# General Symmetry Breaking Constraints

Toby Walsh[1]

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

**Abstract.** We present some general constraints for breaking symmetries in constraint satisfaction problems. These constraints can be used to break symmetries acting on variables, values, or both. We also consider symmetry breaking constraints to deal with conditional symmetries, and symmetries acting on set and other types of variables.

## 1 Introduction

Symmetry occurs in many constraint satisfaction problems. We must deal with symmetry or we will waste much time visiting symmetric solutions, as well as parts of the search tree which are symmetric to already visited parts. One mechanism to deal with symmetry is to add constraints which eliminate symmetric solutions [1]. Crawford *et al.* have presented [2] a simple method for breaking any type of symmetry between variables. 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. Whilst this method was defined for symmetries of Boolean variables, it lifts immediately to symmetries of non-Boolean variables. In this paper, we show how this basic method can be extended further to value symmetries, to symmetries involving both variables and values, and to symmetries acting on set and other types of variables. This provides what we believe is the first *generic* method for constructing symmetry breaking constraints that works with *any* type of symmetry.

## 2 Background

A constraint satisfaction problem consists of a set of variables, each with a domain of values, and a set of constraints specifying allowed combinations of values for given subsets of variables. A solution is an assignment of values to variables satisfying the constraints. Finite domain variables take one value from a given finite set. Set variables take sets of such values and are typically defined by a lower bound on the definite elements and an upper bound on the definite and potential elements. Systematic constraint solvers explore partial assignments enforcing some level of local consistency property. We consider two of the most common local consistencies: arc consistency and bound consistency. Given a constraint $C$ on finite domain variables, a *support* is assignment to each variable of a value in its domain which satisfies $C$. A constraint $C$ on finite domains variables is *generalized arc consistent* ($GAC$) iff for each variable, every value in its domain belongs to a support. Given a constraint $C$ on set variables, a *bound support* on $C$ is an assignment of a set to each set variable between its lower and upper bounds which satisfies $C$. A constraint $C$ is *bound consistent* ($BC$) iff for each set variable $S$, the values in $ub(S)$ belong to $S$ in at least one bound support and the values in $lb(S)$ belong to $S$ in all bound supports.

## 3 Variable symmetry breaking

A variable symmetry $\sigma$ is a bijection on variables that preserves solutions. That is, if $\{X_i = v_i \mid 1 \leq i \leq n\}$ is a solution, then $\{X_{\sigma(i)} = v_i \mid 1 \leq i \leq n\}$ is also. Crawford *et al.* [2] show how to break *all* variable symmetry by posting the constraint $\textsc{LexLeader}(\sigma, [X_1, \ldots, X_n])$ for each variable symmetry $\sigma$. This ensures:

$$[X_1, \ldots, X_n] \leq_{\text{lex}} [X_{\sigma(1)}, \ldots, X_{\sigma(n)}]$$

Where $X_1$ to $X_n$ is any fixed ordering on the variables. To enforce GAC on such a constraint, we can use the lex propagator described in [3] which copes with repeated variables. This takes $O(nm)$ time where $m$ is the maximum domain size. In general, this decomposition into a set of LexLeader constraints hinders propagation. There may be values which can be pruned because they do not occur in all lex leaders which we will not identify by looking at the lex leader constraints individually.

**Theorem 1** *$GAC(\bigwedge_{\sigma \in \Sigma} \textsc{LexLeader}(\sigma, [X_1, \ldots, X_n]))$ is strictly stronger than $GAC(\textsc{LexLeader}(\sigma, [X_1, \ldots, X_n]))$ for all $\sigma \in \Sigma$.*

**Proof:** Clearly it is at least as strong. We show strictness with just two symmetries and four variables. Consider $X_1, X_2, X_4 \in \{0, 1\}$, $X_3 = 1$ and two symmetries defined by the cycles $(1\,4\,2\,3)$ and $(1\,2\,4\,3)$. Then $[X_1, X_2, X_3, X_4] \leq_{\text{lex}} [X_4, X_3, X_1, X_2]$ and $[X_1, X_2, X_3, X_4] \leq_{\text{lex}} [X_2, X_4, X_1, X_3]$) are GAC. However, enforcing GAC on their conjunction prunes 0 from $X_4$ as the only support for $X_3 = 1$ and $X_4 = 0$ that satisfies $[X_1, X_2, X_3, X_4] \leq_{\text{lex}} [X_4, X_3, X_1, X_2]$ is $X_1 = X_2 = 0$, whilst the only support for $X_3 = 1$ and $X_4 = 0$ that satisfies $[X_1, X_2, X_3, X_4] \leq_{\text{lex}} [X_2, X_4, X_1, X_3]$) is $X_1 = 0$ and $X_2 = 1$. $\diamond$

Breaking all symmetry may require an exponential number of LexLeader constraints (e.g. the $n!$ symmetries of $n$ indistinguishable variables). It may therefore be worth developing a global constraint that combines together several LexLeader constraints. For example, the lex chain constraint [4] breaks all row symmetries of a matrix model. This would require an exponential number of LexLeader constraints. Unfortunately, there are also cases where such a combined symmetry breaking constraint is intractable. For example, a global constraint that combines together all the LexLeader constraints for row and column symmetries of a matrix model is NP-hard to propagate completely [5].

## 4 Value symmetry breaking

A value symmetry $\sigma$ is a bijection on values that preserves solutions. That is, if $\{X_i = v_i \mid 1 \leq i \leq n\}$ is a solution, then $\{X_i = \sigma(v_i) \mid 1 \leq i \leq n\}$ is also.

Inspired by [2], we now propose a generic method to break *all* value symmetry. We simply post the constraint ValLexLeader($\sigma, [X_1, \ldots, X_n]$) for each value symmetry $\sigma$. This holds iff:

$$[X_1, \ldots, X_n] \leq_{\text{lex}} [\sigma(X_1), \ldots, \sigma(X_n)]$$

Again $X_1$ to $X_n$ is any fixed ordering on the variables. For example, suppose we have a reflection symmetry on the values 1 to $m$ defined by $\sigma_r(i) = m - i + 1$. This value symmetry holds for the Golomb ruler problem (prob006 in CSPLib). We can break this symmetry with a single ValLexLeader constraint. If $m$ is even, ValLexLeader($\sigma_r, [X_1, \ldots, X_n]$) simplifies to the logically equivalent constraint: $X_1 \leq \frac{m}{2}$. If $m$ is odd, it simplifies into the constraints: $X_1 \leq \frac{m+1}{2}$, if $X_1 = \frac{m+1}{2}$ then $X_2 \leq \frac{m+1}{2}$, if $X_1 = X_2 = \frac{m+1}{2}$ then $X_3 \leq \frac{m+1}{2}$, ..., and if $X_1 = \ldots = X_{n-1} = \frac{m+1}{2}$ then $X_n \leq \frac{m+1}{2}$.

