Breaking symmetries in all different problems

Jean-François Puget
ILOG
9 Avenue de Verdun
94253 Gentilly Cedex, France
puget@ilog.fr

Abstract

Adding symmetry breaking constraints is one of the oldest ways of breaking variable symmetries for CSPs. For instance, it is well known that all the symmetries for the pigeon hole problem can be removed by ordering the variables. We have generalized this result to all CSPs where the variables are subject to an all different constraint. In such case it is possible to remove all variable symmetries with a partial ordering of the variables. We show how this partial ordering can be automatically computed using computational group theory (CGT). We further show that partial orders can be safely used together with the GE-tree method of [Roney-Dougal *et al.*, 2004]. Experiments show the efficiency of our method.

1 Introduction

A symmetry for a Constraint Satisfaction Problem (CSP) is a mapping of the CSP onto itself that preserves its structure as well as its solutions. If a CSP has some symmetries, it may be the case that all symmetrical variants of every dead end encountered during the search must be explored before a solution can be found. Even if the problem is easy to solve, all symmetrical variants of a solution are also solutions, and listing all of them may just be impossible in practice. Those observations have triggered a lot of interest for the detection and removal of symmetries in the constraint programming community. Adding symmetry breaking constraints is one of the oldest ways of breaking variable symmetries for (CSPs). For instance, it is shown in [Crawford *et al.*, 1996] that all variable symmetries could be broken by adding one lexicographical ordering constraint per symmetry. Unfortunately, this method is not tractable in general, as there may be an exponential number of symmetries. It has been shown that in general there is no way to break all symmetries of a problem with a polynomial number of constraints[Roy and Luks, 2004]. In [Flener *et al.*, 2002], a linear number of constraints are used to break symmetries for matrix problems. As expected, since there are a polynomial number of constraints, not all symmetries are broken. However, a polynomial number of constraints may be sufficient for breaking all symmetries in some special cases. For instance, in [Puget, 1993], it is shown that a linear number of constraints can break all symmetries for the pigeon hole problem: one simply needs to order the variables. In this paper we consider a more general class of problems: *all different problems*. These are CSPs such that the variables are subject to an all different constraint among other constraints. We show in section 4 that for such CSPs, all variable symmetries can be broken with at most $n-1$ binary constraints, where n is the number of variables.

In [Roney-Dougal *et al.*, 2004] a general purpose method for breaking all value symmetries is given: the GE-tree method. We show in section 5 that this method can be safely combined with symmetry breaking constraints, under some conditions on the order in which the search tree is traversed.

In section 6, we apply our method to some complex CSPs. We summarize our findings and discuss some possible generalizations in section 7.

2 Symmetries, Graphs and CSPs

The symmetries we consider are permutations, i.e. one to one mappings (bijections) from a finite set onto itself. Without loss of generality, we can consider permutations of I^n, where I^n is the set of integers ranging from 0 to $n-1$. For instance, we can label the variables of a graph with integers, such that any variable symmetry is completely described by a permutation of the labels of its variables. This is formalized as follows.

2.1 Computational Group Theory

Let S^n be the set of all permutations of the set I^n. The image of i by the permutation σ is denoted i^σ. A permutation $\sigma \in S^n$ is fully described by the vector $[0^\sigma, 1^\sigma, \dots, (n-1)^\sigma]$. The product of two permutations σ and θ is defined by $i^{(\sigma\theta)} = (i^\sigma)^\theta$.

Given $i \in I^n$ and a permutation group $G \subseteq S^n$, the *orbit* of i in G, denoted i^G, is the set of elements to which i can be mapped to by an element of G:

$$i^G = \{i^\sigma | \sigma \in G\}$$

Given $i \in I^n$ and a permutation group $G \subseteq S^n$, the *stabilizer* of i in G, denoted i_G, is the set of permutations of G that leave i unchanged:

$$i_G = \{\sigma \in G | i^\sigma = i\}$$

2.2 CSP and symmetries

A *constraint satisfaction problem* $\mathcal{P}$ (CSP) with n variables is a triple $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ where $\mathcal{V}$ is a finite set of variables $(v_i)_{i \in I^n}$, $\mathcal{D}$ a finite set of finite sets $(\mathcal{D}_i)_{i \in I^n}$, and every constraint in $\mathcal{C}$ is a subset of the cross product $\bigotimes_{i \in I^n} \mathcal{D}_i$. Without loss of generality, we can assume that $\mathcal{D}_i \subseteq I^k$ for some k.

