Notes de cours de Méthodes exactes : techniques classiques de noyaux bornes inférieures

Sabrina Ouazzani-Chahdi, M2 MOCA, Université Montpellier II

Décembre 2012

Résumé

Dans la détermination de l'existence ou non d'un noyau polynômial pour un problème donné, cette partie du cours s'attache à présenter des outils de recherche de noyaux de bornes inférieures exponentielles. Nous nous intéresserons en particulier à deux grandes familles d'algorithmes, que sont d'une part les algorithmes de distillation et de composition, et les transformations paramétrées d'autre part.

1 Exemple introductif

Lorsque l'on étudie un problème de kernelization, il semble naturel de vouloir obtenir le noyau le plus « faible » possible. Cependant, est-il toujours possible de limiter la taille du noyau à une fonction polynômiale ?

La réponse est bien entendu négative. Nous pouvons citer à titre d'illustration le problème récemment fermé de Edge clique-cover (Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk et Magnus Wahlstrom, 2011), ou encore le cas de Longest Path. Attardons nous sur ce dernier.

Énoncé : Soit G=(V, E) un graphe et $k \in \mathbb{N}$ un paramètre. G contient-il un chemin de taille k ?

Ce problème est NP-Complet.

Supposons qu'il existe algorithme de kernelization A qui retourne un noyau de taille polynomiale pour ce problème, et construisons une instance (G, k) avec t instances différentes, càd $(G, k) = (G_1, k) \oplus \cdots \oplus (G_t, k)$.

(G, k) admet donc un chemin de longueur k si et seulement si l'une des t instances admet un tel chemin.

Cela revient à identifier en temps polynômial un plus long chemin parmi ces graphes. Ce qui contredit le fait que longest past soit NP-Complet (sous l'hypothèse $P \neq NP$).

Nous étudierons dans la suite les résultats clés connus sur les principales variantes de ce type de structure.

2 Algorithmes de distillation et de composition

Un algorithme de ou-distillation A
pour un problème de décison Q sur l'alphabet Σ est un algorithme qui :
- en entrée, reçoit une séquence $(x_1, \ldots, x_t)$ d'instances (càd $x_i \in \Sigma^* \forall i \in [t]$) ;
- en sortie, retourne une instance y telle que :
  1. $y \in Q$ si et seulement si $\exists i \in [t], x_i \in Q$;
  2. la taille de y est polynômiale en la taille de la plus grande des instances d'entrée ;
- possède une complexité polynômiale en $\sum_{i=1}^{t} |x_i|$.

Conjecture 1 (Bodlaender, Downey, Fellows, Hermelin)**.** *Aucun problème NP-Complet n'est ou-distillable.*

Théorème 1 (Fortnow et Santhanam)**.** *S'il existe un problème NP-Complet ou-distillable, alors la hiérarchie polynômiale s'écrase au troisième niveau ($PH = \sum_p^3$).*

Généralisons cette définition aux problèmes paramétrés.

Un algorithme de ou-composition A
pour un problème de paramétré $(Q, \kappa) \subseteq \Sigma^* \times \mathbb{N}$ est un algorithme qui :
- en entrée, reçoit une séquence $((x_1, k), \ldots, (x_t, k))$ d'instances (càd $(x_i, k) \in \Sigma^* \times \mathbb{N}\ \forall i \in [t]$) ;
- en sortie, retourne une instance (y, k') telle que :
  1. $(y, k') \in (Q, \kappa)$ si et seulement si $\exists i \in [t], (x_i, k) \in (Q, \kappa)$;
  2. k' est polynômial en k ;
- possède une complexité polynômiale en $\sum_{i=1}^{t} |x_i| + k$.

Théorème 2 (Bodlaender, Downey, Fellows, Hermelin)**.** *Soit $(Q, \kappa) \in \Sigma^* \times \mathbb{N}$, un problème paramétré ou-composable tel que le problème non-paramétré $\tilde{Q} \in \Sigma^*$ est NP-Complet. Si (Q, κ) admet un noyau polynômial, alors $\tilde{Q}$ est ou-distillable.*

À titre d'illustration, revenons à l'exemple introductif et remarquons alors que l'existence d'un noyau polynômial pour longest path impliquerait l'existence d'un algorithme de ou-composition.

