Séance 2 : les classes de complexité FPT et XP

Alix Munier-Kordon et Claire Hanen

Décembre 2023

Lorsqu'un problème Π est NP$-$complet, on considère (sous l'hypothèse courante P $\neq$ NP) qu'il n'existe pas d'algorithme de complexité polynomiale dans la taille de l'instance qui puisse résoudre toutes les instances. La question est alors l'existence de paramètres associés au problème Π qui rendraient le problème plus facile à résoudre. L'objet de cette séance est d'introduire les bases de la complexité paramétrée sur 3 problèmes classiques de la théorie des graphes, suivi d'une définition plus formelle d'un problème paramétré, et des classes de complexité XP et FPT.

1 Première étude de trois problèmes paramétrés classiques

On commence par un rappel de vocabulaire. Soit $G = (V, E)$ un graphe non orienté et les sous-ensembles de sommets $V' \subseteq V$ et $E' \subseteq E$ tels que les arêtes de E' relient des sommets de V'.

— Alors si E' est maximal par rapport à V', autrement dit E' contient toutes les arêtes de E qui relient des sommets de V', alors $G' = (V', E')$ est un **sous-graphe** de G ; on le note $G' = (V', E)$.

— Sinon, $G' = (V', E')$ est un **sous-graphe partiel** de G.

On rappelle également que $n = |V|$ désigne le nombre de sommets et $m = |E|$ le nombre d'arêtes. On considère généralement que m est en $\mathcal{O}(n^2)$.

On considère les versions paramétrées de trois problèmes classiques :

$k-$**Vertex Coloring**

Input : Un graphe non orienté $G = (V, E)$;

Question : Existe-t-il une coloration des sommets de G en au plus k couleurs de sorte que pour toute arête $\{x, y\} \in E$, les sommets x et y sont de couleur différente ?

$k-$**Independent Set**

Input : Un graphe non orienté $G = (V, E)$;

Question : Existe-t-il sous-ensemble $X \subseteq V$ de taille minimum k, dont les sommets ne sont pas reliés par des arêtes, ie. $\forall (x, y) \in X^2$, $\{x, y\} \notin E$?

Un ensemble indépendant est aussi appelé un **stable**.

$k-$**Vertex Cover**

Input : Un graphe non orienté $G = (V, E)$;

Question : Existe-t-il un sous-ensemble $X \subseteq V$ de taille maximum k, tel que toute arête $\{x, y\} \in E$ vérifie $x \in X$ ou $y \in X$?

Les versions non paramétrées de ces trois problèmes sont NP-complets. Ainsi, trouver un algorithme exact polynomial pour toutes leurs instances semble peu probable. Mais sont-ils vraiment d'égale difficulté ? Peut-on les analyser plus finement ?

$k-$**Vertex Coloring** est un problème NP-complet pour $k = 3$ (voir exercice 3 feuille 1). Donc même pour des valeurs bornées de k (sauf $k = 2$) on a peu de chances de trouver un algorithme efficace.

Si on considère un algorithme « brute force »pour énumérer les solutions possibles : on a k^n colorations en k couleurs possibles de tous les sommets. Pour chacune d'elles, on peut vérifier en $\mathcal{O}(m)$ si les arêtes ont des extrémités de couleur différente. Ainsi, un tel algorithme énumératif a une complexité de $\mathcal{O}(n^2 k^n)$. On observe que pour $k = 2$, ou $k = 3$, cette complexité est exponentielle en n.

$k-$Independent Set Il est possible, en $\mathcal{O}(k^2)$ de vérifier qu'un sous ensemble X de k sommets est un ensemble stable (en vérifiant pour chaque couple de sommets $(x, y) \in X^2$ que $\{x, y\} \notin E$). Or il y a $\binom{n}{k}$ sous-ensembles de taille k dans un ensemble de taille $n = |V|$ et $\binom{n}{k} \leq n^k$. L'algorithme qui énumère les sous-ensembles de taille k et vérifie pour chacun d'eux s'ils forment un stable a donc une complexité $\mathcal{O}(k^2 n^k)$.

On observe que pour $k = 3$ on obtient une complexité $\mathcal{O}(n^3)$, qui est donc polynomiale. Cependant, comme le degré du polynôme croit avec k, cela n'est concrètement utilisable que pour des petites valeurs de k.

$k-$Vertex Cover On considère la propriété suivante.

Propriété 1 *Soit $G = (V, E)$ un graphe non orienté, $X \subseteq V$ un sous ensemble de sommets de k élements et un sommet $x \in X$. L'ensemble X est une couverture de G si et seulement si $X - \{x\}$ est une couverture de taille $k - 1$ du sous-graphe $G_x = (V - \{x\}, E)$.*

Soit $\{x, y\} \in E$ une arête de G. On observe que si X est une couverture de G alors $x \in X$ ou $y \in X$. L'algorithme récursif 1 VCB est basé sur cette remarque et la propriété 1.

Algorithm 1 VCB(G, k)

Require: $G = (V, E)$, un entier k
Ensure: Retourne True ssi il existe une couverture de sommets de G de cardinalité k.

if $k = 0$ ou G n'a pas d'arête **then**
    return (G ne possède pas d'arête ($E = \emptyset$))
end if
Choisir une arête $\{x, y\} \in E$
Soient les sous-graphes $G_x = (V - \{x\}, E)$ et $G_y = (V - \{y\}, E)$
if VCB$(G_x, k - 1)$ **then**
    return True
end if
return VCB$(G_y, k - 1)$

Propriété 2 *La complexité de l'algorithme 1 est $\mathcal{O}(2^k(m + n))$.*

Ici, le degré du polynome ne varie pas, mais la constante multiplicative augmente (beaucoup) avec k.

