Increasing Symmetry Breaking by Preserving Target Symmetries

Jimmy Lee and Jingying Li

Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, N.T.,Hong Kong
{jlee,jyli}@cse.cuhk.edu.hk

Abstract. Breaking the exponential number of all symmetries of a constraint satisfaction problem is too costly. In practice, we often aim at breaking a subset of the symmetries efficiently, which we call target symmetries. In static symmetry breaking, the goal is to post a set of constraints to break these target symmetries in order to reduce the solution set and thus also the search space. Symmetries of a problem are all intertwined. A symmetry breaking constraint intended for a particular symmetry always breaks more than just the intended symmetry as a side-effect. Different constraints for breaking the same target symmetry can have different side-effects. Conventional wisdom suggests that we should select a symmetry breaking constraint that has more side-effects by breaking more symmetries. While this wisdom is valid in many ways, we should be careful where the side-effects take place. A symmetry is preserved by a constraint iff its symmetry classes are either entirely removed from the solution set or retained as solutions by the constraint. We give theorems and examples to demonstrate that it is beneficial to post symmetry breaking constraints that preserve the target symmetries and restrict the side-effects to only non-target symmetries as much as possible. The benefits are in terms of the number of symmetries broken and the extent to which a symmetry is broken (or eliminated), resulting in a smaller solution set and search space. Extensive experiments are also conducted to confirm the feasibility and efficiency of our proposal empirically.

1 Introduction

Symmetries are common in Constraint Satisfaction Problems. Several methods [9, 4] are proposed to avoid the exploration of search space segments with assignments that can be generated by representatives of symmetry classes. One common way is to add dedicated constraints statically at the modeling stage to eliminate symmetries [22], such as the LEXLEADER method [3]. A symmetry breaking constraint leaves only canonical solutions with their symmetrically-equivalent solutions eliminated. It prunes the search space in two ways: remove the symmetrically equivalent search branches, and trigger constraint propagation with other constraints and *vice versa*. They affect both the size of the solution set and the search tree of the problem.

There are some tractable classes of symmetries [7] and also methods to simplify the constraints [21]. However, in general more symmetry breaking constraints need to be posted in order to eliminate more symmetries. When the propagation overhead of symmetry breaking constraints outweighs the time saved in exploring the search space, there is no longer better efficiency [20]. Eliminating all symmetries are too costly. Usually we only eliminate a subset of them, which we call *target symmetries*. Jefferson *et al*. [15] has given algorithms to generate a good set of target symmetries. While posting constraints for only the target symmetries, we show we can actually eliminate more symmetries and achieve a smaller solution set by carefully choosing the constraints.

When and how the choice of symmetry breaking constraints will affect the number of solutions are discussed systematically. We point out how the side-effects of symmetry breaking constraints in breaking other symmetries are common. To eliminate the same symmetry set, we have alternative choices which cause different side-effects, breaking or eliminating extra symmetries. We formally define the situation in which a symmetry is not removed by a symmetry breaking constraint: *preservation*. A symmetry is preserved by a constraint iff its symmetry classes are either entirely removed from the solution set or retained as solutions by the constraint. Although, at first sight, constraints that break more symmetries as side-effects seem to be good choices, we should distinguish between side-effects on target symmetries and those on non-target symmetries.

We propose that symmetry breaking constraints aiming at some target symmetries should be selected to preserve other target symmetries and restrict the side-effects to non-target symmetries as much as possible. We analyze through the solution reduction ratio to show why preserving target symmetries can actually help us to eliminate more symmetries and thus are better choices. By carefully choosing symmetry breaking constraints to achieve specific side-effects, we achieve smaller solution set size and better efficiency. We also give observations on other factors that we should pay attention to when choosing symmetry breaking constraints.

A running example is given throughout the paper to demonstrate our ideas and results. Experimental results on four problems in the literature confirm empirically that models constructed using symmetry preservation achieve better efficiency up to one order of magnitude both in terms of runtime and search space.

2 Background

A *constraint satisfaction problem* (CSP) is a triple $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$, consisting of a set of variables $\mathcal{V}$, each $v \in \mathcal{V}$ with a finite domain of possible values $\mathcal{D}(v)$ and a set of constraints $\mathcal{C}$, each defined over a subset of variables specifying the allowed combination of values. An *assignment* gives each variable a value from its domain. A *solution* α is an assignment that satisfies all constraints. We use $sol(\mathcal{P})$ to denote the set of all solutions of $\mathcal{P}$.

A *symmetry* for a CSP is a bijection on the set of all assignments that maps solutions to solutions, and thus also non-solutions to non-solutions. Two common types of symmetry are variable symmetry and value symmetry. A *variable symmetry* is a bijective mapping σ on the indices of variables. If $[X_1, ..., X_n] = [d_1, ..., d_n]$ is a solution then $[X_{\sigma(1)}, ..., X_{\sigma(n)}] = [d_1, ..., d_n]$ is also a solution. A *value symmetry* is a bijective mapping θ on the values. If $[X_1, ..., X_n] = [d_1, ..., d_n]$ is a solution then $[X_1, ..., X_n] = [\theta(d_1), ..., \theta(d_n)]$ is also a solution. There are also constraint symmetries [2] that act on both variables and values. Our results are general and work with every kind of symmetries.

The *symmetry group* G_Σ of a set of symmetries Σ is formed by closing Σ under composition. A *symmetry class* of G_Σ is a subset S of symmetrically-equivalent assignments of $\mathcal{P}$. If an assignment $\alpha \in S$, then $\forall \sigma \in G_\Sigma$, $\sigma(\alpha) \in S$. A symmetry class contains either all solutions or no solutions. Symmetry classes of G_Σ partition $sol(\mathcal{P})$, with solutions in the same symmetry class mapped to one another under symmetries in G_Σ. Without loss of generality, by symmetry classes, we refer to the ones in $sol(\mathcal{P})$. A simple running example with variable and value symmetries will be used to illustrate various concepts throughout the paper.

Example 1. The Diagonal Latin Square problem (DLS(n)) aims to assign numbers 1 to n to the cells of an $n \times n$ board with no numbers occurring more than once in each row, column and the 2 diagonals. For convenience, we call the one extending from left top diagonal 1, and that from left bottom diagonal 2. We use a matrix model $[X_{ij}]$ of n^2 variables, each representing a cell and with domain $\{1, ..., n\}$. Problem constraints consist of ALLDIFF [24] on each row, column and the 2 diagonals.

The variable symmetries of DLS include the geometric symmetry group G_{geo} of size 8: horizontal reflection $\sigma_{rx}(X_{ij}) = X_{i(n+1-j)}$, vertical reflection σ_{ry}, diagonal reflections $\sigma_{d1}(X_{ij}) = X_{ji}$ and σ_{d2}, rotational symmetries σ_{r90}, σ_{r180} and σ_{r270}, and the identity symmetry σ_{id}. The values in DLS are interchangeable, which means the permutation of values G_{val} preserves solution. The following 4 solutions of DLS(5) are in the same geometric symmetry class. Solution $\sigma_{rx}(\alpha)$ can be obtained by flipping α over the vertical axis.

