Symmetry Breaking Constraints: Recent Results[*]

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

Abstract

Symmetry is an important problem in many combinatorial problems. One way of dealing with symmetry is to add constraints that eliminate symmetric solutions. We survey recent results in this area, focusing especially on two common and useful cases: symmetry breaking constraints for row and column symmetry, and symmetry breaking constraints for eliminating value symmetry.

Introduction

Symmetry occurs in many constraint satisfaction and optimisation problems (Gent, Petrie, and Puget 2006). For example, suppose we have a proper coloring of a graph, and we permute the colors then we will obtain another symmetric coloring. Symmetries are problematic as they increase the size of the search space. We will waste time visiting symmetric solutions. Worse still, we will waste even more time visiting the (many failing) parts of the search tree which are symmetric to already visited states. A common and effective method to deal with symmetry is to add constraints which eliminate some, but not all symmetric solutions (e.g. (Puget 1993; Crawford et al. 1996; Shlyakhter 2001; Aloul, Sakallah, and Markov 2003; Puget 2005c; Law and Lee 2006; Walsh 2006a)). In this paper, we survey recent results on symmetry breaking constraints. We hope that this survey is of wider interest as many of the results are likely to translate to other domains like planning, model checking and heuristic search.

Symmetry breaking constraints have a number of good and bad properties. On the positive side, a few simple constraints can often eliminate most if not all symmetry in a problem quickly and easily. In addition, propagation between the problem and the symmetry breaking constraints can reduce search considerably. On the negative side, we pick out particular solutions in each symmetry class, and

this may conflict with the direction of the branching heuristic. An alternative to posting static symmetry breaking constraints is a more dynamic approach that modifies the search method to ignore symmetric states (Backofen and Will 1999; Gent and Smith 2000; Fahle, Schamberger, and Sellmann 2001; Sellmann and Hentenryck 2005; Puget 2006). The advantage of a dynamic method is that it can reduce the conflict with the branching heuristic. However, we may also get less propagation. The results reported here on static symmetry breaking constraints typically have implications for dynamic methods as most dynamic methods are equivalent to adding static symmetry breaking constraints "on the fly".

There are several aspects of symmetry breaking that this survey does not cover. We do not discuss methods to identify symmetry in a problem. See, for instance, (Puget 2005a; Mears et al. 2008). We suppose that the symmetries are known in advance as they often are, and our challenge is merely to deal with them. The survey also does not cover other active area of research like the intersection of symmetry reasoning and nogood learning. See, for instance, (Benhamou et al. 2010; Chu et al. 2011). For background material on constraint satisfaction in general, see a text like (Rossi, van Beek, and Walsh 2006).

An Example

We consider a combinatorial problem studied in several earlier works on symmetry breaking.

Running example: *The Equidistant Frequency Permutation Array (EFPA) problem (Huczynska et al. 2009) is a challenging problem in coding theory. The goal is to find a set of v code words, each of length $q\lambda$ such that each word contains λ copies of the symbols 1 to q, and each pair of code words is Hamming distance d apart. For example, for $v = 4$, $\lambda = 2$, $q = 3$, $d = 4$, one solution is:*

$$
\begin{array}{cccccc}
0 & 2 & 1 & 2 & 0 & 1 \\
0 & 2 & 2 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 1 & 2 \\
0 & 0 & 1 & 1 & 2 & 2 \\
\end{array}
\qquad \text{(a)}
$$

This problem has applications in communication theory, and is related to other combinatorial problems like finding orthogonal Latin squares. We can model this problem by a v by $q\lambda$ array of decision variables with domains 1 to q.

A *symmetry* of such a problem is a bijection σ that maps solutions onto solutions (Cohen et al. 2006). We consider two special classes of symmetries: *variable symmetries* which act just on variables, and *value symmetries* which act just on values. Both types of symmetry can be found in our model of the EFPA problem.

Running example: *This model of the EFPA problem has variable symmetry since we can permute the variables in any two rows, or in any two columns and still have a solution. For example, we can swap the first column with the third without changing the frequency of symbols in each row or the Hamming distance between any two rows:*

$$
\begin{array}{cccccc}
1 & 2 & 0 & 2 & 0 & 1 \\
2 & 2 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 1 & 2 \\
1 & 0 & 0 & 1 & 2 & 2 \\
\end{array} \quad (b)
$$

The model also has value symmetry since we can interchange values throughout any solution. For example, the value symmetry θ swaps the value 0 with 2 throughout solution (a)*, again without changing the frequency of symbols in each row or the Hamming distance between any two rows:*

$$
\begin{array}{cccccc}
2 & 0 & 1 & 0 & 2 & 1 \\
2 & 0 & 0 & 1 & 1 & 2 \\
2 & 1 & 2 & 0 & 1 & 0 \\
2 & 2 & 1 & 1 & 0 & 0 \\
\end{array} \quad (c)
$$

Symmetry Breaking Constraints

A simple but effective method to deal with symmetry is to add constraints which eliminate some (but not all) symmetric solutions in each symmetry class.

