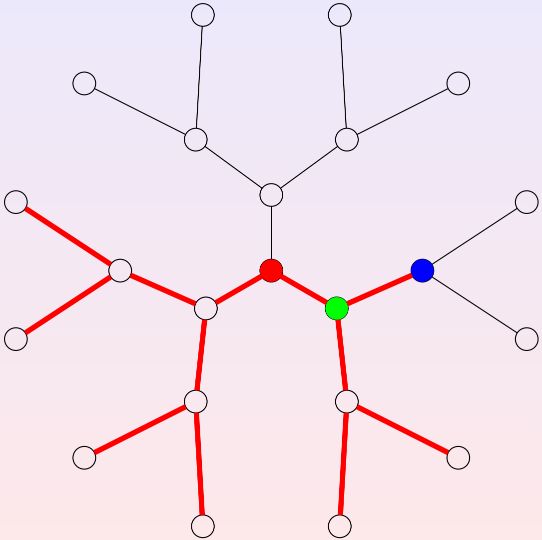
Encerclement dans un arbre



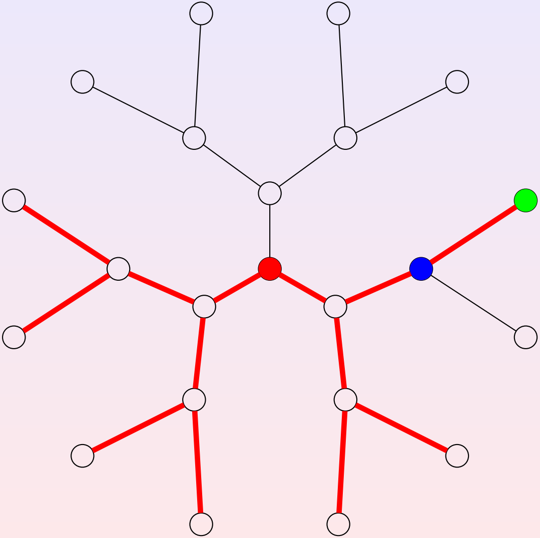
Encerclement dans un arbre



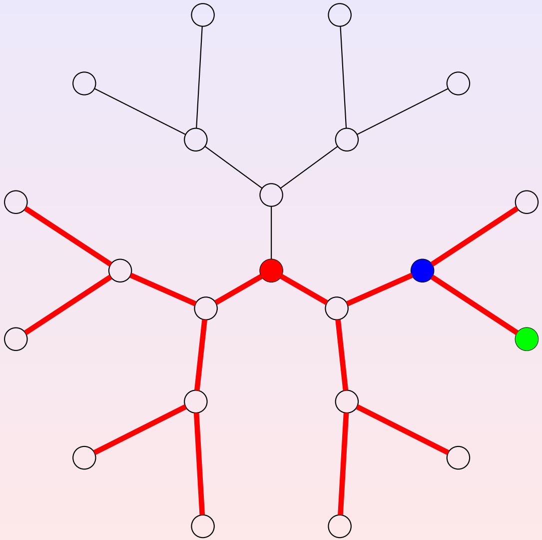
Encerclement dans un arbre



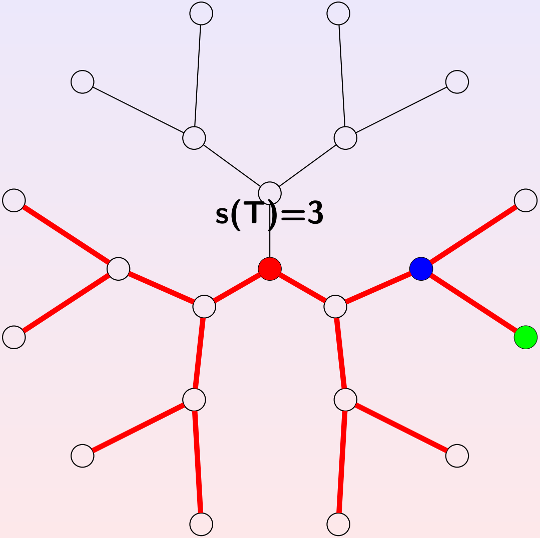
Encerclement dans un arbre



Encerclement dans un arbre



Encerclement dans un arbre



Encerclement visible

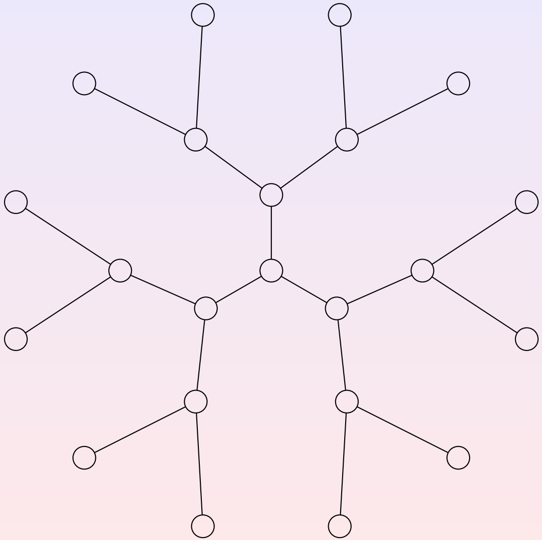
Visibilité du fugitif

