Variable Symmetry Breaking in Numerical Constraint Problems[1]

Alexandre Goldsztejn[a], Christophe Jermann[b], Vicente Ruiz de Angulo[c],
Carme Torras[c]

[a]*alexandre.goldsztejn@irccyn.ec-nantes.fr*
IRCCyN, CNRS, Nantes, France
[b]*christophe.jermann@univ-nantes.fr*
LINA, Université de Nantes / CNRS, Nantes, France
[c]*surname@iri.upc.edu*
IRI, CSIC / UPC, Barcelona, Spain

Abstract

Symmetry breaking has been a hot topic of research in the past years, leading to many theoretical developments as well as strong scaling strategies for dealing with hard applications. Most of the research has however focused on discrete, combinatorial, problems, and only few considered also continuous, numerical, problems. While part of the theory applies in both contexts, numerical problems have specificities that make most of the technical developments inadequate.

In this paper, we present the RLEX constraints, partial symmetry-breaking inequalities corresponding to a relaxation of the famous LEX constraints extensively studied in the discrete case. They allow (partially) breaking any variable symmetry and can be generated in polynomial time. Contrarily to LEX constraints that are impractical in general (due to their overwhelming number) and inappropriate in the continuous context (due to their form), RLEX constraints can be efficiently handled natively by numerical constraint solvers. Moreover, we demonstrate their pruning power on continuous domains is almost as strong as that of LEX constraints, and they subsume several previous work on breaking specific symmetry classes for continuous problems. Their experimental behavior is assessed on a collection of standard numerical problems and the factors influencing their impact are studied. The results confirm RLEX constraints are a dependable counterpart to LEX constraints for numerical problems.

Keywords: Constraint programming, Symmetries, Numerical constraints, Variable symmetries

---

[1]This paper is an extended version of [1] presented at the conference CP 2011.

1. Introduction

Numerical constraint solvers are nowadays beginning to be competitive and even to outperform, in some cases, classical methods for solving systems of equations and inequalities over the reals. As a consequence, their application has raised interest in fields as diverse as neurophysiology and economics [2], biochemistry, crystallography, robotics [3] and, more generally, in those related to global optimization [4]. Symmetries naturally occur in many of these applications, and it is advisable to exploit them in order to reduce the search space and, thus, to increase the efficiency of the solvers.

Numerical solvers follow the Branch&Prune scheme, similarly to discrete constraint solvers: At each iteration, a sub-domain is selected, pruned according to the constraints, and then split into several sub-domains to be further explored. The main differences with discrete solvers are that (sub)domains being continuous only their boundaries are contracted, and that a sub-domain is declared to be a solution whenever it reaches some prescribed computational precision. Because of the resemblance of the solving processes, it is tempting to port symmetry breaking methods designed for discrete Constraint Satisfaction Problems (CSPs) to numerical ones.

Considerable work on symmetry breaking has been done for discrete CSPs in the last decades [5, 6, 7]. Two main symmetry-breaking strategies have been pursued: 1) to devise specialized search algorithms that avoid symmetric portions of the search space [8, 9]; and 2) to add *symmetry-breaking constraints* (SBCs) that filter out redundant subspaces [10, 11]. Contrarily, there exists very little work on symmetry breaking for numerical problems. For cyclic variables permutations, an approach divides the initial space into boxes and eliminates symmetric ones before the solving starts [12]. SBCs have also been proposed, but only for either specific problems [13, 14] or specific symmetry classes [15, 16, 17], and often only partially breaking the considered symmetries.

In this paper, we propose the first general SBCs for numerical constraint problems that (partially) break any variable symmetry. These SBCs take the form of simple binary inequalities of the form $x_i \leq x_j$ where x_i and x_j are two distinct variables of the problem and $i < j$. Thus, at most $\frac{n(n-1)}{2}$ inequalities are generated to deal with any symmetry. They can be generated in polynomial time knowing a generator of the symmetry group, using classical group theory algorithms. Moreover we demonstrate that these SBCs are suitable and optimal for numerical problems, i.e., they enclose tightly an asymmetric search subspace and thus have a better, or similar, pruning power than other SBCs.

The outline of the paper is as follows: Section 2 provides the necessary background on numerical problems and symmetries; Section 3 introduces our SBCs as a relaxation of the lexicographic-ordering based SBCs [18] widely used by the discrete CSP community, and it also establishes the good properties of this relaxation; Section 4 introduces the state-of-the-art and compares our SBCs to existing alternatives; Section 5 assesses the practical interest of our SBCs on a benchmark of standard problems and analyzes the factors influencing their impact. Section 6 concludes the paper with future research directions.

2. Principles of Variable Symmetry Breaking

A CSP is defined as a triple $\langle x, d, c \rangle$, where $x = (x_1, \ldots, x_n)$ is a list of variables, $d = (d_1, \ldots, d_n)$ is a list of domains for the variables, and $c = (c_1, \ldots, c_m)$ is a list of constraints. The focus of this paper is on numerical CSPs (NCSPs), whose variables are continuous, and thus domains are subsets of $\mathbb{R}$, typically represented as a set of intervals (box) $\mathbf{x}$. To conform mathematical notations in use for numerical problems, the same symbols are used for variables and their valuations, i.e., x will often denote a point in $\mathbb{R}^n$. For the same reason, we adopt a functional notation for the evaluation of a constraint $c_i : \mathbb{R}^n \to \{\text{true}, \text{false}\}$ and for the evaluation of the conjunction of the constraints $c : \mathbb{R}^n \to \{\text{true}, \text{false}\}$. Hence a solution of a NCSP is a point $x \in \mathbf{x}$ that satisfies $c(x)$, and its solution set is $\chi = \{x \in \mathbf{x} : c(x)\}$.

A bijective function $s : \mathbb{R}^n \to \mathbb{R}^n$ is a *symmetry* of a (N)CSP if it maps solutions to solutions[2], i.e., for any $x \in \mathbf{x}$ such that $c(x) = \text{true}$, $s(x) \in \mathbf{x}$ and $c(s(x)) = \text{true}$. We say x and $s(x)$ are *symmetric points* and, in case they are solutions, *symmetric solutions*. The symmetries of a (N)CSP form a group for the composition law. This symmetry group is denoted Σ in the following. Though the identity function is forcibly part of Σ, since it is a trivial symmetry of any CSP, it is not considered in the following as it is irrelevant to break.

In this paper, we consider only symmetries that are permutations of variables. Let $\mathcal{S}_n$ be the set of all permutations of $\{1, \ldots, n\}$. The image of i by a permutation σ is denoted by i^σ. Any permutation σ is completely defined by the image of each integer in $\{1, \ldots, n\}$, and it is usually described as a vector $[1^\sigma, 2^\sigma, \ldots, n^\sigma]$. A symmetry s is a *variable symmetry* iff there is a permutation $\sigma \in \mathcal{S}_n$ such that for any point $x \in \mathbf{x}$, $s(x) = (x_{1^\sigma}, \ldots, x_{n^\sigma})$. We identify such symmetries with their associated permutations and denote both by σ in the following. Consequently, the group of variable symmetries of a CSP is isomorphic to a permutation subgroup of $\mathcal{S}_n$, which are both identified and denoted by Σ in the following. The application of a variable symmetry σ to a point x is denoted by x^σ, this notation being extended to sets (discrete or continuous) of points $X \subseteq \mathbf{x}$ by $X^\sigma = \{x^\sigma : x \in X\}$.

Example 1. The 3-cyclic roots problem consists in finding all $(x_1, x_2, x_3) \in \mathbb{R}^3$ satisfying $(x_1 + x_2 + x_3 = 0) \wedge (x_1 x_2 + x_2 x_3 + x_3 x_1 = 0) \wedge (x_1 x_2 x_3 = 1)$. This problem has six variable symmetries (including identity):

$$\Sigma = \{[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]\}, \tag{1}$$

i.e., $\Sigma = \mathcal{S}_3$. Indeed, all the variables are *syntactically* interchangeable within all the constraints by the commutativity laws of product and sum.

We can define the *symmetry class* of any point $x \in \mathbf{x}$ as the set of the images $s(x)$ for all the symmetries s of the problem. In the case of variable symmetries,

---

[2]Nothing is required for non-solution points, i.e., we consider only *solution symmetries* [19].

it can be noted $x^\Sigma = \{x^\sigma : \sigma \in \Sigma\}$. Provided that the domain satisfy the symmetries, i.e. $\mathbf{x} = \mathbf{x}^\sigma$ for all $\sigma \in \Sigma$, being in the same class is an equivalence relation and thus the symmetry classes form a partition of $\mathbf{x}$.

The symmetries of a (N)CSP are broken when a single representative in each symmetry class is retained. To this end, it is possible to add *symmetry-breaking constraints* (SBCs) which exclude all but a single representative in each symmetry class [5, 7, 20]. Crawford *et al.* [18] proposed *lexicographic ordering constraints* (LEX) to address variable symmetry breaking. Recall that given x and y in $\mathbb{R}^n$, the lexicographic order is defined inductively as:

$$\begin{aligned} \text{for } n = 1, \quad & x \preceq_{lex} y \;\equiv\; (x_1 \le y_1) \\ \text{for } n > 1, \quad & x \preceq_{lex} y \;\equiv\; (x_1 < y_1) \;\vee\; \Big((x_1 = y_1) \;\wedge\; (x_{2:n} \preceq_{lex} y_{2:n})\Big) \end{aligned} \qquad (2)$$

where, given a list $z = (z_1, z_2, \ldots, z_n)$, $z_{i:j}$ denotes the sublist $(z_i, z_{i+1}, \ldots, z_j)$.

