Exploiting Functional Constraints in Automatic Dominance Breaking for Constraint Optimization

Extended Abstract

Jimmy H. M. Lee
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Allen Z. Zhong
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Dominance relations [3, 7] in Constraint Optimization Problems (COPs) describe relations between two full assignments where one is known to be subordinate compared with another concerning satisfiability and/or objective value. *Dominance breaking*, which adds additional constraints to remove dominated full assignments, is known to be effective in a range of problems [1, 6, 9, 14] but also demands sophisticated insights into specific problem structures. Chu and Stuckey [3, 4] presents a generic method for deriving dominance breaking constraints for COPs, in which manual effort are required to select mappings, identify sufficient conditions, and simplify the constraints. Mears and de la Banda [12] automate the derivation process to a certain extent based on automated symmetry detection, but it still requires manual selection of symmetries to be effective.

Lee and Zhong [11, 10] introduce *automatic dominance breaking* for a class of COPs. The idea is to construct auxiliary *generation CSPs* for a target COP P and derive dominance breaking constraints automatically from the solutions of generation CSPs. Figure 1 presents the overall workflow of the method as follows:

1. Analyze the objective and constraints of P and construct generation CSPs.
2. Enumerate all solutions of the generation CSPs using a constraint solver.
3. Generate one dominance breaking constraint for each solution of the CSPs.
4. Add all generated constraints to the COP P.
5. Solve the COP with extra constraints by a constraint solver.

Different from other approaches that require manual interventions, the method generates constraints in the form of nogoods [8]. The generation CSPs solve for pairs (θ, θ') of partial assignments of P, and constraints in the CSPs guarantee that all full assignments extending from θ' are dominated by some full assignments of P. Therefore, the negation of θ', which is a *dominance breaking nogood* $\neg\theta'$, can be added to P to reduce the search space.

The key step of automatic dominance breaking is to build generation CSPs *automatically* based on the COP P. Lee and Zhong [11, 10] identify a class of *efficiently checkable* objectives and constraints, based on which constraints in generation CSPs are given. Yet, the method is restrictive to COPs with only efficiently checkable objectives and constraints. For example, in order to apply automatic dominance breaking to a COP, the objective is required to be either a separable function or a submodular function [11]. This prevents the use of automatic dominance breaking for COPs with varying objectives and constraints, especially the ones with nested function calls.

▶ **Example 1.** Consider a simple COP that minimizes $\max(z_1, z_2) + 4z_3$ subject to the constraint $2z_1 - 3z_2 * z_3 \leq 5$, where $z_1, z_2, z_3 \in \{1, 2, 3\}$. The objective with the max function and the constraint with the multiplication are not efficiently checkable, and therefore the method of automatic dominance breaking cannot be applied to the COP.

Functional expressions, such as $\max(z_1, z_2)$ and $z_2 * z_3$, are ubiquitous in problem modelling, but objectives and constraints with functional expressions are usually not efficiently checkable.

28th International Conference on Principles and Practice of Constraint Programming (CP 2022).
Editor: Christine Solnon; Article No. 6; pp. 6:1–6:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

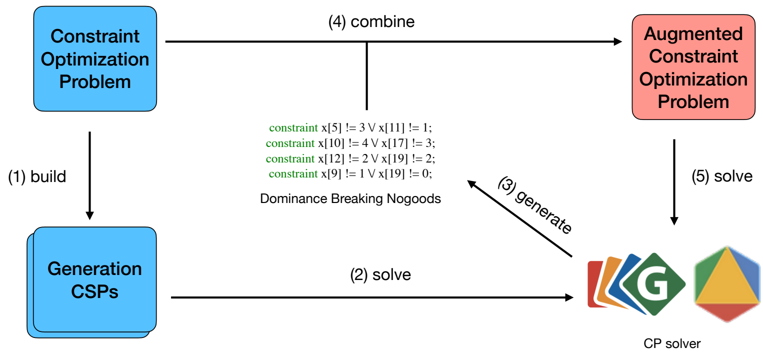

Figure 1 Workflow of Automatic Dominance Breaking for a Class of COPs

We observe that COPs specified in a high-level modeling language [5, 13] are usually normalized/flattened into a form with only standard constraints.

