Symmetries of Symmetry Breaking Constraints

George Katsirelos[1] and **Toby Walsh**[2]

Abstract. Symmetry is an important feature of many constraint programs. We show that *any* problem symmetry acting on a set of symmetry breaking constraints can be used to break symmetry. Different symmetries pick out different solutions in each symmetry class. This simple but powerful idea can be used in a number of different ways. We describe one application within model restarts, a search technique designed to reduce the conflict between symmetry breaking and the branching heuristic. In model restarts, we restart search periodically with a random symmetry of the symmetry breaking constraints. Experimental results show that this symmetry breaking technique is effective in practice on some standard benchmark problems.

1 INTRODUCTION

Symmetry occurs in many real world problems. For instance, certain machines in a scheduling problem might be identical. If we have a valid schedule, we can permute these machines and still have a valid schedule. We typically need to factor such symmetry out of the search space to be able to find solutions efficiently. One popular way to deal with symmetry is to add constraints which eliminate symmetric solutions (see, for instance, [15, 18, 4, 14, 19, 21]). Such symmetry breaking is usually simple to implement [5, 6] and is often highly efficient and effective in practice. Even for problems with many symmetries, a small number of symmetry breaking constraints can often eliminate much or all of the symmetry. But where do such symmetry breaking constraints come from? We show here that we can apply symmetry to symmetry breaking constraints themselves to generate potentially new symmetry breaking constraints.

There are a number of applications of this simple but powerful idea. We give one application in the area of restart based methods. Restarting has proven a powerful technique to deal with branching mistakes in backtracking search [9]. One problem with posting symmetry breaking constraints is that they pick out particular solutions in each symmetry class, and branching heuristics may conflict with this choice. *Model restarts* is a technique to deal with this conflict [10]. We periodically restart search with a new model containing different symmetry breaking constraints. Our idea of applying symmetry to symmetry breaking constraints provides a systematic method to generate different symmetry breaking constraints to be used within model restarts. Different symmetries pick out different solutions in each symmetry class. Restarting search with a different symmetry of the symmetry breaking constraints may therefore permit symmetry breaking constraints to be posted that do not conflict with the branching heuristic. Our experimental results show that model restarts is indeed effective at reducing the conflict between branching heuristics and symmetry breaking.

[1] CRIL-CNRS, Lens, France, email: gkatsi@gmail.com
[2] NICTA and UNSW, Sydney, Australia, email: toby.walsh@nicta.com.au

2 SYMMETRY

We consider two common types of symmetry (see [1] for more discussion). A *variable symmetry* is a permutation of the variables that preserves solutions. Formally, a variable symmetry is a bijection σ on the indices of variables such that if $X_1 = d_1, \ldots, X_n = d_n$ is a solution then $X_{\sigma(1)} = d_1, \ldots, X_{\sigma(n)} = d_n$ is also. A *value symmetry* is a permutation of the values that preserves solutions. Formally, a value symmetry is a bijection θ on the values such that if $X_1 = d_1, \ldots, X_n = d_n$ is a solution then $X_1 = \theta(d_1), \ldots, X_n = \theta(d_n)$ is also. In [17], these are called *global value symmetries* as their action on values is the same for all variables. Symmetries can more generally act on both variables and values. Our results apply also to such symmetries. As the inverse of a symmetry and the identity are symmetries, the set of symmetries forms a group under composition.

We will use a simple running example which has a small number of symmetries. This example will demonstrate that we can use symmetry itself to pick out different solutions in each symmetry class.

Running Example. *A magic square is a labelling of a n by n square with the numbers 1 to n^2 so that the sums of each row, column and diagonal are equal (prob019 in CSPLib [8]). The most-perfect magic square problem is to find a magic square in which every 2 by 2 square has the same sum, and in which all pairs of integers $n/2$ apart on either diagonal have the same sum. We model this as a CSP with $X_{i,j} = k$ iff the square (i, j) contains k. One solution for $n = 4$ is:*

14	11	5	4
1	8	10	15
12	13	3	6
7	2	16	9

(1)

This is one of the oldest known most-perfect magic squares, dating from a 10th century temple engraving in Khajuraho, India.

This problem has several symmetries. First, there are the 8 symmetries of the square: the identity mapping, the rotations 90° clockwise, 180° and 270°, and the reflections in the vertical, horizontal and diagonal axes. For example, applying the symmetry σ_v that reflects the square in its vertical axis to (1) gives a symmetric solution:

4	5	11	14
15	10	8	1
6	3	13	12
9	16	2	7

(2)

The problem also has a value symmetry θ_{inv} that inverts values, mapping i onto $n^2 + 1 - i$. For instance, applying θ_{inv} to (1), generates another symmetric solution:

3	6	12	13
16	9	7	2
5	4	14	11
10	15	1	8

(3)

We can also combine the value and variable symmetries. For example, if we apply the composition of the last two symmetries (that is, $\theta_{inv} \circ \sigma_v$) to (1), we reflect the solution in the vertical axis and invert all values giving another symmetric solution:

13	12	6	3
2	7	9	16
11	14	4	5
8	1	15	10

(4)

Note that (4) is itself the reflection of (3) in the vertical axis. The problem thus has 16 symmetries in total. ♣

3 SYMMETRY BREAKING

One common way to deal with symmetry is to add constraints to eliminate symmetric solutions [15]. We shall show that we can in fact break symmetry by posting a *symmetry* of any such symmetry breaking constraints.

