

Complexité paramétrée (2) :

Recherche de noyaux - bornes supérieures

Christophe PAUL
(CNRS - LIRMM)

October 5, 2011

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

- Règles de réduction en SUNFLOWER

- Analyse de la structure de la solution

- Analyse par comptage

Programmation linéaire

Règles de réduction globales

- Réduction en couronne pour VERTEX COVER

- Un noyau linéaire pour le CLUSTER EDITING

- Un noyau linéaire pour FAST à l'aide de couplages

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

Réduction en couronne pour VERTEX COVER

Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

Règles de réduction pour VERTEX COVER

1. Si x est un sommet isolé, alors x n'appartient à aucune solution optimale.

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - x, k)$$

Règles de réduction pour VERTEX COVER

1. Si x est un sommet isolé, alors x n'appartient à aucune solution optimale.

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - x, k)$$

2. Si x est un sommet de degré 1 voisin de y , alors il existe une solution optimale contenant y .

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - \{y, x\}, k - 1)$$

Règles de réduction pour VERTEX COVER

1. Si x est un sommet isolé, alors x n'appartient à aucune solution optimale.

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - x, k)$$

2. Si x est un sommet de degré 1 voisin de y , alors il existe une solution optimale contenant y .

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - \{y, x\}, k - 1)$$

3. **Si x est de degré $\geq k + 1$, alors si G possède une solution de taille k , elle contient le sommet x .**

$$\text{VERTEX COVER}(G, k) = \text{VERTEX COVER}(G - x, k - 1)$$

Lemme [Buss]

Si G possède un VERTEX COVER de taille k , alors le graphe réduit possède au plus $k^2 + k$ sommets au plus k^2 arêtes.

Lemme [Buss]

Si G possède un VERTEX COVER de taille k , alors le graphe réduit possède au plus $k^2 + k$ sommets au plus k^2 arêtes.

Preuve

1. *Le graphe réduit possède au plus k^2 arêtes.*

Lemme [Buss]

Si G possède un VERTEX COVER de taille k , alors le graphe réduit possède au plus $k^2 + k$ sommets au plus k^2 arêtes.

Preuve

1. *Le graphe réduit possède au plus k^2 arêtes.*

Si S est un VERTEX COVER, alors toute arête est incidente à un sommet de S .

Or $d(x) \leq k$ et $\Rightarrow k^2$ arêtes au plus.

Lemme [Buss]

Si G possède un VERTEX COVER de taille k , alors le graphe réduit possède au plus $k^2 + k$ sommets au plus k^2 arêtes.

Preuve

1. *Le graphe réduit possède au plus k^2 arêtes.*
2. *Le graphe réduit possède au plus $k^2 + k$ sommets.*

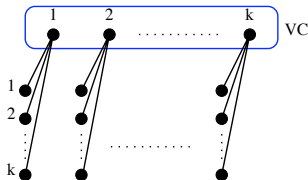
Lemme [Buss]

Si G possède un VERTEX COVER de taille k , alors le graphe réduit possède au plus $k^2 + k$ sommets au plus k^2 arêtes.

Preuve

1. Le graphe réduit possède au plus k^2 arêtes.
2. Le graphe réduit possède au plus $k^2 + k$ sommets.

S possède au plus k sommets de degré $\leq k$
 \Rightarrow il y a au plus $k^2 + k$ sommets



Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

Réduction en couronne pour VERTEX COVER

Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

Définition

Soit (Q, κ) un problème paramétré sur l'alphabet Σ . Une fonction calculable en temps polynomiale

$$K : \Sigma^* \rightarrow \Sigma^*$$

est une **kernalisation** de (Q, κ) s'il existe une fonction

$$h : \mathbb{N} \rightarrow \mathbb{N}$$

telle que pour tout $x \in \Sigma^*$, on a

$$(x \in Q \Leftrightarrow K(x) \in Q)$$

$$|K(x)| \leq h(\kappa(x)) \quad \text{et} \quad \kappa(K(x)) \leq \kappa(x)$$

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

- ▶ $(2) \Rightarrow (1)$: Soit K la kernalisation de (Q, κ) . Considérons l'algorithme \mathcal{A} suivant:
 1. calculer $K(x)$
 2. décider si $K(x) \in Q$ avec un algorithme \mathcal{A}'

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

- (2) \Rightarrow (1) : Soit K la kernalisation de (Q, κ) . Considérons l'algorithme \mathcal{A} suivant:

1. calculer $K(x)$
2. décider si $K(x) \in Q$ avec un algorithme \mathcal{A}'

Le calcul de $K(x)$ se fait en $|x|^{O(1)}$

L'algorithme \mathcal{A}' coûte $|K(x)|^{O(1)}$

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

► $(2) \Rightarrow (1)$: Soit K la kernalisation de (Q, κ) . Considérons l'algorithme \mathcal{A} suivant:

1. calculer $K(x)$
2. décider si $K(x) \in Q$ avec un algorithme \mathcal{A}'

Le calcul de $K(x)$ se fait en $|x|^{O(1)}$

L'algorithme \mathcal{A}' coûte $|K(x)|^{O(1)}$

\Rightarrow Puisque $|K(x)| \leq h(\kappa(x))$, l'algorithme \mathcal{A} est **FPT**.

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

- ▶ $(1) \Rightarrow (2)$: Soit \mathcal{A} un algorithme **FPT** pour (Q, κ) de complexité $f(\kappa(|x|)).p(|x|)$

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

- (1) \Rightarrow (2) : Soit \mathcal{A} un algorithme **FPT** pour (Q, κ) de complexité $f(\kappa(|x|)).p(|x|)$

Si $Q = \emptyset$ ou $Q = \Sigma^*$, alors (Q, κ) a une kernalisation triviale.