Given a variable symmetry σ, Crawford *et al.* define the corresponding SBC

$$\text{LEX}_\sigma(x) \;\equiv\; x \preceq_{lex} x^\sigma. \qquad (3)$$

Intuitively, this constraint imposes a total order on the symmetric points, hence allowing to retain a single one w.r.t. a given symmetry σ. One such constraint is thus imposed for each of the symmetries of a problem in order to break them all, yielding the complete symmetry-breaking constraint

$$\text{LEX}_\Sigma(x) \;\equiv\; \bigwedge_{\sigma \in \Sigma} \text{LEX}_\sigma(x) \qquad (4)$$

Example 2. The LEX constraints for the symmetries of the 3-cyclic-roots problem are:

$$\begin{aligned} (x_1, x_2, x_3) \preceq_{lex} (x_1, x_3, x_2) \;, \quad & (x_1, x_2, x_3) \preceq_{lex} (x_2, x_1, x_3) \\ (x_1, x_2, x_3) \preceq_{lex} (x_2, x_3, x_1) \;, \quad & (x_1, x_2, x_3) \preceq_{lex} (x_3, x_1, x_2) \\ (x_1, x_2, x_3) \preceq_{lex} (x_3, x_2, x_1). & \end{aligned}$$

LEX constraints can be efficiently propagated over finite domains (e.g., [21]). However, this approach has the critical drawback that the number of symmetries in Σ can be factorial in the number of variables even in some simple situations (e.g., $|\mathcal{S}_n| = n!$). While in general the corresponding SBCs cannot be simplified to a polynomial number of constraints (in line with the NP-hardness of symmetry breaking proved in [18] and further explored in [20]), simplifying specific symmetry groups into tractable ones has been the topic of several works in the symmetry breaking literature (e.g., [18, 11, 22]). For instance, Crawford *et al.* noted that when $\Sigma = \mathcal{S}_n$, the $n!$ terms in LEX_Σ can be simplified to the $n-1$ inequalities $\bigwedge_{i \in \{1, \ldots, n-1\}} x_i \le x_{i+1}$.

3. Symmetry-Breaking Constraints for Numerical CSPs

In this section we first explain the reason why LEX SBCs are impractical for numerical CSPs. We also point out the intuition behind the partial SBCs we introduce, and we discuss their properties in the rest of the section.
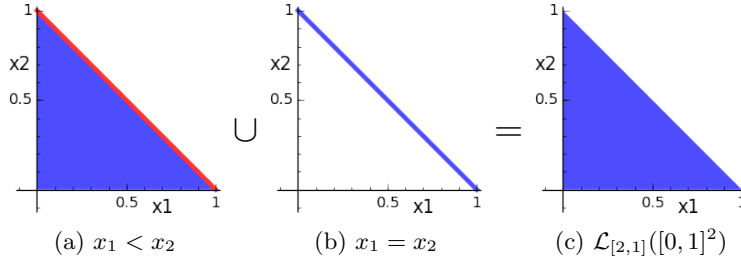
Figure 1: The LEX$_{[2,1]}$ constraint components in 2D (blue=included, red=excluded).

3.1. LEX *intractability and impracticality for NCSPs*

The first and foremost reason why LEX SBCs are intractable, whether domains are continuous or discrete, is because they can be overwhelmingly numerous. However, even when the symmetry group is of reasonable size or when LEX SBCs can be reduced to a reasonable number, they remain impractical in the context of NCSPs.

A geometric interpretation of LEX constraints highlights their complexity and indicates the intuition behind the relaxation we propose in the next section. Let us denote $\mathcal{L}_\sigma(\mathbf{x})$ (resp. $\mathcal{L}_\Sigma(\mathbf{x})$) the solution set of a LEX constraint breaking a symmetry σ (resp. a whole symmetry group Σ) on a domain $\mathbf{x}$, i.e.,

$$\mathcal{L}_\sigma(\mathbf{x}) := \{x \in \mathbf{x} : \text{LEX}_\sigma(x)\}, \tag{5}$$
$$\mathcal{L}_\Sigma(\mathbf{x}) := \{x \in \mathbf{x} : \text{LEX}_\Sigma(x)\}, \tag{6}$$

For convenience, we will omit the domain $\mathbf{x}$ when it spans the whole set of real vectors $\mathbb{R}^n$.

Breaking the symmetries of a NCSP by adding the SBC LEX$_\Sigma(x)$ to its constraints allows restricting its search space to $\mathcal{L}_\Sigma(\mathbf{x})$, hence retaining only one solution by symmetric class. We first remark that LEX$_\Sigma(x)$ being the conjunction of all LEX$_\sigma(x)$, $\mathcal{L}_\Sigma(\mathbf{x})$ is thus the intersection of all $\mathcal{L}_\sigma(\mathbf{x})$, i.e.,

$$\mathcal{L}_\Sigma(\mathbf{x}) = \bigcap_{\sigma \in \Sigma} \mathcal{L}_\sigma(\mathbf{x}) \tag{7}$$

Second, the recursive definition of the lexicographic relation in Equation (2) indicates that the form of each LEX$_\sigma(\mathbf{x})$ is the union of 1) an open half-space $(x_1 < x_{1^\sigma})$ and 2) a portion of its boundary "plane" $(x_1 = x_{1^\sigma})$, this portion being itself recursively defined similarly as the union of 3) a half-plane $(x_2 < x_{2^\sigma})$ and 4) a portion of its boundary "line", etc. Hence, the structure of $\mathcal{L}_\sigma(\mathbf{x})$ becomes more and more complicated as the number of variables increases.

Example 3. When $n = 2$ the only possible symmetry is the interchangeability of x_1 and x_2, i.e., permutation $[2, 1]$. It yields the LEX constraint

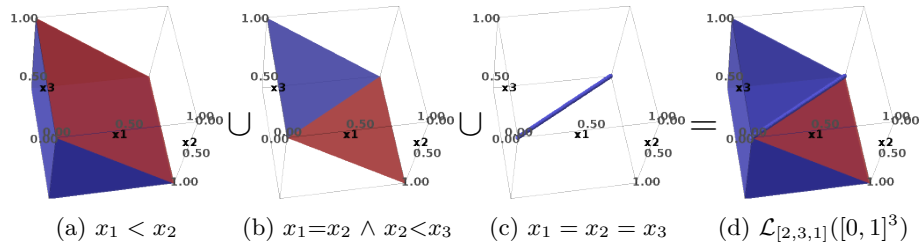$$x_1 < x_2 \vee (x_1 = x_2 \wedge (x_2 < x_1 \vee (x_2 = x_1))), \tag{8}$$

5

Figure 2: The LEX$_{[2,3,1]}$ constraint components in 3D (blue=included, red=excluded).

which boils down to $x_1 \leq x_2$, i.e., the union of 1) the open half-plane $x_1 < x_2$ with 2) the line $x_1 = x_2$, as depicted in Figure 1.

Things get more complicated when $n = 3$. Consider for instance permutation $[2, 3, 1]$ and the corresponding LEX SBC

$$x_1 < x_2 \vee (x_1 = x_2 \wedge (x_2 < x_3 \vee (x_2 = x_3 \wedge (x_3 < x_1 \vee (x_3 = x_1))))), \quad (9)$$

which simplifies to

$$(x_1 < x_2) \ \vee \ (x_1 = x_2 \wedge x_2 < x_3) \ \vee \ (x_1 = x_2 = x_3), \quad (10)$$

i.e., the union of 1) the open half-space $(x_1 < x_2)$ with 2) the open half-plane $(x_1 = x_2 \wedge x_2 < x_3)$ and with 3) the line $(x_1 = x_2 = x_3)$ as depicted in Figure 2.

Though it becomes impossible to depict the geometry of a LEX constraint when $n > 3$, one can easily imagine its general structure, and thus its complexity in the numerical context.

As a consequence, the geometric shape of $\mathcal{L}_\Sigma(\mathbf{x})$ is the intersection of a (potentially factorial) number of complex asymmetric subspaces $\mathcal{L}_\sigma(\mathbf{x})$, resulting in a very complex subspace having the form of a polyhedral cone, whose essential complexity lies on the boundaries while its main part is the simple intersection of open half-spaces $x_i < x_j$, whose number is at most $\frac{n(n-1)}{2}$ in dimension n.

Example 4. $\mathcal{L}_{\mathcal{C}_3}$ for the cyclic symmetry group in 3D is the intersection of two subspaces corresponding to LEX$_{[2,3,1]}$ and LEX$_{[3,1,2]}$, as depicted in Figure 3.

In conclusion, the complexity of LEX SBCs[3], which makes them impractical for NCSPs, lays in the boundaries of their solution sets while these boundaries are somehow negligible in numerical domains (this will be formalized in Subsection 3.3). In the following subsection, we propose a relaxation of LEX constraints that consists in accepting all their boundaries, leading to partial SBCs much better suited to NCSPs.

---

[3]Note that various formulations of LEX constraints are proposed in the literature, but in the continuous context they are either equivalent to the considered formulation (3) and thus define an identical, complex, half-open, subspace, or they are not equivalent (see Section 4.4 for more details).
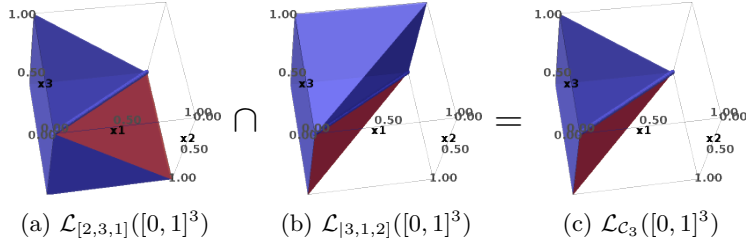
(a) $\mathcal{L}_{[2,3,1]}([0,1]^3)$ (b) $\mathcal{L}_{[3,1,2]}([0,1]^3)$ (c) $\mathcal{L}_{\mathcal{C}_3}([0,1]^3)$

Figure 3: The LEX constraint components for the cyclic symmetry group in 3D. In blue, inner components, in red, excluded boundaries.