α	$\sigma_{rx}(\alpha)$	$\sigma_{d1}(\alpha)$	$\sigma_{rx} \circ \sigma_{d1}(\alpha)$
1 2 3 4 5	5 4 3 2 1	1 2 5 3 4	4 3 5 2 1
2 4 5 3 1	1 3 5 4 2	2 4 3 1 5	5 1 3 4 2
5 3 2 1 4	4 1 2 3 5	3 5 2 4 1	1 4 2 5 3
3 1 4 5 2	2 5 4 1 3	4 3 1 5 2	2 5 1 3 4
4 5 1 2 3	3 2 1 5 4	5 1 4 2 3	3 2 4 1 5

Define $row([X_{ij}]) \equiv [X_{11}, ..., X_{1n}, X_{21}, ..., X_{2n}, ..., X_{n1}, ..., X_{nn}]$. To eliminate σ_{rx}, we can post the LEXLEADER constraint: $row([X_{ij}]) \leq_{lex} [X_{1n}, ..., X_{11}, X_{2n}, ..., X_{21}, ..., X_{nn}, ..., X_{n1}]$. From problem constraints we infer $X_{11} \neq X_{1n}$ and simplify the LEXLEADER constraint to:

$$X_{11} < X_{1n} \tag{1}$$

To eliminate σ_{d1}, we can post constraint $row([X_{ij}]) \leq_{lex} [X_{11}, ..., X_{n1}, X_{12}, ..., X_{n2}, ..., X_{1n}, ..., X_{nn}]$, which can be simplified since $X_{11} = X_{11}$ and $X_{1n} \neq X_{n1}$:

$$[X_{12}, ..., X_{1n}] \leq_{lex} [X_{21}, ..., X_{n1}] \tag{2}$$

3 Effects of Symmetry Breaking Constraints

This section reports our observations and views based on existing results from the literature. We first introduce some definitions related to symmetry breaking constraints. We are concerned with the actual set of symmetries out of the whole symmetry group on which we post symmetry breaking constraints. We systematically discuss the effects of symmetry breaking constraints on the final solution set size as split into two cases.

3.1 Properties of Symmetry Breaking Constraints

We first introduce some useful concepts for the rest of the paper. Suppose a CSP $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ contains symmetries Σ and C^{sb} is a set of symmetry breaking constraints such that $\mathcal{P}' = (\mathcal{V},\mathcal{D},\mathcal{C} \cup C^{sb})$. Here and throughout, $sol(\mathcal{P}, C^{sb})$ is a short hand for $sol(\mathcal{P}')$, which is $sol((\mathcal{V}, \mathcal{D}, \mathcal{C} \cup C^{sb}))$.

We adapt the following definitions from Katsirelos and Walsh [18]. A set of symmetry breaking constraints C^{sb} *breaks* a symmetry $\sigma \in \Sigma$ iff there exist a solution $\alpha \in sol(\mathcal{P}')$ such that $\sigma(\alpha) \notin sol(\mathcal{P}')$. A set of symmetry breaking constraints C^{sb} *eliminates* a symmetry $\sigma \in \Sigma$ iff for each solution $\alpha \in sol(\mathcal{P}')$, $\sigma(\alpha) \notin sol(\mathcal{P}')$. We define a stronger version of elimination. C^{sb} *fully eliminates* a symmetry $\sigma \in \Sigma$ iff for each solution $\alpha \in sol(\mathcal{P}')$, $\forall k \in \mathbb{Z}$, if $\sigma^k(\alpha) \neq \alpha$ then $\sigma^k(\alpha) \notin sol(\mathcal{P}')$. This definition is to ensure there is at most one solution left by C^{sb} in each symmetry class of $G_{\{\sigma\}}$. C^{sb} *breaks/(fully) eliminates* a symmetry set Σ iff C^{sb} breaks/(fully) eliminates each $\sigma \in \Sigma$. A set of symmetry breaking constraints is *sound/complete* to a symmetry set Σ iff it leaves at least/exactly one solution in each symmetry class of G_Σ.

In the following, we use $\mathcal{C}_\Sigma$ to denote a set of symmetry breaking constraints that eliminates Σ, and let $\mathcal{P}' = (\mathcal{V}, \mathcal{D}, \mathcal{C} \cup \mathcal{C}_\Sigma)$. We say that $\mathcal{C}_\Sigma$ is sound/complete to mean that $\mathcal{C}_\Sigma$ is sound/complete to Σ. Symmetry breaking constraints can be derived by predefining a *canonical variable ordering* [26] and forcing the canonical solution to be always smaller (bigger) than its symmetrically-equivalent counterpart. The canonical variable ordering is the row-wise ordering $row([X_{ij}])$ in the running example. Any permutation on row($[X_{ij}]$) can serve as a possible ordering. We consider other specific orderings by moving X_{1n} and X_{n1} forward in $row([X_{ij}])$ and their simplified symmetry breaking constraints to eliminate σ_{d1} are shown respectively as follows:

$$canonical\ ordering:\ [X_{11}, X_{1n}, X_{n1}, X_{12}, ..., X_{nn}] \rightarrow\ \mathcal{C}_{\sigma_{d1}}:\ X_{1n} < X_{n1} \quad (3)$$
$$canonical\ ordering:\ [X_{11}, X_{n1}, X_{1n}, X_{12}, ..., X_{nn}] \rightarrow\ \mathcal{C}_{\sigma_{d1}}:\ X_{n1} < X_{1n} \quad (4)$$

These two canonical orderings result in the same constraint to eliminate σ_{rx} as $row([X_{ij}])$. We are interested in whether the solution set size is affected by picking other canonical ordering or using other methods to eliminate symmetries Σ. There are two possibilities: a symmetry group is eliminated *entirely* or *partially*.

3.2 Eliminating a Symmetry Group Entirely

We can easily see that $\mathcal{C}_\Sigma$ is complete to Σ if $\mathcal{C}_\Sigma$ is sound to Σ and Σ is a symmetry group, i.e. $\Sigma = G_\Sigma$.

Theorem 1. *Given a CSP, any sound set of symmetry breaking constraints eliminating the same symmetry group results in exactly the same solution set size.*

Proof. A sound set of symmetry breaking constraints eliminating a symmetry group is complete. Symmetry classes formed in the solution space under a symmetry group are fixed. By picking exactly one solution from each symmetry class of G, we gain a solution set with the same size as the number of symmetry classes, no matter what constraints we use to eliminate the symmetries. □

Many CSPs have exponentially-sized symmetry group. There exist efficient methods [6, 7] that eliminate all symmetries in some tractable classes of symmetries such as piecewise variable and value symmetry. Also in special cases such as all different problems [21], a linear number of constraints can eliminate all symmetries. However, eliminating the whole symmetry group is intractable and too costly in general. We aim to eliminate a subset [20]. We call this subset of symmetries we intend to eliminate as *target symmetries*. In practice, target symmetries are usually a generator set [1] for which there exist efficient methods to eliminate and can be generated automatically [15].