It is an interesting open question how to simplify ValLexLeader constraints for other types of value symmetries. We conjecture that we may be able to apply some of Puget's ideas from [6]. Even without simplification, we can enforce GAC on a ValLexLeader constraint in $O(nm)$ time for any value symmetry. One possibility is to adapt the lex propagator from [7]. Alternatively, we can decompose ValLexLeader into ternary constraints and a set of Boolean variables, $B_i$ which play the role of $\alpha$ in the lex propagator. We identify two important sets of values: $\mathcal{L}$ are those values for which $v < \sigma(v)$, whilst $\mathcal{E}$ are those values for which $v = \sigma(v)$. We post the sequence of constraints, $C(X_i, B_i, B_{i+1})$ for $1 \leq i \leq n$, where $B_1 = 0$ and $C(X_i, B_i, B_{i+1})$ holds iff $B_i = B_{i+1} = 0$ and $X_i \in \mathcal{E}$, or $B_i = 0$, $B_{i+1} = 1$ and $X_i \in \mathcal{L}$, or $B_i = B_{i+1} = 1$. We can enforce GAC on the ternary constraint $C$ using a table constraint, GAC-schema or logical primitives like implication. As the constraint graph of the decomposition is Berge-acyclic, enforcing GAC on each ternary constraint achieves GAC on the whole ValLexLeader constraint.

As with variable symmetry, decomposition into individual symmetry breaking constraints may hinder propagation.

**Theorem 2** *GAC*($\bigwedge_{\sigma \in \Sigma}$ ValLexLeader($\sigma, [X_1, \ldots, X_n]$)) *is strictly stronger than GAC*(ValLexLeader($\sigma, [X_1, \ldots, X_n]$)) *for all* $\sigma \in \Sigma$.

**Proof:** Clearly it is at least as strong. We show strictness with just two symmetries and two variables. Suppose $X_1 \in \{0, 1\}$ and $X_2 \in \{0, 2\}$. Consider a value symmetry $\sigma$ defined by the cycle $(0\,2)$. Then ValLexLeader($\sigma, [X_1, X_2]$)) is GAC. Consider a second value symmetry $\sigma'$ defined by the cycle $(1\,2)$. Then ValLexLeader($\sigma', [X_1, X_2]$)) is GAC. However, enforcing GAC on the conjunction of these two symmetry breaking constraints prunes 2 from $X_2$. $\diamond$

Breaking all value symmetry may again introduce an exponential number of symmetry breaking constraints (e.g. the $n!$ symmetries of $n$ indistinguishable values). It may therefore be worth developing a specialized global constraint that combines together several ValLexLeader constraints. For example, as we discuss later, the global precedence constraint [11] combines together all the ValLexLeader constraints for indistinguishable values.

## 5 Variable and value symmetry breaking

We may have both variable and value symmetries. Consider, for example, a model of the rehearsal problem (prob039 in CSPLib) in which variables represent time slots and values are the pieces practised in each time slot. This model has variable symmetry since any rehearsal can be reversed, as well as value symmetry since any piece requiring the same players is indistinguishable. Can we simply combine the appropriate LexLeader and ValLexLeader constraints?

If each symmetry breaking constraint considers the variables in different orders, it may not be safe to combine them. For example, if $\sigma$ reflects 1 and 2, and $X_1, X_2 \in \{1,2\}$ then LexLeader($\sigma, [X_1, X_2]$) and ValLexLeader($\sigma, [X_2, X_1]$) eliminate the assignment $X_1 = 1$ and $X_2 = 2$, as well as all its symmetries. We assume therefore in what follows that the lexicographical ordering within each symmetry breaking constraint considers the variables $X_1$ to $X_n$ in the same ordering, and is the lifting of the same ordering on values to an ordering on tuples. Variable and value symmetry breaking constraints can then be combined safely.

**Theorem 3** *If $X_1$ to $X_n$ have a set of variable symmetries $\Sigma$ and a set of value symmetries $\Sigma'$ then posting* LexLeader($\sigma, [X_1, \ldots, X_n]$) *for all $\sigma \in \Sigma$ and* ValLexLeader($\sigma', [X_1, \ldots, X_n]$) *for all $\sigma' \in \Sigma'$ leaves one or more assignments in each equivalence class.*

**Proof:** Consider any assignment. Pick the lex leader under $\Sigma$ of this assignment. By construction, this satisfies LexLeader($\sigma, [X_1, \ldots, X_n]$) for all $\sigma \in \Sigma$. Now consider the lex leader under $\Sigma'$ of this current assignment. By construction, this satisfies ValLexLeader($\sigma', [X_1, \ldots, X_n]$) for all $\sigma' \in \Sigma'$. This also moves us down the lexicographical order on tuples. However, we may no longer satisfy LexLeader($\sigma, [X_1, \ldots, X_n]$). We therefore pick the lex leader under $\Sigma$ of our current assignment. We again must move down the lexicographical order on tuples. This process terminates as the lexicographical order is well founded and bounded by $[0, \ldots, 0]$ where 0 is the least value. We terminate with an assignment that satisfies both the LexLeader and ValLexLeader constraints. $\diamond$

Such symmetry breaking may leave one or *more* assignments in each equivalence class. Consider, for example, a Boolean problem in which both variables and values have reflection symmetry. Then the assignments $[0, 1, 1]$ and $[0, 0, 1]$ are symmetric, and both satisfy the appropriate LexLeader and ValLexLeader constraints. Thus, whilst we can post variable symmetry breaking constraints and value symmetry breaking constraints separately, they may not break all symmetry. We need to consider symmetry breaking constraints for the combined variable and value symmetries. We give such constraints in the next section.