An *assignment* is a member of $\mathcal{S}$, i.e. a vector of values $(a_i)_{i \in I^n}$ such that $a_i \in \mathcal{D}_i$ for all $i \in I^n$, and is denoted $(v_i = a_i)_{i \in I^n}$. A *partial assignment* is sub vector of an assignment.

A *solution* to $(\mathcal{V}, \mathcal{D}, \mathcal{C})$ is an assignment that is consistent with every member of $\mathcal{C}$.

Given a permutation σ of I^n, we define a variable permutation on (partial) assignments as follows:

$$((v_i = a_i)_{i \in I^n})^\sigma = ((v_{i^\sigma} = a_i)_{i \in I^n})$$

Such permutation is called a *variable symmetry* if it maps solutions to solutions.

Given a permutation θ of I^k, we define a value permutation on (partial) assignments as follow:

$$((v_i = a_i)_{i \in I^n})^\theta = ((v_i = a_i^{\theta^{-1}})_{i \in I^n})$$

Such permutation is called a *value symmetry* if it maps solutions to solutions.

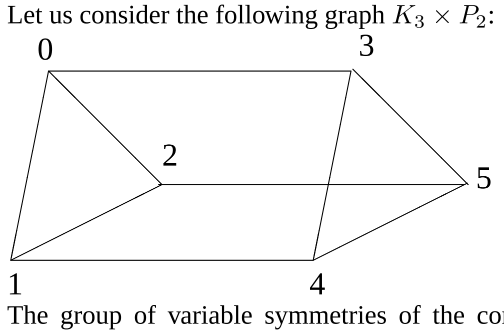
2.3 A graph coloring example

Let us introduce an example that will be used throughout the paper. We say that a graph with m edges is *graceful* if there exists a labeling f of its vertices such that:

- $0 \leq f(i) \leq m$ for each vertex i,
- the set of values $f(i)$ are all different,
- the set of values $abs(f(i), f(j))$ for every edge (i, j) are all different.

A straightforward translation into a CSP exists where there is a variable v_i for each vertex v_i, see [Lustig and Puget, 2001]. The variable symmetries of the problem are induced by the automorphism of the graph. There is one value symmetry, which maps v to $m - v$. More information on symmetries in graceful graphs is available in [Petrie and Smith, 2003], [Petrie, 2004].

Let us consider the following graph $K_3 \times P_2$:



The group of variable symmetries of the corresponding CSP is equivalent to the group of symmetries of the graph. Such group can be computed by packages such as Nauty[Mc Kay, 1981]. This group G is:

$$\{[0,1,2,3,4,5], [0,2,1,3,5,4], [1,0,2,4,3,5],$$
$$[1,2,0,4,5,3], [2,0,1,5,3,4], [2,1,0,5,4,3],$$
$$[3,4,5,0,1,2], [3,5,4,0,2,1], [4,3,5,1,0,2],$$
$$[4,5,3,1,2,0], [5,3,4,2,0,1], [5,4,3,2,1,0]\}$$

3 Breaking variable symmetries

Without loss of generality, we can assume that domains are subsets of I^k for some k, with the usual ordering on integers.

3.1 Lex leader constraints

Adding constraints is one of the oldest methods for reducing the number of variable symmetries of a CSP[Puget, 1993]. In [Crawford *et al.*, 1996], it is shown that all the variable symmetries of any CSP can be broken by the following constraints.

$$\forall \sigma \in G, \ \mathcal{V} \preceq \mathcal{V}^\sigma \tag{1}$$

For a given σ, the constraint $(\mathcal{V} \preceq \mathcal{V}^\sigma)$ is semantically equivalent to the disjunction of the constraints:

$$\begin{array}{c} v_0 < v_{0^\sigma} \\ v_0 = v_{0^\sigma} \wedge v_1 < v_{1^\sigma} \\ \vdots \\ v_0 = v_{0^\sigma} \wedge \ldots \wedge v_{i-1} = v_{(i-1)^\sigma} \wedge v_i < v_{i^\sigma} \\ \vdots \\ v_0 = v_{0^\sigma} \wedge \ldots \wedge v_{n-2} = v_{(n-2)^\sigma} \wedge v_{n-1} < v_{(n-1)^\sigma} \\ v_0 = v_{0^\sigma} \wedge \ldots \wedge v_{n-2} = v_{(n-2)^\sigma} \wedge v_{n-1} = v_{(n-1)^\sigma} \end{array}$$

If the last constraint is omitted, the set of constraints is denoted $\mathcal{V} \prec \mathcal{V}^\sigma$.