Soit $x_1 \in Q$ et $x_0 \in \Sigma^* \setminus Q$

Soit l'algorithme \mathcal{A}' suivant :

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

- (1) \Rightarrow (2) : Soit \mathcal{A} un algorithme **FPT** pour (Q, κ) de complexité $f(\kappa(|x|)).p(|x|)$

Si $Q = \emptyset$ ou $Q = \Sigma^*$, alors (Q, κ) a une kernalisation triviale.

Soit $x_1 \in Q$ et $x_0 \in \Sigma^* \setminus Q$

Soit l'algorithme \mathcal{A}' suivant :

1. Exécuter les $p(|x|).|x|$ premières étapes de \mathcal{A}
2. \mathcal{A} stoppe et accepte (réfute) $x \Rightarrow K(x) = x_1$ ($K(x) = x_0$)
3. Sinon (\mathcal{A} ne s'arrête pas en $p(|x|).|x|$ étapes) $K(x) = x$

Théorème

Pour tout problème paramétré (Q, κ) , les conditions suivantes sont équivalentes

1. $(Q, \kappa) \in \mathbf{FPT}$
2. Q est décidable et (Q, κ) possède un kernel.

Preuve

► $(1) \Rightarrow (2)$:

1. Exécuter les $p(|x|) \cdot |x|$ premières étapes de \mathcal{A}
2. \mathcal{A} stoppe et accepte (réfute) $x \Rightarrow K(x) = x_1$ ($K(x) = x_0$)
3. Sinon (\mathcal{A} ne s'arrête pas en $p(|x|) \cdot |x|$ étapes) $K(x) = x$

K est bien une kernalisation:

- $K(x)$ est calculé en temps polynomial
- $|K(x)| \leq |x_0| + |x_1| + f(\kappa(x))$ car

$$p(|x|) \cdot |x| \leq f(\kappa(x)) \cdot p(|x|)$$

Théorème

VERTEX COVER ne possède pas de noyau formé par un graphe de taille $1,36k$ sommets à moins que $\mathbf{P} = \mathbf{NP}$.

Théorème

VERTEX COVER ne possède pas de noyau formé par un graphe de taille $1,36k$ sommets à moins que $\mathbf{P} = \mathbf{NP}$.

Preuve

Supposons qu'un tel noyau existe. Alors cet ensemble de $1,36k$ sommets serait une approximation polynomiale de la solution optimale.

⇒ Impossible par des résultats de la théorie de l'approximabilité (Théorème PCP), à moins que $\mathbf{P} = \mathbf{NP}$.

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

- Règles de réduction en SUNFLOWER

- Analyse de la structure de la solution

- Analyse par comptage

Programmation linéaire

Règles de réduction globales

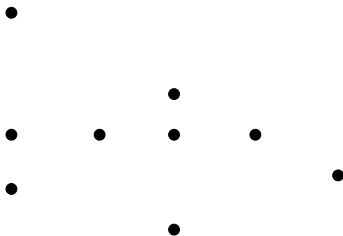
- Réduction en couronne pour VERTEX COVER

- Un noyau linéaire pour le CLUSTER EDITING

- Un noyau linéaire pour FAST à l'aide de couplages

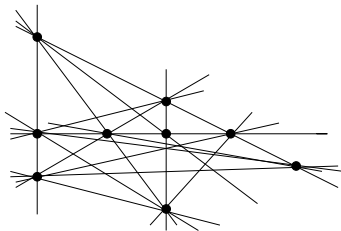
Courverture d'un ensemble de points par des lignes

- ▶ **Entrée** : Un ensemble S de n points dans le plan
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il k lignes couvrant l'ensemble S ?



Couverture d'un ensemble de points par des lignes

- ▶ **Entrée** : Un ensemble S de n points dans le plan
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il k lignes couvrant l'ensemble S ?

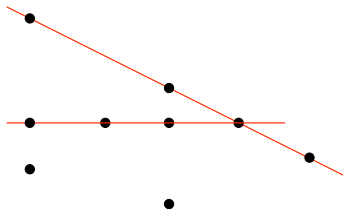


Observation

L'ensemble des lignes nécessaires pour couvrir S est un sous-ensemble des lignes générées par l'ensemble des paires de points de S

Courverture d'un ensemble de points par des lignes

- ▶ **Entrée** : Un ensemble S de n points dans le plan
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il k lignes couvrant l'ensemble S ?



Règle de réduction

Si une ligne contient au moins $k + 1$ points, alors elle appartient la solution (si elle existe)

e.g. Ici $k = 3$

Couverture d'un ensemble de points par des lignes

- ▶ **Entrée** : Un ensemble S de n points dans le plan
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il k lignes couvrant l'ensemble S ?

Théorème

Le problème de la couverture de points par des lignes admet un noyau quadratique.

Preuve :

Si l'instance est positive, alors l'instance réduite peut être couverte par au plus k lignes contenant chacune au plus k points
 \Rightarrow l'instance réduite contient au plus k^2 points.

Exercice

Montrer que 3-HITTING SET possède un noyau de taille k^3 à l'aide de règles du type sunflower. Notez que 3-HITTING SET est une généralisation de vertex cover, qui est en fait 2-HITTING SET. Il faut donc généraliser le lemme de Buss.

