COMPLEXITÉ PARAMÉTRÉE

"des outils efficaces pour la résolution exacte de problèmes difficiles"

Christophe Paul

Journée scientifique du LIRMM
15 juin 2010

Un peu d'histoire de la complexité

Théorie de la NP-Complétude

- Théorème de Cook (1971) : SAT est NP-COMPLET
  (2-SAT $\in$ **P**)
- 21 problèmes NP-complets de Karp (1972)

Idée admise : les problèmes NP-complets sont tous équivalents !

Un peu d'histoire de la complexité

Théorie de la NP-Complétude

- Théorème de Cook (1971) : SAT est NP-COMPLET
  (2-SAT $\in$ **P**)
- 21 problèmes NP-complets de Karp (1972)

Idée admise : les problèmes NP-complets sont tous équivalents !

Complexité de SAT en fonction de différents paramètres

1. **nombre n de variables** — 2^n affectations possibles
   $O(1,49^n)$ pour 3-SAT

Un peu d'histoire de la complexité

Théorie de la NP-Complétude

- Théorème de Cook (1971) : SAT est NP-COMPLET
  (2-SAT $\in$ **P**)
- 21 problèmes NP-complets de Karp (1972)

Idée admise : les problèmes NP-complets sont tous équivalents !

Complexité de SAT en fonction de différents paramètres

1. **nombre n de variables** — 2^n affectations possibles
   $O(1,49^n)$ pour 3-SAT
2. **nombre m de clauses** — $O(1,24^m)$

Un peu d'histoire de la complexité

Théorie de la NP-Complétude

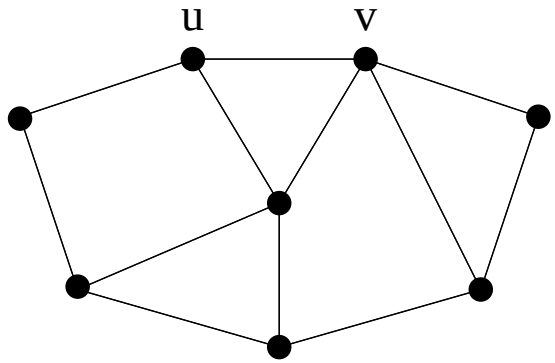
- Théorème de Cook (1971) : SAT est NP-COMPLET
  (2-SAT $\in$ **P**)
- 21 problèmes NP-complets de Karp (1972)

Idée admise : les problèmes NP-complets sont tous équivalents !

Complexité de SAT en fonction de différents paramètres

1. **nombre n de variables** — 2^n affectations possibles
   $O(1,49^n)$ pour 3-SAT
2. **nombre m de clauses** — $O(1,24^m)$
3. **longueur l de la formule** — $O(1,08^l)$