In our example, the constraints given by [Crawford *et al.*, 1996] are

$$\begin{array}{rcl} (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_0,v_1,v_2,v_3,v_4,v_5) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_0,v_2,v_1,v_3,v_5,v_4) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_1,v_0,v_2,v_4,v_3,v_5) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_1,v_2,v_0,v_4,v_5,v_3) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_2,v_0,v_1,v_5,v_3,v_4) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_2,v_1,v_0,v_5,v_4,v_3) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_3,v_4,v_5,v_0,v_1,v_2) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_3,v_5,v_4,v_0,v_2,v_1) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_4,v_3,v_5,v_1,v_0,v_2) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_4,v_5,v_3,v_1,v_2,v_0) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_5,v_3,v_4,v_2,v_0,v_1) \\ (v_0,v_1,v_2,v_3,v_4,v_5) & \preceq & (v_5,v_4,v_3,v_2,v_1,v_0) \end{array}$$

3.2 A polynomial number of constraints

The number of constraints (1) can grow exponentially with the number of variables $\mathcal{V}$. Using the fact that the variable are subject to an all different constraint, we can significantly reduce the number of symmetry breaking constraints. Let us consider one of the symmetries of our example, namely:

$$\sigma = [0,2,1,3,5,4]$$

The constraint breaking this symmetry is

$$(v_0, v_1, v_2, v_3, v_4, v_5) \preceq (v_0, v_2, v_1, v_3, v_5, v_4)$$

Since $v_0 = v_0$ is trivially true, and since $v_1 = v_2$ cannot be true because of the all different constraint, this constraint can be simplified into:

$$v_1 < v_2$$

This simplification is true in general and can be formalized as follows. Given a permutation σ, let $s(\sigma)$ be the smallest i such that $i^\sigma \neq i$, and let $t(\sigma)$ be equal to $(s(\sigma))^\sigma$.

Lemma 1. *Given a CSP where the variables $\mathcal{V}$ are subject to an all different constraint, and a variable symmetry group G for this CSP, then all variable symmetries can be broken by adding the following constraints:*

$$\forall \sigma \in G, v_{s(\sigma)} < v_{t(\sigma)} \tag{2}$$

Proof. By definition $k^\sigma = k$ for all $k < s(\sigma)$, and $s(\sigma)^\sigma \neq s(\sigma)$. Let us look at the constraint $\mathcal{V} \preceq \mathcal{V}_\sigma$. There is an all different constraint on the variables $\mathcal{V}$, which means that $v_i = v_{i^\sigma}$ if and only if $i^\sigma = i$. In particular, $v_k = v_{k^\sigma}$ for all $k < s(\sigma)$, and $v_{s(\sigma)} \neq v_{(s(\sigma))^\sigma}$. Therefore, only one disjunct for the constraint can be true, namely:

$$v_0 = v_{0^\sigma} \wedge \ldots \wedge v_{s(\sigma)-1} = v_{(s(\sigma)-1)^\sigma} \wedge v_{s(\sigma)} < v_{(s(\sigma))^\sigma}$$

Since $k^\sigma = k$ for $k < s(\sigma)$ and $s(\sigma)^\sigma = t(\sigma)$, this can be simplified into $v_{s(\sigma)} < v_{t(\sigma)}$. □

Note that if two permutations σ and θ are such that $s(\sigma) = s(\theta)$ and $t(\sigma) = t(\theta)$, then the corresponding symmetry breaking constraints are identical. Therefore, it is sufficient to state only one symmetry breaking constraints for each pair i, j such that there exists a permutation σ with $i = s(\sigma)$ and $j = t(\sigma)$.