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

- Règles de réduction en SUNFLOWER

- Analyse de la structure de la solution

- Analyse par comptage

Programmation linéaire

Règles de réduction globales

- Réduction en couronne pour VERTEX COVER

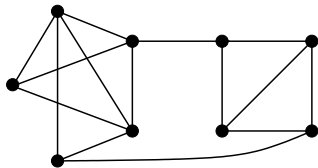
- Un noyau linéaire pour le CLUSTER EDITING

- Un noyau linéaire pour FAST à l'aide de couplages

k -CLUSTER EDITING Problem

Etant donné un graphe $G = (V, E)$, existe-t-il un ensemble $F \subseteq V \times V$ tel que:

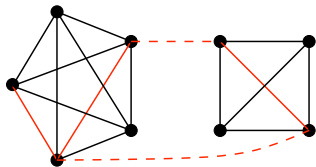
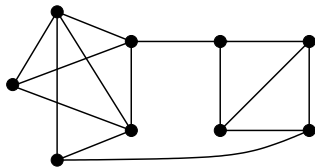
- ▶ $|F| \leq k$ et
- ▶ $H = (V, E \triangle F)$ est une union disjointe de cliques (graphes complets)



k -CLUSTER EDITING Problem

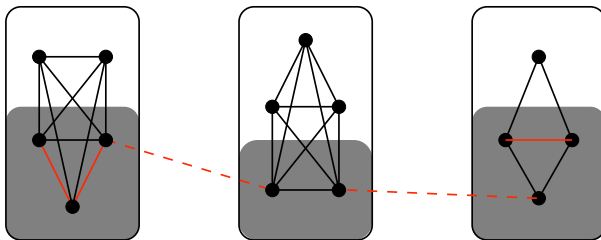
Etant donné un graphe $G = (V, E)$, existe-t-il un ensemble $F \subseteq V \times V$ tel que:

- ▶ $|F| \leq k$ et
- ▶ $H = (V, E \triangle F)$ est une union disjointe de cliques (graphes complets)



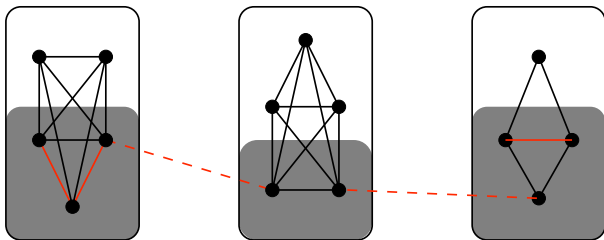
Un sommet est **affecté** s'il est incident à une arête de F .

1. Il y a au plus $2k$ sommets affectés par une k -édition F



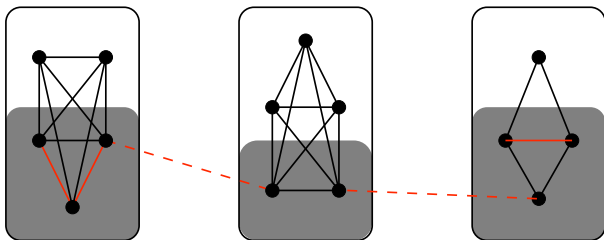
Un sommet est **affecté** s'il est incident à une arête de F .

1. Il y a au plus $2k$ sommets affectés par une k -édition F
2. Si F est une k -édition de $G = (V, E)$, alors $H = (V, E \triangle F)$ possède au plus $2k$ clusters.



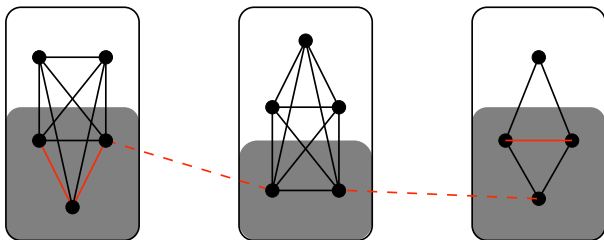
Clique critique

- L'ensemble S des sommets non-affectés d'un cluster de $H = (V, E \triangle F)$ induit une clique module dans G .



Clique critique

- L'ensemble S des sommets non-affectés d'un cluster de $H = (V, E \triangle F)$ induit une clique module dans G .
- Un ensemble S de sommets de G est une **clique critique** si c'est une clique module maximale.

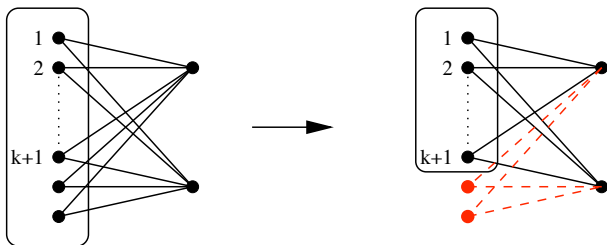


Règles de réduction

1. Supprimer les composantes connexes de G qui sont des cliques.

Règles de réduction

1. Supprimer les composantes connexes de G qui sont des cliques.
2. Si G possède un clique critique K de taille $|K| > k + 1$, alors supprimer $|K| - k + 1$ sommets de K .



Théorème

Le problème k -CLUSTER EDITING admet un noyau de taille $4k^2 + 2k$. Ce noyau est calculable en temps $O(n + m)$

Preuve

Dans une instance réduite:

1. Chaque cluster est bipartitionné en une clique critique et un ensemble de sommets affectés;

Théorème

Le problème k -CLUSTER EDITING admet un noyau de taille $4k^2 + 2k$. Ce noyau est calculable en temps $O(n + m)$

Preuve

Dans une instance réduite:

1. Chaque cluster est bipartitionné en une clique critique et un ensemble de sommets affectés;
2. Il y a au plus $2k$ clusters, $2k$ sommets affectés;

Théorème

Le problème k -CLUSTER EDITING admet un noyau de taille $4k^2 + 2k$. Ce noyau est calculable en temps $O(n + m)$

Preuve

Dans une instance réduite:

1. Chaque cluster est bipartitionné en une clique critique et un ensemble de sommets affectés;
2. Il y a au plus $2k$ clusters, $2k$ sommets affectés;
3. Il y a au plus $2k$ cliques critiques, chacune de taille au plus $k + 1$.

