Sets of Symmetry Breaking Constraints

Barbara M. Smith
Cork Constraint Computation Centre, University College Cork, Ireland
b.m.smith@4c.ucc.ie

Abstract

[Puget, 2004] has shown that if the symmetry in a constraint satisfaction problem acts only on the variables and there is also an allDifferent constraint on the variables, the symmetry can be eliminated by adding a small number of constraints, linear in the number of variables. In this paper, Puget's procedure for finding a set of symmetry-breaking constraints is extended to find all possible distinct sets. It is shown that there can be exponentially many distinct sets of symmetry breaking constraints, leading to the need to choose between them. The choice depends on how the problem will be solved, and specifically on the variable order. Since a variable order can lead to a choice of symmetry-breaking constraints, it seems plausible that the same variable order should be used during search; however, experiments with a graceful graph problem do not show that pairing the symmetry-breaking constraints with a compatible variable order in this way leads to reduced search.

1 Introduction

If the symmetries of a CSP permute the variables, and not the values, there is a systematic procedure for generating symmetry-breaking constraints, given by Crawford, Ginsberg, Luks and Roy [Crawford *et al.*, 1996]. This procedure requires an order of the variables to be specified, and the symmetry breaking constraints that are derived depend on this order; hence in theory, if there are n variables, there could be $n!$ different sets of symmetry-breaking constraints. In practice, this does not happen; different variable orders can give exactly the same constraints, or different orders can yield equivalent constraints. Taking this into account, there may be only a few distinct sets of symmetry-breaking constraints that we can consider adding to the CSP, or perhaps only one.

This is true in general of these kinds of symmetry and symmetry-breaking constraints. In this paper, the question of how many distinct sets of symmetry-breaking constraints there can be is investigated, specifically for problems where there is an allDifferent constraint on the variables. Puget has shown [Puget, 2004] that in such a case, the symmetry can be broken with a small number of constraints (linear in the number of variables) and gives a procedure for generating them from the symmetry group. Here, the procedure is adapted to generate all the symmetrically distinct sets of constraints. As in [Puget, 2004], instances of the 'graceful graph' problem are used as examples, since these do have variable symmetry and an allDifferent constraint on the variables. Graceful graph problems also have a value symmetry, but for the present purposes that is ignored.

2 Graceful Graphs

A labelling f of the nodes of a graph with q edges is *graceful* if f assigns each node a unique label from $\{0, 1, ..., q\}$ and when each edge xy is labelled with $|f(x) - f(y)|$, the edge labels are all different. (Hence, the edge labels are a permutation of 1, 2, ..., q.) [Gallian, 2003] gives a survey of graceful graphs, i.e. graphs with a graceful labelling, and lists the graphs whose status is known.

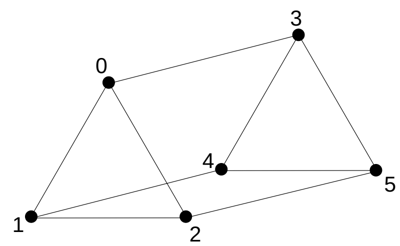

Figure 1: The graph $K_3 \times P_2$: the node numbers correspond to the indices of the CSP variables $x_0, x_1, ..., x_5$.

Figure 1 shows an example of a graph that has a graceful labelling. This graph is an instance of the class of graphs $K_m \times P_n$, consisting of n copies of a m-clique (K_m), with corresponding nodes in each clique joined by a path of length n (P_n). A possible model of this problem as a CSP has a variable for each node, in this case $x_0, x_1, ..., x_5$, corresponding to the node numbering given in Figure 1. The domain of x_i is $\{0..., 8\}$, since there are 9 edges. These are the search variables. We also have a variable for each edge, in this case $d_0, .., d_8$, with domains $\{1..., 9\}$. If edge k joins nodes i and j, we have a constraint $d_k = |x_i - x_j|$. The variables $x_0, x_1, ..., x_5$ are all different, as are $d_0, .., d_{14}$.

As discussed in [Petrie and Smith, 2003], symmetry in the resulting CSP arises in two ways. First there is the symmetry of the graph: in this case, we can permute the nodes within any clique as long as we permute the nodes in the other cliques in the same way; and we can reverse the order of the cliques, so that in this case, the first clique (nodes 0, 1, 2) swaps with the second clique (nodes 3, 4, 5). Second, in any assignment to the node variables, we can replace any value v by $q - v$; this symmetry will be ignored in this paper. For $K_m \times P_n$ graphs, we can eliminate the graph symmetry by adding constraints on the node variables to the model.

3 Examples of Symmetry Breaking Constraints

In this section, two examples are given to show that different variable orders do not necessarily yield different constraints, but may do.

Suppose that we have n variables and the symmetry group acting on them is S_n, i.e all permutations. A graceful graph problem with this symmetry is the problem of finding graceful labellings of a clique. It has been known for a long time, and is easy to show, that the clique K_n is graceful iff $n \leq 4$, so that solving the corresponding CSP is not very interesting in itself; nevertheless, it provides a good illustration.

The CSP model has a variable for each node, $x_0, x_1, ..., x_{n-1}$. If the procedure given in [Crawford *et al.*, 1996] is applied to this problem, taking the variables in lexicographic order, we get a set of symmetry-breaking constraints that can be simplified to $x_0 < x_1 < x_2 ... < x_{n-1}$. (The simplification involves dropping constraints that are implied by combinations of other constraints, through transitivity, e.g. $x_0 < x_2$.)