3.3 Eliminating a Symmetry Group Partially

We demonstrate how the choice of symmetry breaking constraints affects the solution set size when only the target symmetries Σ are eliminated. Representatives selected from each symmetry class of $G_{\{\sigma_i\}}$ ($\sigma_i \in \Sigma$) intersects to form the remaining solutions in the symmetry classes of G_Σ. The final solution set is determined by the intersection of canonical solutions of each symmetry breaking constraints: $sol(\mathcal{P}, \cup_{\sigma_i \in \Sigma} \mathcal{C}_{\sigma_i}) = \cap_{\sigma_i \in \Sigma} sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$. Even if each $\mathcal{C}_{\sigma_i}$ is complete, which means the size of each $sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$ is fixed, picking different canonical solutions of each constraint makes the intersection significantly different.

Example 2. We give a simple example considering again σ_{rx} and σ_{d1} in the DLS problem. We compare the result of combining constraint (1) with either constraint (3) or constraint (4). Picking constraint (3) we obtain $\{X_{11} < X_{1n}, X_{1n} < X_{n1}\}$, a total ordering on the variable sequence $[X_{11}, X_{1n}, X_{n1}]$; picking constraint (4), we obtain $\{X_{11} < X_{1n}, X_{n1} < X_{1n}\}$, making X_{1n} the biggest value out of the three. Because X_{11} and X_{n1} are not ordered in $\{X_{11} < X_{1n}, X_{n1} < X_{1n}\}$, the size of $sol(\mathcal{P}, \{X_{11} < X_{1n}, X_{n1} < X_{1n}\})$ is twice as that of $sol(\mathcal{P}, \{X_{11} < X_{1n}, X_{1n} < X_{n1}\})$.

We want to formulate a set of symmetry breaking constraints to get a minimum intersection, but we should avoid selecting an unsound set that misses solutions. For example in the DLS problem, to eliminate σ_{rx}, we can post $X_{11} < X_{1n}$; to eliminate σ_{r90} (90 degree rotational symmetry), we can post $X_{11} > \max\{X_{1n}, X_{n1}, X_{nn}\}$. Combining the two constraints results in the empty solution set.

The number of symmetry classes of G_Σ is the minimum size for the intersection, since completeness is the best we can achieve for $\mathcal{C}_\Sigma = \cup_{\sigma_i \in \Sigma} \mathcal{C}_{\sigma_i}$. In other words, we can eliminate the symmetry group G_Σ by eliminating each σ_i. This is always possible if we can use table constraints to specify exactly which representatives to retain as solutions. In practice, however, table constraints are difficult to craft and problem-specific, and we are limited by the available symmetry breaking methods in existing constraint programming system. Thus, eliminating each σ_i is not always sufficient to eliminate all symmetries when $\Sigma \neq G_\Sigma$. We need to find practical ways to prove soundness and achieve as small a solution set as possible. Our proposal is based on the side-effects of symmetry breaking constraints.

4 Side-Effects of Symmetry Breaking Constraints

We assume that every symmetry breaking constraint aims at breaking a target symmetry. The *side-effect* of a symmetry breaking constraint is its effect in breaking symmetries other than the target symmetry. We show that side-effects are common in symmetry breaking constraints and study how strong and widespread side-effects could be. Different choices of symmetry breaking constraints aiming at the same target symmetry can have different side-effects. For symmetries that are not affected, we say they are *preserved*. Then we show that selecting constraints that restrict the side-effects to non-target symmetries and preserve other target symmetries can gain us smaller solution set. We support our claims with theoretical analysis, examples and experimental results.

4.1 Side-Effects are Common

Our general study on side-effects are inspired by several examples in the literature. Katsirelos *et al*. [18, 17] shows in examples a constraint eliminating symmetry σ_{rx} also breaks the symmetry σ_{r90}, and the DOUBLELEX constraint eliminates also the value interchangeability in the EFPA problem in addition to the row and column symmetries. We are going to show that these side-effects of symmetry breaking constraints are common and in many cases inevitable. Given a CSP $\mathcal{P}$ with symmetry G_Σ, each pair of solutions in a symmetry class are mapped to each other under some symmetry $\sigma \in G_\Sigma$.

Theorem 2. *Given a CSP $\mathcal{P}$ with symmetries G_Σ. Suppose $\mathcal{C}_{\sigma_i}$ eliminates symmetry $\sigma_i \in G_\Sigma$ and it reduces the number of solutions in one symmetry class of G_Σ to at least two. Then $\exists \sigma_j \in G_\Sigma$ such that $\sigma_i \neq \sigma_j$ and σ_j is broken by $\mathcal{C}_{\sigma_i}$.*

Proof. Suppose $\alpha_1, \alpha_2 \in sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$ and $\sigma_i(\alpha_1) \neq \alpha_1$. Then $\sigma_i(\alpha_1)$ is not in $sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$ and there must exist a symmetry σ_j linking α_2 and $\sigma_i(\alpha_1)$. Since $\alpha_2 \in sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$ but $\sigma_j(\alpha_2)(= \sigma_i(\alpha_1))$ is not in $sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$, σ_j is broken by $\mathcal{C}_{\sigma_i}$ by definition. □

As a symmetry breaking constraint can break other symmetries in addition to its target one, we analyze the possible number of symmetries broken or eliminated as side-effects with the help of Figure 1. Suppose the symmetry group G_Σ is of size m and a symmetry class S is of size n, $n \leq m$. We assume $n = m$ for ease of discussion. A symmetry class is represented as a directed graph and the following notions are used.

- circle node: solution.
- solid arrow directed edge with specific symmetry label: symmetry mapping. If an edge from node 1 to node 2 is labeled with σ_j and suppose node 1 is solution α, then node 2 is $\sigma_j(\alpha)$.
- dash line cut: symmetry breaking constraint. The effect of a symmetry breaking constraint C_{σ_i} on the symmetry class is a *cut* on the graph that partitions the nodes (solutions) into two parts, S_1 with n_1 nodes satisfying the constraint and S_2 with n_2 nodes violating it , as shown in Figure 1(a).
- dot-dash edge: edge labeled with target symmetry σ_i and removed by the cut C_{σ_i}.

Each pair of nodes have edges in between them in both directions. Each node has totally $n - 1$ out-going edges and $n - 1$ in-coming edges, each labeled with a distinct symmetry $\sigma \in G_\Sigma$, as shown in Figure 1(b). The cut by C_{σ_i} removes $2 \times n_1 \times n_2$ edges, among which $2 \times n_1$ edges are labeled with σ_i. $\frac{n_2 - 1}{n_2}$ of the removed edges are labeled with other symmetries. We quantify the number of symmetries broken or eliminated by analyzing the result of the cut in the graph.

– Symmetry σ_j is broken iff there exists an edge labeled with σ_j removed by the cut.
– Symmetry σ_j is eliminated iff no edges in S_1 are labeled with σ_j.