Running example: *To eliminate the value symmetry in the EFPA problem, we can add a* PRECEDENCE *constraint (Law and Lee 2004) to ensure that the values in the bottom row occur in order. That is, the value 0 first occurs on the bottom row before 1, and 1 first occurs before 2. The bottom row of solution* (a) *is 001122 which satisfies this symmetry breaking constraint. The value 0 first occurs in the first position, the value 1 later on in the third position, and the value 2 even later on in the fifth position. Suppose we swap 0 with 2, then the bottom row of solution* (a) *becomes 221100, the bottom row of* (c)*. This violates the* PRECEDENCE *symmetry breaking constraint. The value 1 first occurs in the third position, which is before the first occurrence of the value 0 in the fifth position. Hence, by posting this symmetry breaking constraint, we only find solution* (a)*, and the symmetric solution* (c) *is eliminated.*

Two important properties of symmetry breaking constraints are soundness and completeness. A set of symmetry breaking constraints is *sound* iff it leaves at least one solution in each symmetry class, and *complete* iff it leaves at most one solution in each symmetry class. The PRECEDENCE constraint is, for example, sound and complete. Where do symmetry breaking constraints like this come from in general? The Lex-Leader method offers a generic method for deriving a sound and complete set of symmetry breaking constraints for variable symmetries (Crawford et

al. 1996) and value symmetries (Walsh 2007). The method picks out the lexicographically smallest solution in each symmetry class. For every symmetry σ, it posts a lexicographical ordering constraint:

$$
\langle X_1, \ldots, X_n \rangle \leq_{\text{lex}} \sigma(\langle X_1, \ldots, X_n \rangle)
$$

Where X_1 to X_n is some fixed ordering on the variables in the problem.

Running example: *Consider again the value symmetry, θ that swaps 0 and 2. We can describe this by the mapping $\theta(X) = 2 - X$. The Lex-Leader method eliminates this symmetry with the ordering constraint:*

$$
\langle X_1, \ldots, X_{24} \rangle \leq_{\text{lex}} \langle 2 - X_1, \ldots, 2 - X_{24} \rangle
$$

Where X_1 to X_{24} is some ordering of the 24 decision variables modelling our 4 by 6 instance of the EFPA problem. This simplifies to:

$$
\langle X_1, \ldots, X_{24} \rangle \leq_{\text{lex}} \langle 1, \ldots, 1 \rangle
$$

Suppose we order the variables in the matrix row-wise from top left to bottom right. Then this constraint would accept solution (a)*, but eliminate solutions* (b) *and* (c)*.*

Many static symmetry breaking constraints can be derived from such Lex-Leader constraints. For example, PRECEDENCE constraints to break the symmetry due to interchangeable values can be derived from them (Walsh 2006b). Efficient algorithms have been developed to propagate lexicographical ordering constraints (Frisch et al. 2002; 2006; Katsirelos, Narodytska, and Walsh 2009; Bessiere et al. 2011).

Symmetries of Symmetry Breaking Constraints

One way of constructing new symmetry breaking constraints is to use symmetry itself (Katsirelos and Walsh 2010). Any symmetry acting on a set of symmetry breaking constraints will itself break the symmetry in a problem. This requires us to define the action of a symmetry on a set of symmetry breaking constraints. We defined symmetry as acting on assignments, mapping solutions to solutions. We can lift this definition to constraints. For example, the action of a variable symmetry on a constraint changes the scope of the constraint (that is, the variables on which the constraint acts).

Theorem 1 ((Katsirelos and Walsh 2010)) *Given a set of symmetries Σ of C, if S is a sound (complete) set of symmetry breaking constraints for Σ then $\sigma(S)$ for any $\sigma \in \Sigma$ is also a sound (complete) set of symmetry breaking constraints for Σ.*

Different symmetries pick out different solutions in each symmetry class. In fact, if we have a particular solution in mind, we can pick it out using a suitable symmetry of a set of symmetry breaking constraints. Let $sol(C)$ be the set of solutions of a set of constraints C.

Theorem 2 ((Katsirelos and Walsh 2010)) *Given a set of symmetries Σ of a set of constraints C, a sound set S of symmetry breaking constraints, and any solution A of C, then there is a symmetry $\sigma \in \Sigma$ such that $A \in sol(C \cup \sigma(S))$.*

Applying symmetry to a set of symmetry breaking constraints does not change the symmetries which are eliminated. We say that a set of constraints S *breaks* a symmetry σ of a problem C iff there exists a solution A of $C \cup S$ such that $\sigma(A)$ is not a solution of $C \cup S$, and *completely breaks* a symmetry σ iff for each solution A of $C \cup S$, $\sigma(A)$ is not a solution of $C \cup S$. Given a symmetry group Σ, we say that a set of constraints (completely) breaks Σ iff it breaks every non-identity symmetry in Σ.

