AMAI manuscript No.
(will be inserted by the editor)

Static and Dynamic Structural Symmetry Breaking

Pierre Flener[1,2⋆], **Justin Pearson**[2], **Meinolf Sellmann**[3]

[1] Faculty of Engineering and Natural Sciences
Sabancı University, Orhanlı, Tuzla, TR – 34956 İstanbul, Turkey

[2] Department of Information Technology
Uppsala University, Box 337, SE – 751 05 Uppsala, Sweden
e-mail: {Pierre.Flener,Justin.Pearson}@it.uu.se

[3] Department of Computer Science
Brown University, Box 1910, Providence, RI 02912, USA
e-mail: Meinolf_Sellmann@brown.edu

The date of receipt and acceptance will be inserted by the editor

Abstract We reconsider the idea of structural symmetry breaking for constraint satisfaction problems (CSPs). We show that the dynamic dominance checks used in symmetry breaking by dominance-detection search for CSPs with piecewise variable *and* value symmetries have a static counterpart: there exists a set of constraints that can be posted at the root node and that breaks *all* the *compositions* of these (unconditional) symmetries. The amount of these symmetry-breaking constraints is *linear* in the size of the problem, and yet they are able to remove a super-exponential number of symmetries on both values and variables. Moreover, we compare the search trees under static and dynamic structural symmetry breaking when using fixed variable and value orderings. These results are then generalised to wreath-symmetric CSPs with both variable and value symmetries. We show that there also exists a polynomial-time dominance-detection algorithm for this class of CSPs, as well as a linear-sized set of constraints that breaks these symmetries statically.

1 Introduction

Symmetry breaking for constraint satisfaction problems (CSPs) has been the topic of intense research in recent years, as symmetries naturally arise in many real-life applications. Substantial progress was achieved in many directions, often exhibiting significant speedups, for instance in configuration and network design. It is outside the scope of this introduction to review the wealth of research in this area.

⋆ Most of the work by this author was done while on leave of absence in 2006/07 as a Visiting Faculty Member and Erasmus Exchange Teacher at Sabancı University.

However, it is important to highlight some recent research avenues to position this paper properly.

One of the interesting developments has been the design of general symmetry-breaking schemes such as symmetry breaking by dominance detection (SBDD) and symmetry breaking during search (SBDS). SBDD [3,7] is particularly appealing for our purposes as it combines low memory requirements with a number of dominance checks linearly proportional to the depth of the search tree. It then became natural to study which classes of symmetries for CSPs are tractable, i.e., admit polynomial-time dominance-detection algorithms. This issue was first studied in [18], where symmetry breaking for various classes of value symmetries was shown to take constant time and space. In [16], this result was generalised elegantly to all value symmetries. We next revisited the issue for CSPs with simultaneous piecewise variable *and* value symmetry in [17], where a polynomial-time dominance-detection algorithm was given and the name 'structural symmetry breaking' was coined. The same paper also presented intractability results for set-CSPs under variable and value interchangeability, leaving open whether wreath symmetric CSPs (formally defined later in this paper) with variable and value symmetries were tractable with respect to symmetries. Moreover, some recent interesting results (see [19,12]) have indicated the possibility of automatically detecting certain classes of symmetries. These results taken together offer an opportunity to address the need for more automation, which was presented as one of the main challenges faced by constraint programming in industry [11].

In parallel, researchers have investigated, and this for many years (e.g., [10]) static symmetry breaking, which consists in the idea of adding constraints to a CSP in order to remove symmetries. That is, by breaking symmetries statically, we mean the addition of constraints that leave exactly one representative solution in each equivalence class of solutions. Lexicographic constraints [2] are one traditional way of breaking symmetries in this way.

This paper is an extension and revision of our [5]. It addresses two open issues in structural symmetry breaking. First, it reconsiders CSPs with piecewise interchangeable variables *and* values and studies whether the polynomial-time dominance-detection algorithm of [17] has a static counterpart. In other words, it studies whether there exists a set of constraints that, when added to the CSP at hand, produces a symmetry-free search tree. Second, the paper studies wreath symmetric CSPs with interchangeable variables and values. Note that throughout this paper we focus on unconditional (or global) symmetries, that is we do not handle any symmetries that appear during search. The results of this paper can be summarised as follows:

- We show that the polynomial-time dominance-detection algorithm of [17] has a static counterpart, namely that there exists a set of constraints for CSPs with piecewise symmetric variables *and* values that, when added to the CSP, results in a symmetry-free search tree.
- We establish a clear link between this static structural symmetry breaking (SSSB) scheme and the dynamic structural symmetry breaking (DSSB) scheme of [17] by comparing their search trees under various forms of consistency for

the symmetry-breaking constraints whenever the variable and value orderings are fixed.
- We show that wreath symmetric CSPs, with piecewise interchangeable variables and wreath interchangeable values, pose a tractable dominance detection problem. The dominance check is rather complex, but we also provide a set of constraints that break all these symmetries.
- To our knowledge, this is the first time that, for some classes of symmetries for CSPs, static symmetry breaking has been shown capable of breaking all compositions of variable and value symmetries at the same time, using an amount of symmetry-breaking constraints that is polynomial in the size of the problem, and yet removing, in general, a super-exponential number of symmetries on both values and variables. In fact, the number of constraints is even *linear* in the size of the problem.
- With the case of wreath values closed, the only classes of symmetries for which intractability results have been proven [17] involve variable and value symmetries over set CSPs or 0/1 representations of these as matrix models. The tractability results in this paper thereby also improve our understanding of what can and what cannot make symmetry breaking hard.

The remainder of the paper is organised as follows. Section 2 reviews the basic concepts. Section 3 presents the symmetry-breaking constraints for CSPs with piecewise variable and value interchangeability. Section 4 then establishes a link between static and dynamic symmetry breaking for such piecewise symmetric CSPs. Sections 5, 6, and 7 present the analogous, generalised concepts and results for wreath symmetric CSPs. Finally, Section 8 concludes the paper and discusses future research directions.

2 Basic Concepts

In this section, we fix some standard notation that we will use throughout the paper. Particularly, we define what we understand by piecewise variable and value symmetry, and what in this context we mean by dominance detection.

Definition 1 (CSP, Assignment, Solution)

- *A* constraint satisfaction problem (CSP) *is a triplet* $\langle V, D, C\rangle$*, where* V *denotes the set of variables,* D *denotes the set of possible values for these variables and is called their* domain*, and* $C : (V \to D) \to Bool$ *is a constraint that specifies which assignments of values to the variables are solutions.*
- *An* assignment *for a CSP* $\mathcal{P} = \langle V, D, C\rangle$ *is a function* $\alpha : V \to D$*.*
- *A* partial assignment *for a CSP* $\mathcal{P} = \langle V, D, C\rangle$ *is a function* $\alpha : W \to D$*, where* $W \subseteq V$*. The* scope *of* α*, denoted by* $scope(\alpha)$*, is* W*.*
- *A* solution *to a CSP* $\mathcal{P} = \langle V, D, C\rangle$ *is an assignment* σ *for* $\mathcal{P}$ *such that* $C(\sigma) = \mathtt{true}$*. The set of all solutions to a CSP* $\mathcal{P}$ *is denoted by* $Sol(\mathcal{P})$*.*

We want to reason about a special class of CSPs, namely those where subsets of variables or values are pairwise interchangeable. For instance, imagine an actor

scheduling problem where days are divided into morning and afternoon sessions; actors probably have strong preferences (and thus different fees for the morning and afternoon sessions) but the day of the session may not matter. To provide a formal definition, we first define:

Definition 2 (Partition) *Given a set S and a set of sets $P = \{P_1, \ldots, P_n\}$ such that $S = \bigcup_i P_i$ and the P_i are pairwise non-overlapping, we say that P is a* partition *of S and that each P_i is a* component*, and we write $S = \sum_i P_i$.*

Piecewise interchangeability implies that any reshuffling of variables or values within each component results in the same problem. Consequently, the corresponding permutations cannot mingle elements from different components:

Definition 3 (Piecewise Bijection) *Let $S = \sum_i P_i$ be a partitioned set. A bijection $b : S \to S$ is a* piecewise bijection *over $\sum_i P_i$ if and only if $\{b(e) \mid e \in P_i\} = P_i$.*

Equipped with this notion, we can now define formally:

Definition 4 (Piecewise Symmetric CSP) *A CSP $\mathcal{P} = \langle \sum_k V_k, \sum_\ell D_\ell, C \rangle$ is a* piecewise symmetric CSP *if and only if, for each solution $\alpha \in Sol(\mathcal{P})$, each piecewise bijection τ over $\sum_\ell D_\ell$, and each piecewise bijection σ over $\sum_k V_k$, we have $\tau \circ \alpha \circ \sigma \in Sol(\mathcal{P})$.*

Piecewise symmetric CSPs were studied in [17], where a polynomial-time algorithm was devised to detect symmetric dominance between two partial assignments:

Definition 5 (Dominance Detection) *Given two partial assignments α and β for a piecewise symmetric CSP $\mathcal{P} = \langle \sum_k V_k, \sum_\ell D_\ell, C \rangle$, we say that α* dominates *β if and only if there exist piecewise bijections σ over $\sum_k V_k$ and τ over $\sum_\ell D_\ell$ such that $\alpha(v) = \tau \circ \beta \circ \sigma(v)$ for all $v \in scope(\alpha)$. Then, we call the problem of determining whether α dominates β the* dominance detection problem.