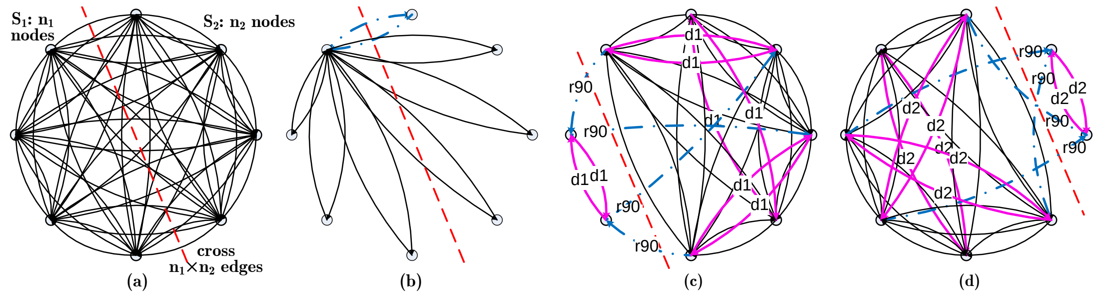

Fig. 1. A Symmetry Class of G_{geo}

 # Broken Symmetries Whenever an edge labeled with σ_j is cut, symmetry σ_j is broken by C_{σ_i}. If an edge labeled with σ_j is cut, there must exist an edge labeled with σ_j from S_1 to S_2 is cut. This is due to the closure of symmetries, which means there must exist a finite path from a node to itself consisting of only edges labeled with σ_j. All solutions in S_1 are in $sol(\mathcal{P}, C_{\sigma_i})$ while no solution in S_2 belong to $sol(\mathcal{P}, C_{\sigma_i})$. Therefore the symmetry σ_j linking a solution $\alpha \in S_1$ to $\sigma_j(\alpha) \in S_2$ is broken. For each node in S_1/S_2, there must exist n_2/n_1 edges labeled with distinct symmetries linking it to nodes in S_2/S_1. At least $\max\{n_1, n_2\}$ symmetries including the target symmetry are broken by the symmetry breaking constraint. *At least half the number of symmetries out of all are broken by a complete symmetry breaking constraint that aims at a single symmetry*.

 # Eliminated Symmetries According to the definition of elimination, symmetries that do not label any edges in S_1 are eliminated. For each solution in S_1, there must exist $n_1 - 1$ symmetries linking it to the others S_1. *At least* $n_1 - 1$ *symmetries are not yet eliminated*. Take Figures 1(c) and (d) as examples, one to two symmetries are not yet eliminated. A smaller n_1 means smaller number of edges $n_1 \times (n_1 - 1)$ are left in S_1. A small size S_1 usually indicates that less symmetries are left un-eliminated.

If we choose different symmetry breaking constraints to eliminate the same target symmetry, the cut is different and respectively the side-effects are different. The side-effects can be different in the number of symmetries broken/eliminated or the specific symmetries broken/eliminated. We demonstrate it using a simple example. To eliminate symmetry σ_{r90} in the DLS example, we can use constraint $X_{11} < \min\{X_{1n}, X_{n1}, X_{nn}\}$ or $X_{1n} < \min\{X_{11}, X_{n1}, X_{nn}\}$. The effect of each constraint on the symmetry class is shown in Figures 1(c) and (d) respectively. Constraint $X_{11} < \min\{X_{1n}, X_{n1}, X_{nn}\}$ eliminates all but σ_{d1} while constraint $X_{1n} < \min\{X_{11}, X_{n1}, X_{nn}\}$ eliminates all but σ_{d2}. We say the constraint *preserves* symmetry σ_{d1}/σ_{d2}.

4.2 Symmetry Preservation

We formally define symmetry preservation in the following:

Definition 1. *A symmetry σ is preserved by a set of symmetry breaking constraints iff elements of each symmetry class of σ are either entirely removed from the solution set or retained as solutions.*

In other words, the cut by symmetry breaking constraints C^{sb} does not remove any edges labeled with σ. For example in Figure 1(c), the edges labeled with symmetry σ_{d1} are not removed. When we say a symmetry σ is preserved by constraints C^{sb}, the intuition is that σ is not removed by C^{sb}. That means σ is still a symmetry in the new problem (the original problem with C^{sb} added). Even if C^{sb} removes all solutions (i.e. making them non-solutions) of σ's symmetry class, σ is not removed in the new problem since now σ still maps all such non-solutions to non-solutions, and is still a symmetry. A set of symmetries Σ is preserved iff each $\sigma \in \Sigma$ is preserved.

There are two nice properties of preservation. First, eliminating some symmetries can eliminate their composition under preservation.

Theorem 3. *Given a CSP $\mathcal{P}$ with symmetry σ_i and σ_j. If symmetry breaking constraint $\mathcal{C}_{\sigma_i}$ preserves symmetry σ_j, then $\sigma_i \circ \sigma_j$ and $\sigma_j \circ \sigma_i$ are eliminated by $\mathcal{C}_{\sigma_i} \cup \mathcal{C}_{\sigma_j}$.*

Proof. As shown in Figure 2, we have three disjoint sets $A = sol(\mathcal{P}, \mathcal{C}_{\sigma_i} \cup \mathcal{C}_{\sigma_j}), B = sol(\mathcal{P}, \mathcal{C}_{\sigma_i}) - sol(\mathcal{P}, \mathcal{C}_{\sigma_i} \cup \mathcal{C}_{\sigma_j}), E = sol(\mathcal{P}) - sol(\mathcal{P}, \mathcal{C}_{\sigma_i})$. Suppose $\alpha \in A$. By definition, symmetry σ_i links solutions (a) from A to E, (b) from B to E and (c) within E. Symmetry σ_j links solutions (a) within B, (b) within E and (c) from A to B. Therefore $\sigma_j(\alpha) \in B$ and $\sigma_i(\alpha) \in E$, $\sigma_i \circ \sigma_j(\alpha) \in E$ and $\sigma_j \circ \sigma_i(\alpha) \in E$. Since A and E are disjoint, both $\sigma_i \circ \sigma_j(\alpha)$ and $\sigma_j \circ \sigma_i(\alpha)$ are not in $sol(\mathcal{P}, \mathcal{C}_{\sigma_i} \cup \mathcal{C}_{\sigma_j})$. □

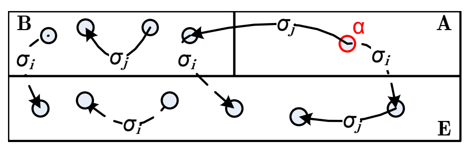

Fig. 2. σ_i preserve σ_j

Similarly, if a set of symmetry breaking constraints $\mathcal{C}_{G_1}$ preserves symmetry group G_2, then $\forall \sigma_i \in G_1, \sigma_j \in G_2$, $\sigma_i \circ \sigma_j$ and $\sigma_j \circ \sigma_i$ are eliminated by $\mathcal{C}_{G_1} \cup \mathcal{C}_{G_2}$. Second, when a symmetry set is preserved, if two symmetry breaking constraints are sound with respect to their target symmetries, their combination is also sound.