Détaillons désormais la preuve du théorème 2 : proposons un algorithme de ou-distillation pour $\tilde{Q}$.

Comment construire cet algorithme ?

Nous distinguons quatre principales étapes.

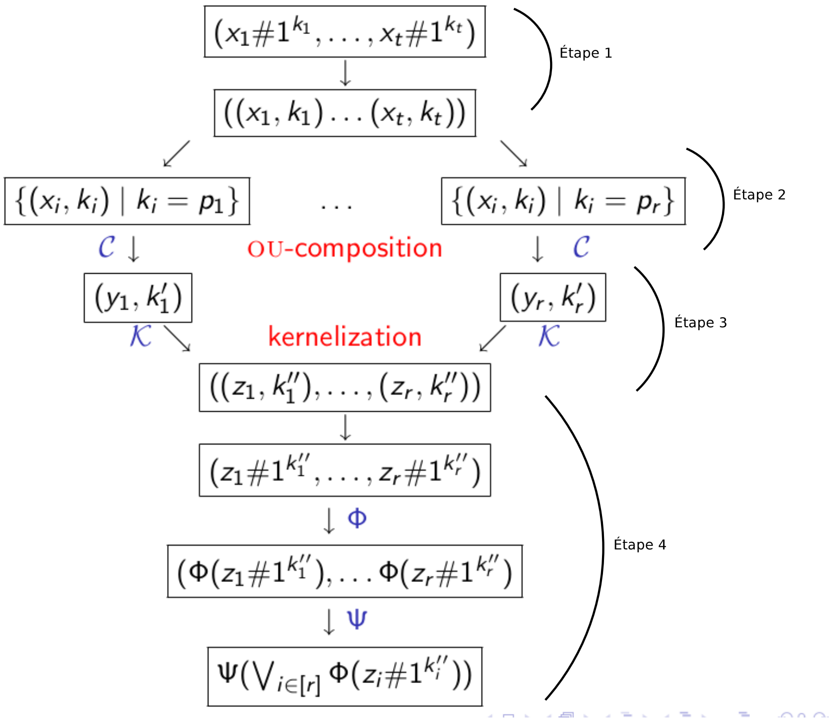

FIGURE 1 – Illustration du cours de la preuve du théorème 2.

1. Partir de $\tilde{Q}$ et le transformer en un problème paramétré Q. Cette transformation est possible en considérant qu'à chaque instance $(x, \kappa(x))$ de problème paramétré, nous pouvons associer une instance non paramétrée constituée d'une concaténation de x, d'un symbole de séparation et du codage unaire du paramètre.
2. Regrouper les instances de Q qui ont le même paramètre afin de pouvoir appliquer un algorithme de ou-composition (par définition, Q est ou-composable).
3. Choisir des instances représentatives pour chaque groupe et appliquer un algorithme de kernelization qui nous renvoie des instances polynômiales en la taille des paramètres.
4. Transformer les instances de Q ainsi obtenues en instances de $\tilde{Q}$, puis appliquer une première transformation polynômiale pour obtenir une conjonction de disjonctions (SAT), puis une deuxième transformation de SAT vers $\tilde{Q}$ pour obtenir une disjonction de conjonctions.

Notons qu'appliquer un algorithme de ou-composition à l'étape 2 permet de garantir que les paramètres restent polynômiaux.

L'étape 3 se fait sous l'hypothèse du théorème 2 sur l'existence d'un noyau polynômial, qui garantit l'existence d'un tel agorithme.

L'existence des deux transformations polynômiales citées à l'étape 4 est garantie par le fait que $\tilde{Q}$ soit NP-Complet.

Comment garantir que cet algorithme est bien ou-distillable ?

Il nous faut vérifier les trois points suivants :
- la complexité de l'algorithme est $\sum_{i=1}^{t} |x_i|$;
- l'instance retournée par l'algorithme est vrai si et seulement si il existe une instance vraie parmi les instances de départ de $\tilde{Q}$;
- la taille de l'instance de $\tilde{Q}$ retournée est polynômiale en la taille de la plus grande des instances d'entrée.