Running Example. *Consider again the most-perfect magic square problem. To eliminate the symmetries of the square, we can post the constraints:*

$$X_{1,1} < \min(X_{1,n}, X_{n,1}, X_{n,n}), \qquad X_{1,n} < X_{n,1} \tag{5}$$

These ensure that the smallest corner is top left, and the bottom left corner is smaller than the top right. This eliminates all degrees of freedom to rotate and reflect the square. Note that (5) eliminates solutions (1) and (4) but leaves their reflections (2) and (3).

To eliminate the value symmetry θ_{inv} and any rotations or reflections of it, we can post:

$$X_{1,1} < n^2 + 1 - \max(X_{1,1}, X_{1,n}, X_{n,1}, X_{n,n}) \tag{6}$$

This ensures that the smallest corner in the magic square is smaller than the smallest corner in any rotation or reflection of the inversion of the solution. Note that the smallest corner cannot equal *the inversion of any corner since the problem constraints of a most-perfect magic square ensure that the inversion of any corner lies inside the square (in fact, on the diagonal). This symmetry breaking constraint eliminates (2) but leaves (3). Thus, of the four symmetric solutions given earlier, only (3) satisfies (5) and (6).* ♣

One of our main observations is that *any symmetry* acting on a set of symmetry breaking constraints will itself break the symmetry in a problem. Different symmetries pick out different solutions in each symmetry class. To show this, we need to consider the action of a symmetry on a set of symmetry breaking constraints. Symmetry is often defined as acting on assignments, mapping solutions to solutions. We need to lift this definition to constraints. The action of a variable symmetry on a constraint changes the variables on which the constraint acts. More precisely, a variable symmetry σ applied to the constraint $C(X_j, \ldots, X_k)$ gives $C(X_{\sigma(j)}, \ldots, X_{\sigma(k)})$. The action of a value symmetry is also easy to compute. A value symmetry θ applied to the constraint $C(X_j, \ldots, X_k)$ gives $C(\theta(X_j), \ldots, \theta(X_k))$.

Running Example. *To illustrate how we can break symmetry with the symmetry of a set of symmetry breaking constraints, we shall construct symmetries of (5) and (6).*

Consider the symmetry σ_v that reflects the square in the vertical axis mapping $X_{1,1}$ onto $X_{n,1}$ (and vice versa), and $X_{1,n}$ onto $X_{n,n}$ (and vice versa). If we apply σ_v to (5) we get:

$$X_{n,1} < \min(X_{n,n}, X_{1,1}, X_{1,n}), \qquad X_{n,n} < X_{1,1} \tag{7}$$

These new symmetry breaking constraints ensure that the smallest corner is now top right, and the bottom right corner is smaller than the top left. This again eliminates all degrees of freedom to rotate and reflect the square. Note that (7) eliminates solutions (2) and (3) but leaves (1) and (4). This is the opposite of posting (5) which would leave (2) and (3), but eliminate (1) and (4).

If we apply σ_v to (6), we again get a constraint that breaks the value symmetry θ_{inv} and any rotations or reflections of it:

$$X_{n,1} < n^2 + 1 - \max(X_{n,1}, X_{n,n}, X_{1,1}, X_{1,n}) \tag{8}$$

This again ensures that the smallest corner in the magic square is smaller than the smallest corner in any rotation or reflection of the inversion of the solution. This eliminates (1) but leaves (4). Thus, of the four symmetric solutions given earlier, only (4) satisfies the symmetry σ_v of (5) and (6).

We can also break symmetry with any other symmetry of the symmetry breaking constraints. For instance, if we apply $\theta_{inv} \circ \sigma_v$ to (5) we get the constraints:

$$n^2 + 1 - X_{n,1} < \min(n^2 + 1 - X_{n,n}, n^2 + 1 - X_{1,1}, n^2 + 1 - X_{1,n})$$
$$n^2 + 1 - X_{n,n} < n^2 + 1 - X_{1,1}$$

