

Comprendre le Potentiel des Propagateurs

Sascha Van Cauwelaert¹

Michele Lombardi²

Pierre Schaus¹

¹ INGI, ICTEAM, Université Catholique de Louvain, Belgique

² DEIS, Université de Bologne, Italie

{sascha.vancauwelaert,pierre.schaus}@uclouvain.be michele.lombardi2@unibo.it

Abstract

Solvers performances can be improved through the cost reduction of propagators. The impact of this reduction has however an uncertain outcome. We propose a simple approach to estimate it in a realistic manner, before starting the actual research endeavor.

Résumé

Un axe de recherche important de notre communauté est la réduction du coût des propagateurs (algorithmes de filtrage de domaines), dans le but d'améliorer les performances des solveurs de contraintes. L'impact de cette réduction reste cependant incertaine. Nous proposons une approche simple permettant d'estimer celui-ci de façon réaliste, avant même de débiter le travail de recherche. Cet article résume le travail présenté dans [6].

Introduction La propagation est une caractéristique spécifique à la programmation par contraintes : elle permet à un solveur de contraintes de réduire l'espace de recherche, apportant parfois un important gain de performance. En pratique, la clé du succès réside dans un bon équilibre entre le temps sauvé en filtrant des valeurs et le temps passé à exécuter les propagateurs. Une quantité non-négligeable de travail de recherche vise à améliorer ce compromis.

Nous considérons le cas particulier pour lequel le but est d'optimiser la performance d'une technique de propagation donnée, sans changer son comportement du point de vue entrée-sortie. 2 façons d'obtenir cet effet sont 1) élaborer des algorithmes plus efficace fournissant le même filtrage ; 2) restreindre l'activation du propagateur en question à l'aide d'une condition nécessaire afin de réduire le nombre d'activations dénuées de filtrage. Pour 1) et 2), obtenir des améliorations est un long processus avec des résultats incertains. Dès lors, posséder des outils permettant de sonder le potentiel de techniques de propagation et d'évaluer l'impact d'améliorations spécifiques est d'un grand intérêt.

Le temps et le nombre d'échecs sont régulièrement utilisés pour comparer l'utilisation d'un propagateur avec

une approche de base. Les 2 approches sont testées sur un ensemble d'instances résolues de manière complète (avec le risque de biaiser l'analyse vers des instances de tailles plus modestes). Aussi, des heuristiques de recherche statiques sont généralement choisies afin de réaliser des comparaisons équitables et rigoureuses. Ceci risque de réduire la portée de l'analyse car des heuristiques dynamiques sont généralement préférées en pratique. Nous proposons d'étendre ce type d'évaluation en : 1) instrumentant le solveur afin de collecter des informations au sujet de la contrainte ; 2) enregistrant et *rejouant* des arbres de recherches afin de permettre des comparaisons équitables avec des heuristiques et des tailles d'instances arbitraires. 2 cas d'étude sont ici considérés : le Raisonnement Énergétique [3, 1] et le Raisonnement en Cardinalité pour l'Empaquetage Revisité [5].

Approche Proposée Formellement, nous désirons évaluer une fonction filtrante ϕ qui lie un ensemble de domaines D_0, \dots, D_{n-1} à un second ensemble de domaines D'_0, \dots, D'_{n-1} tels que $D'_i \subseteq D_i$. Plus spécifiquement, nous voulons estimer le potentiel de deux pistes d'amélioration : 1) améliorer l'efficacité de l'implantation actuelle et 2) restreindre l'activation de ϕ à l'aide d'une condition nécessaire.

Mesure de Performance : Afin d'évaluer les performances de ϕ , nous comparons le temps pour résoudre un problème de satisfaction de contraintes avec et sans ϕ , dénotés respectivement par $M \cup \phi$ et M . Afin de s'assurer que 1) les 2 exécutions explorent le même espace de recherche et que 2) tous les nœuds de recherche sont visités dans le même ordre, nous proposons de « rejouer » l'arbre de recherche.

Technique de Rejeu : Le processus de recherche peut être vu comme l'évaluation de la fonction récursive $traverse(b, M)$ prenant en paramètre la stratégie de recherche b et le problème cible M . Soit $enregistre(b)$ et $rejoue(b)$ 2 stratégies de recherche d'emballage mémorisant et ré-ajoutant les contraintes retournées par b , respectivement. Pour évaluer ϕ , nous exécutons séquentiellement 1) $traverse(enregistre(b), M)$, 2)

$traverse(rejoue(b), M)$ et 3) $traverse(rejoue(b), M \cup \phi)$, et nous comparons les résultats de 2) et 3). Ceci permet 1) d'aborder des instances de tailles arbitraires et 2) d'utiliser n'importe quelle stratégie de recherche.