Le fugitif est visible si, à chaque étape, les agents connaissent sa position (en fait la composante connexe où il se trouve). La stratégie peut donc être orientée d'après la position du fugitif.

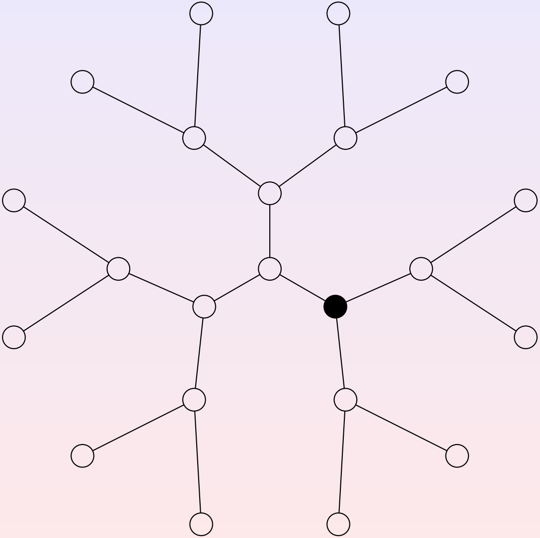
Paramètre associé

Soit **vs**(G) l'encerclement d'un fugitif visible dans le graphe G. De manière évidente, $vs(G) \leq s(G)$.

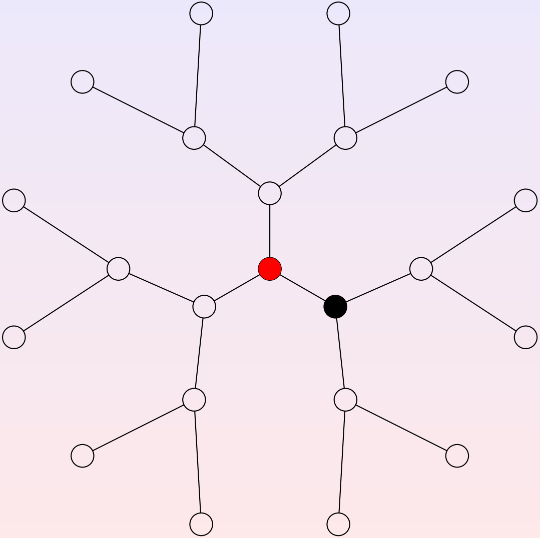
Encerclement d'un fugitif visible dans un arbre



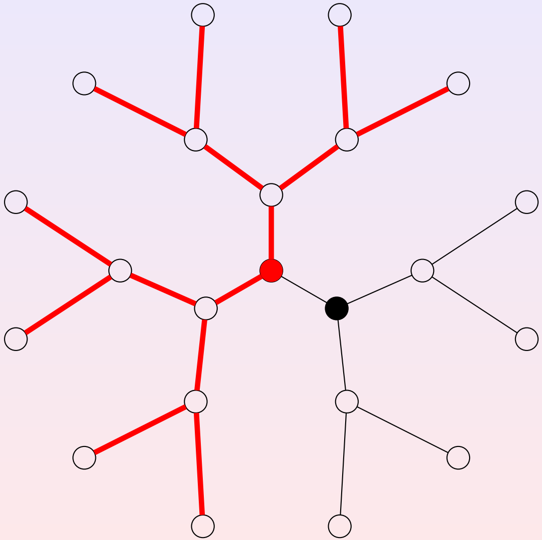
Encerclement d'un fugitif visible dans un arbre