Dominance detection constitutes the core operation of symmetry breaking by dominance detection (SBDD) [3,7], and its tractability immediately implies that we can efficiently limit ourselves to the exploration of symmetry-free search trees only. For piecewise symmetric CSPs, [17] showed that dominance detection is tractable. This was accomplished by *dynamic structural symmetry breaking (DSSB)*, where structural abstractions, so-called *value signatures*, generalise from an exact assignment of values to variables by quantifying how often a given value is assigned to variables in each component:

Definition 6 (Signature) *Given a partial assignment α for a piecewise symmetric CSP $\mathcal{P} = \langle \sum_k V_k, \sum_\ell D_\ell, C \rangle$, the* signature *of value d under α is the tuple that counts, for each variable component V_k, by how many variables in the component the value is taken in α:*

$$\mathrm{sig}_\alpha(d) := (|\{v \in V_k \cap scope(\alpha) \mid \alpha(v) = d\}|)_k$$

where k indexes the different variable components.

In [17], we showed how this structural abstraction allows us to check dominance between partial assignments α and β: We set up a bipartite graph where, for each value d, there is one node on the left and one node on the right. An edge connects two nodes with associated values d and e from the same value component if and only if $\mathrm{sig}_\alpha(d) \leq \mathrm{sig}_\beta(e)$, where $\leq$ denotes the point-wise ordering of two sequences (and not their lexicographic ordering). Then, α dominates β if and only if the bipartite graph contains a perfect matching.

Example 1 Consider the piecewise symmetric CSP $\langle \{v_1, v_2, v_3, v_4\} + \{v_5, v_6\}, \{1, 2\} + \{3, 4\}, C\rangle$ and the partial assignments $\alpha = \{v_1 \mapsto 2, v_2 \mapsto 2, v_3 \mapsto 3, v_5 \mapsto 2\}$ and $\beta = \{v_1 \mapsto 1, v_2 \mapsto 1, v_3 \mapsto 1, v_4 \mapsto 3, v_5 \mapsto 1, v_6 \mapsto 4\}$. The signatures of the values $1, 2, 3, 4$ are $(0, 0)$, $(2, 1)$, $(1, 0)$, $(0, 0)$ under α and $(3, 1)$, $(0, 0)$, $(1, 0)$, $(0, 1)$ under β. See the bipartite graph G of Figure 5 later in this paper (and ignore its caption). The rounded boxes indicate the components of the partition of the value set. There is an edge (d, e) whenever $\mathrm{sig}_\alpha(d) \leq \mathrm{sig}_\beta(e)$. As there exists a perfect matching in G, given by the solid edges, we conclude that α dominates β.

Based on this dominance-detection algorithm, DSSB filters values from domains if and only if setting the respective variable to some value would lead to a symmetric choice point. Since symmetry-based filtering *anticipates* when variable assignments will result in symmetric configurations, within DSSB we have to distinguish two different types of filtering: *ancestor-based filtering* where we compare extensions to the current partial assignment with previously fully expanded search nodes, and *sibling-based filtering* where we compare extensions to the current partial assignment with other such extensions.

When employing these filtering techniques, DSSB leads to symmetry-free search trees while causing only polynomial-time overhead. For a more detailed description of the method and a worst-case asymptotic runtime analysis, we refer the reader to [17].

3 Static SSB for Piecewise Symmetric CSPs

We now show how the idea of structural symmetry breaking, i.e., dominance detection based on signature analysis, can be used to devise a set of symmetry-breaking constraints for piecewise symmetric CSPs. For the first time, we will show that a polynomial, and even linear, amount of symmetry-breaking constraints is able to simultaneously break super-exponentially many compositions of variable and value symmetries efficiently.

3.1 Symmetry-Breaking Constraints

Consider a piecewise symmetric CSP $\langle \sum_{k=1}^{a} V_k, \sum_{\ell=1}^{b} D_\ell, C\rangle$, with $V = \{v_1, \ldots, v_n\} = \sum_{k=1}^{a} V_k$ a set of piecewise interchangeable variables and $D = \{d_1, \ldots, d_m\} = \sum_{\ell=1}^{b} D_\ell$ a set of piecewise interchangeable values. Assume a total ordering of the variables V and the values D.

As it is commonly done in the literature, we can break the variable symmetries within each variable component by requiring that earlier variables take smaller or equal values. To break the value symmetries, we resort to the same structural abstractions as DSSB, namely *value signatures*, which generalise from an exact assignment of values to variables by quantifying how often a given value is assigned to variables in each component. Let the frequency

$$f_i^k = |\{v \in V_k \mid \alpha(v) = d_i\}|$$

denote how often each value d_i is taken under solution α by the variables in each variable component V_k. For a solution α, we then denote by

$$\mathrm{sig}_\alpha(d_i) := (f_i^1, \ldots, f_i^a)$$

the *signature* of d_i under α. Then, for all consecutive values d_i, d_{i+1} in the same value component, we require that their signatures are lexicographically non-increasing, i.e., $\mathrm{sig}_\alpha(d_i) \geq_{\mathrm{lex}} \mathrm{sig}_\alpha(d_{i+1})$. So the problem boils down to computing the signatures of values efficiently. Fortunately, this is an easy task when using the existing global cardinality constraint (gcc) [15].

We summarise the resulting structural symmetry-breaking constraints:

- For each variable component $V_k = \{v_p, \ldots, v_q\}$, there is a variable ordering chain:
$$v_p \leq \cdots \leq v_q \tag{1}$$
hence a total of $n - a$ ordering constraints.
- For each value d_i and each variable component $V_k = \{v_p, \ldots, v_q\}$, the frequencies
$$f_i^k = |\{v \in V_k \mid \alpha(v) = d_i\}|$$
under partial assignment α are calculated by the constraints
$$\mathrm{gcc}(v_p, \ldots, v_q,\ d_1, \ldots, d_m,\ f_1^k, \ldots, f_m^k) \tag{2}$$
for each V_k, hence a total of a global cardinality constraints.
- For each value component $D_\ell = \{d_p, \ldots, d_q\}$, there is an ordering chain for the value signatures:
$$(f_p^1, \ldots, f_p^a) \geq_{\mathrm{lex}} \cdots \geq_{\mathrm{lex}} (f_q^1, \ldots, f_q^a) \tag{3}$$
hence a total of $m - b$ lexicographic ordering constraints.

Note that the number of constraints added is *linear* in the size of the problem, unlike in the more general method in [13], and yet that they are able to break super-exponentially many compositions of variable and value symmetries.

Although value signatures are here defined in essentially the same way as in Definition 6 in the framework of dominance detection, for static symmetry breaking we require them to be *lexicographically* rather than point-wise ordered. Indeed, in dynamic dominance detection, we are not interested in constructing the lexicographically minimal *solution* of any class of symmetrically equivalent solutions, but just in detecting specialisation of *partial* assignments. Also, in static structural symmetry breaking, we require a *total* order between value signatures, but the point-wise order is not total.

3.2 Example

Consider scheduling study groups for two sets of five indistinguishable students each. There are six identical tables with four seats each. Let $\{v_1, \ldots, v_5\}+\{v_6, \ldots, v_{10}\}$ be the partitioned set of piecewise interchangeable variables, one for each student. Let the domain $\{t_1, \ldots, t_6\}$ denote the set of tables, which are fully interchangeable. The structural symmetry-breaking constraints are:

$$\begin{array}{l} v_1 \leq v_2 \leq v_3 \leq v_4 \leq v_5 \\ v_6 \leq v_7 \leq v_8 \leq v_9 \leq v_{10} \\ \mathrm{gcc}(v_1, \ldots, v_5,\ t_1, \ldots, t_6,\ f_1^1, \ldots, f_6^1) \\ \mathrm{gcc}(v_6, \ldots, v_{10},\ t_1, \ldots, t_6,\ f_1^2, \ldots, f_6^2) \\ (f_1^1, f_1^2) \geq_{\mathrm{lex}} \cdots \geq_{\mathrm{lex}} (f_6^1, f_6^2) \end{array}$$

Consider the assignment $\alpha = \{v_1 \mapsto t_1, v_2 \mapsto t_1, v_3 \mapsto t_2, v_4 \mapsto t_3, v_5 \mapsto t_4\} \cup \{v_6 \mapsto t_1, v_7 \mapsto t_2, v_8 \mapsto t_2, v_9 \mapsto t_3, v_{10} \mapsto t_5\}$. Within each variable component, the $\leq$ ordering constraints are satisfied. Having determined the frequencies using the gcc constraints, we observe that the $\geq_{\mathrm{lex}}$ constraints are satisfied, because

$$(2,1) \geq_{\mathrm{lex}} (1,2) \geq_{\mathrm{lex}} (1,1) \geq_{\mathrm{lex}} (1,0) \geq_{\mathrm{lex}} (0,1) \geq_{\mathrm{lex}} (0,0).$$

If student 10 moves from table 5 to table 6, producing a symmetrically equivalent assignment because the tables are fully interchangeable, the $\geq_{\mathrm{lex}}$ constraints are no longer satisfied, because

$$(2,1) \geq_{\mathrm{lex}} (1,2) \geq_{\mathrm{lex}} (1,1) \geq_{\mathrm{lex}} (1,0) \geq_{\mathrm{lex}} (0,0) \not\geq_{\mathrm{lex}} (0,1).$$

If students 9 and 10 swap their assigned tables, producing a symmetrically equivalent assignment because both students are of the same student component, the signatures do not change and the $\geq_{\mathrm{lex}}$ constraints remain satisfied; however, we then have $v_9 \not\leq v_{10}$.