These simplify to:

$$X_{n,1} > \max(X_{n,n}, X_{1,1}, X_{1,n})$$
$$X_{n,n} > X_{1,1}$$

Similarly, if we apply $\theta_{inv} \circ \sigma_v$ to (6) we get the constraint:

$$X_{n,1} > n^2 + 1 - \min(X_{n,1}, X_{n,n}, X_{1,1}, X_{1,n})$$

The three constraints ensure that the largest corner is top right, the bottom right is larger than the top left, and the largest corner is larger than the largest corner in any rotation or reflection of the inversion of the solution. This again prevents us from rotating, reflecting or inverting any solution. Of the four symmetric solutions given earlier, only (2) satisfies $\theta_{inv} \circ \sigma_v$ of (5) and (6). We see therefore that different symmetries of the symmetry breaking constraints pick out different solutions in each symmetry class. ♣

4 THEORETICAL RESULTS

The running example illustrates that we can break symmetry with a symmetry of a set of symmetry breaking constraints. We will prove that this holds in general. Our aim is to show:

> *Any symmetry acting on a set of symmetry breaking constraints itself breaks symmetry. Different symmetries pick out different solutions in each symmetry class.*

We will consider the action of a symmetry on the solutions and symmetries of a set of symmetry breaking constraints. We will also study the action of a symmetry on the soundness and completeness of a set of symmetry breaking constraints, the representative solutions picked out by the symmetry breaking constraints and the symmetries that are eliminated. Finally we will consider what symmetries can be found within a set of symmetry breaking constraints.

4.1 Symmetry and satisfiability

We start with the action of a symmetry on the satisfiability of a set of constraints. This simple result is used in some of the later proofs. We write $\sigma(C)$ for the result of applying the symmetry σ to the set of constraints C.

Proposition 1. *For any symmetry σ, a set of constraints C is satisfiable iff $\sigma(C)$ is satisfiable.*

Proof: Suppose C is satisfiable. Then there exists a satisfying assignment A of C. By considering the action of a symmetry on a set of constraints, we can see that $\sigma(A)$ satisfies $\sigma(C)$. Thus $\sigma(C)$ is satisfiable. The proof reverses easily. □

4.2 Symmetry and solutions

We next consider the action of a symmetry on the solutions of a set of (possibly symmetry breaking) constraints. We write $sol(C)$ for the set of solutions to the set of constraints C.

Proposition 2. *For any symmetry σ and set of constraints C:*

$$sol(\sigma(C)) = \sigma(sol(C))$$

Proof: Consider any solution $A \in sol(\sigma(C))$. We view a solution as a set of assignments. Then $\sigma(C) \cup A$ is satisfiable. As σ is a bijection, there exists a unique B such that $A = \sigma(B)$. Thus $\sigma(C) \cup \sigma(B)$ is satisfiable. Hence $\sigma(C \cup B)$ is satisfiable. By Proposition 1, $C \cup B$ is satisfiable. That is, $B \in sol(C)$. Thus $\sigma(B) \in \sigma(sol(C))$. Hence $A \in \sigma(sol(C))$. The proof reverses directly. □

On the other hand, if we apply a symmetry to a set of constraints with that same symmetry, we do not change the set of solutions.

Proposition 3. *If σ is a symmetry of a set of constraints C then*

$$sol(C) = \sigma(sol(C))$$

Proof: Consider any $A \in \sigma(sol(C))$. Then there exists $B \in sol(C)$ such that $A = \sigma(B)$. Since σ is a symmetry of C, $\sigma(B) \in sol(C)$. That is $A \in sol(C)$. The proof reverses directly. □

4.3 Symmetries under symmetry

The action of a symmetry on a set of constraints also does not change the symmetries of those constraints.

Proposition 4. *If Σ is a symmetry group of a set of constraints C then Σ is also a symmetry group of $\sigma(C)$ for any $\sigma \in \Sigma$.*

Proof: Consider any solution A of $\sigma(C)$ and any $\tau \in \Sigma$. Since $\sigma \in \Sigma$, by Proposition 3, $sol(\sigma(C)) = sol(C)$. Thus $A \in sol(C)$. As $\tau \in \Sigma$, $\tau(A) \in sol(C)$. Hence $\tau(A) \in sol(\sigma(C))$. It follows that τ is a symmetry of $\sigma(C)$. □

4.4 Symmetry and soundness

An important property of a set of symmetry breaking constraints is its soundness. For a problem with symmetries Σ, a set of symmetry breaking constraints is *sound* iff it leaves at least one solution in each symmetry class. All the symmetry breaking constraints used in our running example are sound. The action of a symmetry on a set of symmetry breaking constraints leaves their soundness unchanged.