The set of these pairs can be computed using what is known as the Schreier Sims algorithm [Seress, 2003]. This algorithm constructs a stabilizers chain $G_0, G_1, \ldots, G_n$ as follows:

$$\begin{aligned} G_0 &= G \\ \forall i \in I^n,\ G_i &= (i-1)_{G_{i-1}} \end{aligned}$$

By definition,

$$G_i = \{\sigma \in G : 0^\sigma = 1 \wedge \ldots \wedge (i-1)^\sigma = i-1\}$$
$$G_n \subseteq G_{n-1} \subseteq \ldots G_1 \subseteq G_0$$

The Schreier Sims algorithm also computes set of coset representatives U_i. Those are orbits of i in G_i:

$$U_i = i^{G_i}$$

By definition, U_i is the set of values which i is mapped to by all symmetries in G that leave at least $0, \ldots, (i-1)$ unchanged.

From now on, we will assume that all the groups we use are described by a stabilizers chain and coset representatives.

In our example, the stabilizer chain is :

$$\begin{aligned} G_0 &= G \\ G_1 &= 0_{G_0} = \{[0,1,2,3,4,5],[0,2,1,3,5,4]\} \\ G_2 &= 1_{G_1} = \{[0,1,2,3,4,5]\} \end{aligned}$$

All remaining stabilizers G_3, G_4, G_5 are equal to G_2.

Coset representatives are:

$$\begin{aligned} U_0 &= 0^{G_0} = \{0,1,2,3,4,5\} \\ U_1 &= 1^{G_1} = \{1,2\} \\ U_2 &= 2^{G_2} = \{2\} \\ U_3 &= 3^{G_3} = \{3\} \\ U_4 &= 4^{G_4} = \{4\} \end{aligned}$$

Theorem 2. *Given a CSP with n variables $\mathcal{V}$ such that there exists an all different constraint on these variables, and given coset representatives sets U_i for the variable symmetry group of the CSP, then all the variable symmetries can be broken by at most $n(n-1)/2$ binary constraints. These constraints are given by :*

$$\forall i \in I^n, \forall j \in U_i,\ i \neq j \rightarrow v_i < v_j \tag{3}$$

Proof. By definition, for each element $j \in U_i$, there exists at least one permutation $\sigma \in G_i$ such that $i^\sigma = j$ and $j = t(\sigma)$. The converse is also true. If there exists a permutation σ such that $i = s(\sigma)$ and that $j = t(\sigma)$, then $j \in U_i$. Therefore, the constraints (2) can be rewritten into:

$$\forall i \in I^n, \forall j \in U_i,\ i \neq j \Rightarrow v_i < v_j$$

There are $\sum_{i=0}^{n-1}(|U_i| - 1)$ such constraints. All the permutations of G_i leave the numbers $0, \ldots, i-1$ unchanged. Therefore U_i is a subset of $\{i, \ldots, n-1\}$. Then $|U_i| - 1 \leq n - i - 1$. Therefore, the number of constraints is bounded from above by $\sum_{i=0}^{n-1}(n-i-1) = n(n-1)/2$. □

In our example, these constraints are :

$$v_0 < v_1,\ v_0 < v_2,\ v_0 < v_3,\ v_0 < v_4,\ v_0 < v_5,\ v_1 < v_2$$

Note that some of these constraints are redundant. For instance, the constraint $v_0 < v_2$ is entailed by the first and the last constraints. This remark can be used to reduce the number of constraints as explained in the following section.

3.3 A linear number of constraints

The previous result can be improved by taking into account the transitivity of the $<$ constraints. Given $j \in I^n$, it may be the case that j belongs to several of the sets U_i. In such case, let us define $r(j)$ as the largest i different from j such that j belongs to U_i. If j belongs to no U_i other than U_j, then let $r(j) = j$.

Before stating our main result, let us prove the following.

Lemma 3. *With the above notations, if $j \in U_i$ and $i \neq j$ then $r(j) \in U_i$ and $r(j) < j$*

Proof. Let us assume that $j \in U_i$ and $i \neq j$. By definition of U_i there exists a permutation $\sigma \in G_i$ such that $i^\sigma = j$. Let $k = r(j)$. By definition of $r(j)$, $i \leq k$ and $j \in U_k$. Therefore, there exists a permutation $\theta \in G_k$ such that $k^\theta = j$. Let $\nu = \sigma\theta^{-1}$. Then, $i^\nu = i^{\sigma\theta^{-1}} = j^{\theta^{-1}} = k$. Moreover, $\nu \in G_i$ because $\sigma \in G_i$ and $\theta \in G_k \subseteq G_i$. Therefore,

$k \in U_i$. The fact that $r(j) < j$ is an immediate consequence of the definition of $r(j)$.□

We can now state our main result.