Theorem 3 ((Katsirelos and Walsh 2010)) *Given a problem C and a symmetry group Σ, if S (completely) breaks Σ then $\sigma(S)$ (completely) breaks Σ for any $\sigma \in \Sigma$.*

We have used these ideas as the theoretical basis for a hybrid symmetry breaking method that tries to have the best of both worlds, posting static symmetry breaking constraints dynamically during search according to the direction of the branching heuristic (Katsirelos and Walsh 2010).

Intractability of Breaking Symmetry

The symmetry groups seen in practice can be very large. As a result, symmetry breaking can be computationally challenging. For example, if we have a q interchangeable values (as in our EFPA model), then we have $q!$ symmetries. To eliminate each of these symmetries requires a separate Lex-Leader constraint. As a consequence, the Lex-Leader method is not tractable in general. An alternative is to break symmetry partially by posting just a subset of the Lex-Leader constraints (e.g. (Jefferson and Petrie 2011)).

Crawford *et al.* (1996) proved that breaking symmetry completely by adding constraints to eliminate symmetric solutions is computationally intractable in general. More specifically, they prove that, given a matrix model with row and column symmetries, deciding if an assignment is the smallest in its symmetry class is NP-hard where we append rows together and compare them lexicographically. There is, however, nothing special about appending rows together or comparing assignments lexicographically. We could break symmetry with *any* total ordering over assignments.

Recently we have shown that, under modest assumptions, breaking symmetry remains intractable if we use a different ordering of variables, or even a different ordering over solutions than the lexicographical ordering. More precisely, we show that the problem is NP-hard for any *simple* ordering where, given an assignment, we can compute the position in the ordering in polynomial time and, in the reverse, given a position in the ordering, we can compute the associated assignment in polynomial time.

Theorem 4 ((Walsh 2011)) *Given any simple ordering, there exists a symmetry group such that deciding if an assignment is smallest in its symmetry class according to this ordering is NP-hard.*

Since breaking symmetry appears intractable in general, a major research direction is to identify special cases which occur in practice where the symmetry group is more tractable to break. We consider two such cases: row and column symmetry, and value symmetry.

Row and Column Symmetry

A matrix of decision variables has *row symmetry* iff given a solution, any permutation of the rows is also a solution. Similarly, it has *column symmetry* iff given a solution, any permutation of the columns is also a solution. For example, our model of the EFPA problem has both row and column symmetry. Row and column symmetry occurs in many models with matrices of decision variables (Flener et al. 2001a; 2001b; 2002). We can break row and and column symmetry using the Lex-Leader method. However, this is not practical in general. A n by m matrix has $n!m!$ row and column symmetries, and each symmetry would require a separate lexicographical ordering constraint.

To break all row symmetry, we can post a linear number of constraints that lexicographically order the rows. Similarly, to break all column symmetry we can post a linear number of constraints that lexicographically order the columns. When we have both row and column symmetry, we can post a DOUBLELEX constraint that lexicographical orders both the rows and columns (Flener et al. 2002). In fact, DOUBLELEX constraints can be derived from a subset of the constraints introduced by the Lex-Leader method (Flener et al. 2001a). As a result, DOUBLELEX does not break all row and column symmetry. Nevertheless, it is often highly effective in practice.

Running example: *Consider again solution* (a). *If we order the rows of* (a) *lexicographically, we get a solution in which rows and columns are ordered lexicographically:*

$$
\begin{matrix}
0 & 2 & 1 & 2 & 0 & 1 \\
0 & 2 & 2 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 1 & 2 \\
0 & 0 & 1 & 1 & 2 & 2
\end{matrix}
\quad
\begin{matrix}
\text{order} \\
\Rightarrow \\
\text{rows}
\end{matrix}
\quad
\begin{matrix}
0 & 0 & 1 & 1 & 2 & 2 \\
0 & 1 & 0 & 2 & 1 & 2 \\
0 & 2 & 1 & 2 & 0 & 1 \\
0 & 2 & 2 & 1 & 1 & 0
\end{matrix}
\quad \text{(d)}
$$

Similarly if we order the columns of (a) *lexicographically, we get a different solution in which both rows and columns are again ordered lexicographically:*

$$
\begin{matrix}
0 & 2 & 1 & 2 & 0 & 1 \\
0 & 2 & 2 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 1 & 2 \\
0 & 0 & 1 & 1 & 2 & 2
\end{matrix}
\quad
\begin{matrix}
\text{order} \\
\Rightarrow \\
\text{cols}
\end{matrix}
\quad
\begin{matrix}
0 & 0 & 1 & 1 & 2 & 2 \\
0 & 1 & 0 & 2 & 1 & 2 \\
0 & 1 & 2 & 0 & 2 & 1 \\
0 & 2 & 2 & 1 & 1 & 0
\end{matrix}
\quad \text{(e)}
$$