If we started from another variable order, say $x_{n-1}, x_{n-2}, ..., x_2, x_1, x_0$, we would get a different set of constraints, i.e. $x_{n-1} < x_{n-2} < ... < x_1 < x_0$. However, these constraints are equivalent to the first set: an element of S_n acts on the order $x_0, x_1, ..., x_{n-1}$ to produce $x_{n-1}, x_{n-2}, ..., x_2, x_1, x_0$ and also transforms the first set of constraints into the second. A search for a solution to the first CSP (i.e. with the first set of constraints added) could be transformed into an equivalent search in the second CSP, with the second set of constraints, by applying the same symmetry to the assignments made. Hence, although the $n!$ possible variable orders each yield a different set of symmetry breaking constraints, all these sets are effectively the same.

Now consider the problem of finding graceful labellings of the graph shown in Figure 1. The symmetry of the graph allows the labels of the nodes in each 3-clique to be permuted as long as the labels in the other clique are permuted in the same way; we can also swap the labels of the nodes in one clique with the corresponding nodes in the other clique; and we can combine these transformations. The symmetry group has 12 elements (3! $\times$ 2).

The variable order $x_0, x_1, ..., x_5$ would give the symmetry breaking constraints $x_0 < x_1; x_0 < x_2; x_0 < x_3; x_0 < x_4; x_0 < x_5$ and $x_1 < x_2$. In fact, any variable order in which x_0 and x_1 are placed first (as well as some other orders) will give exactly the same constraints. Also, as with labelling K_n, there are variable orders which will give different but equivalent constraints, e.g. $x_6, x_5, .., x_0$. However, starting from a variable order that placed x_0 and x_4 first, we would get the constraints $x_0 < x_1; x_0 < x_2; x_0 < x_3; x_0 < x_4; x_0 < x_5$ and $x_4 < x_5$. These constraints are different from the first set; there is no element of the symmetry group which will transform one into the other. Abstracting from the specific naming of the nodes, the first set of constraints says that an arbitrary node of the six is constrained to have the smallest label, and an arbitrary ordering is imposed on the labels of the other two nodes in the *same* 3-clique. The other set of constraints similarly selects a specific node to have the smallest label, but imposes an arbitrary ordering on the two nodes in the *other* clique which are not connected to the first node.

The next section discusses deriving the distinct sets of symmetry-breaking constraints by extending the method given in [Puget, 2004].

4 Deriving Symmetry-Breaking Constraints

Finding symmetry-breaking constraints using the method described in [Crawford *et al.*, 1996] requires, in theory, writing down the effect of every element of the symmetry group. (In practice, they are often derived more intuitively.) The method given in [Puget, 2004] is specialised for problems where there is an allDifferent constraint on the variables and in that case gives the same results as [Crawford *et al.*, 1996]. It deals with the original symmetry group and subgroups of that, rather than the individual elements of the group. Puget's method can be used to derive the constraints automatically; however, even if the constraints are derived by hand, it is much less time-consuming than that in the earlier paper.

Recall that the symmetry group in a graceful graph problem acts only on the variables of the CSP (ignoring the complement symmetry). The first step in Puget's method is to partition the variables into orbits: the orbit of a point acted on by a group is the set of points that it can be transformed into by the group elements. In finding graceful labellings of both K_n and $K_3 \times P_2$, any node can be transformed into any other node by the graph symmetry, so the initial partition has just one set, containing all the variables.

Next, we choose a (non-singleton) orbit, if there is more than one, and pick an arbitrary variable in that orbit, say x_0. We impose the constraints that x_0 should be less than any other variable in its orbit.

Next we find the stabiliser of x_0; this is the subgroup of the original symmetry group that leaves x_0 fixed. Intuitively, it is easy to do this in the graceful graphs instances; we can for instance imagine fixing node 0 in Figure 1 and see what transformations of the graph are still possible. Clearly, with node 0 immovable, we can still swap x_1 and x_2 and simultaneously x_4 and x_5. The stabiliser therefore has just two elements, including the identity transformation that does nothing.

We again partition the variables into their orbits in the stabiliser of x_0. Since x_0 is by definition fixed by its stabiliser, its orbit just contains itself, and some other variables may also be fixed by the stabiliser of x_0 (in the case of $K_3 \times P_2$, x_3 is also fixed by the stabiliser of x_0.) The other orbits in this example are $\{x_1, x_2\}$ and $\{x_4, x_5\}$. Again, we choose an orbit and pick an element in the orbit, say x_1, and impose the constraint $x_1 < x_2$.

We repeat these steps, finding the stabiliser of x_0 and x_1 (or the stabiliser of x_1 within the stabiliser of x_0, which amounts to the same thing). In the example of $K_3 \times P_2$, if we fix nodes 0 and 1, the whole graph is fixed, so the new stabiliser contains just the identity, and this is the signal to terminate the process.

If we select x_0 from the first orbit, and then choose the orbit $\{x_1, x_2\}$ and x_1 within that, we get the first set of symmetry-breaking constraints given earlier for $K_3 \times P_2$, i.e. $x_0 < x_1; x_0 < x_2; x_0 < x_3; x_0 < x_4; x_0 < x_5$ and $x_1 < x_2$. However, if we chose the orbit $\{x_4, x_5\}$ in the second stage, and x_4 within that, the last constraint would be replaced by $x_4 < x_5$, i.e. the second set of constraints given earlier.