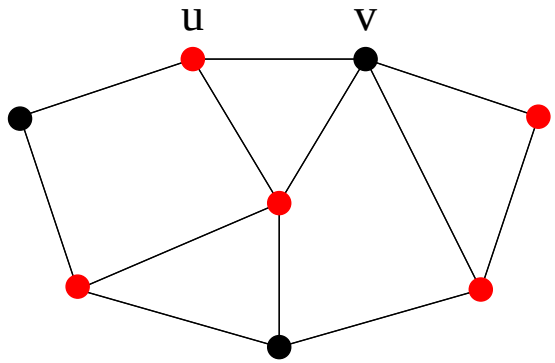
Vertex Cover v.s. Independent Set



k-Vertex Cover

$(n-k)$-Independent Set

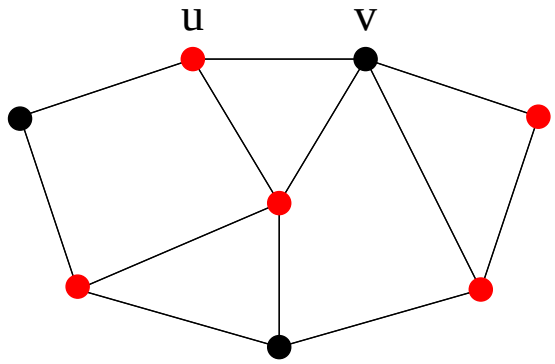
Vertex Cover v.s. Independent Set



k-Vertex Cover

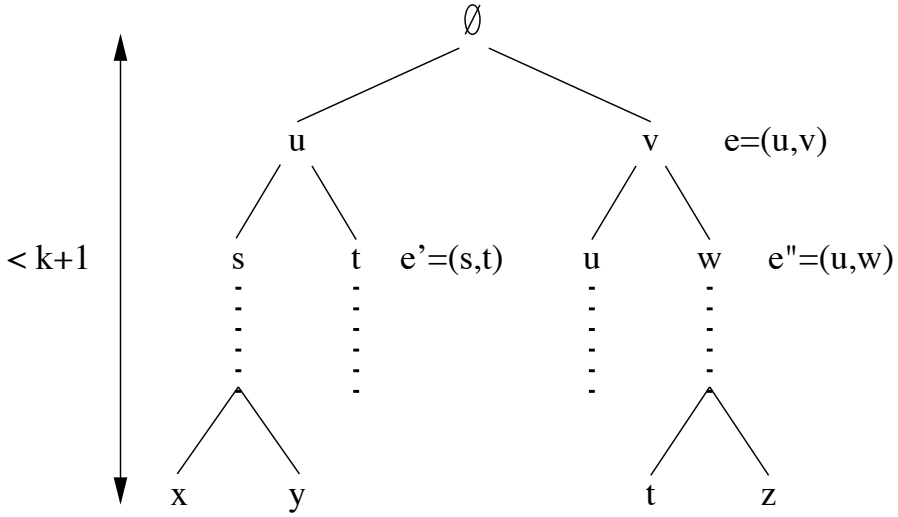
$(n-k)$-Independent Set

Vertex Cover v.s. Independent Set



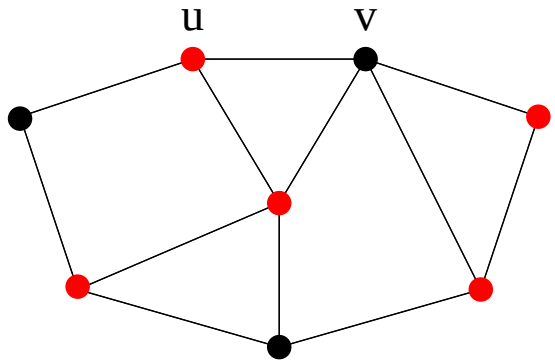
k-Vertex Cover

$(n-k)$-Independent Set

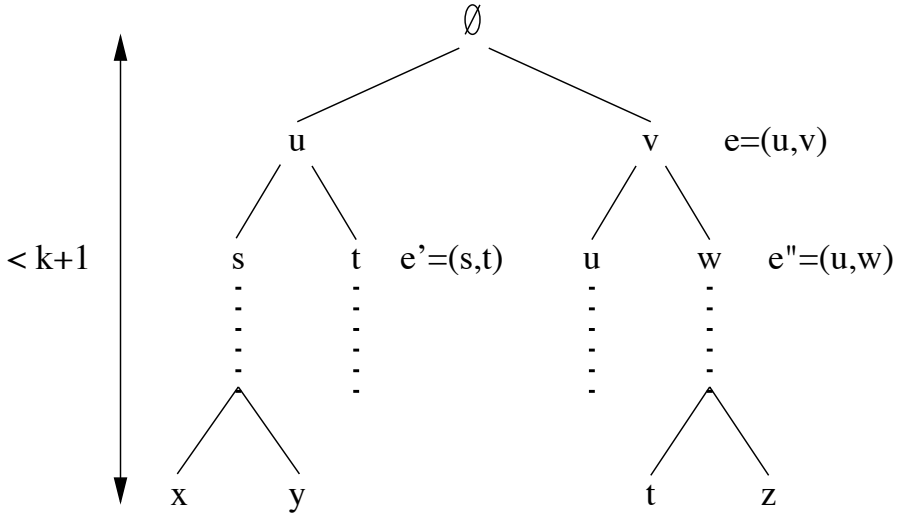


$O(2^k.(m+n))$

Vertex Cover v.s. Independent Set

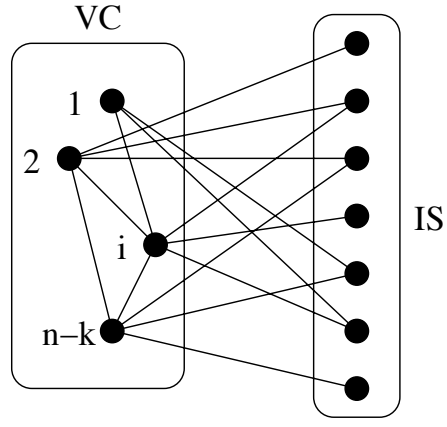


k-Vertex Cover



$O(2^k.(m+n))$

$(n-k)$-Independent Set



$O(2^{(n-k)}.(m+n))$

- Constat: De nombreux problèmes difficiles (NP-COMPLET) peuvent être résolus de manière efficace sur des données pratiques

- Constat: De nombreux problèmes difficiles (NP-COMPLET) peuvent être résolus de manière efficace sur des données pratiques

- *"Mesurer la complexité seulement en fonction de la taille de la donnée signifie ignorer toute information structurelle sur l'instance donnée..."*

  J. Flum and M. Grohe

- Constat: De nombreux problèmes difficiles (NP-COMPLET) peuvent être résolus de manière efficace sur des données pratiques

- *"Mesurer la complexité seulement en fonction de la taille de la donnée signifie ignorer toute information structurelle sur l'instance donnée..."*

  J. Flum and M. Grohe

- *"L'idée fondamentale est de restreindre l'explosion combinatoire, semble-t-il inévitable, qui est responsable de la croissance exponentielle du temps de calcul, à un paramètre spécifique au problème..."*

  R. Niedermeier

Problèmes FPT (Fixed Parameter Tractable)

$\rightsquigarrow$ établir une "dichotomie" / hiérarchie entre les problèmes difficiles

Problèmes FPT (Fixed Parameter Tractable)

$\rightsquigarrow$ établir une "dichotomie" / hiérarchie entre les problèmes difficiles

Un problème Π paramétré par k est FPT s'il peut être résolu par un algorithme de complexité :

$$\boxed{O(f(k).n^{O(1)})}$$

Problèmes FPT (Fixed Parameter Tractable)

$\rightsquigarrow$ établir une "dichotomie" / hiérarchie entre les problèmes difficiles