Proposition 5 (Soundness)**.** *Given a set of symmetries Σ of C, if S is a sound set of symmetry breaking constraints for Σ then $\sigma(S)$ for any $\sigma \in \Sigma$ is also a sound set of symmetry breaking constraints for Σ.*

Proof: Consider any $A \in sol(C \cup S)$ and any $\sigma \in \Sigma$. Now $A \in sol(C)$ and $A \in sol(S)$. But as σ is a symmetry of C, $\sigma(A) \in sol(C)$. Since $A \in sol(S)$, it follows from Proposition 2 that $\sigma(A) \in sol(\sigma(S))$. Thus, $\sigma(A) \in sol(C \cup \sigma(S))$. Hence, there is at least one solution left by $\Sigma(S)$ in every symmetry class of C. That is, $\sigma(S)$ is a sound set of symmetry breaking constraints for Σ. □

4.5 Symmetry and completeness

A set of symmetry breaking constraints may also be complete. For a problem with symmetries Σ, a set of symmetry breaking constraints is *complete* iff it leaves at most one solution in each symmetry class. The action of a symmetry on a set of symmetry breaking constraints leaves their completeness unchanged.

Proposition 6 (Completeness)**.** *Given a set of symmetries Σ of C, if S is a complete set of symmetry breaking constraints for Σ then $\sigma(S)$ for any $\sigma \in \Sigma$ is also a complete set of symmetry breaking constraints for Σ.*

Proof: Consider any $\sigma \in \Sigma$ and $A \in sol(C \cup \sigma(S))$. Now $A \in sol(C)$ and $A \in sol(\sigma(S))$. But as σ is a symmetry of C, so is σ^{-1}. Hence $\sigma^{-1}(A) \in sol(C)$. Since $A \in sol(\sigma(S))$, it follows from Proposition 2 that $\sigma^{-1}(A) \in sol(S)$. Thus $\sigma^{-1}(A) \in sol(C \cup S)$. Hence, there is at most one solution left by $\sigma(S)$ in every symmetry class of C). That is, $\sigma(S)$ is a complete set of symmetry breaking constraints for Σ. □

4.6 Representative solutions

Different symmetries of the symmetry breaking constraints pick out different solutions in each symmetry class. In fact, we can pick out any solution we like by choosing the appropriate symmetry of a set of symmetry breaking constraints.

Proposition 7. *Given a symmetry group 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 $A \in sol(C \cup \sigma(S))$.*

Proof: Since the set of symmetry breaking constraints S is sound, it leaves at least one solution (call it B) in the same symmetry class as A. That is, $B \in sol(C \cup S)$. Hence $B \in sol(C)$ and $B \in sol(S)$. As A and B are in the same symmetry class, there exists a symmetry σ in Σ with $A = \sigma(B)$. Since $B \in sol(S)$, it follows from Proposition 2 that $\sigma(B) \in sol(\sigma(S))$. That is, $A \in sol(\sigma(S))$. As $B \in sol(C)$ and $\sigma \in \Sigma$, it follows that $\sigma(B) \in sol(C)$. That is, $A \in sol(C)$. Hence $A \in sol(C \cup \sigma(S))$. □

We will use this result in the second half of the paper where we consider the conflict between symmetry breaking constraints and branching heuristics. We will exploit the fact that whatever solution the branching heuristic is going towards, there exists a symmetry of the symmetry breaking constraints which does not conflict with this.

4.7 Symmetries eliminated

In certain cases, a set of symmetry breaking constraints completely eliminates a symmetry. We say that a set of symmetry breaking 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 *eliminates* a symmetry σ iff for each solution A of $C \cup S$, $\sigma(A)$ is not a solution of $C \cup S$. Similarly, S breaks (eliminates) a set of symmetries Σ iff S breaks (eliminates) each $\sigma \in \Sigma$.

It is not hard to see that a sound and complete set of symmetry breaking constraints eliminates every non-identity symmetry. However, there are symmetry breaking constraints which break a particular symmetry but do not *eliminate* it.

Running Example. *Consider again one of the symmetry breaking constraints in (7):*

$$X_{1,n} < X_{n,1} \quad (9)$$

This eliminates the symmetry that reflects the magic square in its trailing diagonal. If we take any solution which satisfies (9), then any reflection of this solution in the trailing diagonal is removed by (9). Note that (9) also breaks the symmetry σ_{90} (90° clockwise rotation) since (2) satisfies (9) but σ_{90} of (2) does not. However, (9) does not eliminate σ_{90}. For example, both σ_{270} of (2) and σ_{90} of this solution satisfy (9). ♣

Applying a symmetry to a set of symmetry breaking constraints changes the solutions in each symmetry class accepted by the symmetry breaking constraints. However, it does not change the symmetries broken or eliminated by the symmetry breaking constraints.