We can therefore see that:

- the choice of one variable rather than another within an orbit will give different sets of symmetry-breaking constraints, but the different choices in this case are symmetrically equivalent, since the group (either the original group or the current stabiliser) acts equally on all elements within an orbit;
- the choice of one orbit rather than another will give different sets of symmetry-breaking constraints which are symmetrically distinct, since the group has a different effect on each orbit.

The process gives the same symmetry-breaking constraints as in [Crawford *et al.*, 1996] if we use a pre-specified order of the variables to guide the choice of orbit and the variable within the orbit. In the $K_3 \times P_2$ example, any variable order that has x_0 first would lead to choosing x_0 from the first orbit. Any variable order that has either x_1 or x_2 before both x_4 and x_5 would choose the orbit $\{x_1, x_2\}$ rather than $\{x_4, x_5\}$; and if the order had x_1 before x_2 it would choose $x_1 < x_2$ rather than $x_2 < x_1$. Hence, many different variable orders would yield exactly the same set of constraints in this example.

5 Alternative Sets of Constraints

We can represent schematically the symmetrically distinct choices, that are given by different orbits, and the constraints that they lead to, as shown in the example below, which shows the derivation of the two distinct sets of symmetry-breaking constraints for $K_3 \times P_2$. (Note that the elements of the original group and the stabilisers are not listed, except when the current stabiliser contains only the identity element, shown as $\{i\}$.)

G_0: symmetry group of $K_3 \times P_2$
orbit: $\{x_0, x_1, x_2, x_3, x_4, x_5\}$
constraints: $x_0 < x_1; x_0 < x_2; x_0 < x_3; x_0 < x_4;$
$x_0 < x_5$
G_1: stabilizer of x_0
orbit 1: $\{x_1, x_2\}$
constraints: $x_1 < x_2$
G_2: stabiliser of $x_0, x_1 = \{i\}$

$x_0 < x_1;\quad x_0 < x_2;\quad x_0 < x_3;\quad x_0 < x_4;$
$x_0 < x_5;\quad x_1 < x_2$

orbit 2: $\{x_4, x_5\}$
constraints: $x_4 < x_5$
G_2: stabiliser of $x_0, x_4 = \{i\}$

$x_0 < x_1;\quad x_0 < x_2;\quad x_0 < x_3;\quad x_0 < x_4;$
$x_0 < x_5;\quad x_4 < x_5$

Returning to the clique, K_n, we get a single set of symmetry breaking constraints, apart from symmetric equivalents. At each stage, the stabiliser of the elements fixed so far is the complete symmetry group, S_k, acting on the remaining k elements, and these elements can all be transformed into each other by the action of the stabiliser, so that there is a single orbit.

G_0: symmetry group of K_n = S_n
orbit: $\{x_0, x_1, ..., x_n\}$
constraints: $x_0 < x_1; x_0 < x_2; .....; x_0 < x_n$
G_1: stabilizer of x_0 = S_{n-1} acting on $x_1, x_2, ..., x_n$
orbit: $\{x_1, x_2, ..., x_n\}$
constraints: $x_1 < x_2; x_1 < x_3; ....; x_1 < x_n$
G_2: stabiliser of $x_0, x_1 = S_{n-2}$ acting on $x_2, x_3, ..., x_n$
orbit: $\{x_2, ..., x_n\}$
constraints: $x_2 < x_3; x_2 < x_4; ....; x_2 < x_n$

and so on. As mentioned earlier, the resulting constraints can be simplified into $x_0 < x_1 < x_2 < x_3.... < x_n$.

A more interesting case is $K_3 \times P_3$, shown in Figure 2. In

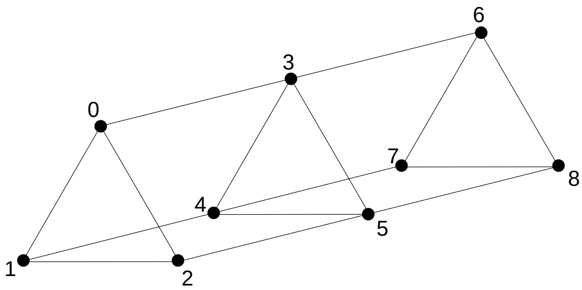

Figure 2: The graph $K_3 \times P_3$

this case, the original symmetry group of the graph partitions the variables into two different orbits, since the nodes in the two outer 3-cliques are interchangeable, as are the nodes in the middle 3-clique, but an outer node cannot be mapped to a middle node. In all, we get 9 distinct sets of symmetry constraints:

G_0: symmetry group of $K_3 \times P_3$
orbit 1: $\{x_0, x_1, x_2, x_6, x_7, x_8\}$
constraints: $x_0 < x_1; x_0 < x_2; x_0 < x_6; x_0 < x_7;$
$x_0 < x_8$
G_1: stabilizer of x_0
orbit 1: $\{x_1, x_2\}$
constraints: $x_1 < x_2$
G_2: stabiliser of $x_0, x_1 = \{i\}$