## 6 Variable/value symmetry breaking

In general, a symmetry $\sigma$ is a bijection on assignments that preserves solutions [8]. We call this a variable/value symmetry to distinguish it from symmetries that act just on the variables or just on the values. Consider, for example, a model for the $n$-queens problem in which we have one variable for each row. This problem has a rotational symmetry $r90$ that maps $X_i = j$ onto $X_j = n - i + 1$. This is neither a variable nor a value symmetry as it acts on both variables and values together. With variable/value symmetries, we need to be careful that the image of an assignment is itself a proper assignment. We say that a complete assignment $[X_1, \ldots, X_n]$ is *admissible* for $\sigma$ if the image under $\sigma$ is also a complete assignment. In particular, the image should assign one value to each variable. Thus $[X_1, \ldots, X_n]$ is admissible iff $|\{k \mid X_i = j, \sigma(i \times j) = k \times l\}| = n$. To return to the 3-queens example, the assignment $[2, 3, 1]$ for $[X_1, X_2, X_3]$ is admissible for $r90$ as its image under $r90$ is $[1, 3, 2]$ which is a complete assignment. However, the assignment $[2, 3, 3]$ is not as its image tries to assign both 1 and 2 to $X_3$. If $[X_1, \ldots, X_n]$ is admissible for $\sigma$ we write $\sigma([X_1, \ldots, X_n])$ for its image under $\sigma$. More precisely, $\sigma([X_1, \ldots, X_n])$ is the sequence $[Y_1, \ldots, Y_n]$ where for each $X_i = j$ and $\sigma(i \times j) = k \times l$, we have $Y_k = l$.

We now propose a *generic* method to break *all* variable/value symmetry. We simply post the constraint $\textsc{GenLexLeader}(\sigma, [X_1, \ldots, X_n])$ for each variable/value symmetry $\sigma$. This holds iff:

$$admissible(\sigma, [X1, \ldots, X_n]) \;\&\; [X_1, \ldots, X_n] \leq_{\text{lex}} \sigma([X_1, \ldots, X_n])$$

Again $X_1$ to $X_n$ is any fixed ordering on the variables. If $\sigma$ is a variable symmetry or a value symmetry, all assignments are admissible and we get the same symmetry breaking constraint as before. Consider the 3-queens problem and the $r90$ rotational symmetry. Suppose $X_1 = 2$, $X_2 \in \{1, 3\}$ and $X_3 \in \{1, 2, 3\}$. Then enforcing GAC on $\textsc{GenLexLeader}(r90, [X_1, X_2, X_3])$ prunes $X_3 = 2$ as this does not extend to an admissible assignment, as well as $X_2 = 3$ and $X_3 = 1$ as the image under $r90$ of any admissible assignment with $X_2 = 3$ or $X_3 = 1$ is smaller in the lexicographical order. As before, decomposition into individual symmetry breaking constraints may hinder propagation so it may be worth developing a specialized global constraint that combines several together.

One way to propagate an individual $\textsc{GenLexLeader}$ constraint is to introduce variables $Y_1$ to $Y_n$ for the image of $X_1$ to $X_n$. We post channelling constraints of the form $X_i = j$ iff $Y_k = l$ for each $i$, $j$ where $\sigma(i \times j) = k \times l$. Finally, we post a lexicographical ordering constraint of the form $[X_1, \ldots, X_n] \leq_{\text{lex}} [Y_1, \ldots, Y_n])$. Enforcing GAC on this decomposition takes just $O(nm)$ time. Unfortunately, this decomposition may hinder propagation.

**Theorem 4** *GAC*$(\textsc{GenLexLeader}(\sigma, [X_1, \ldots, X_n]))$ *is strictly stronger than GAC*$([X_1, \ldots, X_n] \leq_{\text{lex}} [Y_1, \ldots, Y_n])$ *and GAC*$(X_i = j \text{ iff } Y_k = l)$ *for each* $i$, $j$ *where* $\sigma(i \times j) = k \times l$.

**Proof:** Clearly it is at least as strong. To show strictness, consider the 3-queens problem and the symmetry $r90$. Suppose $X_1 \in \{1, 3\}$, $X_2 = 2$ and $X_3 \in \{1, 3\}$. Then enforcing GAC on $\textsc{GenLexLeader}(r90, [X_1, X_2, X_3])$ prunes $X_1 = 3$ and $X_3 = 1$ as these are not lex leaders. However, the decomposition is GAC. ⋄

To enforce GAC on a $\textsc{GenLexLeader}$ constraint, we can adapt the lex propagator with repeated variables [3]. This give an $O(n^2)$ time propagator.

The basic idea is simple but unfortunately we lack space to give full details. The propagator focuses on the first position in the lex constraint where the variables are not ground and equal. We need to test if the variables at this position can be strictly ordered or made equal consistent with admissibility, As in the lex propagator with repeated variables, they can only be made equal if admissibility does not then require the rest of the vector to be ordered the wrong way.

## 7 Conditional symmetry breaking

A conditional symmetry $\sigma$ is a symmetry that preserves solutions subject to some condition [9]. This condition might be a partial assignment (e.g. that $X_1 = X_2 = 1$) or, more generally, an arbitrary constraint (e.g. that $X_1 \neq X_n$). More precisely, a conditional symmetry $\sigma$ is a bijection on assignments such that if $A$ is an admissible solution and $A$ satisfies $C_\sigma$ then $\sigma(A)$ is also a solution. Consider, for example, the all interval series problem (prob007 in CSPLib) in which we wish to find a permutation of the integers from 0 to $n$, such that the difference between adjacent numbers is itself a permutation from 1 to $n$. One model of this problem is a sequence of integer variables where $X_i$ represents the number in position $i$. The problem has a variable reflection symmetry since we can invert any sequence. The problem also has a value reflection symmetry since we can map any value $j$ to $n-j$. Finally, the problem has a conditional symmetry since we can cycle a solution about a pivot to generate a new solution. Consider $n = 4$ and the solution $[3, 2, 0, 4, 1]$. The difference between first and last numbers in the sequence is 2. However, this is also the difference between the 2nd and 3rd numbers in the sequence. We can therefore rotate round to this point to give the new solution $[0, 4, 1, 3, 2]$. This symmetry is conditional since it depends on the values taken by neighbouring variables.