3.3 Analysis

We now establish the correctness and completeness of the introduced symmetry-breaking constraints. The key observation that we need to make is captured by the following lemma:

Lemma 1 *Given an assignment γ to a piecewise symmetric CSP $\langle \sum_k V_k, \sum_\ell D_\ell, C \rangle$, let the associated tuple of signature multisets be $\mathrm{TSM}_\gamma := (\{\mathrm{sig}_\gamma(d) \mid d \in D_\ell\})_\ell$. It holds that assignments α and β are symmetric if and only if $\mathrm{TSM}_\alpha = \mathrm{TSM}_\beta$.*

Proof $\Rightarrow$: Assume there exist piecewise bijections σ over $\sum_k V_k$ and τ over $\sum_\ell D_\ell$ such that $\alpha = \tau \circ \beta \circ \sigma$. Recall that signatures just count how many variables in each component take a given value. Therefore, the permutation of variables within a component cannot change the signatures of values. It follows that

$\mathrm{TSM}_\alpha = \mathrm{TSM}_{\tau\circ\beta\circ\sigma} = \mathrm{TSM}_{\tau\circ\beta}$. But the permutation of values within the same component D_ℓ does not affect the signature multiset $\{\mathrm{sig}_\beta(d) \mid d \in D_\ell\}$. Hence $\mathrm{TSM}_\alpha = \mathrm{TSM}_{\tau\circ\beta} = \mathrm{TSM}_\beta$.

$\Leftarrow$: Now assume that $\mathrm{TSM}_\alpha = \mathrm{TSM}_\beta$. By reversing the previous argument, there exists a piecewise bijection τ over $\sum_\ell D_\ell$ such that $\mathrm{sig}_\alpha(d) = \mathrm{sig}_{\tau\circ\beta}(d)$ for all $d \in D$. Thus, according to [17], there also exists a piecewise bijection σ over $\sum_k V_k$ such that $\alpha = \tau \circ \beta \circ \sigma$.

Theorem 1 *For every solution α to a piecewise symmetric CSP, there exists exactly one symmetric solution that obeys the structural symmetry-breaking constraints.*

Proof At least one: We first show that, for every solution to the original CSP, there exists at least one symmetric solution that also obeys all the additional symmetry-breaking constraints. Denote by $\tau_{\alpha,k}$ the function that ranks the indices of the values in $D_k = \{d_p, \ldots, d_q\}$ according to the signatures over some solution α, i.e., $\mathrm{sig}_\alpha(d_{\tau_{\alpha,k}(p)}) \geq_{\mathrm{lex}} \cdots \geq_{\mathrm{lex}} \mathrm{sig}_\alpha(d_{\tau_{\alpha,k}(q)})$. We obtain a symmetric solution β where we re-order the values in each D_k according to $\tau_{\alpha,k}$. Then, when we denote by $\sigma_{\beta,k}$ the function that ranks the indices of the variables in $V_k = \{v_p, \ldots, v_q\}$ according to β, i.e., $\beta(v_{\sigma_{\beta,k}(p)}) \leq \cdots \leq \beta(v_{\sigma_{\beta,k}(q)})$, we can re-order the variables in each V_k according to $\sigma_{\beta,k}$, and we get a new symmetric solution γ. As we already argued in the proof of Lemma 1, the re-ordering of the variables within each component has no effect on the signatures of the values, i.e., $\mathrm{sig}_\gamma(d) = \mathrm{sig}_\beta(d)$ for all $d \in D$. Thus, γ is also a solution to the original CSP that also obeys all symmetry-breaking constraints.

At most one: Now assume there are two solutions α and β to the piecewise symmetric CSP that both obey all the symmetry-breaking constraints, and such that there exist piecewise variable and value bijections σ and τ such that $\alpha = \tau \circ \beta \circ \sigma$. According to Lemma 1, it then holds that $\mathrm{TSM}_\alpha = \mathrm{TSM}_\beta$. Because of the lexicographic ordering constraints (3) on the value signatures, this implies that the signatures under α and β are identical for all $d \in D$. However, with the signatures of all the values thus fixed, and with the ordering constraints (1) on the variables, there exists exactly one assignment that gives these signatures. Hence α and β must be identical. □

What we have achieved with Theorem 1 is the ability to break statically all piecewise variable and value symmetries in a given CSP. It is very important to note that this theorem is about *solutions*, rather than about partial assignments, hence the level of consistency enforced on the symmetry-breaking constraints does not affect the result.

4 Static versus Dynamic SSB for Piecewise Symmetric CSPs

The advantage of a static symmetry-breaking method lies mainly in its ease of use and its moderate costs per search node. Constraint propagation and incrementality are inherited from the existing $\geq_{\mathrm{lex}}$ and gcc constraints. On the other hand, it is well-known that static symmetry breaking can collide with dynamic variable and

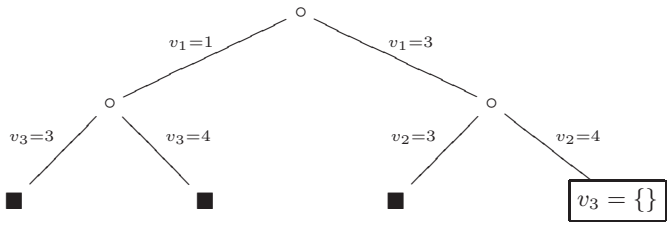


Fig. 1 DSSB search tree. The black-box nodes (■) mark the three solutions; all non-depicted assignments are obtained by propagation.

value orderings, whereas dynamic methods such as SBDD do not suffer from this drawback.

We were interested in studying how dynamic (DSSB) and static structural symmetry breaking (SSSB) actually relate to one another. Particularly, we were curious to know how dynamic structural symmetry breaking (DSSB) and static structural symmetry breaking (SSSB) relate to one another when variable and value orderings are fixed. Before stating our main results, let us consider an insightful example. First, it demonstrates that, when static variable and value orderings are used, DSSB can discard a partial assignment explored by SSSB when the static symmetry-breaking constraints are only used to *prune* the search tree, i.e., when the symmetry-breaking constraints are only used to detect that a partial assignment already violates one of the constraints, but not for filtering (in fact, Theorem 3 will show that the DSSB tree is in general a non-strict subtree of such an SSSB tree). Second, it shows that SSSB can discard a partial assignment explored by DSSB when hyper-arc consistency is enforced on the *conjunction* of the symmetry-breaking constraints (in fact, Theorem 2 will show that such an SSSB tree is in general a non-strict subtree of the DSSB tree).

Example 2 For both cases, we use the piecewise symmetric CSP $\langle \{v_1, v_2, v_3\} + \{v_4\}, \{1, 2\} + \{3, 4\}, C \rangle$, where the constraints C are:

- $v_1, v_2, v_3 \in \{1, 2, 3, 4\}, v_4 \in \{1, 2\}$.
- All variables together must take values 1 and 2 at most once.
- All variables together must take values 3 and 4 at most twice.

The problem only has the following three solutions up to symmetry: $\{v_1 \mapsto 1, v_2 \mapsto 3, v_3 \mapsto 3, v_4 \mapsto 2\}$, $\{v_1 \mapsto 1, v_2 \mapsto 3, v_3 \mapsto 4, v_4 \mapsto 2\}$, and $\{v_1 \mapsto 3, v_2 \mapsto 3, v_3 \mapsto 4, v_4 \mapsto 1\}$.

Consider the SSSB tree when using static symmetry breaking constraints for pruning only. Clearly, the assignment $\{v_1 \mapsto 2\}$ needs to be checked. However, DSSB completely discards this assignment as a symmetric sibling of $\{v_1 \mapsto 1\}$.