$x_0 < x_1;\quad x_0 < x_2;\quad x_0 < x_6;\quad x_0 < x_7;$
$x_0 < x_8;\quad x_1 < x_2$

orbit 2: $\{x_4, x_5\}$
constraints: $x_4 < x_5$
G_2: stabiliser of $x_0, x_4 = \{i\}$

$x_0 < x_1;\quad x_0 < x_2;\quad x_0 < x_6;\quad x_0 < x_7;$
$x_0 < x_8;\quad x_4 < x_5$

orbit 3: $\{x_7, x_8\}$
constraints: $x_7 < x_8$
G_2: stabiliser of $x_0, x_7 = \{i\}$

$x_0 < x_1;\quad x_0 < x_2;\quad x_0 < x_6;\quad x_0 < x_7;$
$x_0 < x_8;\quad x_7 < x_8$

orbit 2: $\{x_3, x_4, x_5\}$
constraints: $x_3 < x_4; x_3 < x_5$
G_1: stabilizer of x_3
orbit 1: $\{x_0, x_6\}$
constraints: $x_0 < x_6$
G_2: stabiliser of x_3, x_0
orbit 1: $\{x_1, x_2\}$
constraints: $x_1 < x_2$
G_3: stabiliser of $x_3, x_0, x_1 = \{i\}$

$x_3 < x_4;\quad x_3 < x_5;\quad x_0 < x_6;\quad x_1 < x_2$

orbit 2: $\{x_4, x_5\}$
constraints: $x_4 < x_5$
G_3: stabiliser of $x_3, x_0, x_4 = \{i\}$

$x_3 < x_4;\quad x_3 < x_5;\quad x_0 < x_6;\quad x_4 < x_5$

orbit 3: $\{x_7, x_8\}$
constraints: $x_7 < x_8$
G_3: stabiliser of $x_3, x_0, x_7 = \{i\}$

$x_3 < x_4;\quad x_3 < x_5;\quad x_0 < x_6;\quad x_7 < x_8$

orbit 2: $\{x_1, x_2, x_7, x_8\}$
constraints: $x_1 < x_2; x_1 < x_7; x_1 < x_8$
G_2: stabiliser of $x_3, x_1 = \{i\}$

$x_3 < x_4;\quad x_3 < x_5; x_1 < x_2;\quad x_1 < x_7;$
$x_1 < x_8$

orbit 3: $\{x_4, x_5\}$
constraints: $x_4 < x_5$
G_2: stabiliser of x_3, x_4
orbit 1: $\{x_0, x_6\}$
constraints: $x_0 < x_6$
G_3: stabiliser of $x_3, x_4, x_0 = \{i\}$

$x_3 < x_4; x_3 < x_5; x_4 < x_5; x_0 < x_6$

orbit 2: $\{x_1, x_7\}$
constraints: $x_1 < x_7$
G_3: stabiliser of $x_3, x_4, x_1 = \{i\}$

$x_3 < x_4; x_3 < x_5; x_4 < x_5; x_1 < x_7$

orbit 3: $\{x_2, x_8\}$
constraints: $x_2 < x_8$
G_3: stabiliser of $x_3, x_4, x_2 = \{i\}$

$x_3 < x_4; x_3 < x_5; x_4 < x_5; x_2 < x_8$

Note that although ten sets of constraints are derived above, the 5th and 8th sets are identical.

6 How Many Sets Can There Be?

Puget showed that variable symmetry in a problem with an allDifferent constraint on the affected variables can be broken by a linear number of binary $<$ constraints. The examples above might suggest that the number of distinct sets of symmetry-breaking constraints might be similarly limited. However, it is not; the following example shows that there can be exponentially many sets. $K_m \times P_2$ is the general class of which $K_3 \times P_2$ is a small example. Under the original symmetry group, all the nodes form a single orbit, but at every later stage, there are two possible orbits, one corresponding to each clique, and hence two choices for the next constraints to add. The following shows the first few levels.

G_0: symmetry group of $K_m \times P_2$
orbit: $\{x_0, x_1, x_2, ..., x_{2m-1}\}$
constraints: $x_0 < x_1; x_0 < x_2; ...; x_0 < x_{2m-1}$
G_1: stabilizer of x_0
orbit 1: $\{x_1, x_2, ..., x_{m-1}\}$
constraints: $x_1 < x_2, x_1 < x_3, ..., x_1 < x_{m-1}$
G_2: stabiliser of x_0, x_1
orbit 1: $\{x_2, x_3, ..., x_{m-1}\}$
constraints: $x_2 < x_3; x_2 < x_4; ..., x_2 < x_{m-1}$
......
orbit 2: $\{x_{m+2}, x_{m+3}, ..., x_{2m-}\}$
constraints: $x_{m+2} < x_{m+3}; x_{m+2} < x_{m+3};$
$...; x_{m+1} < x_{2m-1}$
......
orbit 2: $\{x_{m+1}, x_{m+2}, ..., x_{2m-1}\}$
constraints: $x_{m+1} < x_{m+2}; x_{m+1} < x_{m+2};$
$...; x_{m+1} < x_{2m-1}$
G_2: stabiliser of x_0, x_{m+1}
orbit 1: $\{x_2, x_3, ..., x_{m-1}\}$
constraints: $x_2 < x_3; x_2 < x_4; ..., x_2 < x_{m-1}$
......
orbit 2: $\{x_{m+2}, x_{m+3}, ..., x_{2m-1}\}$
constraints: $x_{m+2} < x_{m+3}; x_{m+2} < x_{m+3};$
$...; x_{m+1} < x_{2m-1}$
......