Soient alors les trois fonctions

1. $f(x) = n^2 3^n$ associéet à $3-$**Vertex Coloring**;
2. $g(x) = 9n^3$ associée à $3-$**Independent Set**;
3. $h(x) = 8n^2$ associéet à $3-$**Vertex Cover**;

La figure 1 montre les courbes de ces trois fonctions les unes par rapport aux autres pour de petites valeurs de n. Qu'en déduisez vous sur les ordres de grandeur ?

FIGURE 1 – Comparaison des fonctions $f(x)$, $g(x)$ et $h(x)$. Sur la courbe de gauche, en vert $g(x)$ et en bleue $f(x)$. Sur celle de droite, en vert $g(x)$ et en rouge $h(x)$.

2 Définition d'un problème paramétré

Deux définitions équivalents existent, sous la forme langage ou problème de décision.

Définition 1 *Un problème paramétré est un langage $L \subseteq \Sigma^* \times \mathbb{N}$, où Σ est un alphabet fini. On note (x, k) une instance de ce problème, où k est le paramètre.*

Définition 2 *Un problème de décision paramétré est caractérisé par une donnée constituée d'une instance d'un problème de décision et d'un entier k correspondant au paramètre, suivi d'une question qui peut ou non faire apparaître le paramètre.*

Définition 3 *Par convention, la* **taille** *d'une instance (x, k) d'un problème paramétré Π est $|(x, k)| = |x| + k$. La valeur $|x|$ de est ici la taille de codage de x.*

Le paramètre k est considéré « codé en unaire », cad que c'est sa valeur qui est considérée, soit une représentation en base 1, et non sa représentation en base 2. Les trois problèmes précédents peuvent être formulés comme des problèmes paramétrés par la valeur k présente dans la description du problème. Quelle est alors la taille d'une instance (x, k) pour ces 3 problèmes ?

En voici un autre exemple. Un graphe est **regulier** si tout des sommets sont de même degré.

$k-$**Clique Regular**

Input : Un graphe non orienté régulier de degré k $G = (V, E)$, un entier $r > 0$;

Question : Existe-t-il une clique de taille r pour G ?

Le problème **Clique Regular** peut être également paramétré par r, on obtient alors :

$r-$**Clique Regular**

Input : Un entier $k > 0$, un graphe non orienté régulier de degré k $G = (V, E)$;

Question : Existe-t-il une clique de taille r pour G ?

La complexité des algorithmes de résolution des problèmes paramétrés sera analysée vis à vis du paramètre. On peut alors caractériser certaines formes particulières de complexité, qui vont conduire à de nouvelles classes de problèmes.

Qu'est-ce qu'un bon paramètre ?

- un entier qui peut être assez petit dans certaines instances des problèmes réels.
- un entier qui, lorsqu'on le code en unaire, va permettre d'avoir des algorihmes de complexité polynomiale pour un paramètre fixé.

Exemples :

1. Quand on cherche à faire un circuit électronique pour réaliser une netlist, le nombre de couches du circuit est bornée (en géneral, max 10).
2. Quand on veut déplacer un ensemble de robots d'un point à un autre sans collision, le nombre de dégré de liberté de chacun des robots est inférieur à 10.
3. Dans un problème d'ordonnancement, il est possible que le nombre effectifs de tâches exécutables à un instant donné soit borné par une constante.

3 Les classes XP et FPT

On definit par la suite deux classes de problèmes en fonction de l'existence d'un algorithme de complexité au pire-cas en fonction de la taille $|(x, k)| = |x| + k$.

Définition 4 ([1]) *Un problème paramétré Π de paramètre k appartient à la classe XP s'il existe :*

- *deux fonctions $f, g : \mathbb{N} \to \mathbb{N}$ calculables ;*
- *un algorithme $\mathcal{A}$ résolvant toute instance (x, k) de Π avec une complexité $\mathcal{O}(f(k)(|x| + k)^{g(k)})$.*

L'algorithme développé pour $k-$**Independent Set** satisfait cette condition : en posant $f(k) = k^2$ et $g(k) = k$, on obtient $k^2 \times n^k = f(k)n^{g(k)}$, ce qui correspond à la définition. On en conclut que $k-$**Independent Set**$\in$XP.

Par contre l'algorithme développé pour $k-$**Vertex Coloring** de complexité $\mathcal{O}(n^2k^n)$ n'est pas de cette nature.

Nous verrons en exercice d'autres exemples.

Définition 5 ([1]) *Un problème paramétré Π de paramètre k appartient à la classe FPT (fixed parameter tractable) s'il existe :*
- *une fonction $f : \mathbb{N} \to \mathbb{N}$ calculable ;*
- *une constante c ;*
- *un algorithme $\mathcal{A}$ résolvant toute instance (x, k) de Π avec une complexité $\mathcal{O}(f(k)(|x| + k)^c)$.*

L'algorithme VCB pour résoudre $k-$**Vertex Cover** satisfait ces conditions. La construction de l'algorithme est du type « arbre de profondeur bornée », qui est très utilisée pour définir des algorithmes FPT. Nous verrons d'autres exemples basé sur le même principe en exercice.

Question : quelle est la relation d'inclusion entre XP et FPT ?

Soit Π un problème paramétré FPT de paramètre k. Si k est borné par une constante, Π peut être résolu par un algorithme polynomial. Les problèmes FPT constituent la classe des problèmes paramétrés les plus faciles à résoudre (par analogie avec les problèmes non paramétrés PTIME).

Références

[1] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer Publishing Company, Incorporated, 1st edition, 2015.

s