Un problème Π paramétré par k est FPT s'il peut être résolu par un algorithme de complexité :

$$O(f(k).n^{O(1)})$$

MAIS quid de la complexité exacte (constante cachée) ?

$$2^{k^{k^{\cdot^{\cdot^{\cdot^{k}}}}}}.n^{O(1)}$$

Problèmes FPT (Fixed Parameter Tractable)

$\rightsquigarrow$ établir une "dichotomie" / hiérarchie entre les problèmes difficiles

Un problème Π paramétré par k est FPT s'il peut être résolu par un algorithme de complexité :

$$O(f(k).n^{O(1)})$$

MAIS quid de la complexité exacte (constante cachée) ?

$$2^{k^{k^{k^{\cdot^{\cdot^{\cdot}}}}}}.n^{O(1)}$$

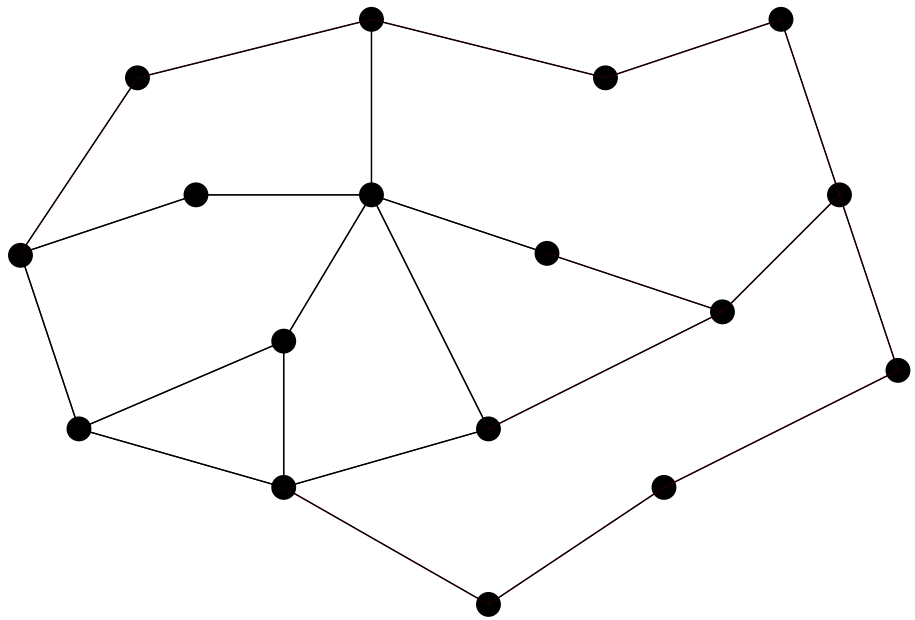
$\rightsquigarrow$ Les problèmes paramétrés se classent dans une hiérarchie

$$\text{FPT} \subseteq \text{W[1]} \subseteq \text{W[2]} \ldots \subseteq W[i] \subseteq \ldots \subseteq \text{W[P]}$$

Exemples de problèmes paramétrés (1)

Test de circuits VLSI (SysMic)

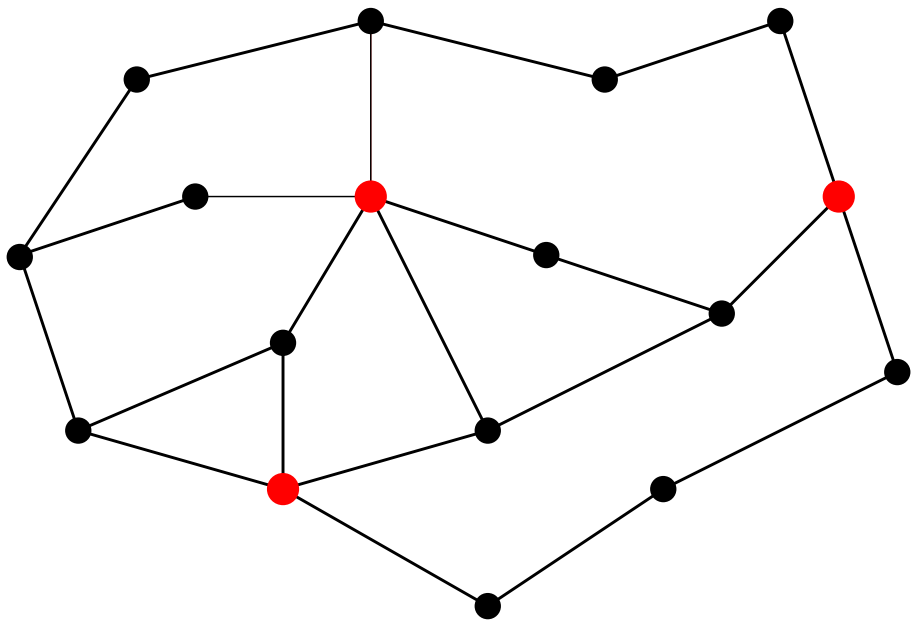
$\rightsquigarrow$ Intégration de BILBOs (Built-in Logic Block Observers) dans les circuits



Exemples de problèmes paramétrés (1)

Test de circuits VLSI (SysMic)

$\rightsquigarrow$ Intégration de BILBOs (Built-in Logic Block Observers) dans les circuits