Theorem 4. *With the above notations, given a CSP with n variables $\mathcal{V}$, such that there exists an all different constraint on these variables, then all variable symmetries can be broken by at most $n-1$ binary constraints. These constraints are given by :*

$$\forall j \in I^n,\ r(j) \neq j \rightarrow v_{r(j)} < v_j \tag{4}$$

Proof. The number of constraints (4) is at most n by definition. Note that $r(0) = 0$ by definition of r, therefore, the number of constraints is at most $n-1$. Let us consider one of the constraints of (3). We are given i and j such that $j \in U_i$ and $i \neq j$. We want to prove that the constraint $c = (v_i < v_j)$ is implied by the constraints (4). Let us consider the sequence $(j, r(j), r(r(j)), r(r(r(j))), \ldots)$. Let us assume that the sequence never meets i. We have that $j \in U_i$ and $i \neq j$. By application of lemma 3, we get $r(j) \in U_i$ and $r(j) < j$. Since $r(j) \neq i$ by hypothesis, lemma 3 can be applied again. By repeated applications of lemma 3 we construct an infinite decreasing sequence of integers all included in U_i. This is not possible as U_i is finite. Therefore, there exists k such that $i = r^k(j)$. Moreover, we have established $r^k(j) \neq r^{k-1}(j), \ldots, r(r(j)) \neq r(j), r(j) \neq j$. Therefore, the constraints $v_{r^k(j)} < v_{r^{k-1}(j)}, \ldots v_{r(r(j))} < v_{r(j)}, v_{r(j)} < v_j$ are constraints of (4). Together they imply $v_{r^k(j)} < v_j$ which is the constraint c. We have proved that the constraints (3) are implied by the constraints (4). Since the set of constraints (4) is a subset of the constraints (3), both sets of constraints are equivalent. Then, by theorem 2, the constraints (4) break all variable symmetries. □

In our example, we get from coset representatives:

$$r(0) = 0,\ r(1) = 0,\ r(2) = 1,\ r(3) = 0,\ r(4) = 0,\ r(5) = 0$$

Therefore, the constraints (4) given by theorem 4 are:

$$v_0 < v_1,\ v_0 < v_3,\ v_0 < v_4,\ v_0 < v_5,\ v_1 < v_2$$

Note that the constraint $v_0 < v_2$ is no longer appearing.

4 Breaking both variable symmetries and value symmetries

In [Roney-Dougal *et al.*, 2004], a general method for breaking all value symmetries is described. This method uses the group of value symmetries of the CSP. We will show that this method can be combined with symmetry breaking constraints when there are both variable symmetries and value symmetries.

4.1 GE-tree and symmetry breaking constraints

We are given a CSP $\mathcal{P}$ with n variables v_i subject to an all different constraint among other constraints. Without loss of generality, we can assume that the domains of the variables are subsets of I^k for some k. It is shown in [Flener *et al.*, 2002] how to transform $\mathcal{P}$ into a new CSP $\mathcal{P}'$ such that all value symmetries of $\mathcal{P}$ become variable symmetries of $\mathcal{P}'$. The idea is to add $n \times k$ additional binary variables x_{ij} (variables with domains equal to $\{0,1\}$). We also add the following channeling constraints:

$$\forall i \in I^n, j \in I^k,\ (x_{ij} = 1) \equiv (v_i = j)$$

These constraints state that the variable x_{ij} equals 1 if and only if the variable v_j equals j. Adding these new variables do not change the solutions of the CSP. Moreover, variable symmetries of $\mathcal{P}$ are equivalent to permutations of the rows of the x_{ij} matrix, whereas value symmetries of $\mathcal{P}$ are equivalent to permutations of the columns of the same matrix.

Let us construct the vector X by concatenating the rows of the matrix x_{ij}. Therefore, the variables x_{ij} are ranked in increasing values of i then increasing values of j in the vector X.