Théorème

Le problème k -CLUSTER EDITING admet un noyau de taille $4k^2 + 2k$. Ce noyau est calculable en temps $O(n + m)$

Preuve

Dans une instance réduite:

1. Chaque cluster est bipartitionné en une clique critique et un ensemble de sommets affectés;
2. Il y a au plus $2k$ clusters, $2k$ sommets affectés;
3. Il y a au plus $2k$ cliques critiques, chacune de taille au plus $k + 1$.

En fait, le k -CLUSTER EDITING admet un noyau linéaire de taille $4k$.

COMPLÉTION EN GRAPHE TRIANGULÉ

- ▶ **Entrée** : Un graphe $G = (V, E)$ (non-orienté)
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il un ensemble de k arêtes F disjoint de E tel que $H = (V, E \cup F)$ est triangulé ?
(H est triangulé ssi tout cycle de longueur au moins 4 possède une corde)

Règles de réduction

1. **[Sunflower]** Soit $xy \notin E$,
 $A_{xy} = \{z \mid xzy \text{ consécutifs dans un cycle sans corde}\}$
Si $|A_{xy}| > 2k$, alors ajouter xy et décrémenter k

COMPLÉTION EN GRAPHE TRIANGULÉ

- ▶ **Entrée** : Un graphe $G = (V, E)$ (non-orienté)
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il un ensemble de k arêtes F disjoint de E tel que $H = (V, E \cup F)$ est triangulé ?

Règles de réduction

1. **[Sunflower]** Soit $xy \notin E$,
 $A_{xy} = \{z \mid xzy \text{ consécutifs dans un cycle sans corde}\}$
Si $|A_{xy}| > 2k$, alors ajouter xy et décrementer k
2. **[Irrelevant vertices]** Supprimer les points d'articulations et les sommets simpliciaux de G
(i.e. les sommets x tels que $N(x)$ est une clique).

COMPLÉTION EN GRAPHE TRIANGULÉ

- ▶ **Entrée** : Un graphe $G = (V, E)$ (non-orienté)
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Existe-t-il un ensemble de k arêtes F disjoint de E tel que $H = (V, E \cup F)$ est triangulé ?

Théorème

Le problème de la complétion en graphe triangulé possède un noyau quadratique.

Preuve : Tout sommet non affecté n'est ni simplicial, ni un séparateur, donc son voisinage contient une arête de F . Pour chaque arête de F , au plus $2k$ voisins non affectés
 \Rightarrow noyau de taille $4k^2 + 2k$

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

- Règles de réduction en SUNFLOWER

- Analyse de la structure de la solution

- Analyse par comptage

Programmation linéaire

Règles de réduction globales

- Réduction en couronne pour VERTEX COVER

- Un noyau linéaire pour le CLUSTER EDITING

- Un noyau linéaire pour FAST à l'aide de couplages

Un kernel quadratique pour (MAXSAT, κ)

- ▶ *Données* : Une formule sous forme normale conjonctive
 $\Phi : (x \vee \bar{y} \vee z) \wedge (u \vee \bar{w} \vee y \vee \bar{z}) \wedge (w \vee y \vee \bar{z}) \wedge (z \vee v)$
- ▶ *Paramètre* : un entier k
- ▶ *Question* : Existe-t'il une affectation satisfaisant au moins k clauses ?

Un kernel quadratique pour (MAXSAT, κ)

- ▶ *Données* : Une formule sous forme normale conjonctive
 $\Phi : (x \vee \bar{y} \vee z) \wedge (u \vee \bar{w} \vee y \vee \bar{z}) \wedge (w \vee y \vee \bar{z}) \wedge (z \vee v)$
- ▶ *Paramètre* : un entier k
- ▶ *Question* : Existe-t'il une affectation satisfaisant au moins k clauses ?

Observation 1

Il existe toujours une affectation satisfaisant $\frac{m}{2}$ clauses.
(Donc on peut poser $k > \frac{m}{2}$.)

Un kernel quadratique pour (MAXSAT, κ)

- ▶ *Données* : Une formule sous forme normale conjonctive
 $\Phi : (x \vee \bar{y} \vee z) \wedge (u \vee \bar{w} \vee y \vee \bar{z}) \wedge (w \vee y \vee \bar{z}) \wedge (z \vee v)$
- ▶ *Paramètre* : un entier k
- ▶ *Question* : Existe-t'il une affectation satisfaisant au moins k clauses ?

Observation 1

Il existe toujours une affectation satisfaisant $\frac{m}{2}$ clauses.
(Donc on peut poser $k > \frac{m}{2}$.)

Preuve : Soit A une affectation quelconque. Si A ne satisfait pas $\frac{m}{2}$ clauses, alors l'affectation complémentaire \bar{A} satisfait $\frac{m}{2}$ clauses.

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$,

$$L = |\Phi_I|$$

Φ_S la formule restreinte aux clauses de taille $< k$,

$$S = |\Phi_S|$$

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$, $L = |\Phi_I|$

Φ_S la formule restreinte aux clauses de taille $< k$, $S = |\Phi_S|$

Observation 2

Si $L \geq k$, alors il existe une affectation de Φ satisfaisant au moins k clauses: (Φ, k) est vraie.