To break such conditional symmetry, we simply need to post a constraint:

$$C_\sigma(X_1, \ldots, X_n) \text{ implies } \textsc{GenLexLeader}(\sigma, [X_1, \ldots, X_n])$$

Consider again the all interval series problem. We can break the conditional symmetries in this problem by posting the constraints:

$$|X_1 - X_{n+1}| = |X_i - X_{i+1}| \text{ implies } \textsc{LexLeader}(rot_i, [X_1, \ldots, X_{n+1}])$$

Where $1 \leq i \leq n$ and $rot_i$ is the symmetry that rotates a sequence by $i$ elements.

One way to propagate such conditional symmetry breaking constraints is to compute the (maximal) set of valid assignments for $C_\sigma$ and the (maximal) set of inconsistent assignments for GenLexLeader. Those assignments in the intersection of these two sets can be pruned. By Theorem 3 in [10], this achieves GAC on the conditional symmetry breaking constraint.

## 8 Indistinguishable values

A common type of value symmetry in which symmetry breaking constraints are effective is when values are interchangeable. For example, in a model of the social golfer problem (prob010 in CSPLib) in which we assign groups to golfers in each week, all values are interchangeable. To break all such symmetry, Law and Lee [11] propose the value precedence constraint:

$$\textsc{Precedence}([v_1, \ldots, v_m], [X_1, \ldots, X_n])$$

This holds iff $\min\{i \mid X_i = v_i \vee i = n+1\} < \min\{i \mid X_i = v_j \vee i = n+2\}$ for all $1 \leq i < j < m$. To propagate this constraint, Law and Lee decompose it into pairwise value precedence constraints, and give a specialized algorithm for enforcing GAC on each pairwise constraint [11]. In [12], we show that this decomposition hinders propagation and give a simple ternary encoding which permits us to achieve GAC in linear time. It is not hard to show that a value precedence constraint combines together the exponential number of ValLexLeader constraints which break the symmetry of indistinguishable values.

## 9 All different problems

Another class of problems on which symmetry breaking constraints are especially effective are those like the all interval series problem in which variables must all take different values. On such problems, Puget has shown that the LexLeader constraints for any (possibly exponentially large) set of variable symmetries simplify down to a linear number of binary inequalities [13]. These can be determined by computing stabilizer chains and coset representatives. For example, on the all interval series problem, the variable reflection symmetry breaking constraint simplifies to $X_1 < X_{n+1}$.

Puget has also shown how to use this method to break value symmetries [6]. We assume we have a surjection problem in which each value is used at least once. (If this is not the case, we introduce "dummy" variables to take the unused values.) We then introduce index variables, $Z_i$ where $Z_j = min\{i \mid X_i = j\}$. These variables are by construction all-different and have a variable symmetry. We can thus break the original value symmetry by adding just a linear number of binary inequalities to break this variable symmetry. For interchangeable values, Puget's method gives the binary symmetry breaking constraints: $X_i = j \rightarrow Z_j \leq i$, $Z_j = i \rightarrow X_i = j$, and $Z_k < Z_{k+1}$ for all $1 \leq i \leq n$, $1 \leq j \leq m$ and $1 \leq k < m$. We observe that these constraints break all value symmetries by ensuring value precedence [11]. However, this decomposition into binary constraints hinders propagation. Consider $X_1 = 1$, $X_2 \in \{1, 2\}$, $X_3 \in \{1, 3\}$, $X_4 \in \{3, 4\}$, $X_5 = 2$, $X_6 = 3$, $X_7 = 4$, $Z_1 = 1$, $Z_2 \in \{2, 5\}$, $Z_3 \in \{3, 4, 6\}$, and $Z_4 \in \{4, 7\}$. Now all the binary implications are arc consistent. However, we can prune $X_2 = 1$ as this violates value precedence.

If we have both variable and value symmetries and an all different problem, we can break variable symmetry with $O(n)$ binary inequalities on the primal model, and break value symmetry with $O(m)$ binary inequalities on the dual. This symmetry breaking on the dual is compatible with symmetry breaking on the primal [6]. However, as in the general case, this may not break all symmetry. Consider an all different problem with 3 variables and 3 values with variable rotation symmetry and value reflection symmetry. Then $[1, 2, 3]$ and $[1, 3, 2]$ are symmetric, but both satisfy the (simplified) variable symmetry breaking constraints that $X_1 < X_2$ and $X_1 < X_3$, and the (simplified) value symmetry breaking constraint (on the dual) that $Y_1 < Y_3$.

## 10 Set variable symmetry breaking

There are many problems involving symmetry which are naturally modelled and effectively solved using set variables. Set variables can themselves eliminate some symmetry (in particular, the ordering between elements). However, we may still be left with symmetry. For example, another model of the social golfers problem introduces a 2 dimensional matrix of set variables, one for each group and each week. As groups and weeks are indistinguishable, these set variables have row and column symmetry. More generally, a set variable symmetry $\sigma$ is a bijection on set variables, $S_1$ to $S_n$ that preserves solutions. To break all such symmetry, we simply post the constraint $\textsc{SetLexLeader}(\sigma, [S_1, \ldots, S_n])$ for each set variable symmetry $\sigma$. This ensures:

$$[S_1, \ldots, S_n] \leq_{\text{lex}} [S_{\sigma(1)}, \ldots, S_{\sigma(n)}]$$

Again, $S_1$ to $S_n$ is any fixed ordering on the set variables. The lexicographical ordering, $\leq_{\text{lex}}$ is the standard lifting of the multiset ordering on set variables to an ordering on tuples of set variables. Note that the multiset ordering on sets is equivalent to the lexicographical ordering on the characteristic function.

We can propagate the $\textsc{SetLexLeader}$ constraint by adapting the propagator for the the lex constraint on finite domain variables described in [3] which copes with repeated variables. Alternatively, we can post a $\textsc{SetLexLeader}$ constraint by means of the following decomposition: $D(S_i, S_{\sigma(i)}, B_i, B_{i+1})$ for $1 \leq i \leq n$ where $B_i$ are Booleans playing the role of $\alpha$ in the lex propagator, $B_1 = 0$, and $D(S_i, S_{\sigma(i)}, B_i, B_{i+1})$ holds iff $B_i = B_{i+1} = 0$ and $S_i = S_{\sigma(i)}$, or $B_i = 0$, $B_{i+1} = 1$ and $S_i <_{\text{mset}} S_{\sigma(i)}$, or $B_i = B_{i+1} = 1$, and $<_{\text{mset}}$ is the multiset ordering on sets.