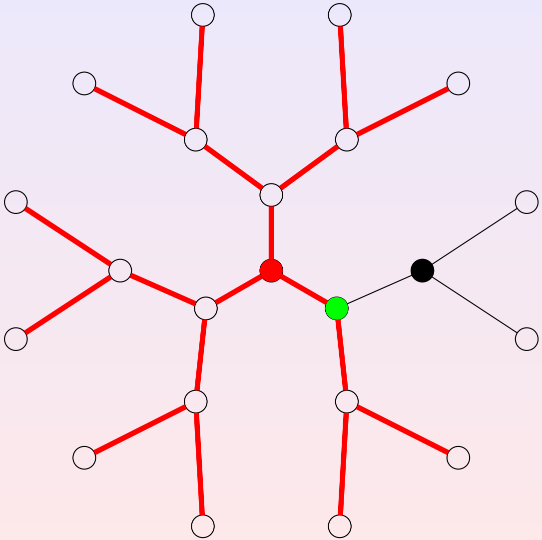
Encerclement d'un fugitif visible dans un arbre



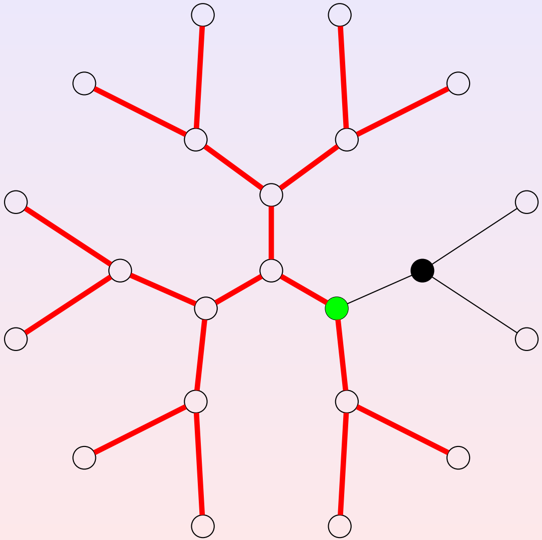
Encerclement d'un fugitif visible dans un arbre



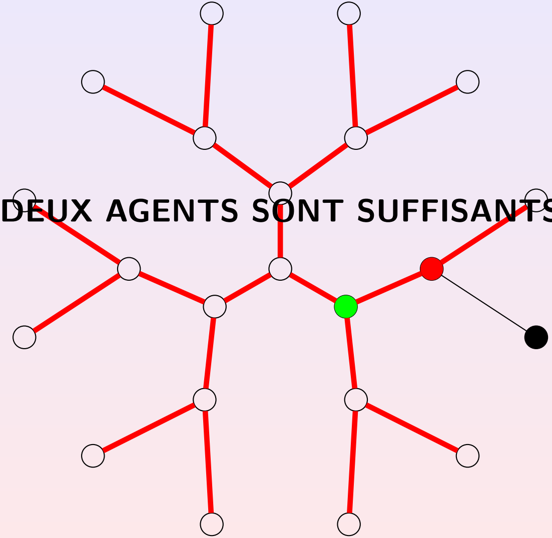
Encerclement d'un fugitif visible dans un arbre



Encerclement d'un fugitif visible dans un arbre



Encerclement d'un fugitif visible dans un arbre



DEUX AGENTS SONT SUFFISANTS

Compléxité et Monotonie

Décider si $\mathbf{s}(G) \leq k$ est NP-difficile

- **Megiddo** *et al*, J.of ACM, 1988
  The complexity of searching a graph.

Décider si $\mathbf{vs}(G) \leq k$ est NP-difficile

- **Seymour et Thomas**, J. of Comb. Th., 1993.
  Graph searching and a min-max theorem for tree-width

Question : le problème est-il dans NP ?
Une stratégie est un certificat. Taille d'une stratégie ?

Compléxité et Monotonie

Décider si $\mathbf{s}(G) \leq k$ est NP-difficile

- **Megiddo** *et al*, J.of ACM, 1988
  The complexity of searching a graph.

Décider si $\mathbf{vs}(G) \leq k$ est NP-difficile

- **Seymour et Thomas**, J. of Comb. Th., 1993.
  Graph searching and a min-max theorem for tree-width

Question : le problème est-il dans NP ?
Une stratégie est un certificat. Taille d'une stratégie ?