(Donc on peut poser $L < k$)

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$,

$$L = |\Phi_I|$$

Φ_S la formule restreinte aux clauses de taille $< k$,

$$S = |\Phi_S|$$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$, $L = |\Phi_I|$

Φ_S la formule restreinte aux clauses de taille $< k$, $S = |\Phi_S|$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Preuve :

- ▶ Si (Φ, k) est vraie alors Φ_I est satisfiable.
- ▶ Si $(\Phi_S, k - L)$ est vraie, alors il reste au moins $k - (k - L)$ variable libres pour chaque clause de Φ_I .

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$,

$$L = |\Phi_I|$$

Φ_S la formule restreinte aux clauses de taille $< k$,

$$S = |\Phi_S|$$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Lemme

La taille de Φ_S est quadratique en k : i.e. $|\Phi_S| \in O(k^2)$

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$,

$$L = |\Phi_I|$$

Φ_S la formule restreinte aux clauses de taille $< k$,

$$S = |\Phi_S|$$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Lemme

La taille de Φ_S est quadratique en k : i.e. $|\Phi_S| \in O(k^2)$

Preuve : Φ_S possède $m - L$ clauses de taille $< k$.

Or $m < 2k$, donc Φ_S contient au plus $2k^2$ littéraux.

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$,

$$L = |\Phi_I|$$

Φ_S la formule restreinte aux clauses de taille $< k$,

$$S = |\Phi_S|$$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Lemme

La taille de Φ_S est quadratique en k : i.e. $|\Phi_S| \in O(k^2)$

Preuve finie ?

Un kernel quadratique pour (MAXSAT, κ) (suite)

$$\Phi = \Phi_I \wedge \Phi_S$$

Φ_I la formule restreinte aux clauses de taille $\geq k$, $L = |\Phi_I|$

Φ_S la formule restreinte aux clauses de taille $< k$, $S = |\Phi_S|$

Observation 3

Si $L < k$, alors (Φ, k) est vraie ssi $(\Phi_S, k - L)$ est vraie.

Lemme

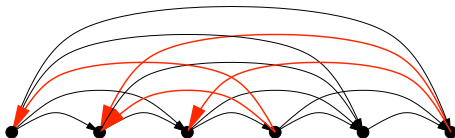
La taille de Φ_S est quadratique en k : i.e. $|\Phi_S| \in O(k^2)$

Preuve finie ?

Il reste à préciser que le kernel $(\Phi_S, k - L)$ est calculable en temps polynomial !

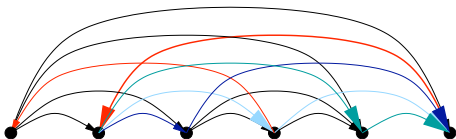
FEEDBACK ARC SET IN TOURNAMENTS (FAST)

- ▶ **Entrée** : Un tournoi $T = (V, \vec{A})$
- ▶ **Paramètre** : $k \in \mathbb{N}$
- ▶ **Question** : Peut-on retourner au plus k arcs de \vec{A} pour obtenir un tournoi transitif ?



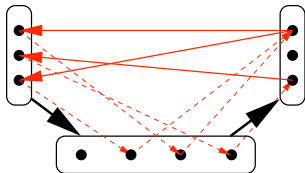
Règles de réduction

1. Si un sommet x n'appartient à aucun triangle, supprimer x .
2. Si un arc \overrightarrow{xy} appartient à au moins $k + 1$ triangles, retourner \overrightarrow{xy}



Règles de réduction

1. Si un sommet x n'appartient à aucun triangle, supprimer x .
2. Si un arc \overrightarrow{xy} appartient à au moins $k + 1$ triangles, retourner \overrightarrow{xy}
3. Si M est un module transitive de taille p tel que $|Z = \{\overrightarrow{xy} \mid x \in N^+(M), y \in N^-(M)\}| < p$, alors retourner tous les arcs de Z .

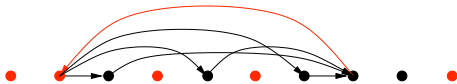


Théorème

Le problème FAST possède un noyau avec $k\sqrt{k}$ sommets.

Preuve :

- ▶ au plus $2k$ sommets affectés, dont la source et le puits de la solution.
- ▶ le *span* d'un arc retour est au plus $2k + 2$ (sinon, il y aura k triangles partageant cet arc)
$$\sum \text{span}(e) \leq 2k^2 + 2k$$



Théorème

Le problème FAST possède un noyau avec $k\sqrt{k}$ sommets.

Preuve :

- ▶ au plus $2k$ sommets affectés, dont la source et le puits de la solution.
- ▶ le *span* d'un arc retour est au plus $2k + 2$ (sinon, il y aura k triangles partageant cet arc) $\sum \text{span}(e) \leq 2k^2 + 2k$
- ▶ chaque module transitif M_i possède au plus m_i arcs retour, chacun de longueur au moins m_i , donc

$$\sum m_i^2 \leq \sum \text{span}(e) \leq 2k^2 + 2k$$

$$\Rightarrow \sum m_i \leq O(k\sqrt{k})$$

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

Réduction en couronne pour VERTEX COVER

Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

Vertex Cover et programmation linéaire

Soit un graphe $G = (V, E)$. Alors le programme linéaire $L_{vc}(G)$ a une solution optimale semi-entière.

$$L_{vc}(G) = \min \sum_{v \in V} x_v \text{ tel que } \begin{cases} x_u + x_v \geq 1 & \forall uv \in E \\ 0 \leq x_v \leq 1 & \forall v \in V \end{cases}$$

Vertex Cover et programmation linéaire

Soit un graphe $G = (V, E)$. Alors le programme linéaire $L_{vc}(G)$ a une solution optimale semi-entière.

$$L_{vc}(G) = \min \sum_{v \in V} x_v \text{ tel que } \begin{cases} x_u + x_v \geq 1 & \forall uv \in E \\ 0 \leq x_v \leq 1 & \forall v \in V \end{cases}$$

Soit $(x_v)_{v \in V}$ une solution optimale demi-entière de $L_{vc}(G)$. Pour $r \in \{0, \frac{1}{2}, 1\}$, on note

- ▶ $V_r = \{v \in V \mid x_v = r\}$ et
- ▶ $G_r = G[V_r]$.