Theorem 4. *Given a set of symmetries $\Sigma = \Sigma_1 \cup \Sigma_2$. If $\mathcal{C}_{\Sigma_1}$ is sound and preserves Σ_2, then $\mathcal{C}_{\Sigma_2}$ being sound implies $\mathcal{C}_{\Sigma_1} \cup \mathcal{C}_{\Sigma_2}$ being sound.*

Proof. Based on the definition of preservation, if $\mathcal{C}_{\Sigma_1}$ preserves Σ_2 then Σ_2 is still a symmetry set of the new problem $\mathcal{P}' = (\mathcal{V}, \mathcal{D}, \mathcal{C} \cup \mathcal{C}_{\Sigma_1})$. No matter what $\mathcal{C}_{\Sigma_2}$ is, as long as it is sound to $\mathcal{P}'$, it is able to regenerate all solutions of $sol(\mathcal{P}, \mathcal{C}_{\Sigma_1})$ via symmetries in Σ_2. Then $sol(\mathcal{P})$ can be completely regenerated from $sol(\mathcal{P}, \mathcal{C}_{\Sigma_1})$ via symmetries in Σ_1 since $\mathcal{C}_{\Sigma_1}$ is sound. □

4.3 Solution Reduction by Symmetry Breaking Constraints

From the previous discussion, we can separate the effect of symmetry breaking constraints on other symmetries into three cases: *eliminate*, *break* and *preserve*. Our goal is to eliminate as many symmetries as possible and thus achieve smaller solution set by eliminating the target symmetries efficiently. What kind of effect is better?

The side-effects are common but we can choose to let the side-effects act on different symmetries. In the following, we show that symmetry breaking constraints that preserve other target symmetries and have more side-effects on non-target symmetries are better in the sense of symmetries eliminated and solution set size. We show how different side-effects may affect the final solution set size through analyzing the *solution reduction* by the symmetry breaking constraints.

The maximum solution reduction ratio $\frac{|sol(\mathcal{P})|}{|sol(\mathcal{P},\mathcal{C}_{\sigma_1})|}$ by a sound symmetry breaking constraint is achieved when it is complete to its target symmetry. Assume constraints are posted one by one and soundness must be ensured at each iteration. If the symmetry classes of σ_2 are left partially by $\mathcal{C}_{\sigma_1}$, the solution reduction ratio by $\mathcal{C}_{\sigma_2}$ in the new problem is smaller than that in the origin problem: $\frac{|sol(\mathcal{P},\mathcal{C}_{\sigma_1})|}{|sol(\mathcal{P},\mathcal{C}_{\sigma_1} \cup \mathcal{C}_{\sigma_2})|} < \frac{|sol(\mathcal{P})|}{|sol(\mathcal{P},\mathcal{C}_{\sigma_2})|}$. The origin solution set size of DLS(5) is 960. Posting constraint (1) $X_{11} < X_{1n}$ to eliminate σ_{ry} results in 480 solutions with reduction ratio 2. Posting it after symmetry σ_{rx} is eliminated by $X_{11} < X_{n1}$ results in 320 solutions out of 480 with reduction ratio 1.5.

This analysis also gives us insight into one reason why the efficiency we gained from eliminating more symmetries becomes weaker. When more constraints are added and their side-effects become stronger, not only the propagation overhead of constraints increases but also the available space that can be pruned becomes smaller. We consider this as an important factor behind partial symmetry breaking [20]. However, we can still achieve maximum solution reduction ratio when $\mathcal{C}_{\sigma_1}$ preserves σ_2. Without explicitly handling the composition symmetries, we are able to eliminate them as side-effects of eliminating the target symmetries when preserving other target symmetries.

A significant advantage of this approach is that, instead of introducing new constraints, we are able to eliminate more symmetries by selecting carefully the symmetry breaking constraints, which potentially entails better runtime.

4.4 Preservation Examples

Based on the previous theoretical analysis, we give three examples of achieving smaller solution set size by preserving symmetries. Two are from a big family of problems that can be modeled into matrix and contain a lot of symmetries.

Geometric Symmetries and Value Interchangeability We consider again the DLS problem and show different side-effects on the geometric symmetry group when eliminating value symmetries. We compare 6 choices of distinct value symmetry breaking constraints: fixing the value of (a) the first row, (b) the first column, (c) middle row, (d) middle column, (e) diagonal 1 and (f) diagonal 2, as shown from left to right in Figure 3. All of the constraints result in the same solution set size according to Theorem 1. The side-effects of each choice is: (a) eliminate G_{geo}, (b) eliminate G_{geo}, (c) eliminate $G_{geo} \setminus \{\sigma_{ry}\}$, (d) eliminate $G_{geo} \setminus \{\sigma_{rx}\}$, (e) eliminate $G_{geo} \setminus \{\sigma_{d1}\}$ and (f) eliminate $G_{geo} \setminus \{\sigma_{d2}\}$. After posting consistent constraints to eliminate G_{geo} for each, the solution set sizes of (a) and (b) are twice of those of the rest. The reason is obvious: the preserved symmetry σ_{ry}/σ_{rx}/σ_{d1}/σ_{d2} is further eliminated in (c)/(d)/(e)/(f) respectively and another half solution reduction is gained.

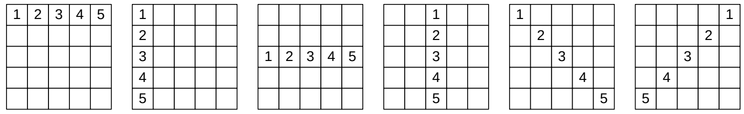

Fig. 3. Different Side-effect of $\mathcal{C}_{G_{val}}$

Matrix Symmetries and Value Interchangeability A common symmetry group in CSP is the matrix symmetries [5]. Since the size of the matrix symmetry group is super-exponential, usually only row and column symmetries are considered as target symmetries. Methods like DOUBLELEX [5] and SNAKELEX [11], built upon LEXLEADER [3], are efficient in eliminating row and column symmetries, but they do not eliminate the matrix symmetries in general. In order to preserve target symmetries, we consider the multiset ordering constraint $\leq_m$ [8].

Enforcing lexicographical ordering in one dimension breaks the symmetries in the other dimension, but enforcing multiset ordering in one dimension preserves the permutation symmetries in the other dimension. However, multiset ordering may not force a unique ordering, and thus eliminates fewer symmetries and is weaker than lexicographical ordering. However, combining it with lex ordering may eliminate more symmetries than combining both lex orderings. Based on Theorem 3, *if multiset ordering determine a unique ordering in one dimension, combining it with lexicographical ordering in the other dimension eliminates the matrix symmetries.* We cannot guarantee multiset ordering achieve unique ordering in many problems, but we show in the following that the solution set left by the combination can be much smaller than the one left by both $\leq_{lex}$.

We take the Cover Array Problem $CA(t, k, g, b)$ from CSPLib prob045 as example and use the integrated model [14], which channels an original model and a compound model. The original model contains a $b \times k$ matrix X of integer variables with domain $\{1..g\}$. The compound model contains a $b \times \binom{k}{t}$ matrix of integer variable with domain $\{1..g^k\}$. The original model contains matrix symmetries and value interchangeability in each column. In the following, we consider tow sets of target symmetries and compare our approach with those in the literature respectively.