3.2. Definition of the RLEX partial SBCs

Partial symmetry-breaking constraints (PSBCs) can be defined by considering a tractable subset of complete symmetry-breaking constraints [22]. The key idea of our approach is instead to simplify every LEX SBC by considering the closure of the geometric space it defines, i.e. the main open half-space together with its boundary. The main half-space of an SBC LEX$_\sigma$ corresponds to the inequality $x_{\hat\sigma} < x_{\hat\sigma^\sigma}$, where $\hat\sigma$ is the smallest integer in $\{1,\dots,n\}$ such that $\hat\sigma \neq \hat\sigma^\sigma$ (e.g., for $\sigma = [1,2,3,5,4]$ we have $\hat\sigma = 4$). Note that this integer is well-defined for all symmetries except the identity permutation, which cannot be broken and is thus disregarded in the following. We define the *relaxed* LEX constraint by

$$\text{RLEX}_\sigma(x) \;\equiv\; x_{\hat\sigma} < x_{\hat\sigma^\sigma} \vee x_{\hat\sigma} = x_{\hat\sigma^\sigma} \;\iff\; x_{\hat\sigma} \leq x_{\hat\sigma^\sigma}. \tag{11}$$

Accordingly, the acceptance of x by the whole set of RLEX constraints is

$$\text{RLEX}_\Sigma(x) \;\equiv\; \bigwedge_{\sigma\in\Sigma} \text{RLEX}_\sigma(x). \tag{12}$$

The following proposition shows that this is indeed a relaxation of LEX, and that it can thus be used as a PSBC.

Proposition 1. $\text{LEX}_\sigma(x) \implies \text{RLEX}_\sigma(x)$

PROOF. Since $i < \hat\sigma$ implies $i = i^\sigma$, we have $x_i = x_{i^\sigma}$ for all $i < \hat\sigma$. Therefore $\text{LEX}_\sigma(x)$, which is $x \preceq_{lex} x^\sigma$, is actually equivalent to $x_{\hat\sigma:n} \preceq_{lex} x_{\hat\sigma:n}{}^\sigma$, i.e., $(x_{\hat\sigma} < x_{\hat\sigma^\sigma}) \vee ((x_{\hat\sigma} = x_{\hat\sigma^\sigma}) \wedge (x_{\hat\sigma+1:n} \preceq_{lex} x_{\hat\sigma+1:n}{}^\sigma))$, which logically implies $(x_{\hat\sigma} < x_{\hat\sigma^\sigma}) \vee (x_{\hat\sigma} = x_{\hat\sigma^\sigma})$, that is $\text{RLEX}_\sigma(x)$. □

Similarly to what we have done for LEX constraints, we define the portions of a domain $\mathbf{x}$ satisfying RLEX constraints as

$$\mathcal{R}_\sigma(\mathbf{x}) \;:=\; \{x \in \mathbf{x} : \text{RLEX}_\sigma(x)\}, \tag{13}$$

$$\mathcal{R}_\Sigma(\mathbf{x}) \;:=\; \{x \in \mathbf{x} : \text{RLEX}_\Sigma(x)\}, \tag{14}$$

For convenience, we will omit the domain $\mathbf{x}$ when it spans the whole set of real vectors $\mathbb{R}^n$.
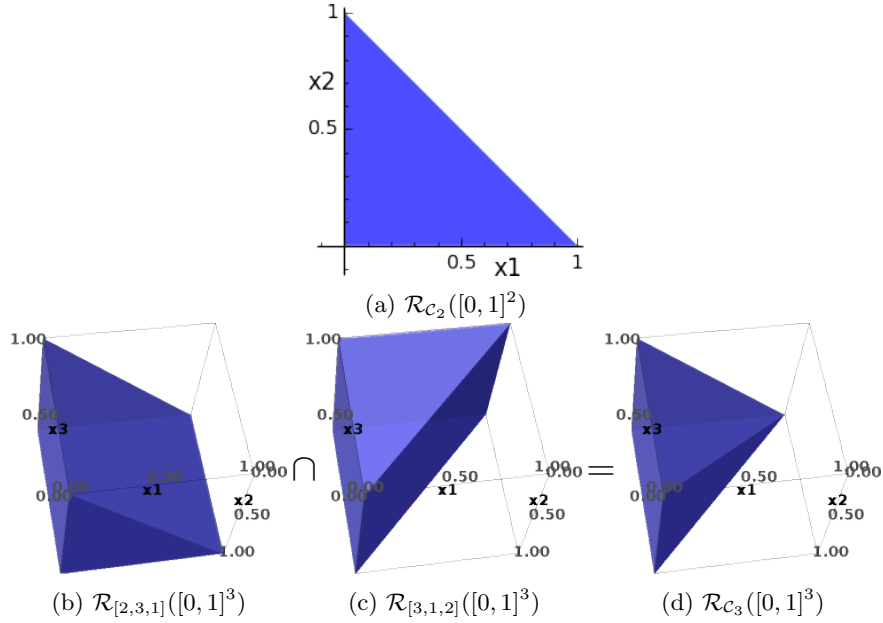
7

Figure 4: RLEX constraints for the cyclic symmetry groups in 2D and 3D.

Example 5. The RLEX constraint for the cyclic symmetry groups $\mathcal{C}_2$ is just RLEX$_{[2,1]} := x_1 \leq x_2$. That for $\mathcal{C}_3$ are RLEX$_{[2,3,1]} := x_1 \leq x_2$ and RLEX$_{[3,1,2]} := x_1 \leq x_3$. The corresponding domains are depicted in Figure 4, which illustrates their relation to the corresponding LEX constraints (see Figures 1 and 3 above).

Also, the RLEX constraints corresponding to Example 2 are the two inequalities $x_1 \leq x_2$ and $x_2 \leq x_3$ ($x_1 \leq x_3$ also appears, but is redundant).

3.3. Properties of the RLEX PSBCs

The RLEX PSBCs are now compared to the LEX SBCs. We first show in Subsection 3.3.1 that RLEX are suitable for NCSPs. We also provide a detailed analysis of the difference between RLEX PSBCs and LEX SBCs in Subsection 3.3.2, indicating that the former are in fact complete SBCs, just as powerful as the latter, in many cases.

3.3.1. On the suitability of RLEX constraints for NCSPs

We now establish some properties of the RLEX PSBCs that make them suitable for breaking variable symmetries in NCSPs, i.e., both reasonable in number whatever the symmetry group, natively handleable by numerical constraint solvers, and optimal in terms of pruning capabilities.

Number of constraints and their generation. There are at most $\frac{n(n-1)}{2}$ RLEX constraints whatever the symmetry group since they all have the form $x_i \leq x_j$

with $i < j$. This number can even be reduced to $n - 1$ using transitivity reduction [23] and according to Theorem 4 in [24].

The naive way of generating RLEX PSBCs is to generate all LEX SBCs and relax them. However, this entails generating all LEX SBCs, which is intractable. The very same indexes of variables are used in [24] where SBCs of the form $x_{\hat{\sigma}} < x_{\hat{\sigma}^\sigma}$ are derived for discrete CSPs including an *alldiff* global constraint. In that paper, a polynomial-time algorithm is proposed to generate these variables index, based on *Shreier-Sims* group theory algorithm [25]. This method can be used to generated RLEX constraints as well. The relationship between RLEX PSBCs and the SBCs proposed in [24] is further investigated in Subsection 4.1.

Pruning optimality for numerical solvers. At the heart of the unsuitability of LEX constraints lies the open-set nature of the subspace they delimit. Numerical constraint solvers based on intervals[4] cannot take advantage of the difference between the subspace defined by the problem constraints and its closure. We now prove that RLEX constraints define just the closure of the subspace defined by LEX constraints, which implies that numerical constraint solvers cannot distinguish LEX and RLEX constraints in general[5].

Proposition 2. $\mathcal{R}_\Sigma = \mathrm{cl}(\mathcal{L}_\Sigma)$

PROOF. Consider the interior and closure sets of the subspace defined by a single LEX constraint. Both $\mathrm{int}(\mathcal{L}_\sigma) = \{x \in \mathbb{R}^n : x_{\hat{\sigma}} < x_{\hat{\sigma}^\sigma}\}$ and $\mathrm{cl}(\mathcal{L}_\sigma) = \{x \in \mathbb{R}^n : x_{\hat{\sigma}} \le x_{\hat{\sigma}^\sigma}\} = \mathcal{R}_\sigma$ obviously hold for any permutation σ. By definition, $\mathcal{R}_\Sigma = \cap_{\sigma \in \Sigma} \mathcal{R}_\sigma$ and $\mathcal{L}_\Sigma = \cap_{\sigma \in \Sigma} \mathcal{L}_\sigma$, therefore $\mathcal{R}_\Sigma \supseteq \mathrm{cl}(\mathcal{L}_\Sigma)$ is a direct consequence of the property that the closure of an intersection is a subset of the intersection of the closures. For the reverse inclusion, note that $y = (1, 2, 3, 4, \ldots, n) \in \mathrm{int}(\mathcal{L}_\Sigma)$, and consider an arbitrary $x \in \mathcal{R}_\Sigma$. For each $\sigma \in \Sigma$ we thus have $y_{\hat{\sigma}} < y_{\hat{\sigma}^\sigma}$ and $x_{\hat{\sigma}} \le x_{\hat{\sigma}^\sigma}$. For $\lambda \in [0, 1]$, define $z^\lambda = \lambda x + (1 - \lambda)y$, and note that for $0 \le \lambda < 1$ we have both $\lambda x_{\hat{\sigma}} \le \lambda x_{\hat{\sigma}^\sigma}$ and $(1 - \lambda)y_{\hat{\sigma}} < (1 - \lambda)y_{\hat{\sigma}^\sigma}$, which implies $z^\lambda_{\hat{\sigma}} < z^\lambda_{\hat{\sigma}^\sigma}$ and thus $z^\lambda \in \mathrm{int}(\mathcal{L}_\sigma)$. As this holds for each $\sigma \in \Sigma$, we conclude that $z^\lambda \in \mathrm{int}(\mathcal{L}_\Sigma)$. As a consequence, $x = \lim_{\lambda \to 1} z^\lambda \in \mathrm{cl}(\mathrm{int}(\mathcal{L}_\Sigma)) \subseteq \mathrm{cl}(\mathcal{L}_\Sigma)$, which proves the reverse inclusion. □

Finally, we remark that RLEX constraints must be more efficient to propagate than LEX constraints. Indeed, they are just binary inequalities while LEX constraints involve all the variables of the symmetries in a large combination of logical operations. Hence, pruning RLEX constraints requires no specific algorithm since simple non-strict inequalities are natively handled by most numerical

---

[4]Interval computations are typically based on closed intervals, computations with open sets being complicated, expensive, and often useless due to outward rounding necessity. Note that typical numerical algorithms do not handle strict inequalities neither.