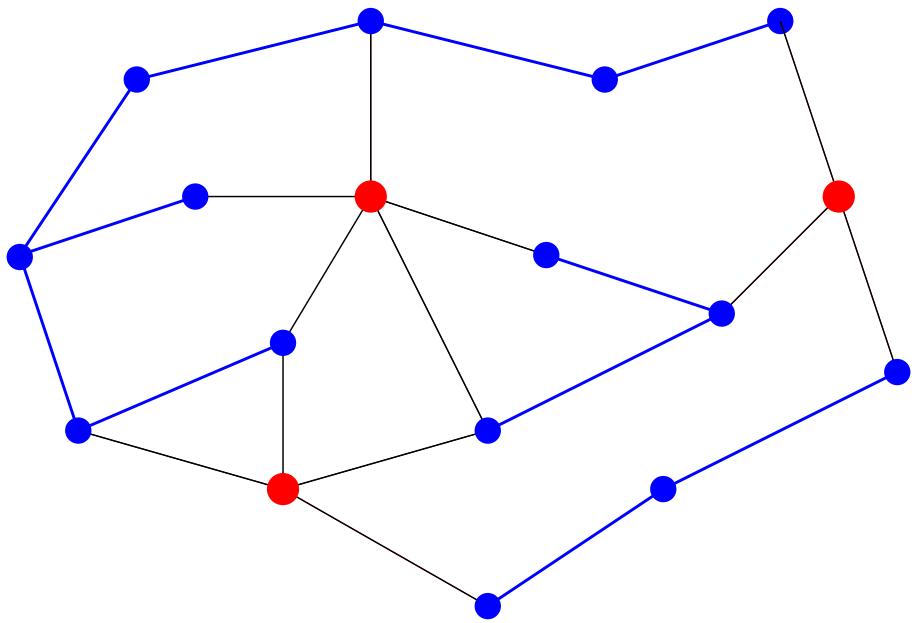


FEEDBACK VERTEX SET: algorithme paramétré en $5^k.n^{O(1)}$

Exemples de problèmes paramétrés (1)

Test de circuits VLSI (SysMic)

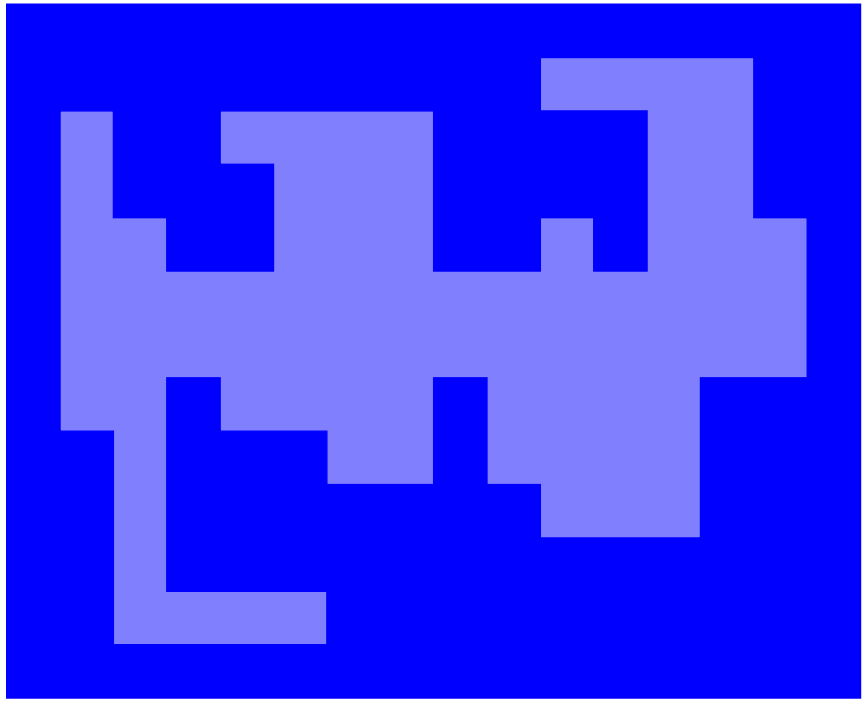
$\rightsquigarrow$ Intégration de BILBOs (Built-in Logic Block Observers) dans les circuits



FEEDBACK VERTEX SET: algorithme paramétré en $5^k.n^{O(1)}$

Exemples de problèmes paramétrés (2)

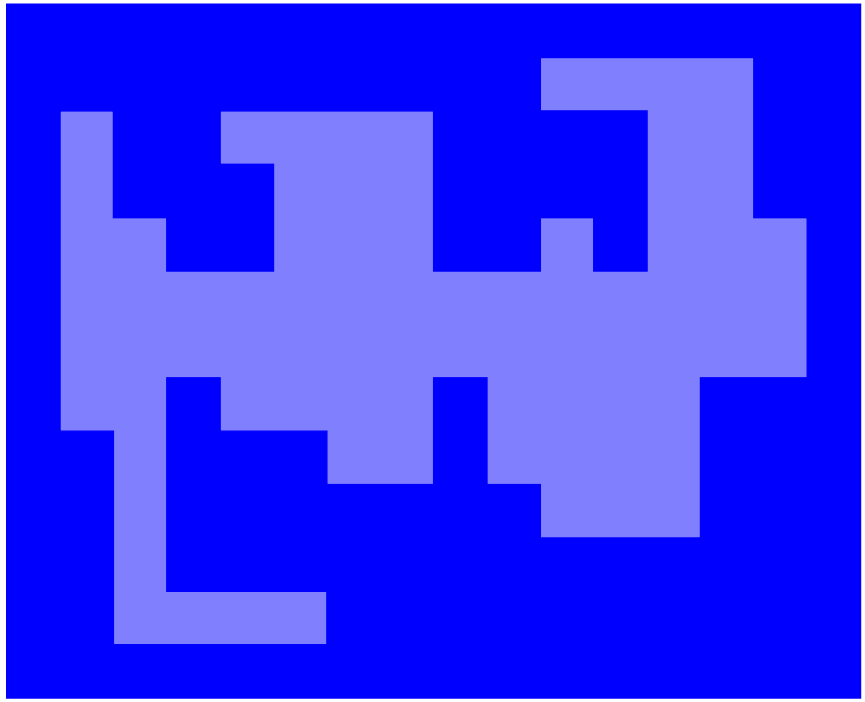
Découpe par un cutter / contrôle qualité de pièces (ROB)