Proposition 8. *Given a problem C with a symmetry group Σ, if S breaks (eliminates) Σ then $\sigma(S)$ breaks (eliminates) Σ for any $\sigma \in \Sigma$.*

Proof: Suppose S breaks Σ. Consider any symmetry $\tau \in \Sigma$. Then there exists a solution A of $C \cup S$ such that $\tau(A)$ is not a solution of $C \cup S$. As A is a solution of C and $\tau \in \Sigma$, $\tau(A)$ is a solution of C. Hence $\tau(A)$ is not a solution of S. By Proposition 2, $\sigma(\tau(A))$ is not a solution of $\sigma(S)$. Since Σ is a group, it is closed under composition. Thus $\tau \circ \sigma \circ \tau^{-1} \in \Sigma$. Hence, as $A \in sol(C)$, $\tau^{-1}(\sigma(\tau(A))) \in sol(C)$. Thus there is a solution of C, namely $\tau^{-1}(\sigma(\tau(A))) \in sol(C)$ such that the symmetry τ of this (which equals $\sigma(\tau(A))$) is not a solution of $\sigma(S)$. Hence, $\sigma(S)$ breaks τ. The proof for when S eliminates Σ follows similar lines. □

4.8 Symmetries of symmetry breaking constraints

We have discussed the action of a symmetry on a set of symmetry breaking constraints. But what can we say about the symmetries of a set of symmetry breaking constraints?

Proposition 9. *If the symmetry breaking constraints S break the symmetries Σ in the set of constraints C then S does not have any symmetry in Σ.*

Proof: Consider any symmetry $\sigma \in \Sigma$. Suppose S has this symmetry. Since S breaks σ there exists a solution A of $C \cup S$ such that $\sigma(A)$ is not a solution of S. Hence, A is a solution of S but $\sigma(A)$ is not. Thus, S does not have any symmetry in Σ. □

The reverse does not necessarily hold. There exist constraints which lack a symmetry which it is sound to post but which do not break that symmetry.

Running Example. *Consider again the most-perfect magic squares problem. Consider the following constraint:*

$$X_{1,1} + X_{n,n} = n^2 + 1 \rightarrow X_{1,1} < X_{n,n}$$

This does not have any variable symmetry since we cannot interchange the two variables. However, this constraint does not break any variable symmetry since $X_{1,1} + X_{n,n} \neq n^2 + 1$ in every most-perfect magic square. ♣

5 MODEL RESTARTS

The idea that the symmetries of symmetry breaking constraints can themselves be used to break symmetry can be used in several different ways. We consider here an application of this idea for tackling the conflict between branching heuristics and symmetry breaking constraints. Symmetry breaking picks out particular solutions in each symmetry class and these may not be the same solutions towards which branching heuristics are directing search. Heller *et al.* propose using *model restarts* [10] to tackle this conflict. Backtracking search is restarted periodically, using a new model which contains different symmetry breaking constraints. By posting different symmetry breaking constraints, we hope at some point for the branching heuristic and symmetry breaking not to conflict. Heller *et al.* do not, however, provide a general method to generate different symmetry breaking constraints after each restart.

Our observations that any symmetry acting on a set of symmetry breaking constraints can be used to break symmetry, and that different symmetries pick out different solutions, provide us with *precisely* the tool we need to perform model restarts to *any* domain (and not just to the domain of interchangeable variables and values studied in [10]). When we restart search, we simply post a different symmetry of the symmetry breaking constraints. We experimented with several possibilities. The simplest was to choose a symmetry at random from the symmetry group. We also tried various heuristics like using the symmetry most consistent or most inconsistent with previous choices of the branching heuristic. However, we observed the best performance of model restarts with a random choice of symmetry so we only report results here with such a choice. This is also algorithmically simple since computer algebra packages like GAP provide efficient algorithms for computing a random element of a group given a set of generators for the group.

Running Example. *We consider the simple problem of finding a magic square of order 5. The following table gives the amount of search needed to find such a magic square when posting one of the rotational symmetries of the symmetry breaking constraints and the default branching heuristic that labels variables in a fixed order. We encoded the problem in BProlog running on a Pentium 4 3.2 GHz processor with 3GB of memory. With magic squares in general, we cannot guarantee that the inverse of the smallest corner is not itself a corner value. We therefore relax the strict inequality in (6), replacing it by a non-strict inequality.*

Symmetry posted of (5) and (6)	Backtracks	Time to solve/s
σ_{id}	658	0.02
σ_{90}	17,143	0.36
σ_{180}	315,267	5.60
σ_{270}	18,808,974	408.85

We see that the different symmetries of the symmetry breaking constraints interact differently with the branching heuristic. Model restarts will help overcome this conflict. Suppose we restart search every 1,000 backtracks and choose to post at random one of these symmetries of (5) and (6). Let t be the average number of branches to find a solution. There is $\frac{1}{4}$ chance that the first restart will post σ_{id} of (5) and (6). In this situation, we find a solution after 658 backtracks. Otherwise we post one of the other symmetries of (5) and (6). We then explore 1,000 backtracks, reach the cutoff and fail to find a solution. As each restart is independent, we restart and explore on average another t more branches. Hence:

$$t \quad = \quad \frac{1}{4}658 + \frac{3}{4}(1000 + t)$$

Solving for t gives $t = 3,658$. Thus, using model restarts, we take just 3,658 backtracks on average to solve the problem. ♣

Note that posting random symmetries of the symmetry breaking constraints is not equivalent to fixing the symmetry breaking and randomly branching. Different symmetries of the symmetry breaking constraints interact in different ways with the problem constraints. Although the problem constraints are themselves initially symmetrical, branching decisions quickly break the symmetries that they have.