[5]Singular cases, e.g., when the considered box has some interval dimensions degenerated into a single real, could in theory make a difference between LEX and RLEX constraints, provided the solver implements LEX constraints filtering which is not the case of any numerical solver we know of.

solvers; and their propagation, following the AC3-like fix-point process in use in most numerical constraint solvers, triggers only constraints depending on two variables at most instead of n at most (entailing all constraints are enqueued).

Search-space reduction factor. In the context of NCSPs, the size of the search space cannot be measured using the number of elements it contains, since there are infinitely many elements in any continuous set S. Instead, we can quantify it using its hypervolume[6], denoted $\operatorname{vol} S$. For example, the search space $([-1,1],[-1,1],[-1,1])$ has a (hyper)volume equal to 8, while its half $([0,1],[-1,1],[-1,1])$ has a (hyper)volume equal to 4.

The aim of this section is to study the hypervolume reduction of the search space provided by the RLEX constraints in comparison to the corresponding LEX constraints. As RLEX is a relaxation of LEX, we have $\mathcal{L}_\Sigma(\mathbf{x}) \subseteq \mathcal{R}_\Sigma(\mathbf{x})$. In fact, we have a more accurate relationship between these sets: Informally, any solution accepted by some RLEX_σ constraint and rejected by the corresponding LEX_σ constraint has to have at least two equal components, namely $x_{\hat{\sigma}} = x_{\hat{\sigma}^\sigma}$. We define the set of vectors that have at least two equal components as $H := \{x \in \mathbb{R}^n : \exists(i,j), i \neq j \wedge x_i = x_j\}$, and note that $\operatorname{vol} H = 0$ since it is only composed of (hyper)planes, i.e., *flat* sets. We therefore see that LEX_Σ and RLEX_Σ solution sets have the same hypervolume, which is proved in the following proposition.

Proposition 3. $\operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}) = \operatorname{vol} \mathcal{R}_\Sigma(\mathbf{x})$.

PROOF. Since $\mathcal{L}_\Sigma(\mathbf{x}) \subseteq \mathcal{R}_\Sigma(\mathbf{x})$ we have $\operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}) \leq \operatorname{vol} \mathcal{R}_\Sigma(\mathbf{x})$. Now note that $x \in \mathcal{R}_\Sigma(\mathbf{x}) \wedge x \notin \mathcal{L}_\Sigma(\mathbf{x})$ obviously implies $x \in H$, so $\mathcal{R}_\Sigma(\mathbf{x}) \subseteq \mathcal{L}_\Sigma(\mathbf{x}) \cup H$ and hence using the rule for measuring unions of sets $\operatorname{vol} \mathcal{R}_\Sigma(\mathbf{x}) \leq \operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}) + \operatorname{vol} H - \operatorname{vol}(\mathcal{L}_\Sigma(\mathbf{x}) \cap H)$. Since $\operatorname{vol} H = 0$, $\operatorname{vol}(\mathcal{L}_\Sigma(\mathbf{x}) \cap H) = 0$ and we obtain $\operatorname{vol} \mathcal{R}_\Sigma(\mathbf{x}) \leq \operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x})$. $\square$

The following proposition quantifies the reduction performed by the LEX constraints, therefore by RLEX constraints as well by Proposition 3, in term of hypervolume.

Proposition 4. *Provided the domain $\mathbf{x}$ is symmetric, i.e. $\mathbf{x} = \mathbf{x}^\sigma$ for all $\sigma \in \Sigma$,*

$$\operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}) = \frac{\operatorname{vol} \mathbf{x}}{|\Sigma|}.$$

PROOF. We suppose that $|\Sigma| \geq 2$, otherwise $\Sigma = \{id\}$ and the statement is trivial. First note that $\operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}) = \operatorname{vol}(\mathcal{L}_\Sigma(\mathbf{x})^\sigma)$ because the map $x \rightarrow x^\sigma$ is an isometry and thus preserves the measure. Now note that for $x \in \mathcal{L}_\Sigma(\mathbf{x})$ and $\sigma, \phi \in \Sigma$, $\sigma \neq \phi$ obviously implies $x^\sigma \neq x^\phi$ or $x \in H$ (indeed if $x^\sigma = x^\phi$ then $x = x^{\phi\sigma^{-1}}$ and hence $x \in H$). Therefore, $\mathcal{L}_\Sigma(\mathbf{x})^\sigma \cap \mathcal{L}_\Sigma(\mathbf{x})^\phi \subseteq H$. Recalling that $\operatorname{vol} H = 0$, we obtain using the rule for measuring unions of sets

$$\operatorname{vol}\left(\bigcup_{\sigma \in \Sigma} \mathcal{L}_\Sigma(\mathbf{x})^\sigma \right) = |\Sigma| \operatorname{vol} \mathcal{L}_\Sigma(\mathbf{x}). \tag{15}$$

---

[6]A generalization of the area in 2D and the volume in 3D, also called Lebesgue measure.

It remains to prove that $\bigcup_{\sigma \in \Sigma} \mathcal{L}_\Sigma(\mathbf{x})^\sigma = \mathbf{x}$. Since the symmetry classes form a partition of $\mathbf{x}$ and $\mathcal{L}_\Sigma(\mathbf{x})$ contains exactly one representative of each symmetry class, we have

$$\bigcup_{x \in \mathcal{L}_\Sigma(\mathbf{x})} \bigcup_{\sigma \in \Sigma} x^\sigma = \mathbf{x}. \tag{16}$$

The proof is concluded by commuting the two unions. $\qquad\qquad\qquad \square$

By Proposition 3, we can conclude that the same hypervolume reduction is enforced by the RLEX constraints. In other words, RLEX and LEX constraints reduce the search space to a portion with identical hypervolume, the difference lying only on the boundaries of the remaining search space. The RLEX PSBCs are optimal in this regard.

3.3.2. A detailed look at the difference between RLEX *and* LEX *filtering powers*

LEX are always SBCs while RLEX are PSBCs, hence the former have in principle a superior filtering power than the latter. However, in a continuous context, RLEX act as SBCs in a set of points occupying the whole domain volume. Indeed, we have already explained that a point retained by RLEX and not retained by LEX must belong to the set H. Note also that all points of a class lies either in H or in $\overline{H}$. Hence RLEX are SBCs in $\overline{H}$. Since H has no volume, the set $\overline{H}$ for which RLEX can be guaranteed to be SBCs has the same volume as the domain. At the end of this section we will further conclude that RLEX are also SBCs in a non-empty subset of H.

The number of symmetric points of x satisfying RLEX constraints is $|\mathcal{R}_\Sigma(x^\Sigma)|$, and we denote by $|x^\Sigma| = |\{x^\sigma : \sigma \in \Sigma\}|$ the number of different elements in the symmetry class of x. Both indicators are indeed properties of the symmetry class of the point rather than properties of the point itself. The pruning power of a PSBC on the symmetry class of a point x is optimal when $|\mathcal{R}_\Sigma(x^\Sigma)| = 1$ and $|x^\Sigma| = |\Sigma|$. Indeed, in this case the PSBC acts as an SBC ($|\mathcal{R}_\Sigma(x^\Sigma)| = 1$) and allows the greatest possible reduction ($\frac{|\mathcal{R}_\Sigma(x^\Sigma)|}{|x^\Sigma|} = \frac{1}{|\Sigma|}$). By abuse of language, we say a point x is RLEX-optimal if it belongs to an optimally reduced symmetry class. Therefore, any point x in $\overline{H}$ is RLEX-optimal since RLEX being SBCs means that $|\mathcal{R}_\Sigma(x^\Sigma)| = 1$, and $x \in \overline{H}$ means x does not have equal components, thus implying that all its permutations are different, i.e., $|x^\Sigma| = |\Sigma|$.

With a more detailed analysis we will show that RLEX-optimality, and therefore the fact RLEX perform as SBCs, can be guaranteed for a set larger than $\overline{H}$, i.e., even in a subset of H. At the same time we will characterize completely RLEX-optimal points.

From a topological point of view, we can divide the subspace $\mathcal{R}_\Sigma(\mathbf{x})$ into two sets: *interior points* and *frontier points*. Interior points are those satisfying the strict part of the inequalities of RLEX (i.e., points satisfying the constraints in [24]). Frontier points are those for which the equality in some of the RLEX holds. The main result we will prove in this subsection is that interior points and optimal points are coincident.

For this purpose, we will rely mainly on an intermediate characterization, Proposition 6, stating that the class of a point is optimal if and only if every symmetry of the point but one is rejected by RLEX. Another interesting result in the way is Proposition 7: any symmetry of an interior point (except identity) will be rejected by RLEX. Combined with the former it implies directly the main result, Proposition 8. The following statement is required by Proposition 6:

Proposition 5. *All elements of x^Σ are generated with the same number of symmetries of Σ, i.e., $|\{\sigma \in \Sigma : x^\sigma = y\}| = |\{\sigma \in \Sigma : x^\sigma = z\}|, \ \forall \ y, z \in x^\Sigma$.*

PROOF. Let $\Xi = \{\sigma \in \Sigma : x^\sigma = y\}$ and $\Upsilon = \{\sigma \in \Sigma : x^\sigma = z\}$ and let $\sigma_1 \in \Xi$, $\sigma_2 \in \Upsilon$ be elements of these sets, i.e., $x^{\sigma_1} = y$, $x^{\sigma_2} = z$. Then $\phi = \sigma_1^{-1} \cdot \sigma_2$ is a permutation that transforms y into z, because $y^\phi = y^{\sigma_1^{-1} \cdot \sigma_2} = (x^{\sigma_1})^{\sigma_1^{-1} \cdot \sigma_2} = x^{\sigma_2} = z$. As a consequence, $\forall \sigma \in \Xi$ we have $x^{\sigma \cdot \phi} = y^\phi = z$. Therefore, every permutation in $\Xi \ \phi$ belongs to Υ. The application $f_\phi(\sigma) = \sigma\phi$ from Σ to Σ is known to be bijective, hence it must also be bijective in the restricted Ξ domain, which implies $|\Xi \ \phi| = |\Xi|$. Therefore Υ has at least $|\Xi|$ symmetries, $|\Upsilon| \geq |\Xi|$. Inverting the roles of y and z and those of Υ and Ξ at the beginning of the proof, we get $|\Upsilon| \leq |\Xi|$, leading to $|\Upsilon| = |\Xi|$. $\square$

In the next step we characterize optimal and non-optimal classes with the following proposition.