We still need to provide a propagator for each quaternary constraint, $D$. This is not trivial as the quaternary constraint is over both set and Boolean variables, and involves notions like whether the sets are multiset ordered. We suppose the set variables are represented by their characteristic function using an $m$-tuple of Boolean variables. We will use the notation $S_{i,k}$ for the Boolean indicating whether $k \in S_i$. If $i = \sigma(i)$ then $D(S_i, S_{\sigma(i)}, B_i, B_{i+1})$ simplifies to the equality constraint: $B_i = B_{i+1}$. If $i \neq \sigma(i)$ then we give a further decomposition. This decomposition exploits the property that the multiset ordering on sets is identical to the lexicographical ordering on the characteristic function. More precisely, we decompose $D(S_i, S_j, B_i, B_{i+1})$ into $E(S_{i,k}, S_{j,k}, B_i, B_{i+1}, A_{i,k}, A_{i,k+1})$ for $1 \leq k \leq m$, and two implication constraints, $A_{i,m+1} = 0$ implies $B_i = B_{i+1}$ and $A_{i,m+1} = 1$ implies $B_{i+1} = 1$, where $A_{i,k}$ are Booleans (again playing a role similar to $\alpha$ in the lex propagator), and $E(S_{i,k}, S_{j,k}, B_i, B_{i+1}, A_{i,k}, A_{i,k+1})$ holds iff $B_i = A_{i,k} = 0$ implies $S_{i,k} \leq S_{j,k}$ and $S_{i,k} < S_{j,k}$ implies $A_{i,k+1} = 1$.

Our decomposition is a simple logical combinations of Booleans which is readily propagated by most solvers. As this decomposition of $D(S_i, S_j, B_i, B_{i+1})$ is Berge-acyclic, enforcing BC (=GAC) on the decomposition enforces BC on the original quaternary constraint. This takes $O(m)$ time as we can enforce BC on each of the $O(m)$ Boolean constraints in constant time. Enforcing BC on the decomposition of SetLexLeader thus takes $O(nm)$ time.

As with finite domain variables, the decomposition into a set of SetLexLeader constraints hinders propagation.

**Theorem 5** *$BC(\bigwedge_{\sigma \in \Sigma}$ SetLexLeader$(\sigma, [S_1, \ldots, S_n]))$ is strictly stronger than $BC($SetLexLeader$(\sigma, [S_1, \ldots, S_n]))$ for all $\sigma \in \Sigma$.*

**Proof:** Clearly it is at least as strong. We show strictness with just two symmetries and purely Boolean domains. Consider $S_1, S_2 \subseteq \{0, 1\}$, $S_3 = \{0\}$, $S_4 = \{0\}$ and two symmetries defined by the cycles $(1\,2)(3\,4)$ and $(1\,3\,2\,4)$. Then $[S_1, S_2, S_3, S_4] \leq_{\text{lex}} [S_2, S_1, S_4, S_3]$ and $[S_1, S_2, S_3, S_4] \leq_{\text{lex}} [S_4, S_3, S_1, S_2])$ are BC. However, enforcing BC on their conjunction prunes 1 from $X_1$. $\diamond$

Decomposition may also introduce an exponential number of SetLexLeader constraints (e.g. the $n!$ symmetries of $n$ indistinguishable set variables). It may therefore be worth developing specialized global constraints that combine together several SetLexLeader constraints.

## 11 Set value symmetry breaking

The values used by some set variables might also be symmetric. Consider again the model of the social golfers problem with a 2 dimensional matrix of set variables, each containing the set of golfers playing in particular group in a given week. As the golfers are indistinguishable, any permutation of the values is also a solution. More precisely, a set value symmetry $\sigma$ is a bijection on values that preserves solutions. That is, if $\{S_i = \{v_{ij} \mid 1 \leq j \leq m_i\} \mid 1 \leq i \leq n\}$ is a solution, then $\{S_i = \{\sigma(v_{ij}) \mid 1 \leq j \leq m_i\} \mid 1 \leq i \leq n\}$ is also. An even more general definition of symmetry would be a bijection on *sets* of values that preserves solutions. However, we focus here on what appears to be the more common case of a bijection on values. To break *all* such set value symmetry, we simply post the constraint SetValLexLeader$(\sigma, [S_1, \ldots, S_n])$ for each value symmetry $\sigma$. This constraint ensures:

$$[S_1, \ldots, S_n] \leq_{\text{lex}} [\sigma(S_1), \ldots, \sigma(S_n)]$$

Again, $S_1$ to $S_n$ is any fixed ordering on the set variables, and $\leq_{\text{lex}}$ is the standard lifting of the multiset ordering on set variables to an ordering on tuples of set variables. To propagate SetValLexLeader we can adapt the lex propagator from [7]. Alternatively, we can use a simple decomposition somewhat similar to that used for ValLexLeader on finite domain variables. We decompose SetValLexLeader$(\sigma, [S_1, \ldots, S_n])$ into $F(S_i, B_i, B_{i+1})$ for $1 \leq i \leq n$ where $B_i$ are Booleans playing the role of $\alpha$ in the lex propagator, $B_1 = 0$, and $F(X_i, B_i, B_{i+1})$ holds iff $B_i = B_{i+1} = 0$ and $S_i = \sigma(S_i)$, or $B_i = 0$, $B_{i+1} = 1$

and $S_i <_{\text{mset}} \sigma(S_i)$, or $B_i = B_{i+1} = 1$. Enforcing BC on this decomposition achieves BC on the SetValLexLeader constraint.