6 EXPERIMENTAL RESULTS

Our experiments are designed to test two hypotheses. The first hypothesis is that model restarts is less sensitive to branching heuristics than posting static symmetry breaking constraints. We test this hypothesis by using either a lexicographic value ordering (which does not conflict with the symmetry breaking constraints) or random value ordering, while using min-domain for variable ordering. We expect that model restarts will show smaller variation between the two value ordering heuristics. The second hypothesis is that model restarts will often explore a smaller search tree than dynamic methods like SBDS due to propagation of the symmetry breaking constraints. We tested two domains that exhibit two different kinds of symmetry: partial variable and value interchangability, and row and column symmetry.

We limit our comparison of dynamic methods to SBDS. Whilst there is a specialized dynamic symmetry breaking method for interchangeable variables and values, experiments in [10] show that this is several orders of magnitude slower than static methods. We also do not compare to methods such as GE-trees [17], as this method is limited to value symmetry and does not deal with the variable/value symmetries in our domains. Finally, we used SBDS to break just generators of the symmetry group as breaking the full symmetry group quickly exhausted available memory. We implemented model restarts and SBDS as well as all the static symmetry breaking constraints in Gecode 2.2.0, and ran all experiments on a 4-core Intel Xeon 5130 with 4MB of L2 cache running at 2GHz.

The first set of experiments uses random graph coloring problems generated in the same way as the previous experimental study in [13]. All values in this model are interchangeable. In addition, we introduce variable symmetry by partitioning variables into interchangeable sets of size at most 8. We randomly connect the vertices within each partition with either a complete graph or an empty graph, and choose each option with equal probability. Similarly, between any two partitions there is equal probability that the partitions are completely connected or independent. Results for graphs with 40 vertices are shown in Table 1.

The second set of experiments uses Equidistant Frequency Permutation Array (EFPA) problem [11]. Given the parameters v, q, λ, d, the objective is to find v codewords of length $q\lambda$, such that each word contains exactly λ occurences of each of q symbols and each pair of words have Hamming distance d. Our model has both row and column symmetry. We implement model restarts by randomly choosing a permutation of the rows and columns and posting lexicographic ordering constraints on the rows and columns of the resulting matrix.

The results support both our hypotheses. Although the model restarts method is not necessarily the best method for any given instance, its performance is most robust. The variability of the runtimes between the lex value ordering and random value ordering is much smaller for model restarts, as well as for SBDS. This suggests that in domains where the branching heuristic interacts more strongly with the problem constraints, model restarts will be more robust than static symmetry breaking constraints. Our second hypothesis, that model restarting tends to explore a smaller search tree than SBDS is also supported by the graph coloring results. SBDS was unable to prove optimality in all but one instance in graph coloring. These results confirm the findings of [10]. In EFPA on the other hand, there are enough solutions in these instances that applying little symmetry breaking with a random value ordering seems to be the best strategy. However, on the harder instances such as 4-6-4-5, using model restarts resolves the conflict between the branching heuristic and the symmetry breaking constraints and achieves parity with SBDS.

7 OTHER RELATED WORK

Crawford *et al.* proposed a general method to break symmetry statically using lex-leader constraints [2]. Like other static methods, the posted constraints pick out in advance a particular solution in each symmetry class. Unfortunately, this may conflict with the solution sought by the branching heuristic. There are a number of symmetry breaking methods proposed to deal with this conflict. For example, dynamic symmetry breaking methods like SBDS posts symmetry breaking constraints dynamically during search [7]. Another dynamic method for breaking symmetry is SBDD [3]. This checks if a node of the search tree is symmetric to some previously explored node. A weakness of such dynamic methods is that we get little or no propagation on the symmetry breaking constraints. It has been shown that propagation between the problem constraints and the static symmetry breaking constraints can reduce search exponentially [20].