Proposition 6. *The class x^Σ of a point $x \in \mathbb{R}^n$ is optimal, i.e.,*

1. *$|x^\Sigma| = |\Sigma|$ and $|\mathcal{R}_\Sigma(x^\Sigma)| = 1$*
   if and only if
2. *$\exists \sigma_1 \in \Sigma : \text{RLEX}_\Sigma(x^{\sigma_1}) = true$ and $\forall \sigma_2 \in \Sigma \setminus \sigma_1, \ \text{RLEX}_\Sigma(x^{\sigma_2}) = false$.*

PROOF. We first prove $1 \Rightarrow 2$ by contradiction: suppose $\exists \ \sigma_1, \sigma_2, \ \sigma_1 \neq \sigma_2$ such that $\text{RLEX}_\Sigma(x^{\sigma_1}) = \text{RLEX}_\Sigma(x^{\sigma_2}) = true$. There are two possibilities. If $x^{\sigma_1} = x^{\sigma_2}$ then, since the two symmetries are different, we have $|x^\Sigma| < |\Sigma|$ and we have arrived to a contradiction. In the opposite case, $x^{\sigma_1} \neq x^{\sigma_2}$, since both symmetries are accepted, we have $|\mathcal{R}_\Sigma(x^\Sigma)| > 1$, which is again a contradiction.

Second, we prove $2 \Rightarrow 1$: since σ_1 is the only symmetry accepted by RLEX, obviously $|\mathcal{R}_\Sigma(x^\Sigma)| = 1$. For the other statement, we note that $|x^\Sigma| = |\Sigma|$ is equivalent to $x^{\sigma_1} \neq x^{\sigma_2} \ \forall \ \sigma_1, \ \sigma_2 \in \Sigma, \sigma_1 \neq \sigma_2$. Since σ_1 is the only permutation of x that is accepted by RLEX, no other permutation can yield the same result as σ_1. Therefore, $|\{\sigma \in \Sigma : x^\sigma = x^{\sigma_1}\}| = 1$. By Proposition 5, this leads to $|\{\sigma \in \Sigma : x^\sigma = x^\phi\}| = 1 \ \forall \phi \in \Sigma$, which implies that each $\phi \in \Sigma$ yields a result different from all the other permutations in the group, i.e., $x^{\sigma_1} \neq x^{\sigma_2} \ \forall \ \sigma_1, \ \sigma_2 \in \Sigma, \sigma_1 \neq \sigma_2$, as was required. $\square$

As a consequence, the class of x is non-optimal iff $\exists \ \sigma_1, \sigma_2, \ \sigma_1 \neq \sigma_2$ such that $\text{RLEX}_\Sigma(x^{\sigma_1}) = \text{RLEX}_\Sigma(x^{\sigma_2}) = true$.

Proposition 7. *x is interior $\Rightarrow \ \forall \ \sigma \in \Sigma \setminus Id, \ \text{RLEX}_\Sigma(x^\sigma) = false$.*

PROOF. Let's define $x \prec_{lex} y$ as in (2), but being equivalent for $n = 1$ to $(x_1 < y_1)$. That is, $x \prec_{lex} y \equiv (x \preceq_{lex} y) \wedge (x \neq y)$ and $y \preceq_{lex} x \equiv \neg(x \prec_{lex} y)$. If a point x is interior then $x_{\hat{\sigma}} < x_{\hat{\sigma}^\sigma} \; \forall \; \sigma \in \Sigma \setminus Id$. This means that $x \neq x^\sigma \; \forall \; \sigma \in \Sigma \setminus Id$, but at the same time $x \preceq_{lex} x^\sigma$, since x satisfies RLEX. Thus, if x is interior, $x \prec_{lex} x^\sigma \; \forall \; \sigma \in \Sigma \setminus Id$. Now we investigate whether x^σ satisfies RLEX $\forall \; \sigma \in \Sigma \setminus Id$. When the RLEX associated to σ^{-1} is applied to x^σ it results in $x^\sigma \preceq_{lex} (x^\sigma)^{\sigma^{-1}} \equiv x^\sigma \preceq_{lex} x$. But, because x is interior, $x \prec_{lex} x^\sigma$, which falsifies that RLEX. Therefore, x^σ does not satisfy RLEX $\forall \; \sigma \in \Sigma \setminus Id$. $\qquad \square$

Let $\mathcal{I}_\Sigma$ denote the set of pairs of variable indexes involved in an RLEX constraint, i.e, $\mathcal{I}_\Sigma := \{(i,j) : x_i \leq x_j$ is a constraint in $RLEX_\Sigma\}$. The next proposition is the main result of this subsection: for a point, being interior is equivalent to being optimal.

Proposition 8. *For any $x \in \mathbf{x}$ such that $RLEX_\Sigma(x) = true$ (i.e., for any $x \in \mathcal{R}_\Sigma(\mathbf{x})$), x is interior $\Leftrightarrow$ the class x^Σ is optimal.*

PROOF. Interior $\Rightarrow$ optimal: Applying Proposition 7 to x we obtain the conditions for Proposition 6 to be be applied with $\sigma_1 = Id$

Optimal $\Rightarrow$ interior: The statement "in an optimal class the member retained by RLEX is interior" is equivalent to "if a point is frontier, its class is non optimal". We will prove the latter statement. So we assume that x is a frontier point, i.e., $\exists \; (i,j) \; \in \mathcal{I}_\Sigma \; : \; x_i = x_j$. To prove the non optimality of x^Σ it is enough to show that $\exists \; \sigma \in \Sigma \setminus Id$ such that $RLEX_\Sigma(x^\sigma) = true$. Let $\delta := \min\{|x_r - x_s| : x_r \neq x_s\}/2$, i.e., half of the smallest non-null difference between any two components in x, and $z = (z_1, \ldots, z_n)$ such that $z_k = x_k$, if $k \neq i$ and $z_k = x_k + \delta$, if $k = i$.

Since $z_i > z_j$ and $(i,j) \in \mathcal{I}_\Sigma$, z must be rejected by RLEX. However, at least one element of its class must by accepted by RLEX. Thus, $\exists \sigma \in \Sigma \setminus Id$ such that $RLEX_\Sigma(z^\sigma) = true$. We can show that this same permutation σ of x is also accepted by $RLEX_\Sigma$. Let $E(i) := \{r : x_r = x_i\}$. We can divide the set of $RLEX_\Sigma$ constraints into two subsets. The first one corresponds to $\{(k,l) \in \mathcal{I}_\Sigma : k^\sigma \in E(i) \; \wedge \; l^\sigma \in E(i)\}$, and for the elements of this set we know that $x_{k^\sigma} = x_{l^\sigma}$ and, therefore, the corresponding constraint for x^σ, $x_{k^\sigma} \leq x_{l^\sigma}$, is satisfied. By construction of z, it is easy to verify that for any (s,r) such that $r \notin E(i)$ or $s \notin E(i)$, if $z_r \leq z_s$ we have also $x_r \leq x_s$. Therefore, in the remaining second set of constraints, $\{(k,l) \in \mathcal{I}_\Sigma : k^\sigma \notin E(i) \; \vee \; l^\sigma \notin E(i)\}$, the fact that z^σ is accepted by RLEX means that $z_{k^\sigma} \leq z_{l^\sigma}$, which in turn implies that $x_{k^\sigma} \leq x_{l^\sigma}$. The conclusion is that all $RLEX_\Sigma$ constraints are satisfied at x^σ and at x, thus its class is not optimal. $\qquad \square$

Example 6. The 4-cyclic roots problem has a formulation similar to that of Example 1 in $\mathbb{R}^4$. It has the following eight symmetries:

$$\Sigma = \quad \{[1,2,3,4], [4,3,2,1], [2,3,4,1], [1,4,3,2],$$
$$[3,4,1,2], [2,1,4,3], [4,1,2,3], [3,2,1,4]\}$$

One can easily check that the non redundant RLEX for this set of permutations is $x_1 \leq x_2$, $x_1 \leq x_3$, $x_1 \leq x_4$ and $x_2 \leq x_4$. An interior point of the constrained space $\mathcal{R}_\Sigma(\mathbf{x})$ must satisfy $x_1 < x_2$, $x_1 < x_3$, $x_1 < x_4$ and $x_2 < x_4$, but can have some components equal, if they are not linked by the relaxed constraints. For instance, $(0.0, 1.5, 1.5, 2.0)$ is an interior point even if $x_2 = x_3$.

A consequence of the last proposition is that frontier points can be identified with non-optimal points. Thus, a frontier point must satisfy at least one of these conditions:

1. It belongs to a class with less than $|\Sigma|$ members.
2. Some of its symmetries are not broken by RLEX$_\Sigma$.

All the possibilities not forbidden by the above statement can be found in the set of frontier points:

- Points whose symmetric points have not been completely eliminated by RLEX, and having $|\Sigma|$ symmetries.

  For example, considering again the 4-cyclic roots problem, $(0.0, 0.0, 1.5, 2.0)$ is a point with eight symmetries, which is retained by RLEX$_\Sigma$ jointly with one of its symmetric points, $(0.0, 0.0, 2.0, 1.5)$.