Now consider the DSSB tree after exploring all partial assignments up to $\alpha = \{v_1 \mapsto 3, v_2 \mapsto 4\}$. This node has to be explored by DSSB since neither of the nogoods $\{v_1 \mapsto 1\}$ and $\{v_1 \mapsto 3, v_2 \mapsto 3\}$ dominates it (see Figure 1).

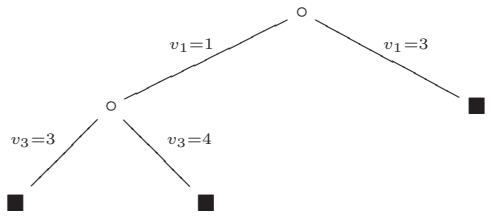


Fig. 2 SSSB search tree when hyper-arc consistency is enforced on the conjunction of the symmetry-breaking constraints; note that it is a subtree of the DSSB tree in Fig. 1.

On the other hand, when using SSSB and enforcing hyper-arc consistency on the conjunction of the symmetry-breaking constraints at the node $\{v_1 \mapsto 3\}$, there is no support for $v_2 \mapsto 4$ and hence the node $\alpha = \{v_1 \mapsto 3, v_2 \mapsto 4\}$ is *not* explored (see Figure 2).

In the following first comparison theorem, we claim that SSSB explores a subtree of the DSSB tree when we enforce hyper-arc consistency on the *conjunction* of the symmetry-breaking constraints. By Example 2, we even know that, in that setting, SSSB sometimes explores a *strict* subtree of the DSSB tree.

Theorem 2 *For piecewise symmetric CSPs, given a fixed variable and value ordering, and posting the symmetry-breaking constraints accordingly, SSSB explores a subtree of the tree explored by DSSB when we enforce hyper-arc consistency on the* conjunction *of the symmetry-breaking constraints.*

Proof To show that a node β in an SSSB tree with hyper-arc consistency on the conjunction of the symmetry-breaking constraints is also in the DSSB tree, we prove the contrapositive: that is, $\beta \notin \text{DSSB}$ implies that $\beta \notin \text{SSSB}$.

Assume that some partial assignment $\beta = \{v_1 \mapsto d_1, \ldots, v_t \mapsto d_t\} \notin \text{DSSB}$. This means that there is some partial assignment α, explored before, that dominates β. We look at the first (in depth-first order) such node α that dominates β.

The fact that α dominates β means that for all $v \in \mathit{scope}(\alpha)$ we have that $\alpha(v) = \tau \circ \beta \circ \sigma(v)$ for some piecewise variable and value bijections σ and τ.

As a reminder, given a conjunction C of constraints, a value d from the domain of a variable v is not filtered while achieving hyper-arc consistency on C iff there exists a solution α to C such that $\alpha(v) = d$ and $\alpha(w) \in \text{dom}(w)$ for all variables w. The assignment α is often referred to as the *support* of $v \mapsto d$. A hyper-arc consistency algorithm thus essentially ensures that there exist supporting assignments for all variables and all values in their domains.

Thus, to prove that $\beta \notin \text{SSSB}$, we have to show that $v_t \mapsto d_t$ has no support, i.e., that no *full* extension β' of β satisfies the symmetry-breaking constraints. We prove this by contradiction.

Assume such a β' exists. Then, applying τ and σ to β' yields a second full assignment:

$$\alpha' = \tau \circ \beta' \circ \sigma$$

that is symmetric to β'. Moreover, α' agrees with α for all $v \in scope(\alpha)$ and hence is a child of α. According to Lemma 1, it then holds that $\mathrm{TSM}_{\alpha'} = \mathrm{TSM}_{\beta'}$.

Now, consider the first variable $v \in V_k$ (recall that we assume that variables are being assigned in order) where α and β disagree, i.e., $\alpha'(w) = \alpha(w) = \beta(w) = \beta'(w)$ for all $w < v$ and when we set $d := \alpha(v) = \alpha'(v)$ and $e := \beta(v) = \beta'(v)$, we have that $d \neq e$. Since α was explored before β and values are also assigned in order, we can infer that $d < e$. However, as $\mathrm{TSM}_{\alpha'} = \mathrm{TSM}_{\beta'}$ and all variables in earlier variable components have been assigned in accordance between α and β, this implies that $\mathrm{sig}_{\alpha'}(f)_h = \mathrm{sig}_{\beta'}(f)_h$ for all values f and variable components $h < k$, and also $\mathrm{sig}_{\alpha'}(d)_k >_{\mathrm{lex}} \mathrm{sig}_{\beta'}(d)_k$. As β' satisfies constraints (1) and (3), there is no match for the signature $\mathrm{sig}_{\alpha'}(d)$ in the signature multiset $\{\mathrm{sig}_{\beta'}(f) \mid f \in D_\ell\}$ when $d \in D_\ell$. Therefore, $\mathrm{TSM}_{\alpha'} \neq \mathrm{TSM}_{\beta'}$. Contradiction. □

Note that it has been shown that achieving this level of consistency is NP-hard [20]. On the other hand, it is easy to check if a partial assignment violates any *individual* symmetry-breaking constraint. In our following second comparison theorem, we claim that DSSB explores a subtree of the SSSB tree when we use static symmetry-breaking constraints for pruning purposes only. Example 2 showed a case where, in that setting, DSSB explores a strict subtree of the SSSB tree.

Theorem 3 *For piecewise symmetric CSPs, given a fixed variable and value ordering, and posting the symmetry-breaking constraints accordingly, DSSB explores a subtree of the tree explored by SSSB when symmetry-breaking constraints are only used to prune the search tree.*

Proof Proof by contradiction. Assume there exists a node in the DSSB search tree that is pruned by SSSB. Without loss of generality, we may consider the first node in a depth-first search tree where this occurs. We identify this node with the assignment $\beta := \{v_1, \ldots, v_t\} \rightarrow D$.

First assume a variable ordering constraint is violated, i.e., $\beta(v_j) < \beta(v_i)$ for some $1 \leq i < j \leq t$ where v_i and v_j are interchangeable. Consider $\alpha : \{v_1, \ldots, v_i\} \rightarrow D$ such that $\alpha(v_k) := \beta(v_k)$ for all $1 \leq k < i$, and $\alpha(v_i) := \beta(v_j)$. Then, due to the fixed variable and value orderings, α is a node that has been fully explored before β, and α dominates β, which is clear by mapping v_i to v_j. Thus, β is also pruned by DSSB.

Now assume a lexicographic ordering constraint on the value signatures is violated. That is, there is some pair of values d_i and d_j, with $1 \leq i < j$ such that $\mathrm{sig}_\beta(d_i) <_{\mathrm{lex}} \mathrm{sig}_\beta(d_j)$. This means that there is some variable component V_k such that $f_i^k > f_j^k$. We pick *the first* such β in the search tree that violates the lexicographic ordering constraint. We now know that $\mathrm{sig}_\beta(d_i)[\ell] = \mathrm{sig}_\beta(d_j)[\ell]$ for all $\ell < k$ and $\mathrm{sig}_\beta(d_i)[k] + 1 = \mathrm{sig}_\beta(d_j)[k]$ (if $\mathrm{sig}_\beta(d_i)[k] + 1 < \mathrm{sig}_\beta(d_j)[k]$ then there exists an earlier β in the search tree violating the lexicographic ordering constraint). With $s := \max\{p \mid p < t \;\; \& \;\; \beta(v_p) = d_i\}$, we set $\alpha : \{v_1, \ldots, v_{s+1}\} \rightarrow D$ with $\alpha(v_r) := \beta(v_r)$ for all $r \leq s$ and $\alpha(v_{s+1}) := d_i$. Again, due to the fixed variable and value orderings, α is a node that has been fully explored before β, and α dominates β, which is clear simply by mapping d_i to d_j and permuting variables accordingly. Hence, β is also pruned by DSSB. □

Note that Theorems 2 and 3 together revise Theorem 2 of our [5], where we wrongly claimed that the two search trees were always identical when the variable and value orders are fixed. Indeed, we had overlooked the fact that two different levels of consistency were assumed in the two proof directions.

In summary, we conclude that dynamic symmetry breaking draws its strength from its ability to accommodate dynamic variable and value orderings, but causes an unnecessary overhead when these orderings are fixed. In this case, static symmetry breaking offers a much more light-weight method for piecewise symmetric CSPs. This view is also supported by the practical experiments carried out in [9].

5 Wreath Symmetry

We now wish to extend our ability to accommodate more complex symmetry classes than piecewise symmetry only. To this end, we consider a class of CSPs that assign a pair of values (d_1, d_2) from a domain $D_1 \times D_2$ to each variable, where the values in D_1 are piecewise interchangeable and, for a given value in D_1, the values in D_2 are piecewise interchangeable as well. These problems are here called wreath *value*-symmetric CSPs, because the symmetry group corresponds to a *wreath product* of groups [1].