Lemme

Soient un graphe $G = (V, E)$ et $(x_v)_{v \in V}$ une solution optimale demi-entière de $L_{vc}(G)$. Alors

1. $VC(G_{\frac{1}{2}}) \geq |V_{\frac{1}{2}}| / 2$
2. $VC(G_{\frac{1}{2}}) = VC(G) - |V_1|$

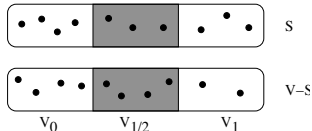
Lemme

Soient un graphe $G = (V, E)$ et $(x_v)_{v \in V}$ une solution optimale demi-entière de $L_{vc}(G)$. Alors

1. $VC(G_{\frac{1}{2}}) \geq |V_{\frac{1}{2}}| / 2$
2. $VC(G_{\frac{1}{2}}) = VC(G) - |V_1|$

Preuve

(i) Si S est un V.C. de G , alors $S_r = S \cap V_r$ est un V.C. de G_r .



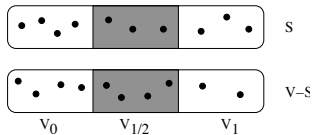
Lemme

Soient un graphe $G = (V, E)$ et $(x_v)_{v \in V}$ une solution optimale demi-entière de $L_{vc}(G)$. Alors

1. $VC(G_{\frac{1}{2}}) \geq |V_{\frac{1}{2}}| / 2$
2. $VC(G_{\frac{1}{2}}) = VC(G) - |V_1|$

Preuve

(i) Si S est un V.C. de G , alors $S_r = S \cap V_r$ est un V.C. de G_r .



(ii) Si S' est un V.C. de $G_{\frac{1}{2}}$, alors $S' \cup V_1$ est un V.C. de G .

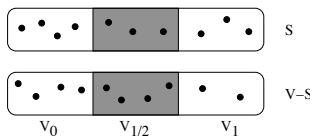
Lemme

Soient un graphe $G = (V, E)$ et $(x_v)_{v \in V}$ une solution optimale demi-entière de $L_{vc}(G)$. Alors

1. $VC(G_{\frac{1}{2}}) \geq |V_{\frac{1}{2}}| / 2$
2. $VC(G_{\frac{1}{2}}) = VC(G) - |V_1|$

Preuve

(i) Si S est un V.C. de G , alors $S_r = S \cap V_r$ est un V.C. de G_r .



(ii) Si S' est un V.C. de $G_{\frac{1}{2}}$, alors $S' \cup V_1$ est un V.C. de G .

$$|S'| + |V_1| \geq VC(G) \geq \sum_{v \in V} x_v = \frac{1}{2}|V_{\frac{1}{2}}| + |V_1|$$

Théorème

k -VERTEX COVER possède un kernel de taille au plus $2k$.

Théorème

k -VERTEX COVER possède un kernel de taille au plus $2k$.

Preuve

- ▶ $k - |V_1| < 0$: G ne possède pas de VC de taille k

Théorème

k -VERTEX COVER possède un kernel de taille au plus $2k$.

Preuve

- ▶ $k - |V_1| < 0$: G ne possède pas de VC de taille k
- ▶ $k - |V_1| = 0$: Si $G_{\frac{1}{2}}$ possède une arête alors G ne possède pas de solution sinon $G_{\frac{2}{2}}$ possède un VC de taille k

Théorème

k -VERTEX COVER possède un kernel de taille au plus $2k$.

Preuve

- ▶ $k - |V_1| < 0$: G ne possède pas de VC de taille k
- ▶ $k - |V_1| = 0$: Si $G_{\frac{1}{2}}$ possède une arête alors G ne possède pas de solution sinon G possède un VC de taille k
- ▶ $k - |V_1| > 0$ et $|V_{\frac{1}{2}}| > 2(k - |V_1|)$: $G_{\frac{1}{2}}$ ne possède pas de VC de taille $k - |V_1| \Rightarrow G$ ne possède pas de VC de taille k .

Théorème

k -VERTEX COVER possède un kernel de taille au plus $2k$.

Preuve

- ▶ $k - |V_1| < 0$: G ne possède pas de VC de taille k
- ▶ $k - |V_1| = 0$: Si $G_{\frac{1}{2}}$ possède une arête alors G ne possède pas de solution sinon G possède un VC de taille k
- ▶ $k - |V_1| > 0$ et $|V_{\frac{1}{2}}| > 2(k - |V_1|)$: $G_{\frac{1}{2}}$ ne possède pas de VC de taille $k - |V_1| \Rightarrow G$ ne possède pas de VC de taille k .
- ▶ Sinon, le noyau est l'instance $(G_{\frac{1}{2}}, k - |V_1|)$.