- Considering row and column symmetries as target symmetries, we compare with the popular approach DOUBLELEX, denoted as **dLex**. Now we explain our choice of constraints. The number of columns k is smaller than the number of rows b in satisfiable Cover Array instances $CA(t, k, g, b)$. Since multiset ordering is relatively weaker, we choose to post it between columns. We simulate the multiset ordering constraint [8] using the Global Cardinality Constraint [25]: $\forall i, 0 \leq i \leq k, gcc([X_{1i}, \ldots, X_{bi}], [1, \ldots, g], [O_{1i}, \ldots, O_{gi}])$. Multiset ordering between columns of the original model $\forall 1 \leq i < j \leq k, [X_{1i}, \ldots, X_{bi}] \leq_m [X_{1j}, \ldots, X_{bj}]$ is achieved by enforcing lex ordering on the cardinality variable sequences $\forall 1 \leq i < j \leq k, [O_{1i}, \ldots, O_{gi}] \leq_{lex} [O_{1j}, \ldots, O_{gj}]$. We denote it as **mLex**.
- Considering row, column and value symmetries as target symmetries, we compare with the combination of DOUBLELEX and PRECEDENCE [19] constraints on each column, denoted as **dLex-V**. PRECEDENCE [19] is a global constraint to eliminate value interchangeability. The value interchangeability of each column are corresponding to variable interchangeability of the cardinality variable sequence $[O_{1i}, \ldots, O_{gi}]$ of each column i. We can simply enforce an ordering on the cardinality variable sequence $O_{1i} \leq \cdots \leq O_{gi}$ for each column i to break the value

symmetries. The combination of these value symmetry breaking constraints and mLex is denoted as **mLex-V**.

The value symmetry breaking constraints $\forall i, 1 \leq i \leq k, O_{1i} \leq \cdots \leq O_{gi}$ *in mLex-V preserve both row and column permutations*. For any solution obeying the constraints, permutating the rows does not change the number of occurrences of values $[O_{1i}, \ldots, O_{gi}]$ in each column i. Moreover, each column has the same ordering constraints on the cardinality variables and permutating the columns still obeys the constraints. When multiset ordering and value symmetry breaking constraint are complete respectively, the linear-size set of constraints mLex-V eliminates all symmetries of size $b! \times k! \times (g!)^k$.

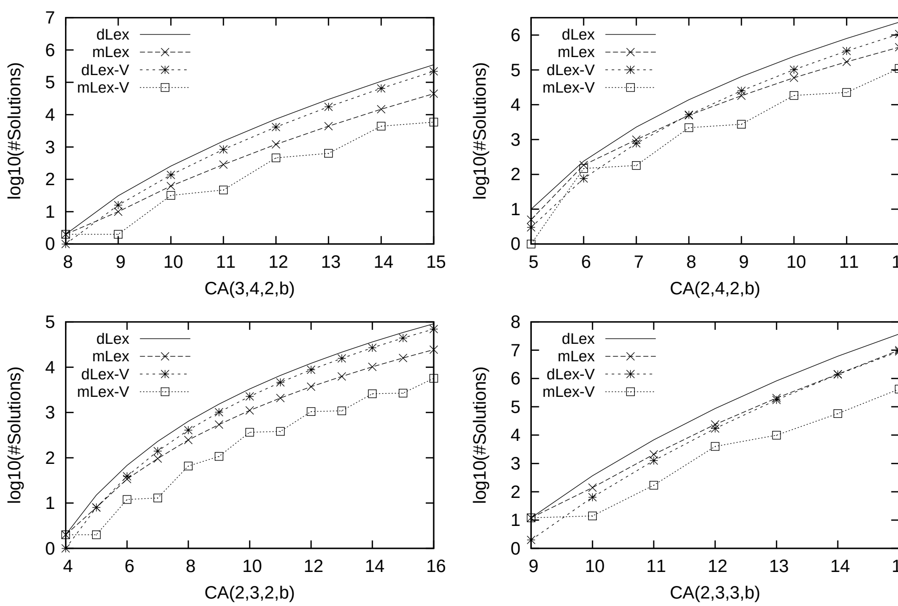

Fig. 4. Comparison of #Solutions in Cover Array Problem

Experiments are conducted on a Sun Blade 1000 (900MHz) running ILOG Solver 6.0. Variables are labeled in row-wise ordering. We conduct experiments on four instances of $CA(t, k, g, b)$ with different problem sizes. They vary in (1) the number of columns k in the original model, (2) the number of columns $\binom{k}{t}$ in the compound model and (3) domain size g in the original model and g^t in the compound model. We show the growth of solution set size in *log scale* in base 10 in Figure 4 and runtime in Figure 5 as the number of rows b increases.

We can see that mLex/mLex-V always have fewer solutions than dLex/dLex-V, which supports that preservation can achieve smaller solution set size. The distance between the curve of dLex and dLex-V is relatively smaller than that between mLex and mLex-V. The value symmetry breaking constraints in mLex-V preserve the row and column permutations such that combining them achieves good solution reduction. In all instances, the growth of dLex and dLex-V are similar, but the growth of mLex is smaller than both. Although mLex has more solutions than dLex-V in small instances, the solution set size of mLex becomes smaller at certain point. Constraints selected under preservation may even break or eliminate more symmetries as side-effects than those intended for a bigger set of target symmetries.

Interestingly, mLex-V has zig-zag curves. When b is even in instances with $g = 2$, the solution set size is relatively bigger. We conjecture that the occurrence of each value in each column are equal with high probability when b has factor g and less value symmetries are broken.

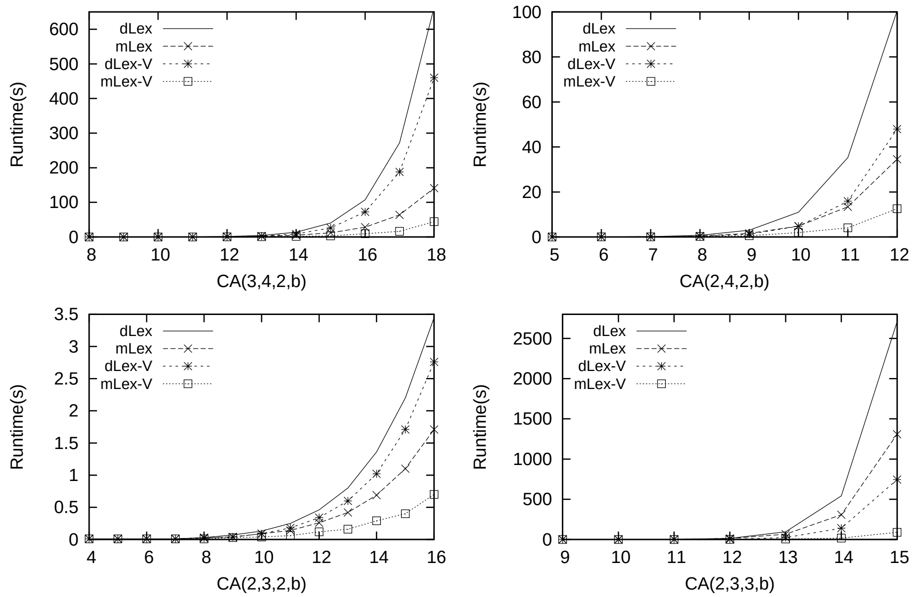

Fig. 5. Comparison of Runtime in Cover Array Problem

Comparing the runtime of each approach in Figure 5, although the performance of mLex/mLex-V is not as good in small instances, it is better as b grows. The runtime of mLex is smaller than dLex-V when $g = 2$ but bigger when $g = 3$.

