Efficient Structural Symmetry Breaking for Constraint Satisfaction Problems

Pierre Flener[1]*, Justin Pearson[1], Meinolf Sellmann[2], Pascal Van Hentenryck[2], and Magnus Ågren[1] **

[1] Department of Information Technology
Uppsala University, Box 337, SE – 751 05 Uppsala, Sweden
{pierref,justin,agren}@it.uu.se

[2] Department of Computer Science
Brown University, Box 1910, Providence, RI 02912, USA
{sello,pvh}@cs.brown.edu

Abstract. Symmetry breaking for constraint satisfaction problems (CSPs) has attracted considerable attention in recent years. Various general schemes have been proposed to eliminate symmetries. In general, these schemes may take exponential space or time to eliminate all the symmetries. We identify several classes of CSPs that encompass many practical problems and for which symmetry breaking for various forms of value and variable interchangeability is tractable using dedicated search procedures or symmetry-breaking constraints that allow nogoods and their symmetrically equivalent solutions to be stored and checked efficiently.

1 Introduction

Many constraint satisfaction problems (CSPs) naturally exhibit symmetries. Symmetry breaking may drastically improve performance (e.g., [2, 13, 15, 23]). An important contribution in this area has been the development of various general schemes for symmetry breaking *during* search in CSPs (e.g., SBDS [1, 12] and SBDD [7, 10, 15]). Unfortunately, in general, these schemes may require exponential resources to break all the symmetries. Indeed, some schemes may require exponential space to store all the nogoods generated through symmetries, while others may take exponential time to discover whether a partial assignment is symmetric to one of the existing nogoods. As a consequence, practical applications often place limits on how many nogoods can be stored and/or which symmetries to break. Other than eliminating symmetries by re-modelling the problem (e.g., [22]), another important approach is to break symmetries by adding constraints *before* search starts (e.g., [5, 14]). Unfortunately, in general, a super-exponential number of constraints may be needed to break all the symmetries. For instance, the lex-leader scheme of [5] adds one constraint per symmetry, but the number of symmetries is often super-exponential (an $m \times n$

* Currently on leave of absence at Sabancı University in İstanbul, Turkey.
** The authors' names are ordered according to the Swedish alphabet.

matrix with fully interchangeable rows and columns has $m! \cdot n!$ variable symmetries). As a consequence, practical applications often add only some of these symmetry-breaking constraints (e.g., [8, 21]).

We approach symmetry breaking from a different, orthogonal standpoint. *Our goal is to identify classes of CSPs that are practically relevant and for which symmetry breaking is tractable, that is polynomial in time and space, using dedicated search procedures or new symmetry-breaking constraints.* We identified several such classes whose CSPs feature various forms of value and variable interchangeability and encompass many practical problems [24, 20]. For some of them, symmetry breaking can even be performed during search with a *constant* overhead [24] with respect to both time and space at every node explored. In [20], we introduced the name *structural symmetry breaking* for this approach, which allows not only the efficient representation of symmetric nogoods but also the derivation of new, efficient static symmetry-breaking constraints, as shown in [9]. We believe that these notions are helpful to derive many other classes of tractable symmetries. In this short paper, we summarise some of our on-going work started in [24, 25, 20, 9].

2 Results

We first fix some definitions and notation. A *constraint satisfaction problem* (*CSP*) is a triplet $\langle V, D, C\rangle$, where V denotes the set of variables, D denotes the set of possible values for these variables and is called their *domain*, and $C : (V \to D) \to Bool$ is a constraint that specifies which assignments of domain values to the variables are solutions. An *assignment* for a CSP $\mathcal{P} = \langle V, D, C\rangle$ is a function $\alpha : V \to D$. A *partial assignment* for a CSP $\mathcal{P} = \langle V, D, C\rangle$ is a partial function $\alpha : V \to D$ whose scope is denoted by $scope(\alpha)$. If the domain D is the power-set of some other set, we say that the CSP is a *set-CSP*. A *solution* to a CSP $\mathcal{P} = \langle V, D, C\rangle$ is an assignment σ for $\mathcal{P}$ such that $C(\sigma) = \mathbf{true}$.

2.1 Value Interchangeability

We say that a CSP $\mathcal{P} = \langle V, D, C\rangle$ is *fully value interchangeable* if whenever α is a solution and $\tau : D \to D$ is a bijection, then $\tau \circ \alpha$ is also a solution.

Symmetry definitions are analysed in some detail in [4]. There, a symmetry acts on variable/value pairs within partial assignments. An important distinction is made between symmetries of the CSP and symmetries on the solution set. In this paper, we do not require the full machinery of [4], as we are only interested in breaking certain symmetries of CSPs. In each case, we define the action of a symmetry in terms of its action on solutions. How such symmetry is discovered is left as an open issue, although some preliminary work has been done in [3, 6, 11, 16, 18, 25] for instance.