We still need to provide a propagator for each ternary constraint, $F$. This is again not trivial as it involves a set variable and notions like strict multiset ordering. We can again decompose it into some simple Boolean constraints on the characteristic function representation of the set variable. We identify two important sets of values: $\mathcal{L}$ are those values for which $v < \sigma(v)$, whilst $\mathcal{E}$ are those value for which $v = \sigma(v)$. We then decompose $F(S_i, B_i, B_{i+1})$ into $G(S_{i,k}, B_i, B_{i+1}, A_{i,k}, A_{i,k+1})$ for $1 \leq k \leq m$, and two implication constraints, $A_{i,m+1} = 0$ implies $B_i = B_{i+1}$ and $A_{i,m+1} = 1$ implies $B_{i+1} = 1$, where $A_{i,k}$ are Booleans (again playing a role similar to $\alpha$ in the lex propagator), and $G(S_{i,k}, B_i, B_{i+1}, A_{i,k}, A_{i,k+1})$ holds iff ($B_i = A_{i,k} = 0$ and $k \notin \mathcal{L} \cup \mathcal{E}$) implies $S_{i,k} = 0$ and ($k \in \mathcal{L}$ and $S_{i,k} = 1$) implies $A_{i,k+1} = 1$. Our decomposition is a simple logical combination of Booleans which is readily propagated by most solvers. As this decomposition is Berge-acyclic, enforcing BC (=GAC) on the decomposition enforces BC on the original ternary constraint, $F(S_i, B_i, B_{i+1})$. This takes $O(m)$ time as we can enforce BC on the $O(m)$ Boolean implication constraints in constant time. Hence, enforcing BC on SetValLexLeader takes $O(nm)$ time. This is optimal. Nevertheless, it may be worth developing a specialized global constraint that combines together several SetValLexLeader constraints as decomposition into individual constraints may hinder propagation, and as there may be an exponential number of SetValLexLeader constraints. For example, the precedence constraint for set variables [11] combines together all the SetValLexLeader constraints for breaking the symmetry of indistinguishable set values.

## 12 Set variable and value symmetry breaking

Problems can contain both set variable and set value symmetry. Consider again the model of the social golfers problem with a 2 dimensional matrix of set variables, each containing the set of golfers playing in particular group in a given week. As the weeks, groups and golfers are all interchangeable, the set variables in this model have row and column symmetry, and their values have permutation symmetry. As with finite domain variables, we can combine together symmetry breaking constraints for set variables and set values, provided all the symmetry breaking constraints use the same ordering of set variables.

**Theorem 6** *Suppose $S_1$ to $S_n$ have set variable symmetries $\Sigma$ and value symmetries $\Sigma'$. Then posting* SetLexLeader$(\sigma, [S_1, \ldots, S_n])$ *for all $\sigma \in \Sigma$ and* SetValLexLeader$(\sigma', [S_1, \ldots, S_n])$ *for all $\sigma' \in \Sigma'$ leaves one or more assignments in each equivalence class.*

**Proof:** Similar to the proof with finite domain variables. The only difference is that we now consider moving down the lexicographical order on tuples of sets. This is well founded and bounded by $[\{\}, \ldots, \{\}]$ where $\{\}$ is the empty set. To see that we may not break all symmetry, consider a problem in which set variables and values both have reflection symmetry. Then the assignments $[\{0\},\{1\},\{1\}]$ and $[\{0\},\{0\},\{1\}]$ are symmetric, and both satisfy the appropriate SetLexLeader and SetValLexLeader constraints. $\diamond$

In fact, complete symmetry breaking in this case is intractable in general (assuming P$\neq$NP). For instance, if we have set variables and values that are indistinguishable, then the problem is isomorphic to breaking all row and column symmetries of an 0/1 matrix model, and this is NP-hard [5].

## 13 Set variable/value symmetry breaking

Symmetries can act on set variables and their values simultaneously. Consider a model of the peaceable coexisting armies of queens problem [14] in which we have a set variable for each row of the chessboard containing the positions of the white queens along the row. As in [14], we do not place the black queens and just keep a count on the number of squares not attacked by white queens. This model has a rotational symmetry $r90'$ that maps $i \in S_j$ onto $n-i+1 \in S_i$. This symmetry acts on both set variables and values together. More generally, we will consider a set variable/value symmetry to be a bijection on set variable membership constraints that preserves solutions. That is, if $\{i \in S_j \mid 1 \leq j \leq n\}$ is a solution then $\{k \in S_l \mid i \in S_j,\ \sigma(i \times j) = k \times l\}$ is also. Note that the mapping of a value just depends on the set variable to which it is assigned. An even more general definition would be when the mapping depends on both the value, the set variable, and the other values assigned to the set variable.

Given a complete assignment to the sequence of set variables, $[S_1 \ldots, S_n]$ we write $\sigma([S_1,\ldots,S_n])$ for its image under $\sigma$. More precisely, $\sigma([S_1,\ldots,S_n])$ is the sequence $[T_1,\ldots,T_n]$ where for each $\sigma(i \times j) = k \times l$, we have $i \in S_j$ iff $k \in T_l$. To break all set variable/value symmetries, we simply post the constraint SetGenLexLeader($\sigma, [S_1,\ldots,S_n]$) for each such symmetry $\sigma$. This ensures:

$$[S_1,\ldots,S_n] \leq_{\text{lex}} \sigma([S_1,\ldots,S_n])$$

Consider the 5 by 5 peaceable armies of queens problem in which $\{1,4\} \subseteq S_1 \subseteq \{1,4,5\}$, $S_2 = \{2\}$, $S_3 = \{\}$, $\{\} \subseteq S_4 \subseteq \{1\}$ and $\{1\} \subseteq S_5 \subseteq \{1,2,3,4\}$. Enforcing BC on SetGenLexLeader($r90', [S_1,S_2,S_3,S_4,S_5]$) will reduce the upper bound on $S_1$ to its lower bound, $\{1,4\}$ to ensure that the placement of white queens is ordered less than its rotation.

As set variable/value symmetry generalizes both set variable and set value symmetry, decomposition into individual symmetry breaking constraints may hinder propagation. We observe, however, that breaking all such symmetry is intractable in general. To propagate an individual SetGenLexLeader constraint, we introduce variables $T_1$ to $T_n$ for the image of $S_1$ to $S_n$. We post channelling constraints of the form $i \in S_j$ iff $k \in T_l$ for each $i$ and $j$, where $\sigma(i \times j) = k \times l$. Finally we post a lexicographical ordering constraint on the set variables: $[S_1,\ldots,S_n] \leq_{\text{lex}} [T_1,\ldots,T_n]$. We can again propagate this by adapting the propagator for the lex constraint on finite domain variables to set variables [3]. Alternatively, we can use a decomposition similar to that used for the

SetLexLeader constraint. Enforcing BC on this decomposition takes $O(nm)$. Unfortunately, decomposition hinders propagation as it ignores the repeated variables in the lexicographical ordering constraint.