So in constructing a set of symmetry-breaking constraints, we can flip backwards and forwards between the two cliques, adding constraints between the variables of either clique.

At each point where there is a choice between two orbits, the first constraint resulting from either choice will never arise if the other orbit is chosen instead. For instance, when there is a choice between the orbits $\{x_1, x_2, ..., x_{m-1}\}$ and $\{x_{m+1}, x_{m+2}, ..., x_{2m-1}\}$, we get a constraint $x_1 < x_2$ in the first case, and $x_{m+1} < x_{m+2}$ in the second. Since x_1 and x_{m+1} have the same stabiliser within the stabiliser of x_0, neither variable appears in any subsequent non-singleton orbit along either branch, hence no further constraints involving these variables can arise. Depending on the later choices, some of the constraints added at this point may later become redundant due to transitivity: for instance, if we choose $\{x_1, x_2, ..., x_{m-1}\}$ at this stage and $\{x_2, x_3, ..., x_{m-1}\}$ at the next, the constraints $x_1 < x_3, ..., x_1 < x_{m-1}$ are subsumed by $x_1 < x_2, x_2 < x_3; x_2 < x_4; ..., x_2 < x_{m-1}$. But the first constraint added at each stage will not become redundant, and as already shown, it would never arise if the alternative choice of orbit were made. Hence every set of symmetry-breaking constraints, resulting from choosing between two orbits at each stage of the algorithm, is different in at least one constraint from any other set resulting from making different choices. There are $m - 2$ levels of binary choice between two orbits (the last choice is between $\{x_{m-2}, x_{m-1}\}$ and $\{x_{2m-2}, x_{2m-1}\}$) and hence 2^{m-2} different sets of symmetry-breaking constraints can be derived. (As already shown, $K_3 \times P_2$ has 2 distinct sets of symmetry-breaking constraints.)

7 Which Constraints To Choose?

If there is more than one set of possible symmetry-breaking constraints, which should we choose? This is not a question that has hitherto been much considered. If users derive symmetry breaking constraints systematically, they are likely to start from a lexicographic order based on some numbering of the variables which reflects in some way the structure of the problem, and find just one set of constraints. It is now clear that there can be many distinct sets of symmetry breaking constraints, and choosing arbitrarily may not lead to the most efficient way of solving the problem at hand.

A further complication is that the symmetry breaking constraints interact with the search strategy, so that neither can be chosen independently. It can be expected that if the variable ordering used during the search for solutions is incompatible with the symmetry-breaking constraints, in some sense, finding solutions can be delayed rather than made more efficient. The aim in adding symmetry-breaking constraints to the CSP is to forbid all but one solution from each symmetry equivalence class; the variable ordering also induces an order on the solutions in each equivalence class. If the only solution allowed by the constraints appears late in the induced order of symmetrically equivalent solutions, then search effort may be wasted in considering partial solutions that could lead to complete solutions to the original problem but are now excluded by the symmetry-breaking constraints. In this way, adding symmetry-breaking constraints could be counter-productive if used with an incompatible variable ordering.

Since the procedure for deriving symmetry-breaking constraints in [Crawford *et al.*, 1996] requires a variable order to be specified (as does the procedure described earlier), it is generally felt that the same order should be used for the search, if a static variable order is to be used, or as a tie-breaker if a dynamic order such a smallest domain is used. In this section, the sets of symmetry-breaking constraints derived earlier for the $K_3 \times P_3$ problem are investigated, and their interaction with a static order. Two questions of interest are: is it possible to decide which symmetry-breaking constraints are best, and is it true that the variable order used to derive the constraints should also be used for the search?

For each of the nine representative sets of symmetry-breaking constraints in this problem, we can find a variable order that would give that set. In the results given below, these variable orders are used as static variable orders during the search for all solutions. The variable orders found are, of course, not unique; here, the lexicographically smallest variable order has been chosen. Choosing a different variable order might give a more efficient search. Nevertheless, the results may give some indications of the answers to these questions.

The sets of constraints as given earlier are:

- A: $x_0 < x_1; x_0 < x_2; x_0 < x_6; x_0 < x_7; x_0 < x_8;$ $x_1 < x_2$
- B: $x_0 < x_1; x_0 < x_2; x_0 < x_6; x_0 < x_7; x_0 < x_8;$ $x_4 < x_5$
- C: $x_0 < x_1; x_0 < x_2; x_0 < x_6; x_0 < x_7; x_0 < x_8;$ $x_7 < x_8$
- D: $x_3 < x_4; x_3 < x_5; x_0 < x_6; x_1 < x_2$
- E: $x_3 < x_4; x_3 < x_5; x_0 < x_6; x_4 < x_5$
- F: $x_3 < x_4; x_3 < x_5; x_0 < x_6; x_7 < x_8$
- G: $x_3 < x_4; x_3 < x_5; x_1 < x_2; x_1 < x_7; x_1 < x_8$
- H: $x_3 < x_4; x_3 < x_5; x_4 < x_5; x_1 < x_7$
- I: $x_3 < x_4; x_3 < x_5; x_4 < x_5; x_2 < x_8$

Variable orders that would lead to the constraints A, for instance, must have x_0 as the first variable, so that we choose the first orbit ($\{x_0, x_1, x_2, x_6, x_7, x_8\}$), at the top level rather than the second ($\{x_3, x_4, x_5\}$), and x_0 rather than x_1, x_2, x_6, x_7 or x_8 from the first orbit. It must then have x_1 before x_2, x_4, x_5, x_7 or x_8, so that we choose the first orbit as the second level, and x_1 rather than x_2. A possible order is x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, but many other orders would give the same constraints, and many more would give equivalent constraints.