All three solutions are thus in the same row and column symmetry class. Note that both (d) *and* (e) *satisfy the* DOUBLELEX *constraint. Therefore* DOUBLELEX *can leave multiple solutions in each symmetry class and is not a complete symmetry breaking method.*

In fact, DOUBLELEX is a very incomplete method for breaking symmetry. It can leave $n!$ symmetric solutions in an $2n$ by $2n$ matrix model. In addition, it only partially provides tractability. Whilst it is polynomial to check a DOUBLELEX constraint given a complete assignment, it is not tractable to propagate it completely given a partial assignment. That is, pruning all symmetric values is NP-hard.

Theorem 5 ((Katsirelos, Narodytska, and Walsh 2010)) *Propagating the* DOUBLELEX *constraint is NP-hard.*

We have identified three tractable cases where we can break all row and column symmetry in polynomial time (Katsirelos, Narodytska, and Walsh 2010): (1) a matrix with

a bounded number of rows (or columns); (2) a 0/1 matrix model where every row sum is 1; (3) an all-different matrix where all entries are different. The first two are perhaps the most interesting and useful cases. We have, for example, used the first case as the basis of a complete method for breaking row and column symmetry. We used this to measure how many symmetries are left by DOUBLELEX in practice. See Table 1 for some results that demonstrate DOUBLELEX actually leaves very few symmetric solutions in practice despite the worst case result that it can leave a factorial number of such solutions. More recently, Yip and Van Hentenryck (2011) have turned this theoretical result into a complete and efficient method for breaking all row and column symmetry in matrix models with a small number of rows (or columns). Another interesting research direction is to identify other constraints that can be effectively added to DOUBLELEX to increase the amount of row and column symmetry broken (Frisch, Jefferson, and Miguel 2003).

(q, λ, d, v)	# symmetry classes	# symmetric solutions	DOUBLELEX # solutions / time
$(3, 3, 2, 3)$	**6**	$1.81 \cdot 10^5$	**6** / 0.00
$(4, 3, 3, 3)$	8	$> 3.88 \cdot 10^7$	16 / 0.01
$(4, 4, 2, 3)$	**12**	$> 5.87 \cdot 10^7$	**12** / 0.00
$(3, 4, 6, 4)$	1427	$> 5.57 \cdot 10^7$	11215 / 5.88
$(4, 3, 5, 4)$	8600	$> 2.03 \cdot 10^7$	61258 / 69.90
$(4, 4, 5, 4)$	9696	$> 5.45 \cdot 10^6$	72251 / 173.72
$(5, 3, 3, 4)$	5	$> 4.72 \cdot 10^6$	20 / 0.36
$(3, 3, 4, 5)$	18	$> 2.47 \cdot 10^7$	71 / 0.17
$(3, 4, 6, 5)$	4978	$> 2.08 \cdot 10^7$	77535 / 167.50
$(4, 3, 4, 5)$	441	$> 6.55 \cdot 10^6$	2694 / 19.37
$(4, 4, 2, 5)$	**12**	$> 6.94 \cdot 10^6$	**12** / 0.02
$(4, 4, 4, 5)$	717	$> 6.27 \cdot 10^6$	4604 / 38.15
$(4, 6, 4, 5)$	819	$> 4.08 \cdot 10^6$	5048 / 69.83
$(5, 3, 4, 5)$	3067	$> 2.39 \cdot 10^6$	20831 / 403.97
$(6, 3, 4, 5)$	15192	$> 2.16 \cdot 10^6$	$1.11 \cdot 10^5$ / 4924.41

Table 1: Equidistant Frequency Permutation Array problems. Number of solutions left when posting DOUBLELEX symmetry breaking constraints.

SnakeLex

An interesting alternative method for breaking row and column symmetry uses a lexicographical ordering which linearizes the matrix in a snake-like manner instead of rowwise. This method appends the first row to the reverse of the second row, and this is appended to the third row, and then the reverse of the fourth row, and so on. Breaking row and column symmetry with this ordering is intractable, as it is with the more conventional lexicographical ordering where matrices are ordered row-wise.