**Theorem 7** *$BC(\text{SetGenLexLeader}(\sigma, [S_1, \ldots, S_n]))$ is strictly stronger than $BC(i \in S_j$ iff $k \in T_l)$ for each $i, j$ where $\sigma(i \times j) = k \times l$, and $BC([S_1, \ldots, S_n] \leq_{\text{lex}} [T_1, \ldots, T_n])$.*

**Proof:** Clearly it is at least as strong. To show strictness, consider $\{\} \subseteq S_1, S_3 \subseteq \{1\}$, $S_2 = \{1\}$ and the rotational set variable symmetry, *rot* which maps $S_1$ to $S_3$, $S_2$ to $S_1$ and $S_3$ to $S_2$. Enforcing BC on $\text{SetGenLexLeader}(rot, [S_1, S_2, S_3])$ reduces the upper bound on $S_3$ to $\{\}$. However, if we let $T_1 = \{1\}$ $\{\} \subseteq T_2, T_3 \subseteq \{1\}$ then the decomposition is BC. $\diamond$

## 14 Symmetry breaking for other representations of sets

We can represent other information about a set variable besides the possible and necessary elements in the set. For example, many solvers include an integer variable to record the cardinality of the set. Such cardinality information can be used when symmetry breaking to permit additional pruning. For instance, suppose we have a set variable, $\{\} \subseteq S_1 \subseteq \{3, 4\}$ with a cardinality of 1 and the permutation set value symmetry, $\sigma_{34}$ that swaps the values 3 and 4. If we post the symmetry breaking constraint, $\text{SetValLexLeader}(\sigma_{34}, [S_1])$ then a propagator can exploit the cardinality information to prune the upper and lower bounds on $S_1$ to give $S_1$ the unique assignment up to set value symmetry of $\{3\}$.

Sadler and Gervet have proposed maintaining upper and lower bounds on a set variable according to the lexicographical ordering [15]. That is, they maintain a set which is lexicographically larger than all possible assignments, as well as one which is lexicographically smaller than all possible assignments. Such lexicographical bounds fit well with the symmetry breaking methods proposed here. For example, suppose we have a pair of set variables, $S_1$ and $S_2$ and the permutation set variable symmetry, $\sigma_{12}$ that swaps $S_1$ and $S_2$. Then the symmetry breaking constraint, $\text{SetLexLeader}(\sigma_{12}, [S_1, S_2])$ simplifies to $S_1 \leq_{\text{lex}} S_2$. We can enforce lexicographical bounds consistency on this ordering constraint simply by making the lexicographical upper bound on $S_1$ equal to the smaller of the two upper bounds, and the lexicographical lower bound on $S_2$ equal to the larger of the two lower bounds.

Another representation used for a set variable is a binary decision diagram [16]. Whilst such a representation requires exponential space in the worst case, it is often manageable in practice and permits the maximum possible pruning. Whilst set variable symmetry breaking constraints fit well with such a decision diagram representation, set value symmetry breaking constraints do not. For set variable symmetry, we can use the same ordering for values within the decision diagram as within the symmetry breaking constraint. This will lead to a compact representation for the symmetry broken set variables. For set value symmetry, different symmetries can map values far apart in the ordering used in the decision diagram. Adding set value symmetry breaking constraints may therefore give exponentially large decision diagrams.

## 15 Symmetry breaking on other variable types

These symmetry breaking methods lift to other types of variables. We consider here multiset variables [17]. Multiset (or bag) variables are useful to model a range of problems. As with set variables, their use can eliminate some but not necessarily all symmetry in the model of a problem. Thus, we may need to break symmetries in models involving multiset variables. For example, one model of the template design problem (prob002 in CSPLib) introduces a multiset variable for each template containing the number of each design. Such multiset variables allow us to ignore the ordering of designs on the template. However, as templates with the same production run length are indistinguishable, certain multiset variables can still be permuted. We thus may have a permutation symmetry, $\sigma$ over some of the multiset variables. To break all such symmetry, we simply post the constraint $\text{MSetLexLeader}(\sigma, [M_1, \ldots, M_n])$ for each multiset variable symmetry $\sigma$. This constraint ensures:

$$[M_1, \ldots, M_n] \leq_{\text{lex}} [M_{\sigma(1)}, \ldots, M_{\sigma(n)}]$$

Again, $M_1$ to $M_n$ is any fixed ordering on the multiset variables. The lexicographical ordering, $\leq_{\text{lex}}$ is the lifting of the multiset ordering on multiset variables to an ordering on tuples of set variables. Note that the multiset ordering is equivalent to the lexicographical ordering on the occurrence representation of the multiset. We can propagate the MSetLexLeader constraint by adapting the propagator for the lex constraint on finite domain variables [3] or with an encoding similar to that proposed for the SetLexLeader constraint. The extensions to multiset value and multiset variable/value symmetries are similar. Another promising computation domain for constraint programming is the domain of graphs. Graph variables can be used in domains like bioinformatics to represent combinatorial graph problems [18]. These symmetry breaking methods also lift to deal with symmetries of such graph variables.

## 16 Related work

Puget proved that symmetries can always be eliminated by the additional of suitable constraints [1]. Crawford *et al.* presented the first general method for constructing such symmetry breaking constraints [2]. However, their lex-leader method was restricted to variable symmetries. In their recent survey chapter on symmetry in constraint programming for the Handbook of Constraint Programming, Gent, Petrie and Puget write that "... generalization of lex-leader to deal with value symmetries would be valuable, even if restricted to some special cases ..." [19]. We provide here this generalization for arbitrary value symmetries (as well as generalizations in several other directions like symmetries acting simultaneously on variables and values, and symmetries acting on set variables).