Such problems arise naturally in a variety of applications, e.g., in resource allocation and scheduling. Consider, for example, the problem of scheduling a meeting where different groups must meet some day of the week in some room, subject to constraints. The days are interchangeable and, on a given day, the rooms are also interchangeable. Problems like this can be modelled as wreath value-interchangeable CSPs:

Definition 7 (Wreath Bijection) *Given two partitions $S_1 = \sum_p S_p^1$ and $S_2 = \sum_q S_q^2$, let $S = S_1 \times S_2$ denote their Cartesian product. A bijection $\tau : S \to S$ is a* wreath bijection *over $S_1 \times S_2$ if and only if there exists a piecewise bijection τ_1 over $\sum_p S_p^1$ and piecewise bijections τ_2^s over $\sum_q S_q^2$ for each $s \in S_1$, such that $\tau(\langle s, t\rangle) = \langle \tau_1(s), \tau_2^s(t)\rangle$.*

Definition 8 (Wreath Value Symmetry) *Given domain partitions $D_1 = \sum_p D_p^1$ and $D_2 = \sum_q D_q^2$, a CSP $\mathcal{P} = \langle V, D_1 \times D_2, C\rangle$ is called* wreath value-symmetric CSP *if and only if, for each solution $\alpha \in Sol(\mathcal{P})$ and each wreath bijection τ over $D_1 \times D_2$, we have $\tau \circ \alpha \in Sol(\mathcal{P})$.*

Note that the notion of wreath symmetry allows us to tackle much more refined symmetries than what can be expressed by piecewise symmetries only. In Figure 3, we show an example that illustrates the increased expressiveness of wreath value symmetry.

The reader should not confuse wreath *variable* symmetry (not discussed in this paper, but in [6]) with piecewise row and column symmetry [4] in a matrix of variables: under wreath variable symmetry, the rows are piecewise interchangeable (as under piecewise row symmetry), but the cells of each row are piecewise interchangeable in independent fashion (contrary to piecewise column symmetry), such as the groups of each week in the social golfer problem [3].

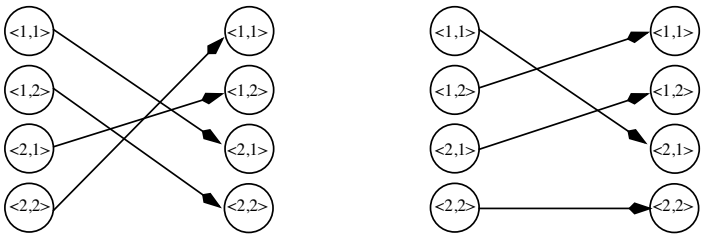


Fig. 3 Permutations on the domain $\{1, 2\} \times \{ 1, 2\}$. With the help of wreath symmetry, we can express that the permutation (a) is a valid symmetry while (b) is not. Piecewise symmetry does not allow us to make that distinction.

6 Dynamic SSB for Wreath Symmetric CSPs

In the following, we present, illustrate, and analyse a dominance detection algorithm for CSPs with piecewise variable symmetry and wreath value symmetry, simply called *wreath symmetric CSPs* hereafter.

6.1 The Dominance Detection Algorithm

Consider a wreath symmetric CSP $\langle \sum_{k=1}^{a} V_k, D_1 \times D_2, C \rangle$, with $V = \{v_1, \ldots, v_n\} = \sum_{k=1}^{a} V_k$ a set of piecewise interchangeable variables and $D_1 \times D_2$ a set of wreath interchangeable values, with $D_1 = \{d_1, \ldots, d_{m_1}\}$ and $D_2 = \{e_1, \ldots, e_{m_2}\}$ each having piecewise interchangeable elements.

Given partial assignments α and β, the dominance detection algorithm attempts to construct a piecewise bijection σ over $\sum_{k=1}^{a} V_k$ and piecewise bijections τ_1 and τ_2^e for all $e \in D_1$ such that $\alpha(v) = \tau \circ \beta \circ \sigma(v)$ for all $v \in \mathit{scope}(\alpha)$, where τ denotes the wreath bijection based on τ_1 and the τ_2^e. By definition, if the algorithm succeeds in finding such piecewise bijections, then α dominates β. Our algorithm will be based on the following core observation:

Remark 1 The piecewise bijection τ_1 can map $\tau_1(e_1) = d_1$ only if there exists a piecewise bijection $\tau_2^{e_1}$ such that

$$\begin{aligned} s_\beta^k(e_1, e_2) &:= |\{v \in V_k \mid \beta(v) = \langle e_1, e_2 \rangle\}| \\ &\geq |\{v \in V_k \mid \alpha(v) = \langle d_1, \tau_2^{e_1}(e_2) \rangle\}| =: s_\alpha^k(d_1, \tau_2^{e_1}(e_2)) \end{aligned}$$

for all k, where $s_\gamma^k(a, b)$ denotes the number of variables in component V_k that partial assignment γ maps to $\langle a, b \rangle$.

Now, for any partial assignment γ and values $f_1 \in D_1$ and $f_2 \in D_2$, we define

$$\mathrm{sig}_\gamma(\langle f_1, f_2 \rangle) := (s_\gamma^1(f_1, f_2), \ldots, s_\gamma^a(f_1, f_2)).$$

Then, in order to compute the subset of possible mappings that τ_1 could make, for each p and every pair $d_1, e_1 \in D_p^1$, our algorithm sets up the bipartite graph with

Initialise $\bar{G}$ as the empty graph
for all value components p **do**
  for all values $d, e \in S_p$ **do**
    if $G(d, e)$ contains a perfect matching **then**
      Add (d, e') to $\bar{G}$
    end if
  end for
end for
Return true if and only if $\bar{G}$ contains a perfect matching

Algorithm 1: Dominance detection for wreath symmetric CSPs.

the node sets $N_1(d_1, e_1) := \{d \mid d \in D_2\} = D_2$ and $N_2(d_1, e_1) := \{e' \mid e \in D_2\}$ as the set of primed copies of the values in D_2, and with the edge set

$$A(d_1, e_1) := \{(d_2, e_2') \in N_1(d_1, e_1) \times N_2(d_1, e_1) \mid \exists\, q :\ d_2, e_2 \in D_q^2 \ \&\ \mathrm{sig}_\alpha(\langle d_1, d_2\rangle) \leq \mathrm{sig}_\beta(\langle e_1, e_2\rangle)\}.$$

With Remark 1, observe that $\tau_1(e_1) = d_1$ is only ever feasible if a perfect matching in the bipartite graph $G(d_1, e_1) := (N_1(d_1, e_1) + N_2(d_1, e_1), A(d_1, e_1))$ exists. Consequently, to compute τ_1, we set up the bipartite graph $\bar{G} := (\bar{N}_1 + \bar{N}_2, \bar{A})$ with the node sets $\bar{N}_1 := \{d \mid d \in D_1\} = D_1$ and $\bar{N}_2 := \{e' \mid e \in D_1\}$ as the set of primed copies of the values in D_1, and with the edge set

$$\bar{A} := \{(d_1, e_1') \in \bar{N}_1 \times \bar{N}_2 \mid \exists\, p :\ d_1, e_1 \in D_p^1 \ \&\ G(d_1, e_1) \text{ has a perfect matching}\}.$$

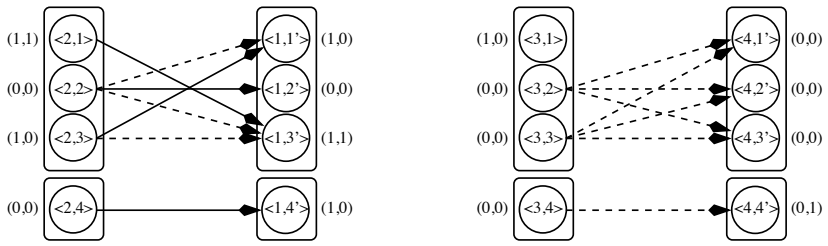
The algorithm decides that α dominates β if and only if $\bar{G}$ contains a perfect matching. The procedure is summarised as Algorithm 1.

6.2 Example

Assume we are given a wreath symmetric CSP $\langle\{v_1, v_2, v_3, v_4\}+\{v_5, v_6\}, (\{1, 2\}+\{3, 4\}) \times (\{1, 2, 3\} + \{4\}), C\rangle$ and partial assignments $\alpha = \{v_1 \mapsto \langle 2, 3\rangle, v_2 \mapsto \langle 2, 1\rangle, v_3 \mapsto \langle 3, 1\rangle, v_5 \mapsto \langle 2, 1\rangle\}$ and $\beta = \{v_1 \mapsto \langle 1, 4\rangle, v_2 \mapsto \langle 1, 1\rangle, v_3 \mapsto \langle 1, 3\rangle, v_4 \mapsto \langle 3, 2\rangle, v_5 \mapsto \langle 1, 3\rangle, v_6 \mapsto \langle 4, 4\rangle\}$.