Theorem 6 ((Narodytska and Walsh 2012)) *Deciding if an assignment is smallest in the snake-lex ordering up to row and column permutation is NP-hard.*

The SNAKELEX constraint, which is derived from the Lex-Leader method using such a snake-lex ordering, has been shown to a promising alternative to DOUBLELEX

(Grayland, Miguel, and Roney-Dougal 2009). To break column symmetry, SNAKELEX ensures that the first column is lexicographically smaller than or equal to both the second and third columns, the reverse of the second column is lexicographically smaller than or equal to the reverse of both the third and fourth columns, and so on up till the penultimate column is compared to the final column. To break row symmetry, SNAKELEX ensures that each neighbouring pair of rows, $X_{1,i}, \ldots, X_{n,i}$ and $X_{1,i+1}, \ldots, X_{n,i+1}$ satisfy the entwined lexicographical ordering:

$$\langle X_{1,i}, X_{2,i+1}, X_{3,i}, X_{4,i+1}, \ldots \rangle \leq_{\text{lex}}$$
$$\langle X_{1,i+1}, X_{2,i}, X_{3,i+1}, X_{4,i}, \ldots \rangle$$

DOUBLELEX breaks the subset of the row and column symmetries that swap neighbouring rows and columns. SNAKELEX, by comparison, breaks a strict superset of these symmetries. Not surprisingly, it is a little more expensive to propagate but this greater cost is often worthwhile in pruning search. Like DOUBLELEX, SNAKELEX is also an incomplete symmetry breaking method that can leave an exponential number of symmetric solutions (Katsirelos, Narodytska, and Walsh 2010).

Gray Code Ordering

Another interesting alternative method for breaking symmetry is to eliminate all but the smallest symmetric solution in some *other* ordering than the lexicographical ordering. For example, we have seen some promise in using the Gray code ordering (Narodytska and Walsh 2012). The Gray code ordering is a total ordering over assignments used in error correcting codes. The 4-bit Gray code orders assignments as follows:

0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100,

1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000

Such an ordering may pick out different solutions in each symmetry class. This can reduce the conflict between symmetry breaking, the problem constraints and the branching heuristic. The Gray code ordering has some properties that may make it useful for symmetry breaking. For example, neighbouring assignments in the ordering only differ at one position, and flipping one bit reverses the ordering of the subsequent bits. Note that there is no renaming of the variables and values that maps the ordering of solutions in the Gray code ordering onto that of the lexicographical ordering. Nevertheless, breaking row and column symmetry with this ordering over solutions remains intractable.

Theorem 7 ((Narodytska and Walsh 2012)) *Deciding if an assignment is smallest in the Gray code ordering up to row and column permutation is NP-hard.*

Experiments suggest some promise for symmetry breaking methods using the Gray code ordering (Narodytska and Walsh 2012). See Table 2 for some example results on a benchmark domain, the maximum density still life problem. This is prob032 in CSPLib (Gent and Walsh 1999). Given a n by n sub-matrix of the infinite plane, we want to find the maximum density pattern which does not change from generation to generation. This problem has the 8 symmetries of

the square as we can rotate or reflect any still life to obtain a new one. We broke all symmetries in the problem with either the lexicographical or Gray code orderings, finding the smallest (lex, gray) or largest (anti-lex, anti-gray) solution in each symmetry class.

Symmetry breaking	$n = 4$	5	6	7	8
none	176	1,166	12,205	231,408	5,867,694
gray	91	446	5,702	123,238	2,507,747
anti-lex	84	424	5,473	120,112	2,416,266
lex	33	406	2,853	87,781	1,982,698
anti-gray	26	269	2,288	38,476	1,073,659

Table 2: Backtracks required to find and prove optimal the maximum density still life of size n by n.

Value Symmetry

A second important class of symmetries which are more tractable to break than symmetries in general are value symmetries. When a problem has many value symmetries, the Lex-Leader method again introduces too many symmetry breaking constraints to be practical. Puget has proposed an effective alternative that maps value symmetries into variables symmetries (Puget 2005b). His method channels into some auxiliary variables which, as in (Smith, Stergiou, and Walsh 2000), permit the (symmetry breaking) constraints to be easily specified. Z_j records the first index using each value j with constraints of the form:

$$X_i = j \quad \Rightarrow \quad Z_j \leq i$$
$$Z_j = i \quad \Rightarrow \quad X_i = j$$

Puget then breaks the variable symmetry on the Z_j by posting binary ordering constraints. For example, if the values 1 to q are interchangeable, we can strictly order the Z_j:

$$Z_1 < Z_2 < Z_3 < \ldots < Z_q$$

This ensures that the first occurrence of 1 is before that of 2, that of 2 is before 3, etc. This is the condition enforced by the PRECEDENCE constraint mentioned previously. Puget proved that we can break *any* number of value symmetries (and not just those due to interchangeable values) with a linear number of ordering constraints on the Z_j. This means that we can break any number of value symmetries in polynomial time. Unfortunately, this decomposition into ordering constraints on Z_j hinders propagation, and we may not prune all symmetric values. This is not surprising as pruning all symmetric values is intractable in general.