- Points for which all symmetries are broken and belonging to classes with less than $|\Sigma|$ members. For these classes, RLEX are equivalent to LEX.

  Examples for the 4-cyclic roots problem are the points $(0.0, 1.0, 0.0, 1.0)$ and $(-2.0, 5.0, 5.0, 5.0)$, which are the only members of their classes (of cardinality two and four, respectively) in the constrained subspace.

- Points that belong to classes with less than $|\Sigma|$ members for which symmetries are not completely broken.

  An example for the 4-cyclic roots problem is $(-5.0, -5.0, 2.0, -5.0)$, a point of a four-members class that has two representatives in the constrained subspace, namely the same point and $(-5.0, -5.0, -5.0, 2.0)$.

In sum, the frontier of the constrained subspace $\mathcal{R}_\Sigma(\mathbf{x})$ is the union of two non-disjoint sets of points belonging to classes with either less than $|\Sigma|$ members, or whose symmetries have not been entirely broken.

Finally we want to remark that the SBC character of RLEX is not limited to $\overline{H}$, or even to interior points, a superset of $\overline{H}$. Let's outline the situation in this respect. RLEX always are, of course, SBCs in $\overline{H}$, which has the same volume as the domain. RLEX are also SBCs for two subsets of H: a) Interior points belonging to H ; b) Frontier points with all symmetries broken. Moreover, at least one of these subsets is not empty. If the set of RLEX imposes a total order on the variables, they are SBCs in the whole domain including the set of frontier points, which coincides with H (because in this case RLEX and LEX are equivalent) and, thus, the set in b) is not empty. If RLEX do not impose a total order then the set in a) is not empty, because there are interior points with variables not related by RLEX constraints with equal values. The conclusion is that RLEX are SBCs in a set always strictly greater than $\overline{H}$.

14

4. Related work

The related work we present in this section concerns only variable symmetry breaking. It can be divided into two categories. The first one (Sections 4.1, 4.2 and 4.3) includes the simplified SBCs or PSBCs for general symmetries developed in the discrete constraint programing field. In general, they are not suited to numerical problems and they have never been applied to numerical solvers. The second group of related works (Sections 4.5 and 4.6) are those specifically addressing symmetries in numerical problems. They deal only with particular cases of symmetries and/or problems.

4.1. Puget's SBCs

In the case where a CSP contains an all-different constraint, Puget has proved in [24] that the LEX SBCs can be drastically simplified. In fact, as shown in Subsection 3.1, the complexity of LEX SBCs lays in their boundaries, which correspond to some equality between two distinct variables, while the presence of an all-different constraint implies that there is no solution on these boundaries. The comparison between Puget's SBCs and RLEX PSBCs is summarized in Figure 5: An arbitrary LEX SBC is equivalent to Puget's $x_{\hat{\sigma}} < x_{\hat{\sigma}^\sigma}$ SBC under the hypothesis that an all-different constraint is enforced, while this LEX SBC implies the corresponding $x_{\hat{\sigma}} \leq x_{\hat{\sigma}^\sigma}$ RLEX PSBC in general. Since the same indices are involved in both cases, the polynomial algorithm used by Puget to generate his constraints, which is based on the Shreier-Sims algorithm, can also be used for generating RLEX PSBCs. Finally, although less natural than the derivation of RLEX PSBCs given in Section 3.2, the following proposition shows that RLEX PSBCs can be derived directly from Puget's SBCs. It is provided here for clarifying the relationship between Puget's SBCs and RLEX PSBCs.

Proposition 9. *The fact Puget's constraints are SBCs in conjunction with an all-different constraint implies that* RLEX *are PSBCs in general.*

PROOF. Consider a CSP with n variables and an arbitrary variable symmetry group Σ. Let x be an arbitrary variable instantiation. We are going to prove that the constraints RLEX$_\Sigma$ retain at least one element of x^Σ, hence that they are PSBCs, only using the fact that Puget's constraints are SBCs.

Let $\delta := \min\{|x_i - x_j| : x_i \neq x_j\}$, i.e. the smallest non-null difference between any two components in x, and $y = (y_1, \ldots, y_n)$ such that $y_i = x_i + \frac{i\delta}{n}$. Then all the components y_i are different. Indeed, $y_i - y_j = (x_i - x_j) + (i - j)\frac{\delta}{n}$ is not zero for $i \neq j$, since when $x_i - x_j = 0$ the second summand is not null and, otherwise, we have $|i - j| < n$ and $|i - j|\frac{\delta}{n} < \delta \leq |x_i - x_j|$, which implies $(x_i - x_j) \neq -(i - j)\frac{\delta}{n}$.

Because y satisfies the all-different constraint, Puget's SBCs on Σ retain exactly one element of its symmetric class, i.e., there exist $\sigma \in \Sigma$ such that y^σ satisfies all Puget's constraints. We now show that x^σ satisfies all RLEX constraints on Σ as well.

$$x \preceq_{lex} x^{\sigma}$$

$$\Updownarrow$$

$$x_{\hat{\sigma}} < x_{\hat{\sigma}^{\sigma}} \ \bigvee \ \left(\overbrace{x_{\hat{\sigma}} = x_{\hat{\sigma}^{\sigma}}}^{A} \ \bigwedge \ \overbrace{x_{\hat{\sigma}+1:n} \preceq_{lex} x_{\hat{\sigma}+1:n}^{\sigma}}^{B} \right)$$

$$\text{AllDiff}(x) \models \neg A \ \nearrow \qquad\qquad \searrow \ A \wedge B \Rightarrow A$$

$$x_{\hat{\sigma}} < x_{\hat{\sigma}^{\sigma}} \qquad \Longrightarrow \qquad x_{\hat{\sigma}} \leq x_{\hat{\sigma}^{\sigma}}$$

Figure 5: Parallel derivation of Puget's SBCs and RLEX showing both the similitude and the difference of the two approaches.

Consider the application to y^{σ} of an arbitrary Puget's constraint $y_{i^{\sigma}} < y_{j^{\sigma}}$ and suppose that the corresponding RLEX constraint does not hold, i.e., $x_{i^{\sigma}} > x_{j^{\sigma}}$. By definition of y, Puget's constraint $y_{i^{\sigma}} < y_{j^{\sigma}}$ is equivalent to $x_{i^{\sigma}} + \frac{i^{\sigma}\delta}{n} < x_{j^{\sigma}} + \frac{j^{\sigma}\delta}{n}$, which is equivalent to $x_{i^{\sigma}} - x_{j^{\sigma}} < \delta \frac{j^{\sigma}-i^{\sigma}}{n}$. Since we supposed $x_{i^{\sigma}} - x_{j^{\sigma}} > 0$ and by definition of δ, we have $\delta \leq x_{i^{\sigma}} - x_{j^{\sigma}} < \delta \frac{j^{\sigma}-i^{\sigma}}{n}$, but because $j^{\sigma} - i^{\sigma} < n$ this is contradictory. $\qquad\square$

4.2. Jefferson's PSBCs

PSBCs in the discrete context follow two distinct principles: either (1) they completely break a subset of the symmetry group, or (2) they partially break the whole symmetry group[7]. Jefferson and Petrie have investigated the two approaches in [22]: following principle (1), they use LEX constraints to completely break random subsets of permutations, or the generators computed by Nauty [26], or the generators obtained using a modification of Nauty ; following (2), they use RLEX constraints (called ArityOne in their work and informally derived from Puget's SBCs) to partially break all symmetries. Their experiments show that, in the context of discrete CSP, RLEX are quite efficient but not the best PSBCs : the LEX constraints on the generators computed using their modification of Nauty appears more efficient in general. These conclusions cannot however be transposed to the continuous context, since approaches following principle (1) are not suitable for continuous solvers due to the impracticality of LEX constraints in this context (see Section 3.1).

4.3. Friedman's PSBCs

Friedman *et al.* introduce the notion of fundamental domains for (binary) integer programs [27] then for continuous domains [28]. They are defined as

---

[7]Though it is also possible to partially break a subset of the symmetry group, this approach is in general inferior to the two others.

(a) $F_{\hat{c}}(\mathcal{C}_2)$
$\text{FD}_{[2,1]} := 2x_1 + x_2 \le 2x_2 + x1$

(b) $F_{\hat{c}}([2,3,1])$
$\text{FD}_{[2,3,1]} := 4x_1 + 2x_2 + x_3 \le 4x_2 + 2x_3 + x_1$

(c) $F_{\hat{c}}([3,1,2])$
$\text{FD}_{[3,1,2]} := 4x_1 + 2x_2 + x_3 \le 4x_3 + 2x_1 + x_2$
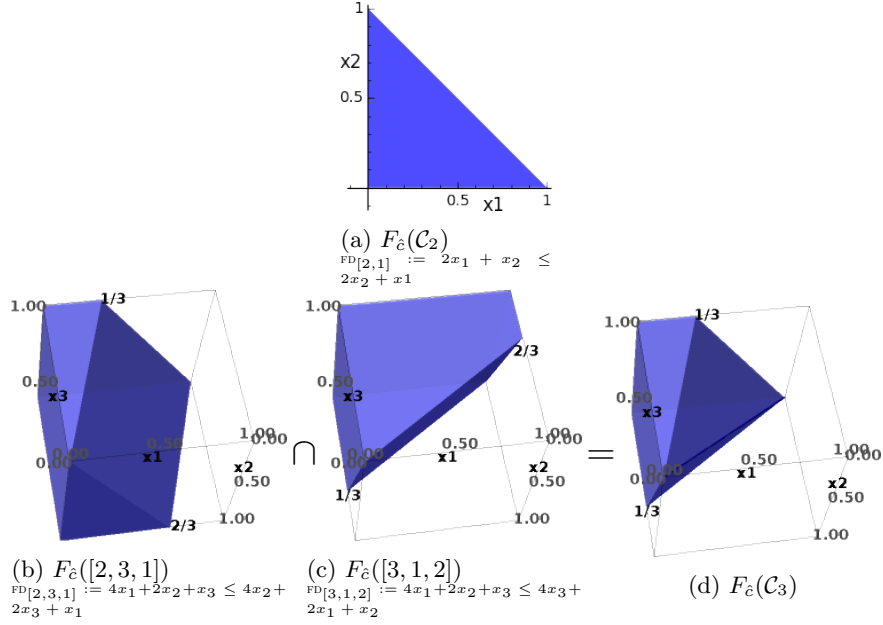
(d) $F_{\hat{c}}(\mathcal{C}_3)$

Figure 6: Universal fundamental domains for the cyclic symmetry group in 2D and 3D.

subsets F of the initial domain D such that $F^\Sigma = D$, i.e., F contains at least one representative of the symmetry class of any point by Σ in D. This definition applies in particular to $\mathcal{L}_\Sigma(D)$ and to $\mathcal{R}_\Sigma(D)$. They also define *minimal* fundamental domains as fundamental domains with no proper closed subset being itself a fundamental domain, a property which also applies to $\mathcal{L}_\Sigma(D)$ and to $\mathcal{R}_\Sigma(D)$ (see Proposition 2).