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

Réduction en couronne pour VERTEX COVER

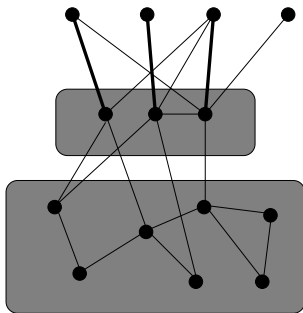
Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

Couronne

Une couronne dans un graphe $G = (V, E)$ est un graphe biparti $B = (I, N, E')$ tel que

1. I est un ensemble indépendant et $N = N(I)$;
2. E' est le sous-ensemble d'arêtes de E entre I et N ;
3. B possède un couplage de taille $|N|$



Lemme

Soit $B = (I, N, E')$ une couronne dans un graphe $G = (V, E)$,
alors $\text{VERTEX COVER}(G) = \text{VERTEX COVER}(G \setminus B) + |N|$

Lemme

Soit $B = (I, N, E')$ une couronne dans un graphe $G = (V, E)$,
alors $\text{VERTEX COVER}(G) = \text{VERTEX COVER}(G \setminus B) + |N|$

Théorème

Si G est un graphe sans couronne admettant un VERTEX COVER de taille k , alors G possède au plus $3k$ sommets.

Lemme

Soit $B = (I, N, E')$ une couronne dans un graphe $G = (V, E)$, alors $\text{VERTEX COVER}(G) = \text{VERTEX COVER}(G \setminus B) + |N|$

Théorème

Si G est un graphe sans couronne admettant un VERTEX COVER de taille k , alors G possède au plus $3k$ sommets.

Remarque

L'intérêt de cette règle est qu'elle **ne dépend pas du paramètre** k . La taille du noyau est donc meilleure. Ce type de règle est en général plus difficile à obtenir.

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

Réduction en couronne pour VERTEX COVER

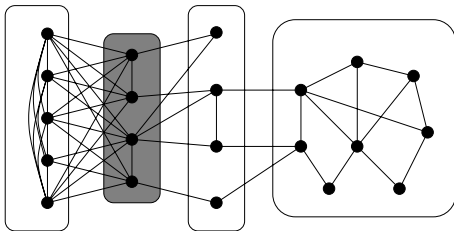
Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

Décomposition en couronne

Une décomposition en couronne d'un graphe $G = (V, E)$ est une partition (C, H, N, X) des sommets V telle que:

1. C est une clique;
2. tout sommet de C est adjacent à tout sommet de H ;
3. H est un séparateur pour C et $N \cup H$;
4. $N = \{v \in V \setminus (H \cup C) \mid \exists u \in H, uv \in E\}$.



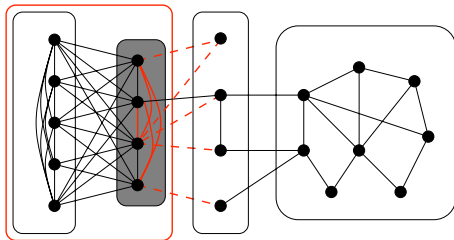
Règle de réduction en couronne

Si (G, k) une instance de CLUSTER EDITING admet une décomposition en couronne (C, H, N, X) telle que

$$|C| \geq |H| + |N| - 1$$

alors transformer (G, k) en (G', k') avec

$G' = G - (C \cup H)$ et $k' = k - |E \cap H^2| - |E \cap (H \times N)|$.



Règle de réduction en couronne

Si (G, k) une instance de CLUSTER EDITING admet une décomposition en couronne (C, H, N, X) telle que

$$|C| \geq |H| + |N| - 1$$

alors transformer (G, k) en (G', k') avec

$$G' = G - (C \cup H) \text{ et } k' = k - |E \cap H^2| - |E \cap (H \times N)|.$$

Théorème

Si (G, k) est une instance de CLUSTER EDITING sans décomposition en couronne, alors G possède au plus $6k$ sommets.

Un premier exemple: VERTEX COVER

Définitions

Autres exemples de noyaux quadratiques

Règles de réduction en SUNFLOWER

Analyse de la structure de la solution

Analyse par comptage

Programmation linéaire

Règles de réduction globales

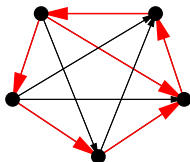
Réduction en couronne pour VERTEX COVER

Un noyau linéaire pour le CLUSTER EDITING

Un noyau linéaire pour FAST à l'aide de couplages

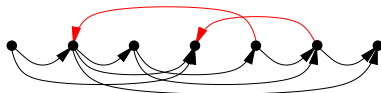
FAST

Observation : un tournoi est transitif (ou acyclique) ssi il ne contient pas de triangle (orienté).



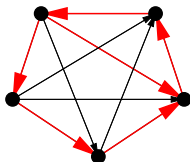
Définitions / terminologie

- On note $T_\sigma = (V, A, \sigma)$ un tournoi dont les sommets sont ordonnés selon une permutation σ



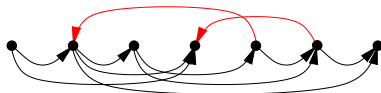
FAST

Observation : un tournoi est transitif (ou acyclique) ssi il ne contient pas de triangle (orienté).