Theorem 8 ((Walsh 2007)) *Given a set of value symmetries and a partial assignment, pruning all symmetric values is NP-hard.*

We have, however, identified a common case where pruning all symmetric values is tractable. If values partition into a fixed number of sets and values within each set are interchangeable then we can prune all symmetric values in polynomial time using a global REGULAR constraint (Pesant 2004) or one of its decompositions (Quimper and Walsh

2006; 2007). When there is a single set of interchangeable values, we can prune all symmetric values using a global PRECEDENCE constraint (Law and Lee 2004) or its decomposition (Walsh 2006b).

Combinations of Symmetries

Another important question is how we deal with with combinations of symmetries. We have already seen that we can take a symmetry group that is tractable to break completely (row symmetry), combine it with another symmetry group that is tractable to break completely (column symmetry), and end up with a symmetry group (row and column symmetry) that is intractable to break and where we lack good methods to eliminate *all* symmetric solutions in general.

Another challenging combination is when we have both variable and value symmetries. How do we break both types of symmetry simultaneously? We can use the Lex-Leader method but this is not practical when we have many variable and value symmetries. Unfortunately, Puget's method for breaking value symmetry on the Z_j variables is not compatible in general with breaking variable symmetry via the Lex-Leader method on the X_i variables. This corrects Theorem 6 and Corollary 7 in (Puget 2005b) which claim that the two methods are compatible if we use the same ordering of variables in each method.

Theorem 9 ((Katsirelos, Narodytska, and Walsh 2010))
There exist problems on which posting lex-leader constraints to break variable symmetries and applying Puget's method to break value symmetries remove all solutions in a symmetry class irrespective of the orderings on variables used by both methods.

There is, however, an important and common case where the two methods for breaking variable and value symmetry are compatible. If values partition into interchangeable sets then it is always consistent to post symmetry breaking constraints using both methods provided we index their variables identically in both cases (Sellmann and Hentenryck 2005; Flener et al. 2006; Law et al. 2007). This is again a case where tractability is limited as pruning all symmetric values is NP-hard.

Open Problems

A number of important and challenging questions about symmetry breaking constraints remain including:

Identifying tractable cases: Are there other common types of symmetry which occur in practice that are polynomial to break? For instance, suppose we are coloring the edges of a graph, and we have a clique of interchangeable vertices. This induces a particular symmetry on the decision variables representing the colors of the different edges in the graph. How do we efficiently break this large symmetry group?

Exploring other orderings: Are there other orderings besides the lexicographical ordering with which we can break symmetry effectively? We have seen some promise for the Gray code ordering. There are, however, many other orderings we could consider. For example, Frisch

et al. have proposed the multiset ordering (Frisch et al. 2003). Even though this is only a partial ordering, it has also shown some promise. In addition, can we use a different ordering in each symmetry class?

Reducing branching conflict: One of the major issues with symmetry breaking constraints is that they may conflict with the direction of the branching heuristic. How can we retain the benefits of static symmetry breaking constraints and of dynamic branching heuristics, whilst avoiding such conflict? How do we get the best of static and dynamic symmetry breaking methods? How do we choose static symmetry breaking constraints that align with a dynamic branching heuristic?

Studying combinations of symmetries: How do we eliminate combinations of symmetries effectively? There are issues of both soundness (which symmetry breaking constraints can be combined together safely?), completeness (how we do eliminate all combinations of symmetry?) and tractability (which symmetry groups that are tractable to break individually combine together to give a symmetry group which is also tractable to break?).

Dominance detection and elimination: in constraint optimisation problems, the notion of dominance plays a very similar role to symmetry. One (partial) solution dominates another if it is sure to be at least as good in quality. How do we identify dominating solutions? Is there a generic method like Lex-Leader for adding constraints that remove dominated solutions? What are interesting and useful tractable cases?

References

Aloul, F.; Sakallah, K.; and Markov, I. 2003. Efficient symmetry breaking for Boolean satisfiability. In *Proc. of 18th International Joint Conf. on AI*, 271–276. IJCAI.

Backofen, R., and Will, S. 1999. Excluding symmetries in constraint-based search. In Jaffar, J., ed., *Proc. of the 5th International Conf. on Principles and Practice of Constraint Programming*, LNCS 1713, 73–87. Springer.

Benhamou, B.; Nabhani, T.; Ostrowski, R.; and Saïdi, M. 2010. Enhancing clause learning by symmetry in sat solvers. In *Proc. of 22nd IEEE International Conf. on Tools with Artificial Intelligence (ICTAI 2010)*, volume 1, 329–335. IEEE Computer Society.

Bessiere, C.; Narodytska, N.; Quimper, C.-G.; and Walsh, T. 2011. The alldifferent constraint with precedences. In Achterberg, T., and Beck, J. C., eds., *Proc. of 8th International Conf. on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2011)*, *LNCS* 6697, 36–52. Springer.

Chu, G.; Stuckey, P.; de la Banda, M.; and Mears, C. 2011. Symmetries and lazy clause generation. In Walsh, T., ed., *Proc. of 22nd International Joint Conf. on Artificial Intelligence (IJCAI-2011)*, 516–521. IJCAI/AAAI.