Let us consider a value symmetry θ for $\mathcal{P}$. Then θ is a permutation of the matrix columns. This symmetry is broken by the constraint:

$$X \preceq X^\theta \tag{5}$$

Let X_i be the variables in the i-th row of the matrix. The value symmetry θ maps variables in a given row to variables in the same row. This is formalized as follows.

$$\begin{aligned} X_i &= (x_{i0}, x_{i1}, \ldots, x_{i(k-1)}) \\ (X^\theta)_i &= (x_{i0^{\theta^{-1}}}, x_{i1^{\theta^{-1}}}, \ldots, x_{i(k-1)^{\theta^{-1}}}) \end{aligned}$$

From the definition of $\preceq$, we have that (5) is equivalent to the disjunction of the following constraints:

$$\begin{gathered} X_0 \prec (X^\theta)_0 \\ X_0 = (X^\theta)_0 \wedge X_1 \prec (X^\theta)_1 \\ \vdots \\ X_0 = (X^\theta)_0 \wedge \ldots \wedge X_{i-1} = (X^\theta)_{i-1} \wedge X_i \prec (X^\theta)_i \\ \vdots \\ X_0 = (X^\theta)_0 \wedge \ldots \wedge X_{n-2} = (X^\theta)_{n-2} \wedge X_{n-1} \prec (X^\theta)_{n-1} \\ X_0 = (X^\theta)_0 \wedge \ldots \wedge X_{n-2} = (X^\theta)_{n-2} \wedge X_{n-1} = (X^\theta)_{n-1} \end{gathered}$$

Let us compare lexicographically X_i with $(X^\theta)_i$. Let a_i be the value assigned to v_i. Then $x_{i(a_i)} = 1$ and $x_{ij} = 0$ for $j \neq a_i$. Similarly, $x_{i(j)^{\theta^{-1}}} = 1$ if and only if $j^{\theta^{-1}} = a_i$, i.e. $a_i^\theta = j$. Therefore, $X_i = (X^\theta)_i$ if and only if $a_i = (a_i)^\theta$, and $X_i \prec (X^\theta)_i$ if and only if $a_i < (a_i)^\theta$.

We then have that (5) is equivalent to the disjunction of the constraints:

$$\begin{gathered} a_0 < (a_0)^\theta \\ a_0 = (a_0)^\theta \wedge a_1 < (a_1)^\theta \\ \vdots \\ a_0 = (a_0)^\theta \wedge \ldots \wedge a_{i-1} = (a_{i-1})^\theta \wedge a_i < (a_i)^\theta \\ \vdots \\ a_0 = (a_0)^\theta \wedge \ldots \wedge a_{n-2} = (a_{n-2})^\theta \wedge a_{n-1} < (a_{n-1})^\theta \\ a_0 = (a_0)^\theta \wedge \ldots \wedge a_{n-2} = (a_{n-2})^\theta \wedge a_{n-1} = (a_{n-1})^\theta \end{gathered}$$

Let us now consider one of the disjunct, namely:

$$a_0 = (a_0)^\theta \wedge \ldots \wedge a_{i-1} = (a_{i-1})^\theta \wedge a_i < (a_i)^\theta$$

This means that θ leaves invariant $a_0, a_1, \ldots, a_{i-1}$. In such case a_i must be minimal among the values that any such θ can map it to. We have therefore proved the following result.

Lemma 5. *With the above notations, a_i is the minimum of its orbit in the group of symmetries that leave $a_0, a_1, \ldots a_{i-1}$ unchanged.*

This is equivalent to the GE-tree method for breaking all value symmetries [Roney-Dougal *et al.*, 2004], when the variables and the values are tried in an increasing order during search.

From [Crawford *et al.*, 1996], it is safe to add all possible symmetry breaking constraints (1) on $\mathcal{P}'$. In particular, it is safe to state all the constraints (1) for the variable symmetries of $\mathcal{P}$ together with all the constraints (5). By lemma 5, the set of constraints (5) is equivalent to the GE-tree method for breaking value symmetries. We have just proved the following result.

Theorem 6. *Given a CSP, its group of variable symmetries G_1, and its group of value symmetries G_2, then the combination of the GE-tree method for breaking value symmetries with the symmetry breaking constraints (1) computes a set of solutions $\mathcal{S}$ such that:*

$$\forall S \in sol(\mathcal{P}), \exists \sigma \in G_1, \exists \theta \in G_2, \exists S' \in \mathcal{S},\ S^{\sigma\theta} = S'$$

Theorem 4 in section 3 says that the set of all those constraints (1) is equivalent to the constraints (4) when there is an all different constraints on all the variables $\mathcal{V}$. This yields the following result.