Piecewise variable and value symmetries Piecewise variable and value symmetries are identified as a tractable class of symmetries in CSPs [7]. There exists a linear-size set of constraints [6] that remove super-exponential number of symmetries. This set of constraints [6] is inclusively a case of symmetry preservation: the value symmetry breaking constraints preserve the variable symmetries and therefore all composition symmetries are eliminated as side-effects.

5 Interactions with Problem Constraints

We have discussed how to achieve smaller solution set and thus better efficiency when intending to eliminate the same target symmetries. In this section, we give observations on ways to further reduce the search space. This results in extra advantages even when we post symmetry breaking constraints that obtain the same solution set size.

Further Simplification Sometimes, a symmetry breaking constraint can be simplified to an equivalent but cheaper one with the help of problem constraints, e.g. when there is an ALLDIFF constraint on all the variables [21]. Reduction rules [10] can be applied to reduce the number and arity of a conjunction of symmetry breaking constraints. We find that the degree of simplification can be different when choosing different alternatives. For example, to eliminate symmetry σ_{d1} in the DLS problem, we can post either LexLeader constraint (2) $[X_{12}, \ldots, X_{1n}] <_{lex} [X_{21}, \ldots, X_{n1}]$ or inequality constraints (3) $X_{1n} < X_{n1}$ or (4) $X_{n1} < X_{1n}$. Each choice cuts half of the solution set. Constraint (3) and (4) are preferred since it is cheaper and likely to entail better runtime.

To achieve simpler constraints, we can try different symmetry breaking constraints, especially those derived from canonical variable orderings that start with variables bounded by more problem constraints and other symmetry breaking constraints. The choice also affects how well symmetry breaking constraint interacts with problem constraints in terms of pruning.

Increasing Constraint Propagation Symmetry breaking constraints are constraints and have interactions with other constraints. A number of new global constraints [13, 16] that combines the symmetry breaking constraints with problem constraints are introduced to increase constraint propagation. We give an example of increasing constraint propagation by simply choosing another symmetry breaking constraint.

Example 3. Among various choices to eliminate the value symmetries of DLS(5), we pick three for discussion: fix the value of the first row, the middle column or the main diagonal. Variables in the center is constrained by four AllDiff constraints; variables in the diagonal line is constrained by three AllDiff constraints; and the rest by two AllDiff constraints. Figure 6 shows the remaining search space after the propagation.

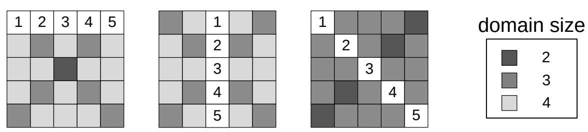

Fig. 6. remaining search space of DLS(5) - eliminate G_{val}

With the same solution set size, the remaining search space after propagation once when fixing the main diagonal is less than one hundredth of that when fixing the first row or middle column even in a small instance ($n = 5$). Even with the same number of solutions, different symmetry breaking constraints cooperate variously with problem constraints and other symmetry breaking constraints in terms of propagation. They may trigger different pruning power on variable domains when combined with problem constraints, some reducing dramatically more search space than others. Constraints that share variables with more or tighter problem constraints potentially trigger more propagation.

6 Experimental Results

This section evaluates how our choices of symmetry breaking constraints derived through preservation and the simple tips from Section 5 can influence solution set size and efficiency. In particular, we compare against the common practice reported in the literature. All problems can be formulated as matrix models containing both variable and value symmetries. We denote the symmetry breaking constraints derived from row-wise ordering as RowWise. The experiments are measured on a 3GHz Intel Core2 Duo PC with 3.2 GB RAM running Ubuntu 10.04.1 and Gecode-3.7.0. Depth first search is used and variables are searched in row-wise ordering with value tried in increasing order. We report the number of solutions, runtime and number of fails. Running time is expressed in seconds. Best results are highlighted in bold.

Our choices of symmetry breaking constraints for the first two benchmarks are as described in the first example of Section 4.4. Taking into account the interaction with problem constraints, constraints (5) and (6) in Figure 3 trigger more propagation and leave a smaller search tree. We choose (5) for experiments. Simplified constraint is posted for the un-eliminated symmetry σ_{d1}. The set of symmetry breaking constraint of Our Method is $\{[X_{11}, \ldots, X_{nn}] = [1, \ldots, n], X_{1n} < X_{n1}\}$. We compare with ROWWISE and VAR+OCC in the literature.

Diagonal Latin Square Table 1 shows results of constraints derived from row-wise ordering and those we selected from preservation. *All solutions are searched above the double line and one solution for the rest.* Our method has half the number of solutions as that of ROWWISE. Benefiting from both preservation and the interaction with problem constraints, we have smaller number of fails that imply smaller search tree. With instance $n = 11$, the runtime of searching for one solution of our method is seven orders of magnitude better.

NNQueen The problem is to place n colored queens on a $n \times n$ chessboard, such that no lines contain more than one queen with the same color. We model the problem as an n^2 variable matrix model and enforce an ALLDIFF constraint for each line. It also has geometric symmetries and value interchangeability. We compare with the symmetry breaking constraints by Puget [23]: VAR=$\{X_{11} < X_{1n}, X_{11} < X_{n1}, X_{11} < X_{nn}, X_{12} < X_{21}\}$ to eliminate geometric symmetries and a set of OCC constraints to eliminate value symmetries. Results presented in Table 2 show we gain better efficiency due to eliminating more symmetries.

Table 1. Diagonal Latin Square

n	ROWWISE			Our Method		
	#sol	time	#fails	#sol	time	#fails
5	8	0.001	7	**4**	**0.001**	**1**
6	128	0.029	3000	**64**	**0.004**	**652**
7	171200	12.981	1413K	**85600**	**1.954**	**163K**
8	1	0.002	140	1	**0.001**	**17**
9	1	40.04	4327K	1	**0.001**	**25**
10	1	0.031	2025	1	**0.002**	**175**
11	1	12052	1204124K	1	**0.005**	**339**

Table 2. NNQueen

n	VAR+OCC			Our Method		
	#sol	time	#fails	#sol	time	#fails
5	2	**0.001**	7	**1**	**0.001**	**0**
6	**0**	0.002	96	**0**	**0.001**	**55**
7	4	0.195	6201	**2**	**0.038**	**2496**
8	**0**	65.75	1824K	**0**	**16.47**	**1258K**
9	**0**	10660	232274K	**0**	**2349**	**153952K**

Error Correcting Code - Lee Distance (ECCLD) The problem is from CSPLib prob036. It requires to find the maximum number b of codes of length n drawn from 4 symbols $\{1, 2, 3, 4\}$ such that the Lee distance between any pair of codes is exactly c. The Lee distance between two symbols a and b is $\min\{|a - b|, 4 - |a - b|\}$. We model it into a $b \times n$ matrix with domain $\{1..4\}$. In order to illustrate the effect on solution set size, we transform the optimization problem to a satisfaction one by setting b in advance. Due to limited runtime, we constrain the problem size by limiting b. In general, to which dimension multiset ordering should be applied is problem specific. Different from the Cover Array Problem, we find that the best performance of the combination of multiset ordering and lexicographical ordering is achieved by $\geq_m$R,$\leq_{lex}$C (multiset ordering between rows and lexicographical ordering between columns). We compare with DOUBLELEX [5] and SNAKELEX [11]. We can see from Table 3 that $\geq_m R, \leq_{lex} C$ is better in both solution set size and runtime when $b \leq n$ but worse roughly when $n < b$.

