

Heuristiques boîte-noire de sélection de valeur pour les problèmes d'optimisation en programmation par contraintes

Augustin Delecluse^{1,2}, Pierre Schaus¹

¹ ICTEAM, UCLouvain, Belgique

² TRAIL, Belgique

{augustin.delecluse,pierre.schaus}@uclouvain.be

5 mai 2025

Résumé

La Sélection de Valeur Impactant les Bornes est une heuristique de sélection de valeur efficace mais coûteuse, qui estime l'impact de chaque valeur d'une variable sur l'objectif via un calcul de point fixe. Nous introduisons deux améliorations à son fonctionnement : (1) restreindre le calcul du point fixe aux contraintes situées sur le plus court chemin vers l'objectif, et (2) un regard-avant inversé fixant de manière optimiste la variable objectif à sa meilleure borne. Validées sur des instances académiques et lors de la compétition XCSP³, ces approches réduisent sensiblement les coûts de calcul.

Mots-clés

Programmation par contraintes, sélection de valeur, optimisation

Abstract

Bound-Impact Value Selection is an effective but costly value selection heuristic that estimates the impact of each possible value on the objective through a fixpoint computation. We introduce two improvements to this approach : (1) restricting the fixpoint computation to constraints on the shortest path to the objective, and (2) a reversed look-ahead that optimistically sets the objective variable to its best bound. Validated on academic instances and in the XCSP³ competition, these approaches significantly reduce computation costs.

Keywords

Constraint programming, value selection, optimization

Cet article est un résumé de [7]

1 Introduction

Le paradigme de la Programmation par Contraintes (PPC) permet de résoudre des problèmes combinatoires de façon déclarative. La vision « Graal » de la PPC postule que l'utilisateur se contente de spécifier le problème, et l'ordinateur le résout [4]. Toutefois, afin de garantir une efficacité pratique, cette vision a été nuancée par le mantra historique de la PPC : *PPC = Modélisation + Recherche*. La capa-

cité à programmer des recherches spécifiques demeure cruciale pour réduire la taille de l'arbre de recherche considéré. Au fil du temps, l'intérêt porté à la programmation de la recherche a toutefois diminué grâce à l'apparition de méthodes génériques efficaces [6, 3, 9, 10].

Une procédure de recherche typique effectuée, à chaque nœud, deux choix successifs : la sélection d'une variable non encore assignée, puis le choix d'une valeur que l'on affecte à cette variable dans la branche de gauche. Malgré les nombreuses études consacrées à la conception d'heuristiques de sélection de variables basées sur le principe du *premier-échec*, peu de stratégies "boîte noire" ont été développées pour la sélection de valeur. Dans le contexte de l'optimisation, une méthode générique simple et efficace est la *Sélection de Valeur Impactant les Bornes* (SVIB) [8]. Cette heuristique de *regard-avant* évalue chaque valeur dans le domaine d'une variable : elle assigne successivement la variable à chacune de ces valeurs et calcule le point fixe induit par les contraintes. On mesure alors son impact sur l'objectif — en particulier, l'augmentation de la borne inférieure pour un problème de minimisation. La valeur entraînant l'augmentation la plus faible est sélectionnée pour la branche de gauche.¹

Bien que SVIB propose des solutions de meilleure qualité que d'autres heuristiques plus naïves (par exemple le choix systématique de la valeur minimale du domaine), le gain en guidage de la recherche peut être coûteux à obtenir. Cela est particulièrement vrai lorsque les domaines sont de grande taille ou lorsque le nombre de contraintes impliquées dans le point fixe est élevé. Une approche permettant de diminuer ce surcoût consiste à n'examiner que les valeurs extrêmes (minimum et maximum) du domaine, surtout si ce dernier est trop grand.

2 Contributions et conclusion

Notre travail introduit deux pistes pour réduire le coût de SVIB. La première, appelée *Point Fixe Restreint* (PFR), consiste à calculer le point fixe uniquement sur un ensemble restreint de contraintes. Plus précisément, on ne considère

¹. Cette idée est proche du *branchement fort* utilisé dans les solveurs de Programmation Linéaire en Nombres Mixtes [1].

que les contraintes situées sur les plus courts chemins, dans le graphe de contraintes, qui relie la variable choisie à la variable objectif (chaque contrainte étant reliée aux variables de son domaine). Cette étape allégée n'est utilisée que dans l'heuristique de choix de valeur afin de sélectionner une valeur intéressante; la recherche complète prend, quant à elle, en compte l'ensemble des contraintes. Par exemple, la Figure 1 illustre ce principe pour un Problème du Voyageur de Commerce (PVC) avec un modèle de circuit en PPC. Les variables S_i ($i \in 0..n$), avec $n = 4$, représentent le successeur de chaque nœud dans le tour, D_i la distance entre un nœud et son successeur, et $DTot$ la distance totale, qui constitue l'objectif à minimiser.