Corollary 7. *Given a CSP where the variable are subject to an all different constraint, its group of variable symmetries G_1, and its group of value symmetries G_2, then the combination of the GE-tree method for breaking value symmetries with the symmetry breaking constraints (4) computes a set of solutions $\mathcal{S}$ such that:*

$$\forall S \in sol(\mathcal{P}), \exists \sigma \in G_1, \exists \theta \in G_2, \exists S' \in \mathcal{S},\ S^{\sigma\theta} = S'$$

5 Experimental results

We have implemented an algorithm similar to Nauty[Mc Kay, 1981] for computing graph automorphisms, as well as a Schreier Sims algorithm[Seress, 2003]. These have been used in the following examples. In our implementation, we did not fully implement the GE-tree method, because it requires more computational group algorithms than what we have implemented so far. We simply compute the orbits for the group G of value symmetries. Then, only the minimum element of each orbit is left in the domain of the variable v_0. We will refer to this method as SBC (for Symmetry Breaking Constraints) in order to differentiate it from other methods.

5.1 Graceful graphs

We have tested our approach on the graceful graphs of [Petrie and Smith, 2003]. Variable symmetries are broken by the constraints (4). There is one non trivial value symmetry, which maps a to $e - a$; where e is the number of edges of the graph. Therefore, the orbits for this symmetry are the sets $\{a, e - a\}$, for $0 \leq a \leq e/2$. Therefore, one can restrict the domain of v_0 by keeping one the smallest value in each of these orbits.

For each graph we report the number of solutions of the CSP (sol), the size of the search tree (node) and the time (time) needed to compute all these solutions the running time without symmetry breaking technique (no sym). We also report these figures when the SBC method is used. In this case the running time includes the time needed to perform all the group computations. Running times are measured on a 1.4 GHz Dell Latitude D800 laptop running Windows XP. The implementation is done with ILOG Solver 6.0[ILOG, 2003].

graph	no sym			SBC		
	sol	node	time	sol	node	time
$K_3 \times P_2$	96	1518	0.12	8	83	0.01
$K_4 \times P_2$	1440	216781	13.6	30	1863	0.27
$K_5 \times P_2$	480	34931511	4454	2	53266	6.5
$K_6 \times P_2$				0	1326585	305

Table 1. Computing all solutions for graceful graphs.

The running times are up to 30 times smaller than the ones reported in [Petrie, 2004] for the GAP-SBDD and the GAP-SBDS methods, using a computer about half the speed of ours. This shows that in this example our symmetry breaking constraints are much more efficient than modified search methods. However, we find twice as many more solutions. Let see why on the graph $K_5 \times P_2$. This graph has 10 vertices and 25 edges. We list the values for the variables $v_0, v_1, \ldots, v_9$ for the two solutions:

$$(0, 4, 18, 19, 25, 23, 14, 6, 3, 1)$$

$$(0, 6, 7, 21, 25, 24, 22, 19, 11, 2)$$

Let us apply the non trivial value symmetry to the second one. We get:

$$(25, 19, 18, 4, 0, 1, 3, 6, 14, 23)$$

Let us apply the following variable symmetry to it:

$$[4, 3, 2, 1, 0, 9, 8, 7, 6, 5]$$

This yields the first solution!

This example shows that we did not break all symmetries that are a product of a variable symmetry by a value symmetry. This is so despite the fact that all variable symmetries and all value symmetries are broken.

5.2 Most Perfect Magic Squares

Most perfect magic squares, studied in [Ollerenshaw, 1986], are given as an example of a CSP with convoluted variables symmetries in [Roney-Dougal *et al.*, 2004]. The authors decided to use an inverse representation in order to transform variable symmetries into value symmetries. These were in turn broken with the GE-tree method. In [Ollerenshaw, 1986], it is proven that most perfect magic squares are in a one to one relationship with *reversible squares*. A reversible square of size $n \times n$ (where $n \equiv 0 \bmod 4$) has entries $1 \dots n^2$ such that (i) the sum or the two entries at diagonally opposite corners of any rectangle or sub-square equals the sum of the other pair of diagonally opposite corners (ii) in each row or column, the sum of the first and last entries equals the sum of the next and the next to last number, etc (iii) diametrically opposed numbers sum to $n^2 + 1$.

Any solution is one of $2^{n+1}((n/2)!)^2$ symmetric equivalent[Ollerenshaw, 1986]. For $n = 16$, this is about 2.13e+14.