Évaluer le Potentiel du Propagateur : Soit t_ϕ^+ et t_ϕ^- les temps d'exécution totaux de ϕ pour lesquels les activations ont respectivement conduit à de l'élagage de domaine ou non. Cette information est mesurée en introduisant une fonction d'emballage $stats(\phi)$ qui vérifie la taille des domaines avant et après l'exécution de ϕ . Si $t(b, M)$ est le temps requis pour exécuter $traverse(b, M)$, nous pouvons estimer l'impact de la réduction du temps d'exécution de ϕ par un facteur $\mu \in [0, 1]$ en calculant :

$$t(rejoue(b), M \cup stats(\phi)) - \mu \cdot (t_\phi^+ + t_\phi^-)$$

De même, l'impact de la restriction de l'exécution de ϕ à l'aide d'une condition nécessaire qui empêche une fraction $\mu \in [0, 1]$ des activations inutiles peut être estimé en calculant :

$$t(rejoue(b), M \cup stats(\phi)) - \mu \cdot (t_\phi^-)$$

Cette simple approche linéaire nous permet de comparer des implantations réelles de ϕ avec des homologues fictifs, fournissant ainsi une chance d'explorer quelles valeurs de μ seraient nécessaires pour vaincre l'approche de base M , et donc une meilleure compréhension de l'effort requis pour arriver à une telle fin. Une plus grande sagacité peut être obtenue en opérant cette comparaison sur une panoplie de tests reconnue. Pour ce faire, nous nous reposons sur des *profils de performance* [2]. Ceux-ci sont des fonctions de distribution cumulatives $F(\tau)$ d'une métrique de performance donnée τ (dans notre cas le rapport entre le temps de résolution requis par une approche cible et celui de l'approche de base). En supposant la panoplie de tests assez représentative, $F(\tau)$ peut être interprétée comme une probabilité. Soit ϕ_0, ϕ_1, \dots l'ensemble des implantations de ϕ considérées, et \mathcal{M} l'ensemble des instances de test. Le profil de performance de ϕ_i est donné par :

$$F_{\phi_i}(\tau) = \frac{1}{|\mathcal{M}|} \left| \left\{ M \in \mathcal{M} : \frac{t(rejoue(b), M \cup \phi_i)}{t(rejoue(b), M)} \leq \tau \right\} \right|$$

Pour une description sur la façon d'interpréter ces profils de performance, nous renvoyons le lecteur à l'article [6] que le présent document résume.

Expérimentations Nous avons appliqué notre approche à deux propagateurs, à savoir le *Raisonnement Énergétique* [3, 1] et le *Raisonnement en Cardinalité pour l'Empaquetage Revisité* [5]. Nous avons également considéré les classes de propagateurs fictifs suivantes : ϕ_μ^{cost} , i.e. une implantation pour laquelle le temps est réduit par un facteur μ ; $\phi_{O(f(n))}^{cost}$, i.e. une implantation pour laquelle la complexité temporelle est $O(f(n))$; ϕ_p^{oracle} , i.e. une implantation qui restreint l'exécution de ϕ à l'aide d'une condition nécessaire provoquant des activations inutiles avec une probabilité p .

Nous avons utilisé le solveur *Oscar* [4]. La configuration expérimentale, les résultats et leur analyse sont

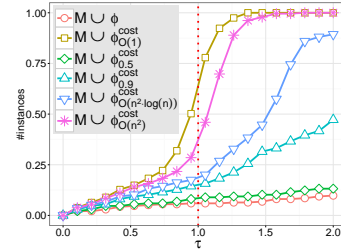


FIGURE 1 – Exemple de profil de performance pour le Raisonnement Énergétique.

donnés dans [6]. Un exemple de profil de performance est donné dans la figure 1.

Conclusion Évaluer les avantages potentiels fournis par la réduction du coût d'une procédure de filtrage est d'un grand intérêt afin de rendre nos efforts de recherche aussi fructueux que possible. De plus, mesurer exactement le gain obtenu par un algorithme de filtrage permet de réduire le biais des évaluations empiriques. Nous avons ici proposé une première étape dans cette direction, à savoir une méthodologie systématique permettant de simuler les performances d'implantations fictives de propagateurs avec un coût réduit.

Références

- [1] Philippe Baptiste, Claude Le Pape, and Wim Nuijten. *Constraint-based scheduling : applying constraint programming to scheduling problems*, volume 39. Springer, 2001.
- [2] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2) :201–213, 2002.
- [3] Jacques Erschler and Pierre Lopez. Energy-based approach for task scheduling under time and resources constraints. In *2nd international workshop on project management and scheduling*, pages 115–121, 1990.
- [4] Oscar Team. *Oscar : Scala in OR*, 2012. Available from <https://bitbucket.org/oscarlib/oscar>.
- [5] François Pelsser, Pierre Schaus, and Jean-Charles Régin. Revisiting the cardinality reasoning for bin-packing constraint. In *Principles and Practice of Constraint Programming*, pages 578–586. Springer, 2013.
- [6] Sascha Van Cauwelaert, Michele Lombardi, and Pierre Schaus. Understanding the potential of propagators. In *Integration of Artificial Intelligence (AI) and Operations Research (OR) techniques in Constraint Programming (CPAIOR 2015)*, June 2015.