Now assume that we consider to have τ_1 map the first-component value 1 to value 2. What are, for instance, the signatures of value $\langle 2, 3\rangle$ under α and of value $\langle 1, 3\rangle$ under β? We see that α assigns exactly one variable to $\langle 2, 3\rangle$, and this variable is in component $\{v_1, v_2, v_3, v_4\}$. According to our definition, it therefore holds that $\mathrm{sig}_\alpha(\langle 2, 3\rangle) = (1, 0)$. On the other hand, β assigns two variables to $\langle 1, 3\rangle$, one from $\{v_1, v_2, v_3, v_4\}$ and one from $\{v_5, v_6\}$. Thus, $\mathrm{sig}_\beta(\langle 1, 3\rangle) = (1, 1)$.

When setting $\tau_1(1) = 2$, all the signatures and the entire graph $G(2, 1)$ are shown in Figure 4(a), which also depicts a perfect matching in $G(2, 1)$, which means that the edge $(2, 1')$ is part of the first-component graph $\bar{G}$, given in Figure 5. In contrast to this existing edge, consider setting $\tau_1(4) = 3$. The corresponding graph $G(3, 4)$ is shown in Figure 4(b): since the node 1 corresponding to value



(a) (b)

Fig. 4 (a) The bipartite graph $G(2,1)$ constructed to assess whether $\tau_1(1) = 2$ is feasible. (b) The bipartite graph $G(3,4)$ constructed to assess whether $\tau_1(4) = 3$ is feasible. An edge between $\langle d_1, d_2\rangle$ and $\langle e_1, e_2'\rangle$ indicates that $\mathrm{sig}_\alpha(\langle d_1, d_2\rangle) \leq \mathrm{sig}_\beta(\langle e_1, e_2\rangle)$. The signatures are given next to the value pairs. The rounded boxes indicate the components of the partition of D_2. Perfect matchings, if any, are given by the solid edges.

$\langle 3,1\rangle$ has no adjacent edge at all, there is no perfect matching in the graph. Indeed, we see that, when setting $\tau_1(4) = 3$, the assignment $\alpha(v_3) = \langle 3,1\rangle$ finds no $v \in \{v_1, v_2, v_3, v_4\}$ and no $e \in \{1,2,3\}$ such that $\beta(v) = \langle 4, e\rangle$. Consequently, the edge $(3,4')$ is not part of $\bar{G}$. However, we see in Figure 5 that $\bar{G}$ contains the perfect matching $M = \{(1,2'),(2,1'),(3,3'),(4,4')\}$.

Now, the perfect matching M on $\bar{G}$ gives us the bijection τ_1 such that $\tau_1(1) = 2$ (from the edge $(2,1')$), $\tau_1(2) = 1$ (from edge $(1,2')$), $\tau_1(3) = 3$, and $\tau_1(4) = 4$. Under this setting, we define τ_2^1 based on the perfect matching in $G(2,1)$ by $\tau_2^1(1) = 3$, $\tau_2^1(2) = 2$, $\tau_2^1(3) = 1$, and $\tau_2^1(4) = 4$ (see Figure 4). Note that this assignment implicitly permutes the variables v for which $\beta(v) = \langle 1, e\rangle$ for some $e \in \{1,2,3,4\}$ while obeying the variable components. In our case, we implicitly get the partial variable bijection σ with $\sigma(v_1) = v_4$, $\sigma(v_2) = v_1$, $\sigma(v_3) = v_2$, and $\sigma(v_5) = v_5$. Note that we never actually need to compute the variable bijection σ that we get by combining the individual re-orderings.

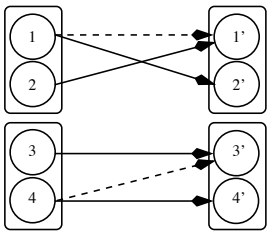


Fig. 5 The first-component bipartite graph $\bar{G}$, containing an edge (d,e) for each feasible mapping $\tau_1(e) = d$. The rounded boxes indicate the components of the partition of D_1. As there exists a perfect matching in $\bar{G}$, given by the non-dotted edges, we conclude that α dominates β.

6.3 Analysis

With the help of this method for dominance detection via structural symmetry breaking, we can show:

Theorem 4 *The dominance detection problem for CSPs with wreath value symmetry and piecewise variable symmetry is tractable.*

Proof First, let us show that the algorithm above is correct, i.e., that it does not detect dominance when there is none. Clearly, the perfect matching M in $\bar{G}$ gives us a piecewise bijection τ_1 over D_1 by setting $\tau_1(e_1) := d_1$ for all $(d_1, e_1) \in M$. Moreover, for each edge (d_1, e_1) in M, the corresponding perfect matching in $G(d_1, e_1)$ gives us a piecewise bijection $\tau_2^{e_1}$ over D_2 that is consistent with setting $\tau_1(e_1) := d_1$. Note that $\tau_2^{e_1}$ *implicitly* assigns one variable from the set $\{v \in V_k \mid \exists\ e_2 : \beta(v) = \langle e_1, e_2 \rangle\}$ to each variable in $\{v \in V_k \mid \exists\ d_2 : \alpha(v) = \langle d_1, d_2 \rangle\}$. This implies that the implicit variable bijections for all matching edges (d_1, e_1) in M do not collide as they map variables from disjoint subsets into disjoint subsets of V. Consequently, we can construct one global piecewise variable bijection σ and one wreath bijection τ such that $\alpha(v) = \tau \circ \beta \circ \sigma(v)$ for all $v \in \mathit{scope}(\alpha)$.

Now, regarding the completeness of our algorithm, assume that α actually dominates β. Denote the corresponding bijections by σ, τ_1, and τ_2^e as before. Under the piecewise variable bijection σ, we find that, for each $\tau_1(e_1) = d_1$, $\tau_2^{e_1}(d_2) = e_2$, and variable component V_k, there must exist at least as many variables in V_k that α maps to $\langle d_1, d_2 \rangle$ as variables in the same V_k that β maps to $\langle e_1, e_2 \rangle$. Consequently, we have that $\mathrm{sig}_\alpha(d_1, d_2) \leq \mathrm{sig}_\beta(e_1, e_2)$, which implies that $\tau_2^{e_1}$ defines a perfect matching in $G(d_1, e_1)$. Then, for each $\tau_1(e_1) = d_1$, there exists the edge (τ_1, d_1) in $\bar{G}$, which shows that $\bar{G}$ has a perfect matching. Thus, our algorithm finds that α dominates β.

Finally, we note that at most $|D_1|^2 + 1$ matchings need to be solved by the algorithm. Consequently, it runs in polynomial time. □

Theorem 4 is theoretically strong in that it subsumes many of previously proven results regarding the tractability of symmetry breaking. As a matter of fact, all the tractability results on breaking piecewise value or variable symmetry of CSPs considered in [18,17,6] follow from Theorem 4. However, from a practical perspective, the algorithm presented is very costly, especially when compared with constant overhead methods for breaking only value symmetries like the ones presented in [18,6]. Consequently, while the point here was to show that, with DSSB, it is even possible to efficiently handle wreath value symmetry and piecewise variable symmetry, in practice one is of course well advised to choose the dominance-detection algorithm just so that it can handle the symmetries that need to be broken.

Note that the dominance checker that we outlined in the proof above can be generalised for wreath tuples with k entries. However, the runtime then turns out to be exponential in k.

Finally, the following new intractability result for set-CSPs follows from Corollary 1 of [17] (intractability of dominance detection for piecewise symmetric set-CSPs), because wreath value interchangeability is piecewise value interchangeability when $|D_2| = 1$:

Corollary 1 *The dominance detection problem for set-CSPs with wreath value symmetry and piecewise variable symmetry is NP-hard.*

7 Static SSB for Full Wreath Symmetric CSPs

We now show that structural symmetry breaking can also be used to devise structural symmetry-breaking constraints for wreath symmetric CSPs. For simplicity, we do so only for piecewise variable symmetry and *full* wreath value symmetry, that is where Definition 7 is restricted to the case where the underlying piecewise bijections are all full bijections. We call such CSPs *full wreath symmetric CSPs* in this paper. It would be easy to generalise this to piecewise bijections, but we do not do so here to keep the notation simple.