There are two results in our [24] that are interesting in light of further developments:

– A compact representation of nogoods, the so-called *abstract nogoods*, allows the efficient checking of nogoods using dominance detection.
– A specialisation of search that results in a search procedure that breaks value symmetry in polynomial time (and even in constant time in the considered case of full value interchangeability), *without* the use of dominance detection. This work is further developed in [19], which gives a procedure that breaks *any* value symmetry in polynomial time.

More complicated classes of CSPs with value symmetry were also studied in our [24], namely the so-called *piecewise value interchangeable CSPs* (defined below) and *wreath value interchangeable CSPs*, and again polynomial-time checkable nogoods and search procedures were developed.

2.2 Value and Variable Interchangeability

The key insight in our [24] was as follows: If a compact representation is given of nogoods and their symmetrically equivalent nogoods, then polynomial-time dominance detection can be obtained.

In our [20], a new class of CSPs with symmetries, involving *both* variable *and* value symmetries, was investigated. Given a partition of a set $E = \sum_k E_k$ (where $\sum$ denotes the disjoint union), we say that a permutation π of E is a *piecewise permutation* with respect to the partition $\sum_k E_k$ if for all k and for all $e \in E_k$, we have $\pi(e) \in E_k$. We then say that a CSP $\langle \sum_k V_k, \sum_\ell D_\ell, C \rangle$ is *piecewise variable and value interchangeable* if whenever α is a solution, σ is a piecewise variable permutation, and τ is a piecewise value permutation, then $\tau \circ \alpha \circ \sigma$ is also a solution.

In our [20], an efficient polynomial-time dominance detection algorithm for piecewise variable and value interchangeable CSPs is given in terms of matchings. This is further developed in our [9] in terms of static symmetry-breaking constraints that allow symmetry breaking with a polynomial number of constraints for piecewise variable and value interchangeable CSPs. Indeed, symmetries within a variable partition $V = \sum_k V_k$ can be broken by simply ordering the variables

$$v_p \leq \cdots \leq v_q$$

within each variable component $V_k = \{v_p, \ldots, v_q\}$, while value symmetry can be broken by exploiting so-called value signatures. Denote by

$$f_h^k = |\{v \in V_k \mid v \in \mathit{scope}(\alpha) \land \alpha(v) = d_h\}|$$

the frequency under which value d_h appears in variable component V_k. Then the *signature* of a value d_h is the tuple $(f_h^1, \ldots, f_h^a)$ of the frequencies under which d_h appears in each of the a variable components. Note that these frequencies can be efficiently calculated by a constraint solver using the global cardinality constraint. Then, for each value partition $D_i = \{d_p, \ldots, d_q\}$, the following chain of lexicographic ordering constraints

$$(f_p^1, \ldots, f_p^a) \geq_{\text{lex}} \cdots \geq_{\text{lex}} (f_q^1, \ldots, f_q^a)$$

is used. These sets of constraints together break the piecewise value and variable symmetry.

3 Conclusions and Further Work

In this short summary, we have traced some of our research themes begun in [24, 25, 20, 9]. The key idea is that when suitable abstractions of classes of symmetrical nogoods are found, symmetry can be broken in polynomial time and often more efficiently than by general methods such as those in [17, 19]. The interesting question of investigating the limits of tractability in symmetry breaking was begun to be addressed in our [20], where certain classes of symmetric set-CSPs are shown to have NP-complete dominance detection problems.

Further research in this area includes finding more general abstractions to break symmetries for more complex classes of groups and a further understanding of the bounds of tractability in symmetry breaking.

Acknowledgements

The authors are partly supported by grant IG2001-67 of STINT, the Swedish Foundation for International Cooperation in Research and Higher Education.