Nine possible variable orders, giving the nine sets of symmetry-breaking constraints, have been derived. These have been used as static variable orders in the search for graceful labellings of $K_3 \times P_3$. Tables 1 to 4 show the results (using ILOG Solver 6.0). For space reasons, the variable orders are given as lists of subscripts, so that 0, 1, 2, 3, 4, 5, 6, 7, 8 represents $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$. Each variable order is labelled A to I, according to the corresponding set of symmetry breaking constraints. In Tables 1 and 3, both allDifferent constraints have been treated as sets of binary $\neq$ constraints; in Tables 2 and 4, bounds consistency has been maintained on the allDifferent constraints on the edge variables. (For this problem, bounds consistency is almost as effective in pruning variable domains as generalized arc consistency, and is much faster.) The final variable order included in the tables, $x_3, x_4, x_5, x_0, x_1, x_2, x_6, x_7, x_8$, is one that a user might choose for this problem, since it reflects the structure of the graph: the first three variables are the most constrained (before adding symmetry-breaking constraints), and it makes intuitive sense to instantiate all the variables in

Variable order	Symmetry-breaking Constraints A	B	C	D	E	F	G	H	I
0, 1, 2, 3, 4, 5, 6, 7, 8 (A)	3,434	885	2,904	6,202	4,939	4,566	975	836	975
0, 4, 1, 2, 3, 5, 6, 7, 8 (B)	273	273	353	797	377	495	323	873	1,797
0, 7, 1, 2, 3, 4, 5, 6, 8 (C)	2,008	3,999	2,074	7,991	2,827	7,133	169	169	456
3, 0, 1, 2, 4, 5, 6, 7, 8 (D)	41	294	36	294	256	1,296	41	36	1,240
3, 0, 4, 1, 2, 5, 6, 7, 8 (E)	1,006	545	1,809	545	260	1,381	1.006	1,809	1,753
3, 0, 7, 1, 2, 4, 5, 6, 8 (F)	20	1,450	20	1.450	716	2,946	20	20	1,879
3, 1, 0, 2, 4, 5, 6, 7, 8 (G)	42	840	37	141	141	399	314	315	314
3, 4, 1, 0, 2, 5, 6, 7, 8 (H)	3,336	2,087	6,398	34	26	116	34	231	1,328
3, 4, 2, 0, 1, 5, 6, 7, 8 (I)	4,113	2,053	7,105	111	55	138	14	154	1,462
3, 4, 5, 0, 1, 2, 6, 7, 8	5,018	2,989	8,436	1,581	1,493	2,187	726	1,519	3,095

Table 1: Number of backtracks to find a graceful labelling of $K_3 \times P_3$, treating the allDifferent constraint on the edge variables as binary $\neq$ constraints.

Variable order	Symmetry-breaking Constraints A	B	C	D	E	F	G	H	I
0, 1, 2, 3, 4, 5, 6, 7, 8 (A)	12	12	12	41	160	27	27	23	27
0, 4, 1, 2, 3, 5, 6, 7, 8 (B)	75	75	54	323	155	215	117	309	550
0, 7, 1, 2, 3, 4, 5, 6, 8 (C)	65	123	68	2,739	17	2,309	86	86	53
3, 0, 1, 2, 4, 5, 6, 7, 8 (D)	16	134	15	134	112	306	16	15	239
3, 0, 4, 1, 2, 5, 6, 7, 8 (E)	200	69	363	69	66	236	200	363	384
3, 0, 7, 1, 2, 4, 5, 6, 8 (F)	9	560	9	560	362	723	9 9	9	576
3, 1, 0, 2, 4, 5, 6, 7, 8 (G)	42	840	37	48	48	152	75	60	75
3, 4, 1, 0, 2, 5, 6, 7, 8 (H)	733	316	1,339	14	11	51	14	100	484
3, 4, 2, 0, 1, 5, 6, 7, 8 (I)	459	338	1,487	49	26	56	5	66	483
3, 4, 5, 0, 1, 2, 6, 7, 8	252	158	226	39	147	162	116	101	589

Table 2: Number of backtracks to find a graceful labelling of $K_3 \times P_3$, maintaining bounds consistency on the allDifferent constraint on the edge variables.

a clique before moving to another clique. (This variable order would give rise to the symmetry-breaking constraints E, if used for that purpose.)

There are several things that can be noticed from the tables. The most obvious is that search effort does vary with both variable order and symmetry-breaking constraints. However, it is difficult to discern any consistent pattern in any of the tables. Significantly, there is no evidence that a set of symmetry-breaking constraints does relatively better with a variable order that could give rise to those constraints than with another variable order. Tables 1 and 2 have been included because one might expect that the link, if any, between a set of symmetry breaking constraints and the corresponding variable order would be stronger than when finding all solutions, but it is still not apparent.