Jefferson *et al,* have proposed GAPLex, a hybrid method that combines together static and dynamic symmetry breaking [12]. However, GAPLex is limited to dynamically posting lexicographical ordering constraints, and to searching with a fixed variable ordering. Puget has proposed "Dynamic Lex", a hybrid method that dynamically posts static symmetry breaking constraints during search which works with dynamic variable ordering heuristics [16]. This method adds symmetry breaking constraints dynamically during search that are compatible with the current partial assignment. In this way, the first solution found during tree search is not removed by symmetry breaking. Dynamic Lex needs to compute the stabilizers of the current partial assignment. This requires a graph isomorphism problem to be solved at each node of the search tree. Whilst Dynamic Lex works with dynamic variable ordering heuristics, it assumes that values are tried in order. Finally Dynamic Lex is limited to posting lexicographical ordering constraints. A comparison with Dynamic Lex is interesting but challenging. For instance, Heller *et al.* [10] did not compare model restarts with Dynamic Lex, arguing:

> *"It is not clear how this method [Dynamic Lex] can be generalized, though, and for the case of piecewise variable and value symmetry, no method with similar properties is known yet."*

8 CONCLUSIONS

We have considered the action of symmetry on symmetry breaking constraints. We proved that any symmetry applied to a set of symmetry breaking constraints gives a (possibly new) set of symmetry breaking constraints that break the same symmetries. In addition, we proved that different symmetries of the set of symmetry breaking constraints will pick out different solutions in each symmetry class. We used these observations to help tackle the conflict between symmetry breaking and branching heuristics. In particular, we applied these ideas to *model restarts* [10]. In this search technique, we periodically restart search with a new model which contains a random symmetry of the symmetry breaking constraints. Experimental results show that this helps keep many of the benefits of posting static symmetry breaking constraints whilst reducing the conflict between symmetry breaking and the branching heuristic. There are other potential applications of these ideas. For example, we are currently de-

	Static posting				Model restarts				SBDS			
	Lex		Random		Lex		Random		Lex		Random	
	opt	t/b	opt	t/b	opt	t/b	opt	t/b	opt	t/b	opt	t/b
1	**13**	**0.11**	**13**	114.99	**13**	1.13	**13**	4.99	13 *	0	14 *	270.42
		387		424 K		2087		13 K		3639 K		4309 K
2	**10**	**0.03**	**10**	354.72	**10**	2.67	-	-	**10**	0.04	**10**	0.77
		31		1364 K		6503		-		229		4539
3	12 *	0.02	-	-	**12**	**3.44**	**12**	137.55	12 *	0	12 *	0.02
		2052 K		-		**7958**		386 K		3178 K		2744 K
4	**14**	6.12	**14**	15.91	**14**	**5.14**	**14**	22.7	14 *	0	20 *	0.01
		25 K		76 K		**13 K**		54 K		3040 K		3188 K
5	**16**	**0.3**	-	-	**16**	26.16	**16**	6.06	16 *	0	16 *	0.01
		730		-		38 K		12 K		3193 K		3083 K
6	**13**	247.85	-	-	**13**	**115.41**	**13**	203.25	14 *	0	15 *	90.27
		1001 K		-		**410 K**		693 K		3313 K		3676 K
7	**8**	0.03	-	-	**8**	0.72	**8**	0.75	**8**	**0.02**	**8**	0.12
		60		-		1980		1794		119		684
8	**17**	**0.1**	-	-	**17**	20.35	**17**	97.9	17 *	0	17 *	0.01
		170		-		59 K		284 K		2820 K		2634 K
9	**20**	**0.22**	**20**	0.48	**20**	111.06	**20**	10.56	20 *	0.01	20 *	0.01
		387		1103		172 K		15 K		3490 K		3442 K
10	**8 ***	0.01	**8 ***	25.29	**8 ***	1.78	**8 ***	5.46	**8 ***	**0**	**8 ***	0.02
		4003 K		4612 K		3460 K		3510 K		3135 K		4309 K

Table 1. Static symmetry breaking constraints vs Model restarts and SBDS on Graph Coloring problems. "opt" is the quality of the solution found (* indicates optimality was not proved), "t" is the runtime in seconds, "b" is the number of backtracks. The best method for a problem instance is in **bold font**.

Instance	Static constraints				Model Restarts				SBDS			
	Lex		Random		Lex		Random		Lex		Random	
	t	b	t	b	t	b	t	b	t	b	t	b
3-3-4-5	**0.01**	**176**	**0.01**	246	0.02	271	0.2	3504	0.06	2060	**0.01**	318
3-4-6-5	0.02	599	0.04	726	0.04	703	0.59	7340	0.03	855	**0.01**	**465**
4-3-3-3	**0**	**34**	0.01	141	0.12	1570	0.08	1013	0.01	51	**0**	57
4-3-4-5	0.02	326	0.14	2392	0.28	3821	0.14	1629	0.16	4177	**0.01**	**42**
4-3-5-4	**0**	91	0.06	1204	0.12	1497	0.23	4055	**0**	52	**0**	**51**
4-4-4-5	0.07	1350	0.03	314	1.77	17 K	6.88	98 K	**0**	**41**	0.04	930
4-4-5-4	0.02	504	0.27	3956	1.01	8941	1.03	10 K	**0.01**	**208**	0.04	790
4-6-4-5	45.43	763 K	-	-	0.27	2612	0.67	5104	6.59	111 K	**0.04**	**629**
5-3-3-4	**0.01**	**190**	0.06	937	6.64	75 K	3.6	63 K	0.14	3113	0.02	382
5-3-4-5	**0.03**	**527**	1.51	21 K	4.34	45 K	1.32	13 K	1.16	26 K	0.12	2109
6-3-4-5	0.25	4538	2.84	40 K	3.9	44 K	9.06	106 K	5.92	114 K	**0.06**	**1380**

