Classes Polynomiales Cachées : de la Théorie à la Pratique *

Achref El Mouelhi Philippe Jégou Cyril Terrioux

LSIS - UMR CNRS 7296
Aix-Marseille Université
13397 Marseille Cedex 20 (France)
{achref.elmouelhi, philippe.jegou, cyril.terrioux}@lsis.org

Résumé

Dans cette contribution, nous introduisons un cadre nouveau qui a pour objet d'étendre le champ des classes polynomiales. Cette approche est basée sur la transformation de CSP et introduit la notion de *classe polynomiale cachée*. Elle part du constat selon lequel les problèmes réels n'appartiennent pas en général à des classes polynomiales mais peuvent parfois, après simplification (des filtrages par cohérence locale par exemple) en faire partie. Ce papier est un résumé de l'article [2].

Abstract

In this paper, we introduce a new framework that aims to extend the scope of tractable classes. This approach is based on the transformation of CSP and introduced the concept of *Hidden Tractable Class*. It starts from the observation that real problems do not belong in general to tractable classes but can sometimes after simplifications (filterings for example) appear in a tractable class. This paper is a summary of [2].

1 Introduction

La plupart des classes polynomiales sont définies sur la base de propriétés très restrictives qui rendent leur présence au sein des instances réelles quasi inexistante. Elles se retrouvent ainsi très peu exploitées en pratique. Pour contourner cette réalité, nous proposons d'étendre le champ des classes polynomiales de CSP en introduisant le concept de *classes polynomiales cachées*. L'idée est que certaines instances qui n'appartiendraient pas à des classes polynomiales, pourraient, après filtrage par exemple, être transformées en instances figurant dans des classes bien connues.

*Ce travail est soutenu par l'Agence Nationale de la Recherche dans le cadre du projet TUPLES (ANR-2010-BLAN-0210).

2 Classes polynomiales cachées

Le cadre formel que nous proposons ici est basé sur la notion de *transformation* d'instances. Nous dirons que certaines instances constituent des classes polynomiales « cachées » découvertes par transformation d'instances. À cette fin, nous proposons un cadre général qui peut être utilisé pour de nombreuses classes polynomiales connues et pour l'essentiel des transformation d'instances de CSP, comme les filtrages par cohérence locale par exemple. Comme transformations classiques de CSP, nous trouvons généralement les suppressions de variables, de valeurs, l'ajout de contraintes ou la suppression de tuples dans les relations de compatibilité. Pour définir formellement cette notion, nous rappelons qu'une *instance de CSP* est un triplet (X, D, C) où X est un ensemble de variables, D un ensemble de domaines finis, avec un domaine $D(x)$ pour chaque variable x, et C un ensemble de contraintes. Chaque contrainte c est définie par un couple $(S(c), R(c))$ où $R(c)$ est une *relation de compatibilité* sur les domaines de l'ensemble $S(c) \subseteq X$ qui est la *portée* de la contrainte c.

Définition 1 *Étant donnée une instance de CSP $P = (X, D, C)$, t est dite une **transformation** de P si $t(P) = (t_{var}(X), t_{dom}(D), t_{cons}(C))$ vérifie :*
- $t_{var}(X) \subseteq X$
- $t_{dom}(D) = \{t_{dom}(D(x')) : x' \in t_{var}(X)$ *et* $t_{dom}(D(x')) \subseteq D(x)\}$
- $\forall c \in C, t_{cons}(c)$ *vérifie :*
  - $t_{cons}(S(c)) = S(c) \backslash \{x \in X : x \notin t_{var}(X)\}$ *et*
  - $t_{cons}(R(c)) \subseteq R(c)[t_{cons}(S(c))]$.
- $\forall c' \in t_{cons}(C)$ *telle que* $c' \notin C$, $t_{cons}(R(c')) \subseteq \prod_{x' \in S(c')} t_{dom}(D(x'))$.

D'autres transformations sont envisageables, mais nous nous limitons à celles-ci dans le cadre de ce travail. Cette définition permet d'introduire la notion de classe cachée :

Définition 2 *Étant données une transformation t et une classe d'instances $\mathcal{P}$, $t(\mathcal{P}) = \{t(P) : P \in \mathcal{P}\}$. Étant donnée une classe $\mathcal{C}$, la classe d'instances* ***mise en évidence par*** *t* ***pour*** *$\mathcal{C}$ est $\mathcal{C}^t = \{P : t(P) \in \mathcal{C}\}$. La classe $\mathcal{P}$ est dite* ***classe cachée*** *de $\mathcal{C}$ pour t, si $\mathcal{P} \subseteq \mathcal{C}^t$ i.e. si $t(\mathcal{P}) \subseteq \mathcal{C}$. Enfin, $\mathcal{P}$ est appelée* ***classe polynomiale cachée de*** *$\mathcal{C}$* ***pour*** *t si $\mathcal{C}$ est une classe polynomiale et si t préserve la cohérence et peut être calculée en temps polynomial.*

Ainsi, si $\mathcal{C}$ est une classe polynomiale et si $\mathcal{P}$ est une classe cachée de $\mathcal{C}$ pour une transformation t calculable en temps polynomial, pour le cas d'une instance $P \in \mathcal{P}$, il sera suffisant d'appliquer t sur P pour avoir une version modifiée de P qui sera traitable en temps polynomial. Au-delà, $\mathcal{P}$ sera aussi considérée comme classe polynomiale. Notre démarche ayant pour objet d'étendre des classes polynomiales à l'aide de transformations d'instances, il est naturel d'étudier la puissance relative de ces transformations. Nous formalisons cela par des comparaisons entre classes mises en évidence par des transformations t pour des classes $\mathcal{C}$.