Cohen, D.; Jeavons, P.; Jefferson, C.; Petrie, K.; and Smith, B. 2006. Symmetry definitions for constraint satisfaction problems. *Constraints* 11(2–3):115–137.

Crawford, J.; Luks, G.; Ginsberg, M.; and Roy, A. 1996. Symmetry breaking predicates for search problems. In *Proc. of the 5th International Conf. on Knowledge Representation and Reasoning, (KR '96)*, 148–159.

Fahle, T.; Schamberger, S.; and Sellmann, M. 2001. Symmetry breaking. In Walsh, T., ed., *Proc. of 7th International Conf. on Principles and Practice of Constraint Programming (CP2001)*, 93–107. Springer.

Flener, P.; Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; Pearson, J.; and Walsh, T. 2001a. Symmetry in matrix models. Technical Report APES-30-2001, APES group. Presented at SymCon'01 (Symmetry in Constraints), CP2001 post-conference workshop.

Flener, P.; Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; and Walsh, T. 2001b. Matrix modelling. Technical Report APES-36-2001, APES group. Presented at Formul'01 Workshop on Modelling and Problem Formulation, CP2001 post-conference workshop.

Flener, P.; Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; Pearson, J.; and Walsh, T. 2002. Breaking row and column symmetry in matrix models. In *8th International Conf. on Principles and Practices of Constraint Programming (CP-2002)*. Springer.

Flener, P.; Pearson, J.; Sellmann, M.; and Hentenryck, P. V. 2006. Static and dynamic structural symmetry breaking. In *Proc. of 12th International Conf. on Principles and Practice of Constraint Programming (CP2006)*. Springer.

Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; and Walsh, T. 2002. Global constraints for lexicographic orderings. In *8th International Conf. on Principles and Practices of Constraint Programming (CP-2002)*. Springer.

Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; and Walsh, T. 2003. Multiset ordering constraints. In *Proc. of 18th International Joint Conf. on Artificial Intelligence (IJCAI-2003)*. IJCAI.

Frisch, A.; Hnich, B.; Kiziltan, Z.; Miguel, I.; and Walsh, T. 2006. Propagation algorithms for lexicographic ordering constraints. *Artificial Intelligence* 170(10):803–908.

Frisch, A.; Jefferson, C.; and Miguel, I. 2003. Constraints for breaking more row and column symmetries. In Rossi, F., ed., *Proc. of 9th International Conf. on Principles and Practice of Constraint Programming (CP2003)*. Springer.

Gent, I., and Smith, B. 2000. Symmetry breaking in constraint programming. In Horn, W., ed., *Proc. of ECAI-2000*, 599–603. IOS Press.

Gent, I., and Walsh, T. 1999. CSPLib: a benchmark library for constraints. Technical report, Technical report APES-09-1999. A shorter version appears in *Proc. of the 5th International Conf. on Principles and Practices of Constraint Programming (CP-99)*.

Gent, I.; Petrie, K.; and Puget, J.-F. 2006. Symmetry in constraint programming. In *Handbook for Constraint Programming*. Elsevier.

Grayland, A.; Miguel, I.; and Roney-Dougal, C. 2009. Snake lex: An alternative to double lex. In Gent, I. P., ed., *Proc. of 15th International Conf. on Principles and*

Practice of Constraint Programming, LNCS 5732, 391–399. Springer.

Huczynska, S.; McKay, P.; Miguel, I.; and Nightingale, P. 2009. Modelling equidistant frequency permutation arrays: An application of constraints to mathematics. In Gent, I. P., ed., *Proc. of 15th International Conf. on Principles and Practice of Constraint Programming (CP 2009)*, LNCS 5732, 50–64. Springer.

Jefferson, C., and Petrie, K. 2011. Automatic generation of constraints for partial symmetry breaking. In Lee, J. H.-M., ed., *Proc. of 17th International Conf. on Principles and Practice of Constraint Programming (CP 2011)*, LNCS 6876, 729–743. Springer.

Katsirelos, G., and Walsh, T. 2010. Symmetries of symmetry breaking constraints. In *Proc. of the 19th European Conf. on Artificial Intelligence (ECAI-2010)*. IOS Press.

Katsirelos, G.; Narodytska, N.; and Walsh, T. 2009. Combining symmetry breaking and global constraints. In Oddi, A.; Fages, F.; and Rossi, F., eds., *Recent Advances in Constraints, 13th Annual ERCIM International Workshop on Constraint Solving and Constraint Logic Programming (CSCLP 2008)*, LNCS 5655, 84–98. Springer.