To reduce the number of lex-leader constraints used, Aloul *et al.* suggest breaking only those symmetries corresponding to generators of the group [20]. Aloul, Sakallah and Markov exploit the cyclic structure of generators to reduce the size of lex-leader constraints from quadratic to linear in the number of variables [21]. The full set of lex-leader constraints can often be simplified. For example, in matrix models with row symmetry, the exponential number of lex-leader constraints simplifies to a linear number of lex row ordering constraints [22, 23]. For matrix models with both row and column symmetry, it is unlikely that we can break all row and column symmetry using a polynomial number of symmetry breaking constraints as this is NP-hard [5]. However, we can break most row and column symmetry using lex constraints to order rows and columns [22, 23]. As a second example, for problems where variables must take all different values, Puget has shown that the lex-leader constraints simplify to just a linear number of binary inequality constraints [13]. Finally, an alternative way to break value symmetry is to convert it into a variable symmetry by channeling into a dual viewpoint and using lex-leader constraints on this dual view [23, 24].

## 17 Conclusions

We have presented some general methods to break symmetries in constraint satisfaction problems. Our symmetry breaking method works with symmetries acting on variables, values, or both. Our methods also work with conditional symmetries, and symmetries acting on set and other types of variables. There exist a number of promising areas for future work. Are they efficient ways to combine together these symmetry breaking constraints for particular types of symmetries (just as value precedence combines together an exponential number of value symmetry breaking constraints)? Are there useful subsets of these symmetry breaking constraints when there are too many to post individually? Are there other types of problems and symmetries where these symmetry breaking constraints simplify dramatically (just as Puget has shown for all different problems)? Finally, can these symmetry breaking methods be used to improve symmetry breaking methods like SBDS and SBDD that work during search?

## References

1. Puget, J.F.: On the satisfiability of symmetrical constrained satisfaction problems. In Proc. of ISMIS'93. LNAI 689, (1993) 350–361
2. Crawford, J., Luks, G., Ginsberg, M., Roy, A.: Symmetry breaking predicates for search problems. In: Proc. of the 5th Int. Conf. on Knowledge Representation and Reasoning, (KR '96). (1996) 148–159
3. Kiziltan, Z.: Symmetry Breaking Ordering Constraints. PhD thesis, Department of Information Science, Uppsala University (2004)
4. Carlsson, M., Beldiceanu, N.: Arc-consistency for a chain of lexicographic ordering constraints. Technical report T2002-18, Swedish Institute of Computer Science (2002)
5. Bessiere, C., Hebrard, E., Hnich, B., Walsh, T.: The complexity of global constraints. In: Proc. of the 19th National Conf. on AI, AAAI (2004)

6. Puget, J.F.: Breaking all value symmetries in surjection problems. In van Beek, P., ed.: Proc. of 11th Int. Conf. on Principles and Practice of Constraint Programming (CP2005), (2005)
7. Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Global constraints for lexicographic orderings. In: 8th Int. Conf. on Principles and Practices of Constraint Programming (CP-2002), (2002)
8. Cohen, D., Jeavons, P., Jefferson, C., Petrie, K., Smith, B.: Symmetry definitions for constraint satisfaction problems. In van Beek, P., ed.: Proc. of 11th Int. Conf. on Principles and Practice of Constraint Programming (CP2005), (2005) 17–31
9. Gent, I., Kelsey, T., Linton, S., McDonald, I., Miguel, I., Smith, B.: Conditional symmetry breaking. In van Beek, P., ed.: Proc. of 11th Int. Conf. on Principles and Practice of Constraint Programming (CP2005), (2005)
10. Bacchus, F., Walsh, T.: Propagating logical combinations of constraints. In: Proc. of 19th IJCAI (2005) 35–40
11. Law, Y., Lee, J.: Global constraints for integer and set value precedence. In: Proc. of 10th Int. Conf. on Principles and Practice of Constraint Programming (CP2004), (2004) 362–376
12. Walsh, T.: Symmetry breaking using value precedence. In: Proc. of the 17th ECAI, European Conf. on Artificial Intelligence, IOS Press (2006)
13. Puget, J.F.: Breaking symmetries in all different problems. In: Proc. of 19th IJCAI (2005) 272–277
14. Smith, B., Petrie, K., Gent, I.: Models and symmetry breaking for peacable armies of queens. In: Proc. of the First Int. Conf. on Integration of AI and OR Techniques in Constraint Programming (CP-AI-OR), (2004) 271–286
15. Sadler, A., Gervet, C.: Hybrid set domains to strengthen constraint propagation and reduce symmetries. In: Proc. of 10th Int. Conf. on Principles and Practice of Constraint Programming (CP2004), (2004)
16. Hawkins, P., Lagoon, V., Stuckey, P.: Solving set constraint satisfaction problems using ROBDDs. Journal of Artificial Intelligence Research **24** (2005) 109–156
17. Walsh, T.: Consistency and propagation with multiset constraints: A formal viewpoint. In Rossi, F., ed.: 9th Int. Conf. on Principles and Practices of Constraint Programming (CP-2003), (2003)
18. Dooms, G., Deville, Y., Dupont, P.: CP(Graph): Introducing a graph computation domain in constraint programming. In van Beek, P., ed.: Proc. of 11th Int. Conf. on Principles and Practice of Constraint Programming (CP2005), (2005)
19. Gent, I., Petrie, K., Puget, J.F.: Symmetry in constraint programming. In: Handbook for Constraint Programming. Elsevier (2006)
20. Aloul, F., Ramani, A., Markov, I., Sakallah, K.: Solving difficult SAT instances in the presence of symmetries. In: Proc. of the Design Automation Conf.. (2002) 731–736
21. Aloul, F., Sakallah, K., Markov, I.: Efficient symmetry breaking for Boolean satisfiability. In: Proc. of the 18th IJCAI (2003) 271–276
22. Shlyakhter, I.: Generating effective symmetry-breaking predicates for search problems. In: Proc. of LICS workshop on Theory and Applications of Satisfiability Testing (SAT 2001). (2001)
23. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Breaking row and column symmetry in matrix models. In: 8th Int. Conf. on Principles and Practices of Constraint Programming (CP-2002), (2002)
24. Law, Y., Lee, J.: Symmetry Breaking Constraints for Value Symmetries in Constraint Satisfaction. Constraints (2006) to appear.