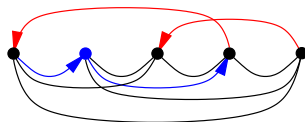
Définitions / terminologie

- On note $T_\sigma = (V, A, \sigma)$ un tournoi dont les sommets sont ordonnés selon une permutation σ



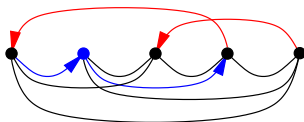
- Un arc \overrightarrow{uv} de T_σ est un arc **retour** si $u <_\sigma v$.
- Soit \overrightarrow{uv} un arc retour $\text{span}(\overrightarrow{uv}) = \{w \in V : u <_\sigma w <_\sigma v\}$

FAST - certificats

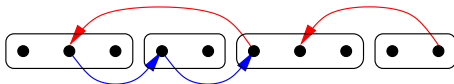


- ▶ Si \vec{uv} est un arc retour de T_σ et $w \in \text{span}(\vec{uv})$ n'est incident à aucun arc retour, $c(\vec{uv}) = \{u, w, v\}$ est un **certificat** pour \vec{uv}
- ▶ Si F est un ensemble d'arcs retours, un **F-certificat** est un ensemble $c(F) = \{c(f) : f \in F\}$ de **certificats arc-disjoints**.

FAST - certificats



- ▶ Si \vec{uv} est un arc retour de T_σ et $w \in \text{span}(\vec{uv})$ n'est incident à aucun arc retour, $c(\vec{uv}) = \{u, w, v\}$ est un **certificat** pour \vec{uv}
- ▶ Si F est un ensemble d'arcs retours, un **F-certificat** est un ensemble $c(F) = \{c(f) : f \in F\}$ de **certificats arc-disjoints**.
- ▶ Une partition ordonnée $\mathcal{P} = \{V_1, \dots, V_l\}$ d'un tournoi T_σ est **saine** si l'ensemble F des arcs retours externes peut être certifié uniquement avec des arcs externes.



FAST - règles de réduction

FAST - règles de réduction

1. [Sommet inutile] Supprimer les sommets n'appartenant à aucun triangle orienté
2. [Partition saine] Si \mathcal{P} est une partition saine de T_σ , retourner les arcs retours externes et décrémenter k en conséquence.

FAST - règles de réduction

1. [Sommet inutile] Supprimer les sommets n'appartenant à aucun triangle orienté
 2. [Partition saine] Si \mathcal{P} est une partition saine de T_σ , retourner les arcs retours externes et décrémenter k en conséquence.
- ▶ Comment calculer une partition saine en temps polynomial ?
 - ▶ Montrer que ces deux règles permettent d'obtenir un noyau de taille $4k$

FAST - conflict packing

Un **conflict packing** est un ensemble **maximal** \mathcal{C} de certificats arcs disjoints. On note $V(\mathcal{C})$ les sommets couverts par \mathcal{C} .

FAST - conflict packing

Un **conflict packing** est un ensemble **maximal** \mathcal{C} de certificats arcs disjoints. On note $V(\mathcal{C})$ les sommets couverts par \mathcal{C} .

Lemme 1

Si \mathcal{C} est un conflict packing d'une instance positive (T, k) de FAST, alors

$$|V(\mathcal{C})| \leq 3k$$

FAST - conflict packing

Un **conflict packing** est un ensemble **maximal** \mathcal{C} de certificats arcs disjoints. On note $V(\mathcal{C})$ les sommets couverts par \mathcal{C} .

Lemme 1

Si \mathcal{C} est un conflict packing d'une instance positive (T, k) de FAST, alors

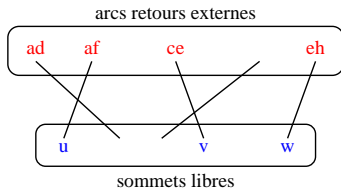
$$|V(\mathcal{C})| \leq 3k$$

Lemme 2

Si \mathcal{C} est un conflict packing d'une instance (T, k) de FAST, alors
 \exists une permutation σ tq $\forall \overrightarrow{uv}$ arc retour de T_σ , $\{u, v\} \subseteq V(\mathcal{C})$

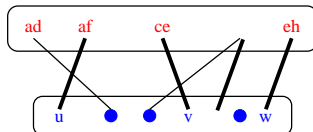
FAST

Lemme 3 : Si le tournoi T possède au moins $4k$ sommets, alors on peut calculer une partition saine en temps polynomial.



FAST

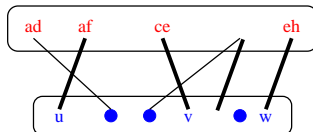
Lemme 3 : Si le tournoi T possède au moins $4k$ sommets, alors on peut calculer une partition saine en temps polynomial.



- il existe un couplage, et donc un vertex cover S de taille k
- \mathcal{P} est la partition obtenue en isolant les sommets libres n'appartenant pas à S (il en existe car $|V| > 4k$).

FAST

Lemme 3 : Si le tournoi T possède au moins $4k$ sommets, alors on peut calculer une partition saine en temps polynomial.

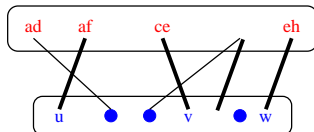


- il existe un couplage, et donc un vertex cover S de taille k
- \mathcal{P} est la partition obtenue en isolant les sommets libres n'appartenant pas à S (il en existe car $|V| > 4k$).

\mathcal{P} est une partition saine.

FAST

Lemme 3 : Si le tournoi T possède au moins $4k$ sommets, alors on peut calculer une partition saine en temps polynomial.



- ▶ il existe un couplage, et donc un vertex cover S de taille k
- ▶ \mathcal{P} est la partition obtenue en isolant les sommets libres n'appartenant pas à S (il en existe car $|V| > 4k$).

\mathcal{P} est une partition saine.

Théorème : FAST (paramétré par la taille k de la solution) admet un noyau de taille au plus $4k$