In Tables 3 and 4, bounds consistency affects the relative ranking of the sets of constraints. Constraints D and G give the best results for several variable orders in Table 3, i.e. for finding all solutions with $\neq$ constraints. However, if bounds consistency is maintained, constraints A and B do much better. Note that the first three sets of constraints force node 0 to have the smallest label. In consequence, it must be labelled 0, but this is not easily discovered from the $\neq$ constraints.

Eight of the nine sets of constraints form three overlapping groups. Sets A, B and C are identical except for the final constraint ($x_1 < x_2$; $x_4 < x_5$; $x_7 < x_8$ respectively). For most of the selected variable orders, A is better than B, which is better than C; A is consistently better than C for all orders in both tables.

Sets D, E, F also match except for the same constraints ($x_1 < x_2$; $x_4 < x_5$; $x_7 < x_8$ respectively). In this case, bounds consistency changes the ranking order: in Table 3, D and E are both better than F, and D is usually better than E; however, with bounds consistency, D becomes worse than the other two, and E and F are broadly similar in performance.

Finally, E, H and I are the same in ordering the node labels of the middle clique, and have an additional constraint ($x_0 < x_6$; $x_1 < x_7$; $x_2 < x_8$ respectively).With $\neq$ constraints, E is consistently better than H and I, and H is usually better than I, except for the orders that have x_7 early, although these are not good orders for any set of constraints. With bounds consistency, I is better than E for all the orders; perhaps in this case, the constraint $x_7 < x_8$ allows useful inferences to be made about the third clique, while node labels in the other cliques are being assigned.

Overall, the picture is quite confusing. It is evidently not possible to choose a best set of symmetry breaking constraints without taking into account the constraint propagation that will act on these constraints (and the others already in the problem). If the allDifferent constraint is treated as a $\neq$ con-

Variable order	Symmetry-breaking Constraints A	B	C	D	E	F	G	H	I
0, 1, 2, 3, 4, 5, 6, 7, 8 (A)	124,694	140,890	207,994	85,478	113,629	136,843	82,323	138,530	148,077
0, 4, 1, 2, 3, 5, 6, 7, 8 (B)	125,103	139,348	210,131	77,245	103,295	127,583	84,370	139,990	149,995
0, 7, 1, 2, 3, 4, 5, 6, 8 (C)	238,390	354,736	336,589	351,703	225,775	401,824	278,133	407,995	297,546
3, 0, 1, 2, 4, 5, 6, 7, 8 (D)	125,220	142,211	209,668	85,880	114,355	137,805	82,494	139,432	148,769
3, 0, 4, 1, 2, 5, 6, 7, 8 (E)	125,308	141,862	213,354	75,465	102,865	128,765	83,288	136,571	151,150
3, 0, 7, 1, 2, 4, 5, 6, 8 (F)	242,143	366,689	348,953	363,602	260,900	424,596	284,224	420,820	300,730
3, 1, 0, 2, 4, 5, 6, 7, 8 (G)	125,206	142180	209,742	86,076	114,552	137,983	82,664	139,660	148,850
3, 4, 1, 0, 2, 5, 6, 7, 8 (H)	124,264	141,597	213,233	75,552	102,639	128,646	83,189	136,748	150,260
3, 4, 2, 0, 1, 5, 6, 7, 8 (I)	137,990	151,185	222,969	84,106	113,097	135,150	87,162	148,109	153,809
3, 4, 5, 0, 1, 2, 6, 7, 8	141,992	140,712	228,760	70,263	97,132	124,893	82,980	121,352	151,333

Table 3: Number of backtracks to find all graceful labellings of $K_3 \times P_3$, treating the allDifferent constraint on the edge variables as binary $\neq$ constraints.

Variable order	Symmetry-breaking Constraints A	B	C	D	E	F	G	H	I
0, 1, 2, 3, 4, 5, 6, 7, 8 (A)	13,026	14,477	19,947	32,609	32,145	32,262	30,238	47,694	21,582
0, 4, 1, 2, 3, 5, 6, 7, 8 (B)	16,083	14,266	25,739	30,908	29,465	27,398	31,789	47, 458	23,436
0, 7, 1, 2, 3, 4, 5, 6, 8 (C)	41,913	45,394	60,349	78,571	54,644	65,372	61,039	68,409	36,745
3, 0, 1, 2, 4, 5, 6, 7, 8 (D)	12,803	14,045	19,530	33,797	32,465	32,702	30,938	48,923	21,427
3, 0, 4, 1, 2, 5, 6, 7, 8 (E)	14,382	12,953	21,556	33,904	31,222	30,938	33,964	50,975	22,167
3, 0, 7, 1, 2, 4, 5, 6, 8 (F)	38,488	25,199	56,735	83,383	56,735	64,113	62,637	58,455	37,025
3, 1, 0, 2, 4, 5, 6, 7, 8 (G)	12,838	14,235	19,608	33,707	32,547	32,764	31,004	48,959	21,268
3, 4, 1, 0, 2, 5, 6, 7, 8 (H)	15,834	13,482	22,155	33,078	32,217	30,367	33,729	50,372	21,989
3, 4, 2, 0, 1, 5, 6, 7, 8 (I)	12,375	13,799	19,618	37,231	34,895	33,426	35,283	58,496	27,991
3, 4, 5, 0, 1, 2, 6, 7, 8	10,552	14,086	21,501	31,569	28,741	25,943	31,478	36,322	20,880

Table 4: Number of backtracks to find all graceful labellings of $K_3 \times P_3$, maintaining bounds consistency on the allDifferent constraint on the edge variables.

straint, sets D and G do well (for these variable orders), which suggests that heterogeneous constraints, that involve different parts of the problem, are a good choice. On the other hand, if bounds consistency is maintained, constraints A and B do better. We might conclude that constraints that allow strong conclusions to be drawn (e.g. that node 0 must be labelled 0) are a good choice, provided that the level of propagation does allow the conclusion.