Objectif : minimiser le nombre de changements de direction

Exemples de problèmes paramétrés (2)

Découpe par un cutter / contrôle qualité de pièces (ROB)



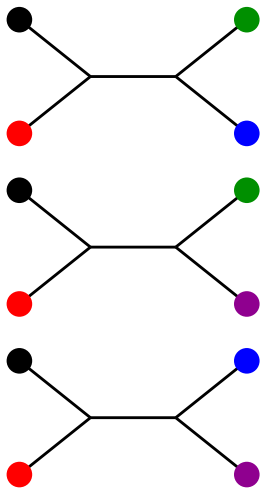
Objectif : minimiser le nombre de changements de direction

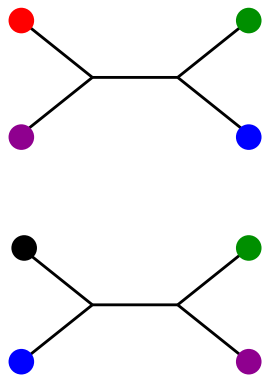
Discrete Milling with Turn Cost

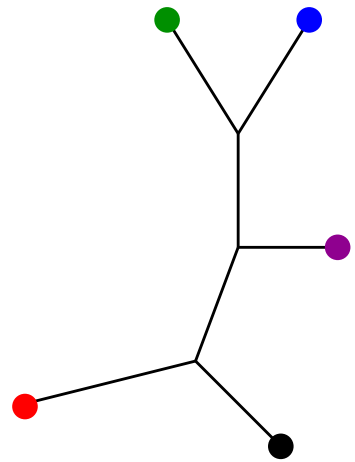
algorithme paramétré en $O(2^{k^2 \log k}.n)$

Exemples de problèmes paramétrés (3)

Reconstruction phylogénétique (MAB)

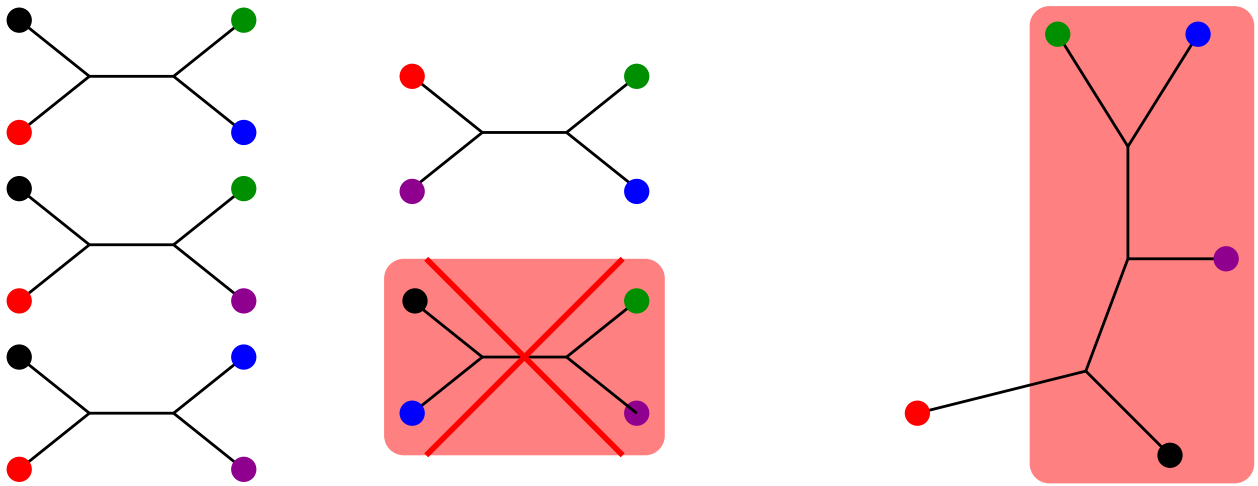






Exemples de problèmes paramétrés (3)

Reconstruction phylogénétique (MAB)



Minimum Quartet Inconsistency

algorithme paramétré en $4^k.n + n^4$

Exemples de problèmes paramétrés (4)

Intelligence Artificielle - Contraintes (COCONUT)

$$(a \vee b) \wedge (\overline{a} \vee c) \wedge (\overline{c} \vee d) \wedge (\overline{a} \vee \overline{d}) \wedge (\overline{b} \vee \overline{e}) \wedge (\overline{c} \vee e) \wedge (a \vee \overline{e})$$