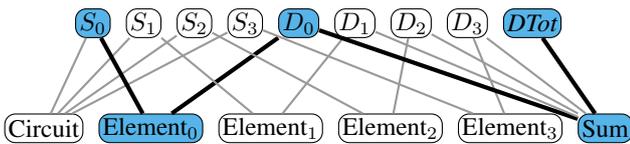


FIGURE 1 – Contraintes et variables pour un PVC à 4 nœuds. Les éléments en bleu indiquent le plus court chemin reliant la variable S_0 à l'objectif $DTot$. Seules les deux contraintes mises en évidence sont propagées lors de l'examen des valeurs de S_0 dans le cadre de PFR.

La seconde amélioration, nommée *Regard-Avant Inverse* (RAI), part de la valeur de l'objectif. Au lieu de fixer successivement la variable sélectionnée à chacune de ses valeurs (comme dans SVIB), on fixe de manière optimiste la variable objectif à sa borne inférieure (dans le cas d'une minimisation), puis on calcule le point fixe. La valeur effectivement sélectionnée pour la variable provient alors des valeurs qui subsistent après le filtrage des domaines. Si une inconsistance est détectée, on augmente la borne de l'objectif, et l'on recommence jusqu'à ce que plus aucune variable ne soit vidée de son domaine.

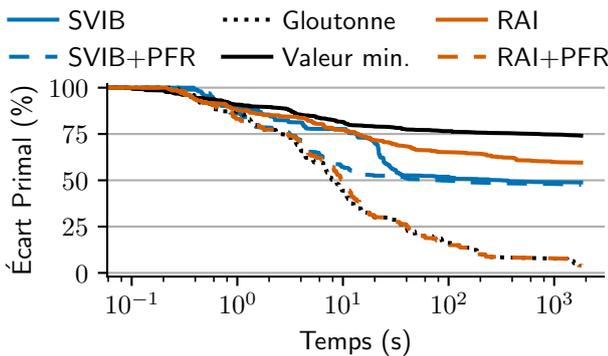


FIGURE 2 – Écart primal [5] des heuristiques au cours du temps, sur le PVC. Un écart de 100% indique qu'aucune solution n'est encore trouvée, et un écart de 0% que la meilleure solution est trouvée. L'approche RAI+PFR a le même comportement qu'une recherche gloutonne.

L'article principal [7] détaille le comportement de ces heuristiques et propose des exemples guidés. Les expérimenta-

tions, menées tant sur des problèmes académiques que sur 18 problèmes de la compétition XCSP³ 2023 [2], montrent que l'ajout de PFR dans la procédure de *regard-avant* et l'utilisation de RAI réduisent sensiblement les coûts de calcul, rendant moins pertinentes les restrictions antérieures imposées à SVIB. En particulier, elles ont le même comportement qu'une sélection gloutonne sur le PVC, comme indiqué dans la Figure 2.

Références

- [1] David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. Finding cuts in the tsp (a preliminary report), 1995.
- [2] Gilles Audemard, Christophe Lecoutre, and Emmanuel Lonca. Proceedings of the 2023 xcsp3 competition. *arXiv preprint arXiv :2312.05877*, 2023.
- [3] Gilles Audemard, Christophe Lecoutre, and Charles Prud'Homme. Guiding backtrack search by tracking variables during constraint propagation. In *International Conference on Principles and Practice of Constraint Programming*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.
- [4] Roman Barták. Constraint programming : In pursuit of the holy grail. In *Proceedings of the Week of Doctoral Students (WDS99)*, volume 4, pages 555–564. MatFyzPress Prague, 1999.
- [5] Timo Berthold. Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6) :611–614, 2013.
- [6] Frédéric Boussemart, Fred Hemery, Christophe Lecoutre, and Lakhdar Sais. Boosting systematic search by weighting constraints. In *ECAI*, volume 16, page 146, 2004.
- [7] Augustin Delecluse and Pierre Schaus. Black-box value heuristics for solving optimization problems with constraint programming (short paper). In *30th International Conference on Principles and Practice of Constraint Programming (CP 2024)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024.
- [8] Jean-Guillaume Fages and Charles Prud'Homme. Making the first solution good! In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1073–1077. IEEE, 2017.
- [9] Steven Gay, Renaud Hartert, Christophe Lecoutre, and Pierre Schaus. Conflict ordering search for scheduling problems. In *Principles and Practice of Constraint Programming : 21st International Conference, CP 2015, Cork, Ireland, August 31–September 4, 2015, Proceedings 21*, pages 140–148. Springer, 2015.
- [10] Christophe Lecoutre, Lakhdar Saïs, Sébastien Tabary, and Vincent Vidal. Reasoning from last conflict (s) in constraint programming. *Artificial Intelligence*, 173(18) :1592–1614, 2009.