The natural model for this problem has one variable per cell in the square with entries as values. In addition to the above constraints on entries, there is an all different constraint. Therefore, our variable symmetry breaking constraints can be used. We report for various sizes the time used to compute the symmetry breaking constraints as well as the time for finding all non symmetrical solutions with our SBC method. We also report the results of [Roney-Dougal *et al.*, 2004], obtained with GAP-SBDD and with GE-tree on a computer about half the speed of ours. A direct comparison is difficult because they directly search for most perfect magic squares whereas we search for reversible squares. It is worth comparing the time spent in the symmetry computations though, because these deal with the same symmetry group. Our method spends much less time in symmetry computations because this needs to be done only once, before the search starts.

Method	n	sols	sym	search
SBC	4	3	0.01	0.02
	8	10	0.09	0.39
	12	42	0.44	22.2
	16	35	4.6	275.6
GAP-SBDD	4	3	0.3	0.3
	8	10	5.4	125.4
	12	42	2745	12518
GE-tree	4	3	0.2	0.1
	8	10	0.7	90.0
	12	42	29.1	10901.8

6 Discussion

We have established two major results (i) all variable symmetries can be broken by a linear number of binary constraints if there is an all different constraints on all the variables of the CSP (ii) symmetry breaking constraints of [Crawford *et al.*, 1996] can be safely used in conjunction with the GE-tree method of [Roney-Dougal *et al.*, 2004].

Furthermore, these methods can be fully automated using automorphism packages such as Nauty[Mc Kay, 1981] and computational group theory[Seress, 2003]. We have implemented such algorithms. Experiments on complex problems show that these algorithms are quite efficient.

The results described in this paper can be generalized. First of all, theorem 4 is valid for all CSPs where the variables are subject to an all different constraint. It would be interesting to see if similar results can be obtained for other forms of CSPs.

It is worth mentioning that we presented a method for breaking all variable symmetries, and all value symmetries. However, our method does not break products of both kinds of symmetries. It remains to be seen if a simple combination of variable and value symmetry breaking techniques can break all such symmetries.

Acknowledgements

The author would like to thank Marie Puget and the anonymous referees for their remarks. It greatly helped improving the readability of this paper.

References

[Crawford *et al.*, 1996] Crawford, J., Ginsberg, M., Luks E.M., Roy, A. “Symmetry Breaking Predicates for Search Problems.” In proceedings of KR’96, 148-159.

[Flener *et al.*, 2002] P. Flener, A. M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, J. Pearson, T. Walsh.: “Breaking Row and Column Symmetries in Matrix Models. “ Proceedings of CP’02, pages 462-476, 2002

[ILOG, 2003] ILOG: ILOG Solver 6.0. User Manual. ILOG, S.A., Gentilly, France, Septembre 2003

[Lustig and Puget, 2001] Lustig, I.J., and Puget, J.F. (2001). ”Program Does Not Equal Program: Constraint Programming and its Relationship to Mathematical Programming,” Interfaces 31(6), 29-53.

[Mc Kay, 1981] Mc Kay, B.: “Practical Graph Isomorphism” Congr. Numer. 30, 45-87, 1981

[Ollerenshaw, 1986] Dame Ollenrenshaw, K. “On most perfect or complete 8x8 pandiagonal magic squares” Proceedings Royal Society London, 407, 259-281, 1986.

[Petrie and Smith, 2003] Petrie, K., Smith, B.M. : “Symmetry breaking in graceful graphs.” In proceedings of CP’03, LNCS 2833, 930-934, Springer Verlag, 2003.

[Petrie, 2004] Petrie, K. : ”Combining SBDS and SBDD” Technical report APES-86-2004. Available from http://www.dcs.st-and.ac.uk/˜apes/apesreports.html”

[Puget, 1993] Puget, J.-F.: “On the Satisfiability of Symmetrical Constraint Satisfaction Problems.” Proceedings of ISMIS’93 (1993), 350–361.

[Roney-Dougal *et al.*, 2004] Roney-Dougal C.M., Gent, I.P., Kelsey T., Linton S.: “Tractable symmetry breaking using restricted search trees” To appear in proceedings of ECAI’04.

[Roy and Luks, 2004] Roy. A., Luks, E.: “The complexity of symmetry-breaking formulas”, *Annals of Mathematics and Artificial Intelligence* , 41 (2004), 19-45 (with A. Roy).

[Seress, 2003] Seress, A.: *Permutation Group Algorithms* Cambrige University Press, 2003.