Exemples de problèmes paramétrés (4)

Intelligence Artificielle - Contraintes (COCONUT)

$$(a \vee b) \wedge (\overline{a} \vee c) \wedge (\overline{c} \vee d) \wedge (\overline{a} \vee \overline{d}) \wedge (\overline{b} \vee \overline{e}) \wedge (\overline{c} \vee e) \wedge (a \vee \overline{e})$$

Almost 2-sat

algorithme paramétré en $O(15^k * k * m^3)$

Mais aussi

- propagation de contraintes
- programmation logique / model checking
- bases de données
- système de votes
- optimisation dans les réseaux

Preprocessing - data reduction (kernelisation)

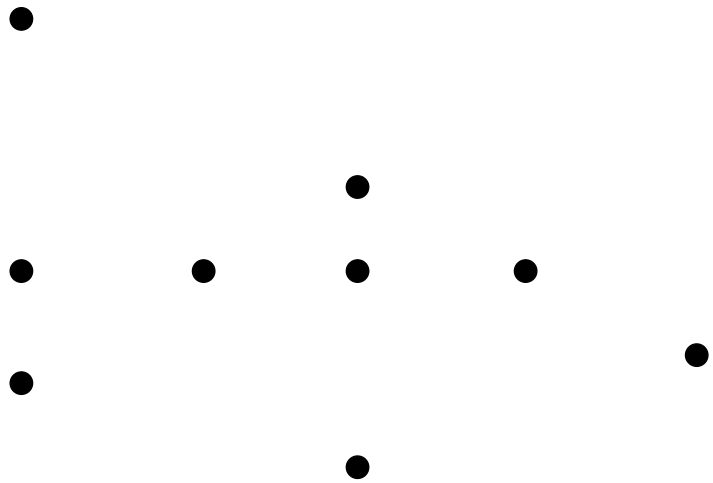
Idée - principe

- simplification / filtrage des données
- réduction de la taille des données

Objectifs

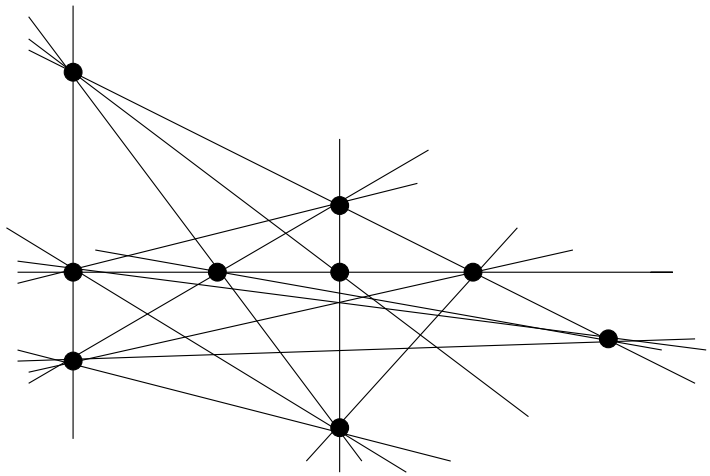
- efficacité
- garantie de l'existence d'une solution
- contrôle de la taille de l'instance réduite

Kernel (1): un problème géométrique



Existe-t-il k lignes couvrant l'ensemble S de points ?

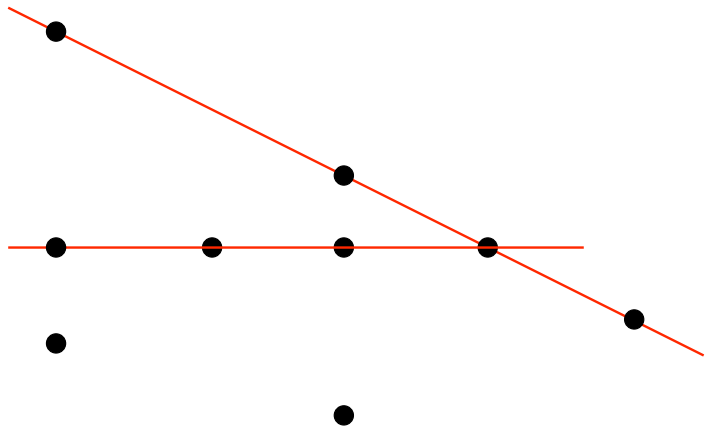
Kernel (1): un problème géométrique



Existe-t-il k lignes couvrant l'ensemble S de points ?

Observation 1: On peut se restreindre aux lignes générées par les paires de points de S

Kernel (1): un problème géométrique

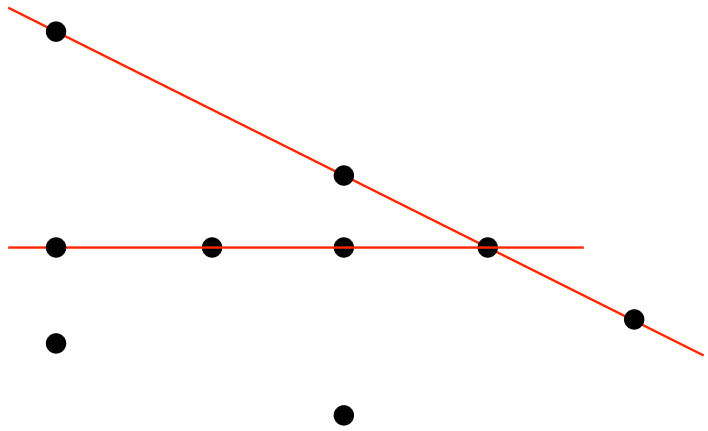


Existe-t-il k lignes couvrant l'ensemble S de points ?