Considering domains restricted to $[0,1]^n$ for simplicity[8], they propose a method for constructing a fundamental domain F_c considering an *ordering vector* $c \in \mathbb{R}^n$ (where n is the number of variables of the problem)[9]:

$$F_c(\Sigma) = \{x \in [0,1]^n : c^T x \le c^T x^\sigma, \forall \sigma \in \Sigma\}. \tag{17}$$

It is easy to see that the constraint $\text{FD}_\sigma := c^T x \le c^T x^\sigma$ is a PSBC for the permutation σ, and $\text{FD}_\Sigma := \bigwedge_{\sigma \in \Sigma} \text{FD}_\sigma$ is a PSBC for the symmetry group Σ.

They introduce the *universal* ordering vector $\hat{c} = (2^{n-1}, 2^{n-2}, \ldots, 2, 1)$ and prove $F_{\hat{c}}$ is a minimal fundamental domain. We illustrate the universal fundamental domain for the cyclic symmetry groups $\mathcal{C}_2$ and $\mathcal{C}_3$ in Figure 6.

As this illustration clearly shows, $F_{\hat{c}}(\Sigma)$ is equivalent to $\mathcal{L}_\Sigma$ and to $\mathcal{R}_\Sigma$ in 2D

---

[8]Note that continuous domains in the form of intervals can be mapped to $[0,1]^n$ by subtracting to each interval its lower-bound and dividing it by its width.

[9]The mentioned paper in fact use the relation $\ge$ instead of $\le$, which is strictly equivalent in terms of symmetry breaking. We change it for an easier comparison to LEX and RLEX constraints.

17

(see Figures 1 page 5 and 4a page 8). However, contrarily to the situation for discrete domains, where they are also equivalent for higher dimensions, $F_{\hat{c}}(\Sigma)$ is in general not directly related to $\mathcal{L}_\Sigma$ (nor to $\mathcal{R}_\Sigma$) in higher dimension. For instance, Figure 4d (page 8) shows $\mathcal{R}_{\mathcal{C}_3}([0,1]^3)$, which is significantly different from $F_{\hat{c}}(\mathcal{C}_3)$ shown in Figure 6d. Still, it has the property of being a (closed) minimal fundamental domain and thus benefits from the same properties as RLEX constraints. For instance, the volume of $F_{\hat{c}}(\mathcal{C}_3)$ can be formally computed and shown to be equal to $\frac{1}{3}$, which is the same volume of $\mathcal{R}_{\mathcal{C}_3}([0,1]^3)$, computed using Proposition 3 and Proposition 4. We however consider RLEX PSBCs superior to FD PSBCs for several reasons:

- The number of FD_σ constraints is by essence equal to the number of symmetries σ in the considered symmetry group, i.e., up to factorial, while there is no obvious simplification of this intractable set of constraint ;

- There is no known efficient (poly-time) way of generating them ;

- Like LEX constraints, each individual FD constraint involves all the variables and is thus costly to prune and propagate ;

- The coefficients in $\hat{c}$ induce a numerical unstability in floating-point computations since adding $2^{n-1}x_{n-1}$ and x_0 can yield cancellation when n is large, leading to inaccurate comparisons.

4.4. Frisch's SBCs

Frisch *et al.* propose in [29, 30] several alternative encodings of LEX constraints. These encodings are equivalent to LEX constraints in discrete domains. Some of them seem appropriate for numerical solvers, but they are in fact not equivalent to LEX constraints in continuous domains.

For instance, the *arithmetic* encoding is in fact equivalent to a fundamental domain F_c with $c = (d^{n-1}, d^{n-2}, \ldots, d, 1)$, where d is the size of the discrete variable domains taken as $\{0, 1, \ldots, d-1\}^n$. Transposed in the normalized continuous domain $[0,1]^n$, this encoding thus becomes just the FD constraints, which we have shown to be different of both LEX and RLEX constraints in the previous section. These constraints are still PSBCs (not SBCs) in the numerical context, but we have explained why RLEX constraints should be preferred.

Another example is the *Harvey* encoding which employs a reification of binary comparison:

$$\text{HLEX}_\sigma := x_1 < \big(x_{1\sigma} + \Delta(x_2 < (x_{2\sigma} + \Delta(\ldots + \Delta(x_n < (x_{n\sigma} + \Delta(true)))\ldots)))\big), \quad (18)$$

where Δ is the reification function which maps $true$ to the integer 1 and $false$ to the integer 0. In the discrete context, HLEX constraints are equivalent to LEX constraints, but it is not the case in the continuous context since there is no gap of size 1.0 between two possible valuations of a variable. HLEX constraints can be adapted by changing the definition of the function Δ, making it closer and closer to LEX constraints in the continuous context as $true$ is mapped to smaller and smaller values $\delta \in (0,1]$, see Figure 7, giving birth to HLEX_δ constraints

which are valid PSBCs in the continuous context[10]. The space they delimit however strictly contains $\mathcal{L}$ and it has a volume strictly larger than $\mathcal{R}$, making HLEX$_\delta$ a weaker (P)SBC than both LEX and RLEX.

(a) $\delta = 1.0$ (b) $\delta = 0.9$ (c) $\delta = 0.5$ (d) $\delta = 0.1$ (e) $\delta = 0.0$
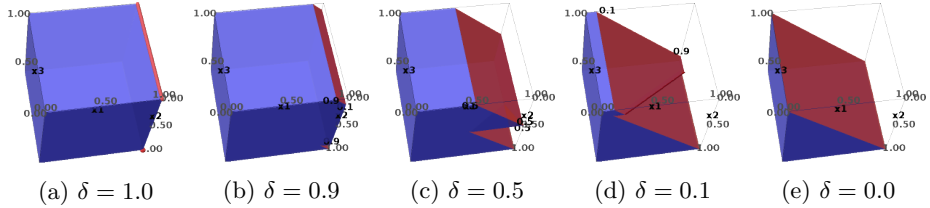
Figure 7: Satisfaction domain of HLEX$^\delta_{[2,3,1]}$ constraint (blue=included, red=excluded).

Again, and similarly to most proposed LEX encodings, the first and foremost reason why such encodings are not appropriate in the continuous context is that they come in factorial number with respect to the number of variables in the worst case, without an efficient (poly-time) method for generating them, and involve all the variables implied in the symmetry in a formula much more complex than RLEX constraints. These are the reasons why we consider RLEX the most appropriate PSBCs in the continuous context.

4.5. Specific PSBCs for NCSPs

The ad-hoc inequalities proposed in [15, 17, 16] to partially break specific classes of variable symmetries in NCSPs are just special cases of the RLEX constraints. For instance, Gasca *et al.* [15] proposed PSBCs $x_i \leq x_{i+1}$ ($i \in \{1, \ldots, n-1\}$) for full permutations ($\Sigma = \mathcal{S}_n$), and PSBCs $x_1 \leq x_i$ ($i \in \{2, \ldots, n\}$) for cyclic permutations ($\Sigma = \mathcal{C}_n$). Similar PSBCs have been proposed for numerical optimization problems with more peculiar symmetry groups, e.g., $\Sigma = \mathcal{C}_2 \times \mathcal{S}_{\frac{n}{2}}$ in [17] and $\Sigma = \prod_i \mathcal{S}_{p_i}$ in [16].

Liberti *et al.* [13, 14] suggest some general PSBCs for MINLP(Mixed Integer NonLinear Programming) problems involving inequality constraints only. However no analysis is provided regarding the completeness and optimality of these PSBCs for this class of problems.

4.6. Symmetric boxes for single-cycle symmetries

A very different strategy from the preceding previous works, not relying on the addition of SBCs, is [12]. From the fact that numerical solvers process only boxes, the authors suggest the utility of the notion of box symmetry, which is the set of points symmetric to those belonging to the box. In the context of variable permutations, the symmetry of a box can be understood as a permutation of its component intervals and, therefore, is another box. Classes of box symmetry are naturally defined, as well as *representatives*, a member of the class

---

[10]HLEX$_0$ is not a valid PSBC since it makes *true* indistinguishable from *false*.

representing the whole class. The approach works by producing a set of representatives, whose symmetries cover the whole domain. The solver is launched with the original problem using each representative as domain, and solutions symmetric to those found within it are generated according to the symmetries of the representative. This approach has been followed in [12] in the context of cyclic permutations. Its main advantage is that the potential burden that SBCs add to the original problem is avoided. However it needs a small program to manage the launching of the representatives. A more serious disadvantage of the algorithm is that the number of representatives grows exponentially with the dimension, which becomes unmanageable when n is large.

5. Experimental results

In this section, we provide experimental evidence of the important performance gains RLEX constraints can bring when solving symmetric NCSPs. For this purpose, we will use a single measure, called *gain* in the following: the ratio of time[11] to solve the problem without RLEX over that to solve the problem with RLEX. Indeed, we expect RLEX constraints to quickly eliminate symmetric portions of the search space, isolating an asymmetric search subspace whose volume is divided by $|\Sigma|$ w.r.t. the initial search space. As a result, we expect to observe gains proportional to $|\Sigma|$.