References

1. R. Backofen and S. Will. Excluding symmetries in constraint-based search. In J. Jaffar, editor, *Proceedings of CP'99*, volume 1713 of *LNCS*, pages 73–87. Springer-Verlag, 1999.
2. N. Barnier and P. Brisset. Solving the Kirkman's schoolgirl problem in a few seconds. In P. Van Hentenryck, editor, *Proceedings of CP'02*, volume 2470 of *LNCS*, pages 477–491. Springer-Verlag, 2002.
3. M. Cadoli and T. Mancini. Detecting and breaking symmetries on specifications. In B. M. Smith, I. P. Gent, and W. Harvey, editors, *Proceedings of SymCon'03*, 2003. Available at `http://4c.ucc.ie/~bms/SymCon03/`.
4. D. Cohen, P. Jeavons, C. Jefferson, K. E. Petrie, and B. M. Smith. Symmetry definitions for constraint satisfaction problems. In P. van Beek, editor, *Proceedings of CP'05*, volume 3709 of *LNCS*, pages 17–31. Springer-Verlag, 2005.
5. J. M. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry-breaking predicates for search problems. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Proceedings of KR'96*, pages 148–159. Morgan Kaufmann, 1996.
6. M. Eriksson. Detecting symmetries in relational models of CSPs. Master's thesis, Computing Science, Department of Information Technology, Uppsala University, Sweden, November 2005. Available as Technical Report 2005-034 at `http://www.it.uu.se/research/reports/2005-034/`.
7. T. Fahle, S. Schamberger, and M. Sellmann. Symmetry breaking. In T. Walsh, editor, *Proceedings of CP'01*, volume 2239 of *LNCS*, pages 93–107. Springer-Verlag, 2001.

8. P. Flener, A. M. Frisch, B. Hnich, Z. Kızıltan, I. Miguel, J. Pearson, and T. Walsh. Breaking row and column symmetries in matrix models. In P. Van Hentenryck, editor, *Proceedings of CP'02*, volume 2470 of *LNCS*, pages 462–476. Springer-Verlag, 2002.
9. P. Flener, J. Pearson, M. Sellmann, and P. Van Hentenryck. Static and dynamic structural symmetry breaking. In F. Benhamou, editor, *Proceedings of CP'06*, LNCS, pages 695–699. Springer-Verlag, 2006.
10. F. Focacci and M. Milano. Global cut framework for removing symmetries. In T. Walsh, editor, *Proceedings of CP'01*, volume 2239 of *LNCS*, pages 77–92. Springer-Verlag, 2001.
11. E. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings of AAAI'91*, pages 227–233, 1991.
12. I. P. Gent and B. M. Smith. Symmetry breaking during search in constraint programming. In *Proceedings of ECAI'00*, pages 599–603, 2000.
13. P. Meseguer and C. Torras. Exploiting symmetries within constraint satisfaction search. *Artificial Intelligence*, 129(1-2):133–163, 2001.
14. J.-F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. In J. Komorowski and Z. Raś, editors, *Proceedings of ISMIS'93*, volume 689 of *LNAI*, pages 350–361. Springer-Verlag, 1993.
15. J.-F. Puget. Symmetry breaking revisited. In P. Van Hentenryck, editor, *Proceedings of CP'02*, volume 2470 of *LNCS*, pages 446–461. Springer-Verlag, 2002.
16. J.-F. Puget. Automatic detection of variable and value symmetries. In P. van Beek, editor, *Proceedings of CP'05*, volume 3709 of *LNCS*, pages 475–489. Springer-Verlag, 2005.
17. J.-F. Puget. An efficient way of breaking value symmetries. In *Proceedings of AAAI'06*. AAAI Press, 2006.
18. A. Ramani and I. L. Markov. Automatically exploiting symmetries in constraint programming. In B. Faltings, A. Petcu, and F. Fages, editors, *Recent Advances in Constraints*, volume 3419 of *LNAI*, pages 98–112. Springer-Verlag, 2005.
19. C. M. Roney-Dougal, I. P. Gent, T. Kelsey, and S. Linton. Tractable symmetry breaking using restricted search trees. In R. L. de Mántaras and L. Saitta, editors, *Proceedings of ECAI'04*, pages 211–215. IOS Press, 2004.
20. M. Sellmann and P. Van Hentenryck. Structural symmetry breaking. In *Proceedings of IJCAI'05*, pages 298–303. IJCAI, 2005.
21. I. Shlyakhter. Generating effective symmetry-breaking predicates for search problems. *Electronic Notes in Discrete Mathematics*, 9, 2001. Proceedings of SAT'01.
22. B. M. Smith. Reducing symmetry in a combinatorial design problem. In C. Gervet and M. Wallace, editors, *Proceedings of CP-AI-OR'01*, 2001.
23. B. M. Smith, S. C. Brailsford, P. M. Hubbard, and H. P. Williams. The progressive party problem: Integer linear programming and constraint programming compared. *Constraints*, 1:119–138, 1996.
24. P. Van Hentenryck, P. Flener, J. Pearson, and M. Ågren. Tractable symmetry breaking for CSPs with interchangeable values. In *Proceedings of IJCAI'03*, pages 277–282. Morgan Kaufmann, 2003.
25. P. Van Hentenryck, P. Flener, J. Pearson, and M. Ågren. Compositional derivation of symmetries for constraint satisfaction. In J.-D. Zucker and L. Saitta, editors, *Proceedings of SARA'05*, volume 3607 of *LNAI*, pages 234–247. Springer-Verlag, 2005.