The tables also allow some comparison of variable orders for this problem (although note that the variable orders presented here are relatively good ones; many orders that are much worse than any of the orders in the tables can be found). The best results overall, for finding all solutions, are from the final variable order, $x_3, x_4, x_5, x_0, x_1, x_2, x_6, x_7, x_8$. This was also the best performance found, out of many other variable orders not shown in the tables, although a systematic investigation has not been practicable. Again, this variable order gives best results with constraints D if bounds consistency is maintained; otherwise, constraints A are best. This variable order is, however, rather poor for finding just one solution. Similarly, of the variable orders selected for the tables, C and F are clearly much worse than others, at least for finding all solutions. These both assign a value to x_7 early in the search, immediately after x_0 and before x_1. Since x_7 is not directly connected to x_0 and x_1, this order could be expected to lead to wasted search. Orders C and F do not do better with constraints C and F respectively than they do otherwise, so again there is little evidence here that a set of symmetry breaking constraints should be used with a variable order that would give rise to it. However, the link with the variable ordering seems far from straightforward, at least in this problem, and warrants further investigation.

8 Conclusions

The procedure described in [Puget, 2004] for devising symmetry breaking constraints when the symmetry acts only on the variables and there is an allDifferent constraint on the variables, has been extended to find all distinct sets of symmetry-breaking constraints. It is shown, using graceful graphs problems, that there can be exponentially many such sets. This leads to the need to choose between them. The choice is complicated by the fact that the search performance resulting from choosing a set of such constraints is affected by the search strategy. It has been recognised that if there is a conflict between the symmetry-breaking constraints and the variable order, then the search effort may increase rather than decrease as a result of symmetry breaking. A plausible assumption is that, since the choice of symmetry-breaking constraints requires choosing one variable rather than another, the same choices should be reflected in the variable order.

Unfortunately, experiments with finding graceful labellings of $K_3 \times P_3$ do not support this assumption; the results do not show a reduction in search effort from pairing a set of symmetry breaking constraints with a compatible variable order in this way. The best choice of symmetry breaking constraints for this problem depends on the level of constraint propagation that will be maintained on all the constraints during search, and also on the variable order. The results suggest that heterogeneous constraints involving different parts of the problem are best if the allDifferent constraint is treated as $\neq$ constraints, whereas if bounds consistency is maintained on this constraint, a set of constraints that will allow constraint propagation to set the value of one of the variables is better.

In some problems, although not the graceful graphs problems considered here, symmetry-breaking constraints allow the derivation of implied constraints that can further reduce search. Frisch, Jefferson and Miguel [Frisch *et al.*, 2004] suggest that when there is a choice between distinct sets of symmetry-breaking constraints, the choice could be guided by considering the implied constraints and their potential effect on the search.

The experiments reported here use static search variable orders. Investigating the interaction between dynamic variable ordering heuristics such as smallest domain and symmetry-breaking constraints would be more complicated, and probably even more confusing. For graceful graphs problems, smallest domain ordering has sometimes proved to be significantly worse than a static ordering. Nevertheless, there may be problem classes where we want to combine symmetry-breaking constraints with dynamic variable ordering. A possibility is to choose the symmetry-breaking constraints during search: the first variable, say x_0, can be determined before the search starts, its orbit found and the appropriate symmetry-breaking constraints imposed; after making an assignment to x_0, the heuristic chooses the second variable, say x_1, and symmetry breaking constraints are again imposed, and so on [1].

However, even if we confine ourselves to static variable orders, further investigation is needed into the interaction between variable order and symmetry-breaking constraints and how to choose between different sets of symmetry-breaking constraints.

References

[Crawford *et al.*, 1996] J. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry-Breaking Predicates for Search Problems. In *Proceedings KR'96*, pages 149–159, November 1996.

[Frisch *et al.*, 2004] A.M. Frisch, C. Jefferson, and I. Miguel. Symmetry-breaking as a Prelude to Implied Constraints: A Constraint Modelling Pattern. In *Proceedings of ECAI 2004,* pages 171–175, 2004.

[Gallian, 2003] J. A. Gallian. A Dynamic Survey of Graph Labeling. *The Electronic Journal of Combinatorics*, (DS6), 2003.

[Petrie and Smith, 2003] K. E. Petrie and B. M. Smith. Symmetry Breaking in Graceful Graphs. Technical Report APES-56-2003, APES Research Group, January 2003.

[Puget, 2004] J.-F. Puget. Breaking symmetries in all different problems. In *Proceedings of SymCon04, the 4th International Workshop on Symmetry in Constraints*, 2004.

[1] This was suggested by one of the reviewers.