7.1 Symmetry-Breaking Constraints

Consider a full wreath symmetric CSP $\langle \sum_{k=1}^{a} V_k, D_1 \times D_2, C \rangle$, with $V = \{v_1, \ldots, v_n\} = \sum_{k=1}^{a} V_k$ a set of piecewise interchangeable variables and $D_1 \times D_2$ a set of wreath interchangeable values, with $D_1 = \{d_1, \ldots, d_{m_1}\}$ and $D_2 = \{e_1, \ldots, e_{m_2}\}$ each having fully interchangeable elements. Assume a total ordering of the variables V, the elements D_1, and the elements D_2. Here are the structural symmetry-breaking constraints:

- For each variable component $V_k = \{v_p, \ldots, v_q\}$, there is a variable ordering chain:

$$v_p \leq_{\text{lex}} \cdots \leq_{\text{lex}} v_q \tag{4}$$

  hence a total of $n - a$ lexicographic ordering constraints.
- For each value (d_i, e_j) and each variable component $V_k = \{v_p, \ldots, v_q\}$, the frequencies

$$f^k_{i,j} = |\{v \in V_k \mid v \in \mathit{scope}(\alpha) \ \ \& \ \ \alpha(v) = (d_i, e_j)\}|$$

  under partial assignment α are calculated by the constraints

$$\text{gcc}(v_p, \ldots, v_q,\ (d_1, e_1), \ldots, (d_{m_1}, e_{m_2}),\ f^k_{1,1}, \ldots, f^k_{m_1,m_2}) \tag{5}$$

  for each V_k, hence a total of a global cardinality constraints.
- For each element d_i, there is an ordering chain for what we call the *signatures* of the (d_i, e_j) values:

$$(f^1_{i,1}, \ldots, f^a_{i,1}) \geq_{\text{lex}} (f^1_{i,2}, \ldots, f^a_{i,2}) \geq_{\text{lex}} \cdots \geq_{\text{lex}} (f^1_{i,m_2}, \ldots, f^a_{i,m_2}) \tag{6}$$

  hence a total of m_1 chains of $m_2 - 1$ lexicographic ordering constraints each.

– There is an ordering chain for what we call the *compound signatures* of the d_i elements:

$$\begin{array}{c}(f^1_{1,1},\ldots,f^a_{1,1},\ f^1_{1,2},\ldots,f^a_{1,2},\ \ldots,\ f^1_{1,m_2},\ldots,f^a_{1,m_2})\\ \geq_{\text{lex}}\\ (f^1_{2,1},\ldots,f^a_{2,1},\ f^1_{2,2},\ldots,f^a_{2,2},\ \ldots,\ f^1_{2,m_2},\ldots,f^a_{2,m_2})\\ \geq_{\text{lex}}\cdots\geq_{\text{lex}}\\ (f^1_{m_1,1},\ldots,f^a_{m_1,1},\ f^1_{m_1,2},\ldots,f^a_{m_1,2},\ \ldots,\ f^1_{m_1,m_2},\ldots,f^a_{m_1,m_2})\end{array} \tag{7}$$

hence one chain of $m_1 - 1$ lexicographic ordering constraints.

Again, we find that the number of constraints added is *linear* in the size of the problem (note that $m_1 \cdot m_2$ is linear in the input size when the domains are given explicitly), and yet they are able to break super-exponentially many compositions of variable and value symmetries as we shall show later in this section.

Note that these constraints specialise into (a specialisation for *full* value symmetry of) the symmetry-breaking constraints of Section 3.1 for piecewise symmetric CSPs. Indeed, the compound signature ordering chain (7) is vacuously true when $m_1 = 1$ while the signature ordering chains (6) then amount to the single signature ordering chain (3). Conversely, the signature ordering chains (6) are vacuously true when $m_2 = 1$ while the compound signature ordering chain (7) then amounts to the signature ordering chain (3). In any case, the variable ordering chains (4) trivially specialise into (1) as we essentially deal with 1-tuples, and the global cardinality constraints (5) trivially specialise into (2).

Finally, note that the constraints above can be adapted to accommodate *piecewise* rather than full wreath value symmetry: The only difference is that the ordering constraints (6) and (7) on the signatures then do not apply at value partition boundaries.

7.2 Example

Consider scheduling study groups for ten students divided into two categories of five indistinguishable students each. There are six tables with four seats each, divided over two rooms containing three tables each. The rooms are indistinguishable, and, within each room, all tables are indistinguishable. Let $\{v_1,\ldots,v_5\} + \{v_6,\ldots,v_{10}\}$ be the set of piecewise interchangeable variables, one for each student. Let the domain $\{r_1,r_2\}\times\{t_1,t_2,t_3\}$ denote the set of tables, which are fully wreath interchangeable. The structural symmetry-breaking constraints are:

$$\begin{array}{l}v_1 \leq_{\text{lex}}\cdots\leq_{\text{lex}} v_5\\ v_6 \leq_{\text{lex}}\cdots\leq_{\text{lex}} v_{10}\\ \text{gcc}(v_1,\ldots,v_5,\ (r_1,t_1),\ldots,(r_2,t_3),\ f^1_{1,1},\ldots,f^1_{2,3})\\ \text{gcc}(v_6,\ldots,v_{10},\ (r_1,t_1),\ldots,(r_2,t_3),\ f^2_{1,1},\ldots,f^2_{2,3})\\ (f^1_{1,1},f^2_{1,1}) \geq_{\text{lex}} (f^1_{1,2},f^2_{1,2}) \geq_{\text{lex}} (f^1_{1,3},f^2_{1,3})\\ (f^1_{2,1},f^2_{2,1}) \geq_{\text{lex}} (f^1_{2,2},f^2_{2,2}) \geq_{\text{lex}} (f^1_{2,3},f^2_{2,3})\\ (f^1_{1,1},f^2_{1,1},\ f^1_{1,2},f^2_{1,2},\ f^1_{1,3},f^2_{1,3}) \geq_{\text{lex}} (f^1_{2,1},f^2_{2,1},\ f^1_{2,2},f^2_{2,2},\ f^1_{2,3},f^2_{2,3})\end{array}$$

Consider the assignment

$$\alpha = \{v_1 \mapsto \langle r_1, t_1\rangle, v_2 \mapsto \langle r_1, t_1\rangle, v_3 \mapsto \langle r_1, t_1\rangle, v_4 \mapsto \langle r_1, t_2\rangle, v_5 \mapsto \langle r_1, t_2\rangle,\\ v_6 \mapsto \langle r_2, t_1\rangle, v_7 \mapsto \langle r_2, t_1\rangle, v_8 \mapsto \langle r_2, t_1\rangle, v_9 \mapsto \langle r_2, t_2\rangle, v_{10} \mapsto \langle r_2, t_3\rangle\}.$$

The $\leq_{\text{lex}}$ variable ordering constraints are satisfied. Having determined the value frequencies using the gcc constraints, we observe that the $\geq_{\text{lex}}$ (compound) signature ordering constraints are all satisfied, because

$$(3,0) \geq_{\text{lex}} (2,0) \geq_{\text{lex}} (0,0) \;\;\&\;\; (0,3) \geq_{\text{lex}} (0,1) \geq_{\text{lex}} (0,1) \;\;\&\\ (3,0,\ 2,0,\ 0,0) \geq_{\text{lex}} (0,3,\ 0,1,\ 0,1).$$

If the two student groups swap their table/room assignments, producing a symmetrically equivalent assignment, namely

$$\beta = \{v_1 \mapsto \langle r_2, t_1\rangle, v_2 \mapsto \langle r_2, t_1\rangle, v_3 \mapsto \langle r_2, t_1\rangle, v_4 \mapsto \langle r_2, t_2\rangle, v_5 \mapsto \langle r_2, t_3\rangle,\\ v_6 \mapsto \langle r_1, t_1\rangle, v_7 \mapsto \langle r_1, t_1\rangle, v_8 \mapsto \langle r_1, t_1\rangle, v_9 \mapsto \langle r_1, t_2\rangle, v_{10} \mapsto \langle r_1, t_2\rangle\},$$

the $\leq_{\text{lex}}$ variable ordering constraints are still satisfied, but the $\geq_{\text{lex}}$ (compound) signature ordering constraints are now violated, because

$$(0,3) \geq_{\text{lex}} (0,1) \geq_{\text{lex}} (0,1) \;\;\&\;\; (3,0) \geq_{\text{lex}} (2,0) \geq_{\text{lex}} (0,0) \;\;\&\\ (0,3,\ 0,1,\ 0,1) \not\geq_{\text{lex}} (3,0,\ 2,0,\ 0,0).$$

7.3 Analysis

Analogously to the case of piecewise symmetric CSPs, we find:

Lemma 2 *Given a full wreath symmetric CSP $\langle \sum_{k=1}^{a} V_k, D_1 \times D_2, C\rangle$, and an assignment γ, let the associated multiset of signature multisets be* $\text{MSM}_\gamma := \{\{\text{sig}_\gamma(\langle d, e\rangle) \mid e \in D_2\} \mid d \in D_1\}$. *It holds that two assignments α and β are symmetric if and only if* $\text{MSM}_\alpha = \text{MSM}_\beta$.

Proof $\Rightarrow$: Assume α and β are symmetric. We observe once more that the permutation of variables within variable components does not affect the signatures of values. Then, for each $d \in D_1$, the permutation of values in D_2 only permutes elements in $\{\text{sig}_\beta(\langle d, e\rangle) \mid e \in D_2\}$, which leaves the multiset as a whole unchanged. The same holds for the permutation of values in D_1 and MSM_β.