Monotonie

Une stratégie est monotone si la zone accessible au fugitif décroit.
Chaque sommet n'est occupé qu'une fois par un agent.
Il n'y a pas de recontamination.

Une stratégie monotone est un certificat de taille polynomiale.

Question : La recontamination aide-t'elle ?
Existe-t'il toujours une stratégie monotone utilisant le nombre optimal d'agent ?

Monotonie

Une stratégie est monotone si la zone accessible au fugitif décroit.
Chaque sommet n'est occupé qu'une fois par un agent.
Il n'y a pas de recontamination.

Une stratégie monotone est un certificat de taille polynomiale.

Question : La recontamination aide-t'elle ?
Existe-t'il toujours une stratégie monotone utilisant le nombre optimal d'agent ?

Monotonie

Une stratégie est monotone si la zone accessible au fugitif décroit.
Chaque sommet n'est occupé qu'une fois par un agent.
Il n'y a pas de recontamination.

Une stratégie monotone est un certificat de taille polynomiale.

Question : La recontamination aide-t'elle ?
Existe-t'il toujours une stratégie monotone utilisant le nombre optimal d'agent ?

La recontamination n'aide pas

Cas d'un fugitif invisible

- **Bienstock et Seymour**, J.of Alg., 1991
  Monotonicity in graph searching.
- **LaPaugh**, J.of ACM, 1993
  Recontamination does not help to search a graph.

Preuves constructives : *LaPaugh*

- stratégie avec $\leq k$ agents
- $\Rightarrow$ stratégie *monotone* avec $\leq k$

La recontamination n'aide pas

Cas d'un fugitif invisible

- **Bienstock et Seymour**, J.of Alg., 1991
  Monotonicity in graph searching.
- **LaPaugh**, J.of ACM, 1993
  Recontamination does not help to search a graph.

Preuves constructives : *Bienstock et Seymour*

- stratégie avec $\leq k$ agents
- $\Rightarrow$ croisade de largeur $\leq k$
- $\Rightarrow$ croisade *monotone* de largeur $\leq k$
- $\Rightarrow$ stratégie *monotone* avec $\leq k$

La recontamination n'aide pas

Cas d'un fugitif visible

- **Seymour et Thomas**, J. of Comb. Th., 1993.
  Graph searching and a min-max theorem for tree-width

Preuve non constructive :

- pas de stratégie monotone avec $\leq k$ agents
- $\Rightarrow$ stratégie pour le fugitif contre $\leq k$ agents "monotones"
- $\Rightarrow$ stratégie pour le fugitif contre $\leq k$ agents
- $\Rightarrow$ pas de stratégie avec $\leq k$

La recontamination n'aide pas

Conséquences

- NP-complet
- lien entre encerclement invisible et largeur linéaire **pw** : $\mathbf{s}(G) = \mathbf{pw}(G) + 1$
- lien entre encerclement visible et largeur arborescente **tw** : $\mathbf{vs}(G) = \mathbf{tw}(G) + 1$

Encerclement non-déterministe

Fugitif invisible
Un Oracle voit le fugitif

Opération supplémentaire

Les agents peuvent demander la position du fugitif à l'oracle.

Séquence d'opérations élémentaires :

- placer un agent sur un sommet du graphe ;
- supprimer un agent d'un sommet du graphe ;
- poser une question à l'oracle.

encerclement q-limité

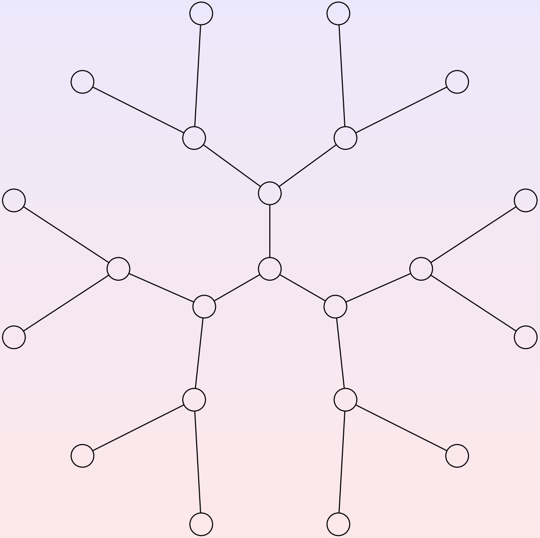
Compromis nombre d'agents / nombre de questions posées

encerclement (non-déterministe) q-limité, $s_q(G)$

- $\mathbf{s}_0(G) = \mathbf{pw}(G) + 1$, encerclement de G ;
- $\mathbf{s}_\infty(G) = \mathbf{tw}(G) + 1$, encerclement visible de G.