▶ **Example 2.** After normalization, the COP in Example 1 can become:

$$
\begin{aligned}
&\text{minimize} && \mathit{obj} \\
&\text{subject to} && \mathit{obj} = y_1 + 4z_3, y_1 = \max(z_1, z_2), \\
& && y_2 \leq 5, y_2 = 2z_1 - 3y_3, y_3 = z_2 * z_3, \\
& && z_1, z_2, z_3 \in \{1, 2, 3\}, y_1, y_2, y_3, \mathit{obj} \in \mathbb{Z}
\end{aligned}
$$

Note that y_1, y_2, y_3 and obj are newly introduced variables, and are defined by *functional constraints* $y_1 = \max(z_1, z_2)$, $y_2 = 2z_1 - 3y_3$, $y_3 = z_2 * z_3$ and $\mathit{obj} = y_1 + 4z_3$ respectively.

In this paper, we propose to exploit standard functional constraints and their properties, such as monotonicity, commutativity and associativity, to identify dominance relations in COPs with various functional expressions. The number of standard constraints is limited. Once we analyze all standard constraints and annotate their properties, we can construct generation CSPs for all possible COPs that can be modelled in a modeling language. In particular, our main contributions are as follows:

- The theory of dominance is generalized to normalized COPs which contain functionally defined variables and functional constraints.
- Utilizing the concept of terms [2], an algorithm is presented for automatic derivation of constraints in generation CSPs based on functional constraints and their properties.
- Theorems are given to characterize important properties such as termination and soundness of the derivation algorithm.
- The proposed method is implemented on top of the MiniZinc compiler [13], and experimentation on various benchmarks confirms that the effectiveness of generated nogoods in pruning the search space and reducing the solving time for problems with ineffective or no known dominance breaking constraints in the literature.
- Case studies are given to demonstrate how to discover compact dominance (symmetry) breaking constraints by studying the nogood patterns of small instances.

Overall, our work gives a theory-backed method to generalize the method of automatic dominance breaking and enable systematic detection of dominance relations in a larger class of COPs with more sophisticated structures.

References

1 Tariq Aldowaisan. A new heuristic and dominance relations for no-wait flowshops with setups. *Computers & Operations Research*, 28(6):563–584, 2001.

2 Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

3 Geoffrey Chu and Peter J Stuckey. A generic method for identifying and exploiting dominance relations. In *International Conference on Principles and Practice of Constraint Programming*, pages 6–22. Springer, 2012.

4 Geoffrey Chu and Peter J Stuckey. Dominance driven search. In *International Conference on Principles and Practice of Constraint Programming*, pages 217–229. Springer, 2013.

5 Alan M Frisch, Warwick Harvey, Chris Jefferson, Bernadette Martinez-Hernandez, and Ian Miguel. Essence: A constraint language for specifying combinatorial problems. *Constraints*, 13(3):268–306, 2008.

6 Lise Getoor, Greger Ottosson, Markus Fromherz, and Björn Carlson. Effective redundant constraints for online scheduling. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*, pages 302–307, 1997.

7 Toshihide Ibaraki. The power of dominance relations in branch-and-bound algorithms. *Journal of the ACM (JACM)*, 24(2):264–279, 1977.

8 George Katsirelos and Fahiem Bacchus. Generalized nogoods in csps. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 1, pages 390–396, 2005.

9 Richard E Korf. Optimal rectangle packing: New results. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 142–149, 2004.

10 Jimmy H. M. Lee and Allen Z. Zhong. Towards more practical and efficient automatic dominance breaking. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, pages 3868–3876, 2021.

11 Jimmy H.M. Lee and Allen Z. Zhong. Automatic generation of dominance breaking nogoods for a class of constraint optimization problems. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 1192–1200, 2020.

12 Christopher Mears and Maria Garcia De La Banda. Towards automatic dominance breaking for constraint optimization problems. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 360–366, 2015.

13 Nicholas Nethercote, Peter J Stuckey, Ralph Becket, Sebastian Brand, Gregory J Duck, and Guido Tack. MiniZinc: towards a standard CP modelling language. In *International Conference on Principles and Practice of Constraint Programming*, pages 529–543. Springer, 2007.

14 Steven Prestwich and J Christopher Beck. Exploiting dominance in three symmetric problems. In *Fourth International Workshop on Symmetry and Constraint Satisfaction Problems*, pages 63–70, 2004.