Observation 2: Si une ligne L contient au moins $k+1$ points, alors elle appartient la solution (si elle existe) (e.g. ici $k = 3$)

$\Rightarrow$ supprimer L et soustraire 1 à k

Kernel (1): un problème géométrique



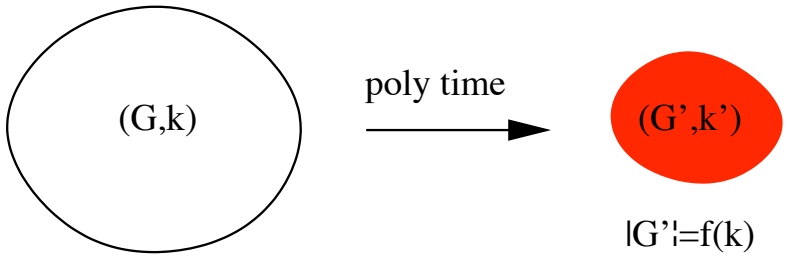
Existe-t-il k lignes couvrant l'ensemble S de points ?

Observation 2: Si une ligne L contient au moins $k+1$ points, alors elle appartient la solution (si elle existe) (e.g. ici $k = 3$)

$\Rightarrow$ supprimer L et soustraire 1 à k

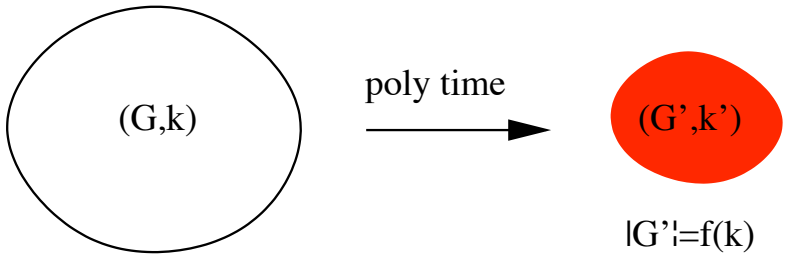
$\Rightarrow$ l'instance réduite contient au plus k^2 points.

Kernel (2)



Une kernelisation pour un problème paramétré Π est un algorithme polynomial qui étant donnée une instance (G, k) calcule une instance équivalente (G', k') telle que $k' = f(k)$.

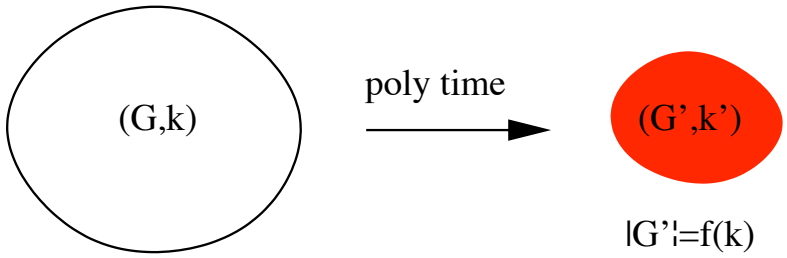
Kernel (2)



Une kernelisation pour un problème paramétré Π est un algorithme polynomial qui étant donnée une instance (G, k) calcule une instance équivalente (G', k') telle que $k' = f(k)$.

Un problème π est FPT ssi il admet une kernelisation

Kernel (2)

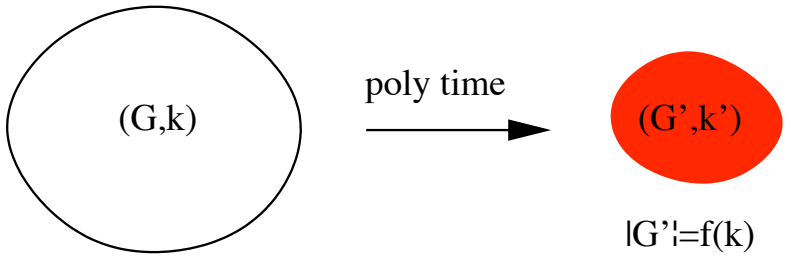


Une kernelisation pour un problème paramétré Π est un algorithme polynomial qui étant donnée une instance (G, k) calcule une instance équivalente (G', k') telle que $k' = f(k)$.

Un problème π est FPT ssi il admet une kernelisation

- Quels problèmes admettent un noyau / kernel de taille polynomiale ?

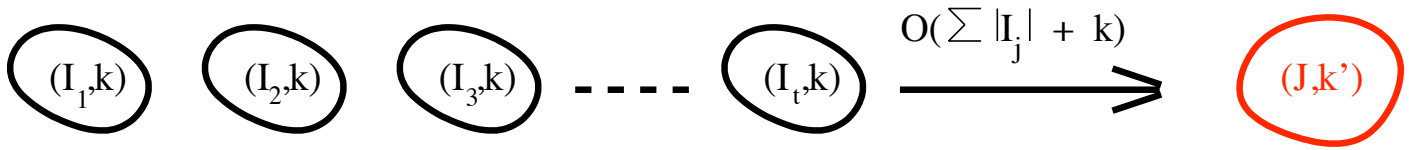
Kernel (2)



Une kernelisation pour un problème paramétré Π est un algorithme polynomial qui étant donnée une instance (G, k) calcule une instance équivalente (G', k') telle que $k' = f(k)$.