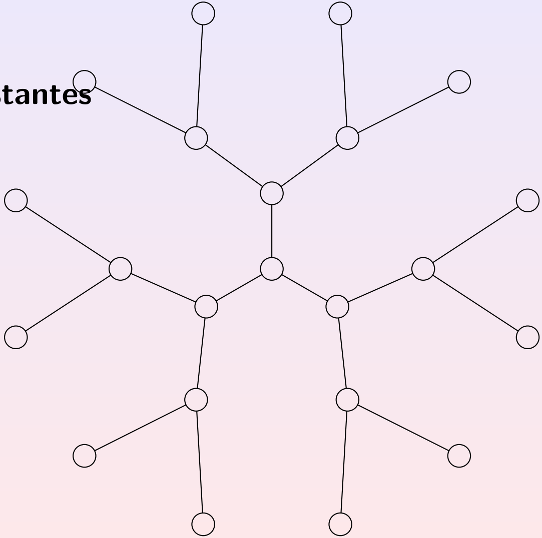
Exemple avec q=2 :

$s_0(T)=3$



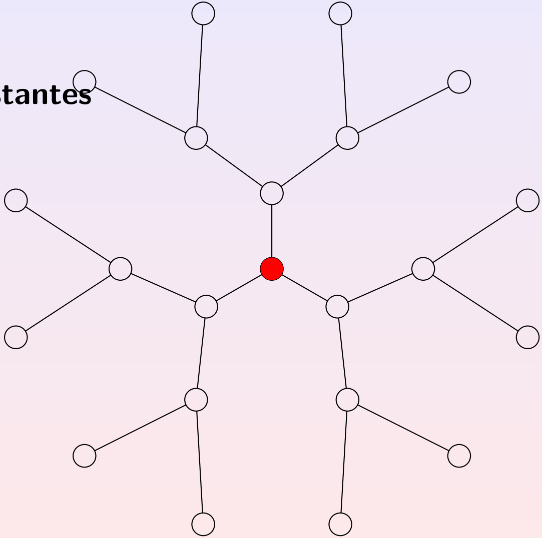
Exemple avec q=2 :

2 questions restantes



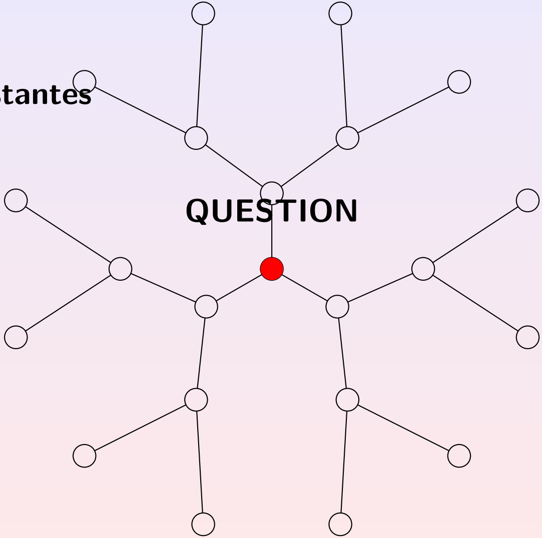
Exemple avec q=2 :

2 questions restantes



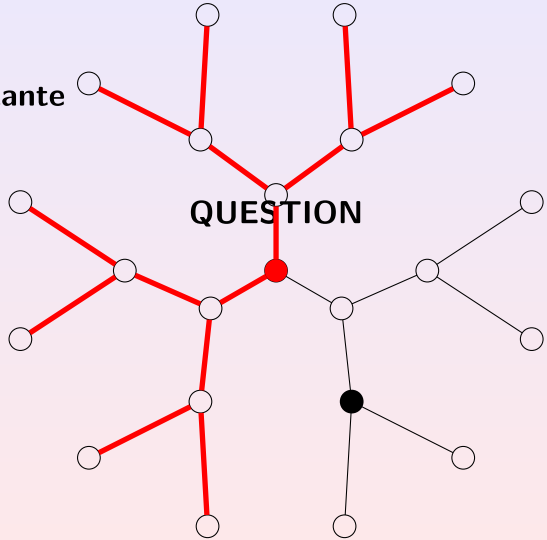
Exemple avec q=2 :