Démonstration. Les deux premiers points sont évidents. En effet, il est clair de voir que chaque étape est polynômiale en la somme des tailles des instances d'origine, et que la disjonction de conjonction couplée à des transformations polynômiales sur les instances de départ nous assure du point deux. Étudions donc le troisième.

L'application de l'algorithme de ou-composition permet de garantir des instances polynômiales en la taille des paramètres de chaque groupe, et donc de la taille du plus grand paramètre. L'application des étapes suivantes (kernelization et transformations polynômiales) permet de conserver cette propriété. □

Nous en déduisons la conclusion suivante :

D'après les théorèmes 1 et 2, **les problèmes ou-composables n'admettent pas de noyau polynômial**.

Corollaire 1. *À moins que $PH = \Sigma_p^3$, longest path n'admet pas de noyau polynômial.*

Lemme 1. *À moins que $PH = \Sigma_p^3$, k-sous-arborescence n'admet pas de noyau polynômial. Mais la version enracinée admet un noyau cubique.*

3 Transformations paramétrées polynomiales

Cependant, tous les problèmes ne sont pas ou-comparables. On se rammène alors à une réduction, en contrôlant le paramètre (de manière à ce qu'il reste polynômial). La méthode est la suivante :

1. Choisir un problème intermédiaire.
2. Montrer que ce problème est ou-comparable (n'admet pas de noyau polynomial).
3. Montrer que ce problème est plus facile que le problème d'origine pour la kernelisation.

Cette procédure est appelée une transformation paramétrée polynomiale (TPP). Définissons là formellement. Soient (P, π) et (Q, κ) deux problèmes paramétrés.

Une Tansformation paramétrée polynômiale
de (P, π) vers (Q, κ) est un algorithme polynomial A qui :
– à toute instance (x, k) de (P, π) associe une instance (x', k') de (Q, κ) ;
– $(x, k) \in (P, \pi)$ si et seulement si $(x', k') \in (Q, \kappa)$ et $k' \leq poly(k)$.

On note $(P, \pi) \leq_{TPP} (Q, \kappa)$.

Théorème 3. *(Bodlaender, Thomassé, Yeo) Soient (P, π) et (Q, κ) deux problèmes paramétrés tels que P est NP-Complet et Q appartient à NP. Si $(P, \pi) \leq_{TPP} (Q, \kappa)$ et si (P, π) n'admet pas de noyau polynomial, alors (Q, κ) n'admet pas de noyau polynomial.*

Démonstration. Procédons par l'absurde en supposant que (Q, κ) admette un noyau polynômial, et montrons que nous pouvons alors construire un algorithme polynômial qui réduit (P, π) à un noyau polynomial.

Comme P est NP-Complet, il peut être réduit à une instance de SAT par une transformation polynomiale.

(Q, κ) admet un noyau polynômial donc sa version non paramétrée $\tilde{Q}$ est ou-distillable (d'après le théorème 2).

Or $(P, \pi) \leq_{TPP} (Q, \kappa)$.

D'où, d'après le schéma de la preuve du théorème 2, on pourrait transformer polynomialement (P, π), instance équivalente à une instance (Q, κ), en une instance de SAT de P, en une instance ou-distillable de $\tilde{Q}$.

Alors (P, π) serait ou-distillable, ce qui contredit le fait qu'il n'admette pas de noyau polynômial. □

4 Perspectives et références

On peut aussi vouloir composer des problèmes NP-Complets vers des problèmes FPT en associant les techniques présentées ci-dessus. Cela mène aux compositions croisées.

Précisions aussi qu'il existe une opération de ET-composition avec un certain nombre de théorèmes similaires à ceux étudiés ici.

Ainsi, il a été montré récemment que Edge clique Cover était à la fois et-composable et ou-composable, d'où le fait qu'il n'admette pas de noyau polynômial.

Les références utilisées ici sont les diaporamas de cours de Christophe Paul ainsi que l'article de Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk et Magnus Wahlstrom intitulé *Clique cover and graph separation : New incompressibility results*, Novembre 2011 (fermant le problème de l'existence d'un noyau polynômial pour edge clique cover).