Katsirelos, G.; Narodytska, N.; and Walsh, T. 2010. On the complexity and completeness of static constraints for breaking row and column symmetry. In Cohen, D., ed., *Proc. of the 16th International Conf. on the Principles and Practice of Constraint Programming (CP 2010)*, LNCS 6308, 305–320. Springer.

Law, Y., and Lee, J. 2004. Global constraints for integer and set value precedence. In *Proc. of 10th International Conf. on Principles and Practice of Constraint Programming (CP2004)*, 362–376. Springer.

Law, Y., and Lee, J. 2006. Symmetry Breaking Constraints for Value Symmetries in Constraint Satisfaction. *Constraints* 11(2–3):221–267.

Law, Y.-C.; Lee, J.; Walsh, T.; and Yip, J. 2007. Breaking symmetry of interchangeable variables and values. In *13th International Conf. on Principles and Practices of Constraint Programming (CP-2007)*. Springer.

Mears, C.; de la Banda, M.; Wallace, M.; and Demoen, B. 2008. A novel approach for detecting symmetries in CSP models. In Perron, L., and Trick, M. A., eds., *Proc. of 5th International Conf. on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2008)*, volume *LNCS 5015*, 158–172. Springer.

Narodytska, N., and Walsh, T. 2012. Beyond lex leader: Breaking symmetry with other orderings. Under review for *Proc. of the 18th International Conf. on the Principles and Practice of Constraint Programming (CP 2012)*, LNCS. Springer.

Pesant, G. 2004. A regular language membership constraint for finite sequences of variables. In Wallace, M., ed., *Proc. of 10th International Conf. on Principles and Practice of Constraint Programming (CP2004)*, 482–295. Springer.

Puget, J.-F. 1993. On the satisfiability of symmetrical con-

strained satisfaction problems. In Komorowski, J., and Ras, Z., eds., *Proc. of ISMIS'93*, LNAI 689, 350–361. Springer.

Puget, J.-F. 2005a. Automatic detection of variable and value symmetries. In van Beek, P., ed., *Proc. of 11th International Conf. on Principles and Practice of Constraint Programming (CP 2005)*, LNCS 3709, 475–489. Springer.

Puget, J.-F. 2005b. Breaking all value symmetries in surjection problems. In van Beek, P., ed., *Proc. of 11th International Conf. on Principles and Practice of Constraint Programming (CP2005)*. Springer.

Puget, J.-F. 2005c. Breaking symmetries in all different problems. In *Proc. of 19th International Joint Conf. on AI*, 272–277. IJCAI.

Puget, J.-F. 2006. Dynamic lex constraints. In Benhamou, F., ed., *12th International Conf. on the Principles and Practice of Constraint Programming (CP 2006)*, LNCS 4204, 453–467. Springer.

Quimper, C.-G., and Walsh, T. 2006. Global grammar constraints. In *12th International Conf. on Principles and Practices of Constraint Programming (CP-2006)*. Springer.

Quimper, C.-G., and Walsh, T. 2007. Decomposing global grammar constraints. In *13th International Conf. on Principles and Practices of Constraint Programming (CP-2007)*. Springer.

Rossi, F.; van Beek, P.; and Walsh, T., eds. 2006. *Handbook of Constraint Programming*. Foundations of Artificial Intelligence. Elsevier.

Sellmann, M., and Hentenryck, P. V. 2005. Structural symmetry breaking. In *Proc. of 19th International Joint Conf. on AI (IJCAI 2005)*, 298–303. IJCAI.

Shlyakhter, I. 2001. Generating effective symmetry-breaking predicates for search problems. In *Proc. of LICS workshop on Theory and Applications of Satisfiability Testing (SAT 2001)*.

Smith, B.; Stergiou, K.; and Walsh, T. 2000. Using auxiliary variables and implied constraints to model non-binary problems. In *Proc. of the 16th National Conf. on AI*, 182–187. AAAI.

Walsh, T. 2006a. General symmetry breaking constraints. In *Proc. of 12th International Conf. on Principles and Practices of Constraint Programming (CP-2006)*. Springer.

Walsh, T. 2006b. Symmetry breaking using value precedence. In *Proc. of the 17th European Conf. on Artificial Intelligence (ECAI-2006)*. IOS Press.

Walsh, T. 2007. Breaking value symmetry. In *Proc. of 13th International Conf. on Principles and Practices of Constraint Programming (CP-2007)*. Springer.

Walsh, T. 2011. On the complexity of breaking symmetry. In *Proc. of The 11th International Workshop on Symmetry in Constraint Satisfaction Problems (SymCon'11)*. Held alongside the 1th International Conf. on the Principles and Practice of Constraint Programming (CP 2011).

Yip, J., and Hentenryck, P. V. 2011. Symmetry breaking via LexLeader feasibility checkers. In Walsh, T., ed., *Proc. of the 22nd International Joint Conf. on Artificial Intelligence (IJCAI-2011)*, 687–692. IJCAI/AAAI.