2 questions restantes



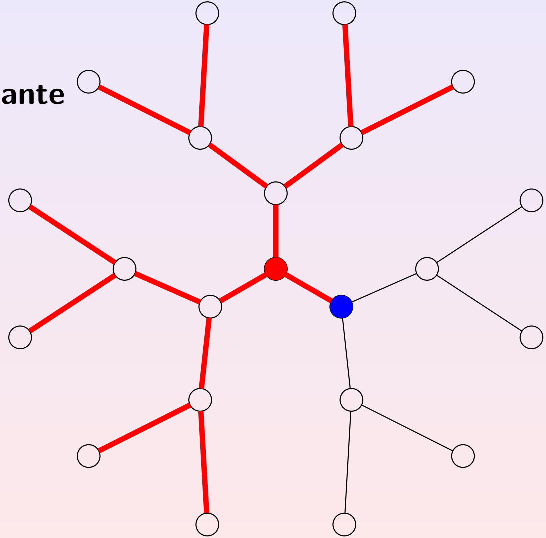
Exemple avec q=2 :

1 question restante



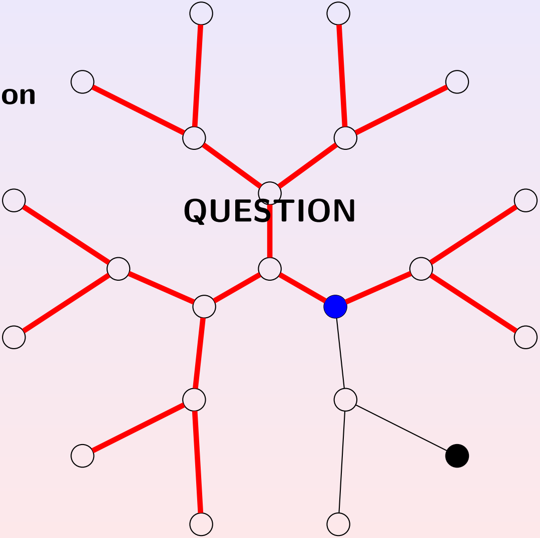
Exemple avec q=2 :

1 question restante



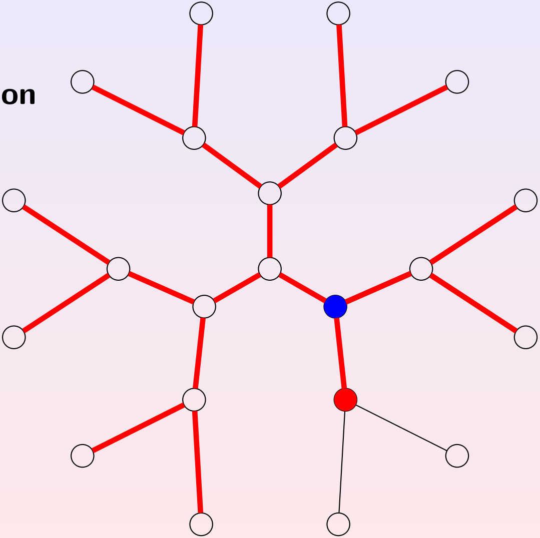
Exemple avec q=2 :

plus de question



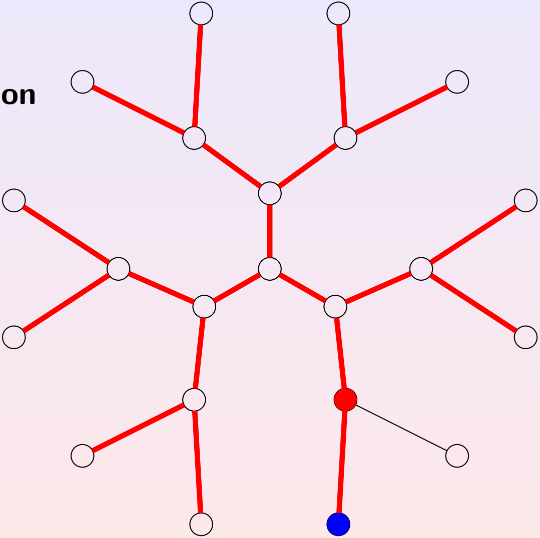
Exemple avec q=2 :