Définition 3 *Étant données deux transformations t_1 et t_2, et deux classes $\mathcal{C}_1$ et $\mathcal{C}_2$, on dit que $\mathcal{C}_2^{t_2}$ est* ***plus grande*** *que $\mathcal{C}_1^{t_1}$ si $\mathcal{C}_1^{t_1} \subseteq \mathcal{C}_2^{t_2}$. De plus, on dit que $\mathcal{C}_2^{t_2}$ est* ***strictement plus grande*** *que $\mathcal{C}_1^{t_1}$ si $\mathcal{C}_1^{t_1} \subsetneq \mathcal{C}_2^{t_2}$, et on dit que $\mathcal{C}_1^{t_1}$ et $\mathcal{C}_2^{t_2}$ sont* ***incomparables*** *si aucune relation entre elles n'existe. Finalement, on dit que $\mathcal{C}_1^{t_1}$ et $\mathcal{C}_2^{t_2}$ sont* ***égales*** *si $\mathcal{C}_1^{t_1} = \mathcal{C}_2^{t_2}$.*

Sur la base de ces relations, il est possible de définir un certain nombre de propriétés qui sont détaillées dans [2], offrant ainsi un cadre d'étude général pour la comparaison de classes et de transformations. Par exemple, on peut facilement constater que pour toute paire de classes $\mathcal{C}_1$ et $\mathcal{C}_2$ telles que $\mathcal{C}_1 \subseteq \mathcal{C}_2$, et pour toute transformation t, on a $\mathcal{C}_1^t \subseteq \mathcal{C}_2^t$. Cette contribution ayant pour objet d'étendre l'intérêt pratique des classes polynomiales, il est naturel d'étudier expérimentalement l'étendue des classes cachées.

3 Le cas de BTP et des filtrages

L'étude de la classe polynomiale BTP [1] semble assez naturelle ici car elle constitue l'une des classes les plus larges qui aient été proposées jusqu'à présent. De même, coupler son étude avec les techniques de filtrage semble tout à fait naturel car ce sont les transformations les plus exploitées en pratique, que ce soit en prétraitement ou bien alors pendant une recherche par des

	BTP	BTP^{AC}	BTP^{PIC}	BTP^{maxRPC}	BTP^{SAC}	BTP^{NIC}	BTP^{SPC}
# inst.	12	191	400	493	550	900	594
# cons.	-	46	47	47	47	83	71

TABLE 1 – Instances et classes cachées de BTP.

algorithmes tels que MAC par exemple. Aussi, nous avons étudié les classes cachées de BTP qui peuvent être découvertes par les filtrages usuels tels que AC, SAC, PIC, $maxRPC$, RPC, PC, SPC ou encore NIC. Nous avons pour cela considéré les instances de la compétition CSP 2008, et dénombré celles qui apparaissent dans des classes cachées. Alors qu'à la base, très peu d'instances figurent dans BTP (précisément 12), la table 1 permet d'observer qu'elles sont finalement plus nombreuses dès lors qu'elles sont recherchées au sein de classes cachées. Ces résultats s'avèrent en adéquation avec les comparaisons théoriques présentées dans la figure 1 où un arc entre deux classes exprime la relation *strictement plus grande* et les pointillés l'*incomparabilité*.

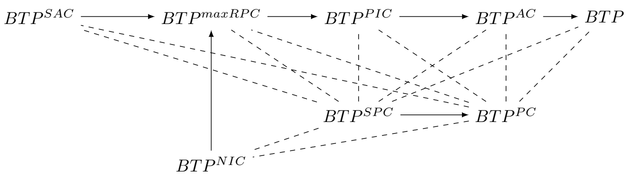

FIGURE 1 – Relations entre classes cachées de BTP.

4 Conclusion

Souvent ignorée des praticiens pour de bonnes raisons, l'étude des classes polynomiales s'est cantonnée à des travaux théoriques. La notion de classe cachée introduite ici pourrait lui donner un impact plus large au sein de la communauté de la Programmation par Contraintes. D'une part, elle permettrait d'offrir une explication à l'efficacité parfois surprenante des solveurs généraux qui se trouve en contradiction avec le verdict délivré par l'analyse de la complexité dans le pire des cas. D'autre part, elle pourrait orienter les recherches théoriques en les dirigeant vers l'élaboration de classes polynomiales plus en adéquation avec les algorithmes de résolution souvent à base de filtrages.

Références

[1] M. Cooper, P. Jeavons, and A. Salamon. Generalizing constraint satisfaction on trees : hybrid tractability and variable elimination. *Artificial Intelligence*, 174 :570–584, 2010.

[2] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. Hidden tractable classes : From theory to practice. In *Proceedings of the 26th IEEE ICTAI*, pages 437–445, 2014.