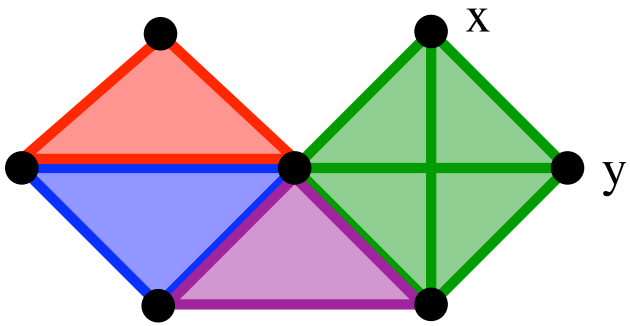
Un problème π est FPT ssi il admet une kernelisation

- Quels problèmes admettent un noyau / kernel de taille polynomiale ?



Théorème [Bodlaender et al.] Tout problème NP-complet composable n'admet pas de noyau polynomial sauf si $PH = \Sigma_p^3$

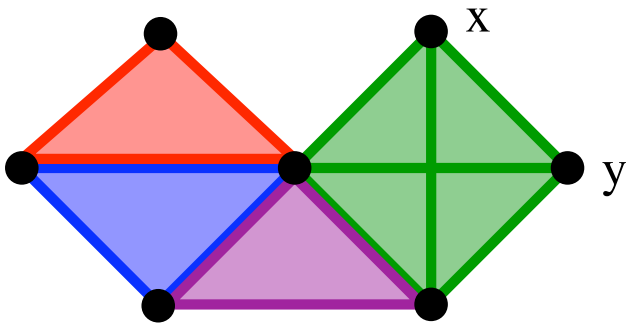
L'exemple de Edge Clique-Cover



Peut-on couvrir les arêtes d'un graphe par au plus k cliques ?

Edge Clique-Cover admet un noyau de taille 2^k sommets.

L'exemple de Edge Clique-Cover



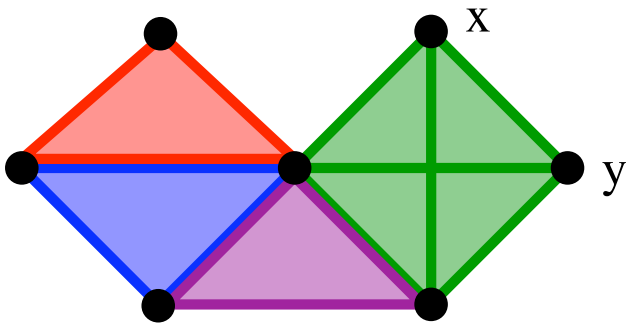
Peut-on couvrir les arêtes d'un graphe par au plus k cliques ?

Edge Clique-Cover admet un noyau de taille 2^k sommets.

Règles de réduction

1. Supprimer les sommets isolés
2. S'il existe deux sommets x et y tels que $N[x] = N[y]$, supprimer l'un des deux sommets.

L'exemple de Edge Clique-Cover



Peut-on couvrir les arêtes d'un graphe par au plus k cliques ?

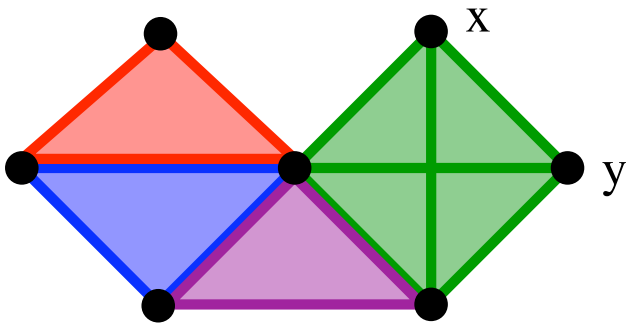
Edge Clique-Cover admet un noyau de taille 2^k sommets.

Supposons qu'il existe k cliques $C_1 \dots C_k$ couvrant E.
A chaque sommet x on asscocie un verteur C de k bits tel que

$$C[x, i] = 1 \iff x \in C_i$$

Obs. : $\forall i \in [k]\ C[x, i] = C[y, i]$ ssi $N[x] = N[y]$

L'exemple de Edge Clique-Cover



Peut-on couvrir les arêtes d'un graphe par au plus k cliques ?

Edge Clique-Cover admet un noyau de taille 2^k sommets.

Supposons qu'il existe k cliques $C_1 \dots C_k$ couvrant E.
A chaque sommet x on asscocie un verteur C de k bits tel que

$$C[x, i] = 1 \iff x \in C_i$$

Obs. : $\forall i \in [k]\ C[x, i] = C[y, i]$ ssi $N[x] = N[y]$

Problème ouvert

Edge Clique-Cover admet-il un noyau polynomial ?

Conclusion

- Algorithmes FPT : techniques efficaces (et souvent assez simples) pour la résolution exacte de problèmes difficiles
- Kernels : cadre théorique pour étudier la notion de pre-processing
- Méthodes opérationelles sur des grandes données (e.g. clustering) en mixant les techniques de kernelisation et arbres de recherche bornée