plus de question



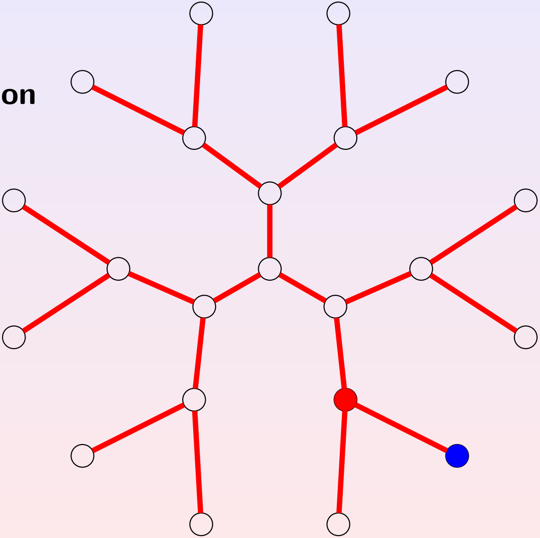
Exemple avec q=2 :

plus de question



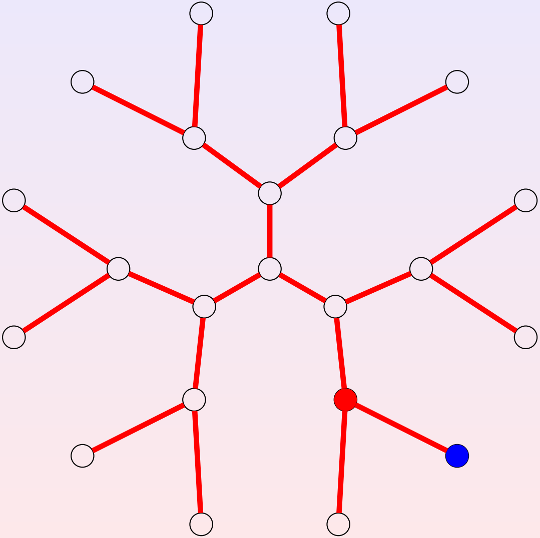
Exemple avec q=2 :

plus de question



Exemple avec q=2 :

$s_2(T)=2$



Encerclement non-déterministe **monotone**

Theorem[Fomin, Fraigniaud, Nisse, 2005] :

1. Interprétation de l'encerclement non-déterministe monotone en terme de décomposition de graphe ;
2. Déterminer si $\mathbf{s}_q(G) \leq k$ est NP-difficile
3. Algorithme exponentiel exact qui calcule $\mathbf{s}_q(G)$

Question : Est-ce que la recontamination aide ?
(la réponse est "Non" pour $q = 0$ et $q = \infty$.)

Notre résultat

Pour tout $q \geq 0$ et tout graphe G, la recontamination n'aide pas à capturer un fugitif dans G en posant au plus q questions.

Remarques :

- Preuve constructive qui unifie les preuves existantes ;
- Décider si $\mathbf{s}_q(G) \leq k$ est dans NP ;
- L'algorithme de Fomin *et al.* calcule $\mathbf{s}_q(G)$.

Idée de la preuve

structure intermédiaire inspirée du tree-labelling [Robertson and Seymour, Graph Minor X] : arbre d'encerclement

Soit G un graphe connexe, $q \geq 0$ et $k \geq 1$.

1. $\exists$ une strategie non-déterministe utilisant $\leq k$ agents et $\leq q$ questions ;
2. $\exists$ un arbre d'encerclement q-branché de larqueur $\leq k$;
3. $\exists$ un arbre d'encerclement monotone q-branché de larqueur $\leq k$;
4. $\exists$ une strategie non-déterministe monotone utilisant $\leq k$ agents et $\leq q$ questions ;

Conclusion et Perspectives

A propos de monotonie

Généraliser notre résultat à d'autres variantes d'encerclement : dans les graphes orientés

A propos de l'encerclement non-déterministe

Algorithme polynomial dans le cas des arbres ?
Algorithmes paramétrés ?