Table 2. Static symmetry breaking constraints vs Model restarts and SBDS on EFPA problems. "t" is the runtime in seconds, "b" is the number of backtracks. The best method for a problem instance is in **bold font**.

veloping methods for dynamically posting a symmetry of the symmetry breaking constraints which does not conflict with the branching heuristic. In the longer term, we would like to exploit symmetries of the nogoods learnt during search. Nogoods are themselves just constraints. We can therefore consider symmetries acting on them.

REFERENCES

[1] D. Cohen, P. Jeavons, C. Jefferson, K.E. Petrie, and B.M. Smith, 'Symmetry definitions for constraint satisfaction problems', *Constraints*, **11**(2–3), 115–137, (2006).

[2] J. Crawford, M. Ginsberg, G. Luks, and A. Roy, 'Symmetry breaking predicates for search problems', in *5th Int. Conf. on Knowledge Representation and Reasoning, (KR '96)*, pp. 148–159, (1996).

[3] T. Fahle, S. Schamberger, and M. Sellmann, 'Symmetry breaking', in *7th Int. Conf. on Principles and Practice of Constraint Programming (CP2001)*, pp. 93–107. (2001).

[4] P. Flener, A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, J. Pearson, and T. Walsh, 'Breaking row and column symmetry in matrix models', in *8th Int. Conf. on Principles and Practices of Constraint Programming (CP-2002)*. (2002).

[5] A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, and T. Walsh, 'Global constraints for lexicographic orderings', in *8th Int. Conf. on Principles and Practices of Constraint Programming (CP-2002)*. (2002).

[6] A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, and T. Walsh, 'Propagation algorithms for lexicographic ordering constraints', *Artificial Intelligence*, **170**(10), 803–908, (2006).

[7] I.P. Gent and B.M. Smith, 'Symmetry breaking in constraint programming', in *ECAI-2000*, pp. 599–603. (2000).

[8] I.P. Gent and T. Walsh, 'CSPLib: a benchmark library for constraints', Technical report, Technical report APES-09-1999, (1999). A shorter version appears in Proc. of CP-99.

[9] C. Gomes, B. Selman, and H. Kautz, 'Boosting combinatorial search through randomization', in *15th National Conf. on Artificial Intelligence*, pp. 431–437. (1998).

[10] D. Heller, A. Panda, M. Sellmann, and J. Yip, 'Model restarts for structural symmetry breaking', in *14th Int. Conf. on the Principles and Practice of Constraint Programming*, pp. 539–544, (2008).

[11] S. Huczynska, P. McKay, I. Miguel, and P. Nightingale, 'Modelling equidistant frequency permutation arrays: An application of constraints to mathematics', in *15th Int. Conf. on the Principles and Practice of Constraint Programming*, pp. 50–64. (2009).

[12] C. Jefferson, T. Kelsey, S. Linton, and K. Petrie, 'Gaplex: Generalised static symmetry breaking', in *6th Int. Workshop on Symmetry in Constraint Satisfaction Problems, held alongside CP-06*, (2006).

[13] Y.-C. Law, J. Lee, T. Walsh, and J. Yip, 'Breaking symmetry of interchangeable variables and values', in *13th Int. Conf. on Principles and Practices of Constraint Programming (CP-2007)*. (2007).

[14] Y.C. Law and J.M.H. Lee, 'Symmetry Breaking Constraints for Value Symmetries in Constraint Satisfaction', *Constraints*, **11**(2–3), 221–267, (2006).

[15] J.-F. Puget, 'On the satisfiability of symmetrical constrained satisfaction problems', in *Proc. of ISMIS'93*, pp. 350–361. (1993).

[16] J-F. Puget, 'Symmetry breaking using stabilizers', in *9th Int. Conf. on Principles and Practice of Constraint Programming (CP2003)*, (2003).

[17] C. Roney-Dougal, I. Gent, T. Kelsey, and S. Linton, 'Tractable symmetry breaking using restricted search trees', in *ECAI-2004*. (2004).

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

[19] T. Walsh, 'Symmetry breaking using value precedence', in *ECAI-2006*, pp. 168–172. (2006).

[20] T. Walsh, 'Breaking value symmetry', in *13th Int. Conf. on Principles and Practices of Constraint Programming (CP-2007)*. (2007).

[21] T. Walsh, 'Breaking value symmetry', in *Proc. of the 23rd National Conf. on AI*, pp. 1585–1588. AAAI, (2008).