Table 3. Error Correcting Code - Lee Distance

(n, c, b)	DOUBLELEX			SNAKELEX			$\geq_m R, \leq_{lex} C$		
	#sol	time	#fails	#sol	time	#fails	#sol	time	#fails
(4,4,8)	32469	**65.7**	**780K**	**29384**	86.9	996K	53972	81.6	928K
(5,2,10)	**87**	**7.98**	**42K**	107	10.9	78K	1040	11.7	80K
(5,6,4)	710731	45.2	426K	748248	29	660K	**213700**	**23.8**	**320K**
(5,6,5)	1441224	148	3378K	1468811	236	5661K	**379456**	**79.6**	**1779K**
(5,6,6)	297476	344	7090K	299821	602	11749K	**76528**	**204**	**3717K**
(6,4,4)	4698842	194	4036K	5061729	225	5341K	**1909044**	**115**	**2270K**
(6,4,5)	29345816	3340	72477K	29668229	3480	81624K	**11166072**	**1772**	**36960K**
(6,8,4)	59158	22.6	1175K	55618	29.8	1822K	**13163**	**12**	**588K**
(8,4,4)	35626714	2172	**48002K**	38629753	2554	63380K	**13403304**	**1108**	73211K

7 Conclusion

There are several methods [12, 18, 15] concerning the choice of the symmetry breaking constraints, such as model restarts [12] and dynamic posting [18] that reduce the conflict with search heuristic, and Jefferson *et al.*'s work [15] that chooses a better subset of symmetries to break, and use Crawford's [3] ordering constraints to generate the constraints. Our approach focuses on choosing better symmetry breaking constraints for a fixed set of target symmetries in the modeling stage so as to increase symmetry breaking or constraint interaction. The previous approach and ours are complementary to each other. Combining the approaches will be interesting future work.

Our goal is to find a set of symmetry breaking constraints that aims at only target symmetries but turns out to eliminate more symmetries. After analyzing the common side-effects of symmetry breaking constraints, we propose to select symmetry breaking constraints that preserve other target symmetries and restrict the side-effects to non-target symmetries. Unfortunately, our methods in the current form require the insight of a human modeler and automating it is non-trivial with our initial experience. The advantages of preserving target symmetries are demonstrated both analytically and empirically. We compare with approaches in the literature for problems in which eliminating all symmetries is not tractable. Without introducing new overhead, our approach can achieve smaller solution set and possibly better efficiency. To achieve symmetry preservation, different choices of symmetry breaking constraint need to be considered. It is interesting to investigate more flexible alternatives of symmetry breaking constraints that can preserve symmetries and propagate well with other constraints.

8 Acknowledgement

We are grateful to the anonymous referees for their comments. The work described in this paper was generously supported by grants CUHK413808 and CUHK413710 from the Research Grants Council of Hong Kong SAR.

References

1. F.A. Aloul, A. Ramani, I.L. Markov, and K.A. Sakallah. Solving difficult sat instances in the presence of symmetry. In *DAC'02*, pages 731–736. ACM, 2002.
2. D. Cohen, P. Jeavons, C. Jefferson, K. Petrie, and B. Smith. Symmetry definitions for constraint satisfaction problems. In *CP'11*, pages 17–31, 2005.
3. J. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry breaking predicates for search problems. In *KR'96*, pages 148–159, 1996.
4. T. Fahle, S. Schamberger, and M. Sellmann. Symmetry breaking. In *CP'01*, pages 93–107, 2001.
5. P. Flener, A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, J. Pearson, and T. Walsh. Breaking row and column symmetries in matrix models. In *CP'02*, pages 187–192, 2002.
6. P. Flener, J. Pearson, M. Sellmann, and P. Van Hentenryck. Static and dynamic structural symmetry breaking. In *CP'06*, pages 37–57. Springer, 2006.
7. P. Flener, J. Pearson, M. Sellmann, P. Van Hentenryck, and M. Ågren. Dynamic structural symmetry breaking for constraint satisfaction problems. *Constraints*, 14(4):506–538, 2009.
8. A.M. Frisch, I. Miguel, Z. Kiziltan, B. Hnich, and T. Walsh. Multiset ordering constraints. In *IJCAI'03*, pages 221–226, 2003.
9. I.P. Gent and B.M. Smith. Symmetry breaking in constraint programming. In *ECAI'00*, pages 599–603, 2000.
10. A. Grayland, C. Jefferson, I. Miguel, and C.M. Roney-Dougal. Minimal ordering constraints for some families of variable symmetries. *AMAI*, 57(1):75–102, 2009.
11. A. Grayland, I. Miguel, and C. Roney-Dougal. Snake lex: An alternative to double lex. In *CP'09*, pages 391–399, 2009.
12. D. Heller, A. Panda, M. Sellmann, and J. Yip. Model restarts for structural symmetry breaking. In *CP'08*, pages 539–544, 2008.
13. B. Hnich, Z. Kiziltan, and T. Walsh. Combining symmetry breaking with other constraints: lexicographic ordering with sums. In *ISAIM'04*, 2004.
14. B. Hnich, S.D. Prestwich, E. Selensky, and B.M. Smith. Constraint models for the covering test problem. *Constraints*, 11(2):199–219, 2006.
15. C. Jefferson and K. Petrie. Automatic generation of constraints for partial symmetry breaking. In *CP'11*, pages 729–743, 2011.
16. G. Katsirelos, N. Narodytska, and T. Walsh. Combining symmetry breaking and global constraints. *Recent Advances in Constraints*, pages 84–98, 2009.
17. G. Katsirelos, N. Narodytska, and T. Walsh. On the complexity and completeness of static constraints for breaking row and column symmetry. In *CP'10*, pages 305–320, 2010.
18. G. Katsirelos and T. Walsh. Symmetries of symmetry breaking constraints. In *ECAI'10*, pages 861–866, 2010.
19. YC Law and J.H.M. Lee. Symmetry breaking constraints for value symmetries in constraint satisfaction. *Constraints*, 11(2):221–267, 2006.
20. I. McDonald and B. Smith. Partial symmetry breaking. In *CP'06*, pages 207–213, 2006.
21. J. Puget. Breaking symmetries in all different problems. In *IJCAI'05*, pages 272–277, 2005.
22. J.F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. *Methodologies for Intelligent Systems*, pages 350–361, 1993.
23. J.F. Puget. Breaking all value symmetries in surjection problems. In *CP'05*, pages 490–504, 2005.
24. J.C. Régin. A filtering algorithm for constraints of difference in csps. In *AAAI'94*, pages 362–367, 1994.
25. J.C. Régin. Generalized arc consistency for global cardinality constraint. In *AAAI'96-Volume 1*, pages 209–215, 1996.
26. B.M. Smith. Sets of symmetry breaking constraints. In *SymCon'05*, 2005.