$\Leftarrow$: Now assume that $\text{MSM}_\alpha = \text{MSM}_\beta$. By reversing the previous argument, there exist a permutation τ_1 over D_1 and for each $d \in D_1$ a permutation τ_2^d over D_2 such that $\text{sig}_\alpha(\langle d, e\rangle) = sig_\beta(\langle \tau_1(d), \tau_2^d(e)\rangle)$ and such that $\{\text{sig}_\alpha(\langle d, e\rangle) \mid e \in D_2\} = \{\text{sig}_\beta(\langle \tau_1(d), \tau_2^d(e)\rangle) \mid e \in D_2\}$. Then it is easy to construct σ and τ such that $\alpha = \tau \circ \beta \circ \sigma$.

Equipped with this insight, we can now establish the counterpart of Theorem 1 for full wreath symmetric CSPs:

Theorem 5 *For every solution α to a full wreath symmetric CSP, there exists exactly one symmetric solution that obeys the structural symmetry-breaking constraints*.

Proof At least one: Given a solution α, we show that there exists at least one symmetrically equivalent solution that also satisfies all the symmetry-breaking constraints. First, for each d, determine a full bijection $\tau^d : D_2 \rightarrow D_2$ such that all the lexicographic ordering constraints (6) on the signatures are satisfied. This can be seen as a wreath bijection acting as the identity on the first component. Second, determine a full wreath bijection τ such that all the lexicographic ordering constraints (7) on the compound signatures are satisfied, the trick at this stage being to carry over the τ^d bijections obtained in the first stage. Doing this will not violate any of the already satisfied constraints (6). Finally, we observe that reordering the variables so that they satisfy all the lexicographic ordering constraints (4) has no effect on any of the signatures, so there exists a solution α' that is symmetric to α and that satisfies all the structural symmetry-breaking constraints.

At most one: Now we prove that any two solutions that satisfy all the structural symmetry-breaking constraints must be identical. According to Lemma 2, there is a fixed multiset of signature multisets MSM_γ for all solutions that are symmetric to solution γ. However, for all $d \in D_1$, the elements in the signature multiset $\{\mathrm{sig}_\alpha(\langle d, e\rangle) \mid e \in D_2\}$ are ordered by the lexicographic ordering constraints (6) on the value signatures. Moreover, the lexicographic ordering constraints (7) on the compound signatures enforce an ordering of all the elements in MSM_γ. In combination with the variable ordering constraints (4), there is but one assignment that fulfils all these constraints for each fixed multiset of signature multisets MSM_γ. □

8 Conclusions

We have shown the great power of structural symmetry breaking on complex cases of simultaneous value and variable interchangeability in CSPs. The results on dynamic symmetry breaking are theoretically significant in that they subsume many of previously proven results regarding the tractability of dominance detection. From a practical perspective, the dynamic algorithms presented are very costly, though, especially when compared with constant-overhead methods for breaking only value symmetries like the ones presented in [18,6]. Consequently, we have exploited the idea of structural symmetry breaking to devise sets of symmetry-breaking constraints that simultaneously break all the compositions of piecewise variable and piecewise or wreath value symmetries. To our knowledge, these are the first identified classes of symmetries for CSPs where a polynomial, yet even a linear number of static symmetry breaking constraints suffices to break a super-exponential number of variable and value symmetries. We have then shown that, in case of static variable and value orderings, the search tree explored by static structural symmetry breaking (SSSB) is a subtree of the one explored by dynamic structural symmetry breaking (DSSB) when we achieve hyper-arc consistency for the conjunction of symmetry breaking constraints, and that the DSSB search tree is a subtree of the SSSB tree when we use constraints for pruning purposes only. Note that the first result implies that SSSB is, in principle, able to guarantee symmetry-free search trees. This is a clear indication that using SSSB is the way to go whenever fixed variable and value orderings can be expected to work well.

With respect to future work, the following questions arise. Can we find general conditions under which a static symmetry-breaking method leads to symmetry-free search trees? Can static structural symmetry breaking be usefully combined with the dynamic lexicographic ordering constraints of [14]?

Acknowledgements:

The authors were partly supported by grant IG2001-67 of STINT, the Swedish Foundation for International Cooperation in Research and Higher Education. Meinolf Sellmann is supported by the National Science Foundation through the Career: Cornflower Project (NSF award number 0644113). Many thanks to Pascal Van Hentenryck, who co-authored a previous version of this paper [5], for his comments on this version. Finally, we appreciate the constructive feedback by Barbara Smith, Chris Jefferson, and the anonymous referees, including those of [5].

References

1. P. Cameron. *Permutation Groups*. Number 45 in London Mathematical Society Student Texts. Cambridge University Press, 1999.
2. J. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry-breaking predicates for search problems. In *Proceedings of KR'96*, pages 148–159. Morgan Kaufmann, 1996.
3. T. Fahle, S. Schamberger, and M. Sellmann. Symmetry breaking. In T. Walsh, editor, *Proceedings of CP'01*, volume 2239 of *LNCS*, pages 93–107. Springer-Verlag, 2001.
4. P. Flener, A. M. Frisch, B. Hnich, Z. Kızıltan, I. Miguel, J. Pearson, and T. Walsh. Breaking row and column symmetries in matrix models. In P. Van Hentenryck, editor, *Proceedings of CP'02*, volume 2470 of *LNCS*, pages 462–476. Springer-Verlag, 2002.
5. P. Flener, J. Pearson, M. Sellmann, and P. Van Hentenryck. Static and dynamic structural symmetry breaking. In F. Benhamou, editor, *Proceedings of CP'06*, volume 4204 of *LNCS*, pages 695–699. Springer-Verlag, 2006.
6. P. Flener, J. Pearson, M. Sellmann, P. Van Hentenryck, and M. Ågren. Dynamic structural symmetry breaking for constraint satisfaction problems. *Constraints*, forthcoming. At `dx.doi.org/10.1007/s10601-008-9059-7`. (Union and extension of [18] and [17].)
7. F. Focacci and M. Milano. Global cut framework for removing symmetries. In T. Walsh, editor, *Proceedings of CP'01*, volume 2239 of *LNCS*, pages 77–92. Springer-Verlag, 2001.
8. I. P. Gent and B. M. Smith. Symmetry breaking during search in constraint programming. In *Proceedings of ECAI'00*, pages 599–603, IOS Press, 2000.
9. D. Heller, A. Panda, M. Sellmann, and J. Yip. Model restarts for structural symmetry breaking. In P. J. Stuckey, editor, *Proceedings of CP'08*, volume 5202 of *LNCS*, pages 539–544. Springer-Verlag, 2008.
10. J.-F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. In J. Komorowski and Z. Raś, editors, *Proceedings of ISMIS'93*, volume 689 of *LNAI*, pages 350–361. Springer-Verlag, 1993.
11. J.-F. Puget. Constraint programming next challenge: Simplicity of use. In M. Wallace, editor, *Proceedings of CP'04*, volume 3258 of *LNCS*, pages 5–8. Springer-Verlag, 2004.

12. J.-F. Puget. Automatic detection of variable and value symmetries. In P. van Beek, editor, *Proceedings of CP'05*, volume 3709 of *LNCS*, pages 475–489. Springer-Verlag, 2005.
13. J.-F. Puget. An efficient way of breaking value symmetries. In *Proceedings of AAAI'06*. AAAI Press, 2006.
14. J.-F. Puget. Dynamic lex constraints. In F. Benhamou, editor, *Proceedings of CP'06*, volume 4204 of *LNCS*, pages 453–467. Springer-Verlag, 2006.
15. J.-C. Régin. Generalized arc-consistency for global cardinality constraint. In *Proceedings of AAAI'96*, pages 209–215. AAAI Press, 1996.
16. C. M. Roney-Dougal, I. P. Gent, T. Kelsey, and S. Linton. Tractable symmetry breaking using restricted search trees. In R. L. de Mántaras and L. Saitta, editors, *Proceedings of ECAI'04*, pages 211–215. IOS Press, 2004.
17. M. Sellmann and P. Van Hentenryck. Structural symmetry breaking. In *Proceedings of IJCAI'05*. 2005.
18. P. Van Hentenryck, P. Flener, J. Pearson, and M. Ågren. Tractable symmetry breaking for CSPs with interchangeable values. In *Proceedings of IJCAI'03*, pages 277–282. Morgan Kaufmann, 2003.
19. P. Van Hentenryck, P. Flener, J. Pearson, and M. Ågren. Compositional derivation of symmetries for constraint satisfaction. In J.-D. Zucker and L. Saitta, editors, *Proceedings of SARA'05*, volume 3607 of *LNCS*, pages 234–247. Springer-Verlag, 2005.
20. T. Walsh. Breaking value symmetry. In Ch. Bessière, editor, *Proceedings of CP'07*, volume 4741 of *LNCS*, pages 880–887. Springer-Verlag, 2007.