Symmetry Breaking

Toby Walsh*

National ICT Australia and School of CSE, University of New South Wales, Sydney, Australia, tw@cse.unsw.edu.au

Symmetry occurs in many problems in aritifical intelligence. For example, in the n-queens problem, the chessboard can be rotated 90°. As a second example, several machines in a factory might have the same capacity. In a production schedule, we might therefore be able to swap the jobs on machines with the same capacity. As a third example, two people within a company might have the same skills. Given a staff roster, we might therefore be able to interchange these two people. And as a fourth example, when configuring a computer, two memory boards might be identical. We might therefore be able to swap them around without changing the performance of the computer.

Symmetries come in many different forms. For instance, there are rotation symmetries (e.g. the 90° rotations of the chessboard in the n-queens problem), reflection symmetries (e.g. the reflection of the chessboard along one of its diagonals), and permutation symmetries (e.g. the interchange of equally skilled personnel in a staff roster). Problems may have many symmetries at once (e.g. the n-queens problem simultaneously has several rotation and reflection symmetries). We can describe such symmetries using group theory. The symmetries of a problem form a group. Their actions are to map solutions (a n-queens solution, a production schedule, a staff roster, a configuration, etc.) onto solutions. We must deal with such symmetry or we will waste much time visiting symmetric solutions. In addition, we must deal with symmetry during our search for solutions otherwise we will visit many search states that are symmetric to those that we have already visited.

One well known mechanism to deal with symmetry is to add constraints which eliminate symmetric solutions [1]. I will describe a general method for breaking any type of symmetry [2, 3]. The basic idea is very simple. We pick an ordering on the variables, and then post symmetry breaking constraints to ensure that the final solution is lexicographically less than any symmetric re-ordering of the variables. That is, we select the "lex leader" assignment. In theory, this solves the problem of symmetries, eliminating all symmetric solutions and pruning many symmetric states. Unfortunately, the set of symmetries might be exponentially large (for example, there are $n!$ symmetries if we have n identical machines in a factory).

By exploting properties of the set of symmetries we can sometimes overcome this problem. I will describe two special cases where we can deal with an expo-

* Thanks to my co-authors, especially Christian Bessiere, Alan Frisch, Emmanuel Hebrard, Brahim Hnich, Zeynep Kiziltan and Ian Miguel. NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

nential number of symmetries in polynomial time. These two cases frequently occur in practice. In the first case, we have a matrix (or array) of decisions variables in which the rows and/or columns are symmetric and can be permuted. In this case, we can lexicographical order the rows and/or columns [4, 5]. In the second case, we have values for our decision variables which are symmetric and can be interchanged. I show how so called "value precedence" can be used to break all such symmetry [6, 7].

Finally, I will end with a discussion of open research questions in this area. Are they efficient ways to combine together these symmetry breaking constraints for particular types of symmetries? Are there useful subsets of these symmetry breaking constraints when there are too many to post individually? Are there problems where the symmetry breaking constraints can be simplified? Can these symmetry breaking methods be used outside of constraint programming?

References

1. Puget, J.F.: On the satisfiability of symmetrical constrained satisfaction problems. In Komorowski, J., Ras, Z., eds.: Proceedings of ISMIS'93. LNAI 689, Springer-Verlag (1993) 350–361
2. Crawford, J., Luks, G., Ginsberg, M., Roy, A.: Symmetry breaking predicates for search problems. In: Proceedings of the 5th International Conference on Knowledge Representation and Reasoning, (KR '96). (1996) 148–159
3. Walsh, T.: General symmetry breaking constraints. In: 12th International Conference on Principles and Practices of Constraint Programming (CP-2006), Springer-Verlag (2006)
4. Shlyakhter, I.: Generating effective symmetry-breaking predicates for search problems. In: Proceedings of LICS workshop on Theory and Applications of Satisfiability Testing (SAT 2001). (2001)
5. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Breaking row and column symmetry in matrix models. In: 8th International Conference on Principles and Practices of Constraint Programming (CP-2002), Springer (2002)
6. Law, Y., Lee, J.: Global constraints for integer and set value precedence. In: Proceedings of 10th International Conference on Principles and Practice of Constraint Programming (CP2004), Springer (2004) 362–376
7. Walsh, T.: Symmetry breaking using value precedence. In: Proceedings of the 17th ECAI, European Conference on Artificial Intelligence, IOS Press (2006)