All experiments are conducted on a dual-core equipped machine (2.5GHz, 4Gb RAM) using the Realpaver [31] constraint solver with default settings, i.e., AC3-like fixed-point propagation loop using Hull-consistency and Box-consistency contracting operators followed by an Interval Newton application, and balanced bisection splitting following a round-robin principle.

We consider a benchmark composed of standard problems picked from [32, 33, 34, 35]. The selected test set has representative characteristics: fixed or scalable dimension, various constraint types and complexity, satisfiable or unsatisfiable problems, discrete or continuous solution sets, and of course various symmetry groups. Tables 1 and 2 provide the characteristics (columns "n" and "$|\Sigma|$") of the considered problems and the experimentally obtained gain (column "*time*") and number of solutions (column "$|\chi|$"), summarizing 68 individual experiments on problems instances with (columns "RLEX") and without (columns "*standard*") our PSBCs. For each scalable problem, the obtained ratios are also graphically depicted in Figure 8 in comparison to their symmetry group order.

We do not compare RLEX performances to the related work mentioned in the previous section because, as explained there, they either do not apply in the continuous context, or are subsumed by RLEX, and are thus identical to RLEX for the classes of symmetry for which they can be applied.

---

[11] The ratio in number of search nodes could also be interesting but it has two drawbacks: First, it does not measure the impact of the additional RLEX constraints in the propagation cost at each node; Second, contrarily to discrete CSPs, the search tree of numerical CSPs explored when SBCs are added is not a subpart of the search-tree without SBCs since split-point location are affected by the SBCs' propagation.

5.1. Global analysis of the results

Without RLEX, the solving time varies from 0.8s to $3\,818\,828$s ($44d04h47m$ $08s$[12]), with a total solving time of $7\,851\,611$s ($90d21h00m11s$) for the 68 experiments. With RLEX, the solving time varies from 0.13s to $132\,070$s ($1d12h41m$ $10s$[13]), with a total solving time of $228\,253.1$s ($2d15h24m13s$) for the 68 experiments. The gain varies from 0.98[14] to $10\,166.2$[15], for an overall gain (ratio of total solving times for the 68 instances) of 34.4. These figures are a bit biased because our test set comprises problems solved in widely different times. In order to allow a more precise comparison, Table 3 partitions our test set into three classes with respect to their solving time without RLEX: *easy* instances solved in less than an hour, *hard* instances solved in less than day and *intractable* instances solved in more than a day. For each category, we provide the total, minimum and maximum solving times and gains. These statistics are complemented by Table 4 which depicts, for the same three classes, the total number of instances, those with gains lower than 1.0, between 1.0 and 10.0 (excluded), between 10.0 and 100.0 (excluded), and larger than 100.0.

A counter-productive performance occur for only 1 run out of the 68 experiments, the instance of problem Eiger at dimension $n = 100$, and the loss in solving time is 2% of $22.3s$, i.e., $0.3s$. A more detailed look at this problem reveals it has only a cyclic symmetry group (order n), very simple binary quadratic constraints and only two fully frontier solutions (see Section 5.2.1 for a detailed analysis of the influence of these features). Another problem in our benchmark has similar characteristics: Vrahatis. As expected, its gains, though never counter-productive, also appear very small in comparison to its symmetry group order. The fact RLEX constraints are inefficient for these two problems is easily explained: the original constraints in both problems are already efficiently filtered (as testified by the very large dimensions that can be tackled, and the fact their solving require a single split) and thus propagating RLEX constraints (numerous due to the large considered dimensions) does not bring useful additional reductions while it induces a small additional cost.

Oppositely, very good performances (i.e., gains greater than 10.0) occur for 33 runs out of 68, 17 of them providing gains even greater than 100.0. Such outstanding gains are obtained for problems Brown, Cpdm, Extended-Powell, Noon and Wright at large enough dimensions. This is again easily explained: these problems have factorial-order symmetry groups and, in general[16], an exponentially growing solution set with n, also exponentially reduced by RLEX.

To sum up, these general statistics demonstrate that RLEX constraints can provide important gains in solving time and seem harmless to use since the additional propagation cost they may induce is only slightly counter-productive in the worst case. In our opinion, this validates their practical relevance.

---

[12]Sparse at dimension $n = 7$

[13]Reimer at dimension $n = 6$

[14]Eiger at dimension $n = 100$

[15]Extended-Powell at dimension $n = 30$

[16]It is not the case for Brown and Noon

Problem	n	$|\Sigma|$	$|\chi|$			$time$ $(seconds)$		
			$standard$	RLEX	$ratio$	$standard$	RLEX	$ratio$
Boon	6	4	8	3	2.7	34.7	11.2	3.1
Brown	5	24	3	3	1.0	0.8	0.3	2.7
	6	120	2	2	1.0	6.9	0.9	7.7
	7	720	3	3	1.0	79.4	2.5	31.8
	8	5 040	2	2	1.0	1 419.4	10.5	135.2
	9	40 320	3	3	1.0	15 965.7	34.8	458.8
Cpdm[a]	3	6	91 105[b]	16 489[b]	5.5	386.7	57.8	6.7
	4	24	233	22	10.5	96.9	5.8	16.7
	5	120	18 819	212	88.7	2 856.8	43.6	65.5
	6	720	30 574	120	254.8	53 587.4	218.8	244.9
Cyclic-	4	8	26 148[b]	3 319[b]	7.9	58.8	7.6	7.7
roots[c]	5	10	10	1	10.0	23.6	2.8	8.4
	6	12	24	7	3.4	3 352.5	284.5	11.8
	7	14	56	4	14.0	135 960.1	11 607.7	11.7
Eiger	100	100	2	2	1.0	22.3	22.6	1.0
	200	200	2	2	1.0	348.4	281.9	1.4
	300	300	2	2	1.0	1 634.7	1 375.0	1.2
	400	400	2	2	1.0	4 927.8	4 224.9	1.2
	500	500	2	2	1.0	11 636.6	10 285.0	1.1
	600	600	2	2	1.0	22 556.3	17 747.4	1.3
Extended	40	2.43E18	1	1	1.0	8.8	1.9	4.6
Freud-	60	2.65E32	1	1	1.0	41.0	8.1	5.1
-enstein	80	8.16E47	1	1	1.0	124.4	23.5	5.3
	100	3.04E64	1	1	1.0	291.5	54.6	5.3
	120	8.32E81	1	1	1.0	638.3	118.5	5.4
	140	1.18E100	1	1	1.0	1 173.2	212.5	5.5
Extended	22	8.78E8	2 048	1	2 048.0	135.5	0.2	677.5
Powell	24	1.15E10	4 096	1	4 096.0	342.8	0.3	1 142.7
	26	1.62E11	8 192	1	8 192.0	854.3	0.4	2 135.7
	28	2.44E12	16 384	1	16 384.0	2 095.5	0.4	5 238.7
	30	3.92E13	32 768	1	32 768.0	5 083.1	0.5	10 166.2
Feigen-	11	11	200	20	10.0	17.2	2.0	8.6
-baum	13	13	522	42	12.4	58.4	5.5	10.6
	15	15	1 368	96	14.2	277.3	24.5	11.3
	17	17	3 572	212	16.8	712.6	57.5	12.4
	19	19	9 350	503	18.5	1 217.0	105.7	11.5

[a]Continuous solution set at $n = 3$, discrete at $n = 4, 5, 6$

[b]Number of enclosing boxes at precision 0.001

[c]No solution at $n = 3$, continuous solution set at $n = 4$, discrete solution set at $n = 5, 6, 7$.

Table 1: Characteristics and results for the selected benchmark (part 1). Timings and ratios rounded to (nearest) one decimal, except when too large.

Problem	n	$|\Sigma|$	$|\chi|$			$time$ ($seconds$)		
			$standard$	RLEX	$ratio$	$standard$	RLEX	$ratio$
Hénon	20	10	104	14	7.4	15.6	2.3	6.8
	30	15	1 082	75	14.4	549.2	50.4	10.9
	40	20	10 808	567	19.1	13 070.3	873.6	15.0
	50	25	112 452	4 527	24.8	212 440.0	10 291.9	20.6
Ikeda	6	3	9	5	1.8	7.0	3.4	2.1
	8	4	17	7	2.4	73.5	24.9	2.9
	10	5	23	7	3.3	471.0	137.8	3.4
	12	6	53	13	4.1	3 072.1	754.9	4.1
	14	7	73	13	5.6	21 806.4	4 447.1	4.9
Noon	5	120	13	3	4.3	35.1	0.7	50.1
	6	720	15	3	5.0	61.5	0.5	123.0
	7	5 040	9	2	4.5	158.0	0.7	225.7
	8	40 320	10	2	5.0	590.4	1.0	590.4
	9	362 880	11	2	5.5	1 607.2	1.5	1 071.5
Reimer	4	4	8	2	4.0	4.1	1.1	3.7
	5	12	24	2	12 .0	2 193.2	242.0	9.0
	6	36	72	2	36.0	3.29E6	132 070.0	24.9
Sparse	4	24	0	0	1.0	1.0	0.1	10.0
	5	120	0	0	1.0	62.6	1.7	36.8
	6	720	0	0	1.0	6 164.7	29.2	211.1
	7	5 040	0	0	1.0	3.82E6	1 845.3	2 070.1
Virasoro	8	8	2 880	1 822	1.58	146 938.6	17 399.1	8.4
Vrahatis	100	100	2	2	1.0	49.1	32.7	1.5
	200	200	2	2	1.0	508.4	249.2	2.0
	300	300	2	2	1.0	2 088.0	832.4	2.5
	400	400	2	2	1.0	5 923.3	1 959.0	3.0
	500	500	2	2	1.0	13 454.0	3 827.3	3.5
	600	600	2	2	1.0	25 346.7	6 324.8	4.0
Wright	7	5 040	128	8	16	11.5	0.2	57.5
	8	40 320	256	9	28.4	117.8	0.5	235.6
	9	362 880	512	10	51.2	1 367.0	1.6	854.4
	10	3 628 800	1 024	11	93	16 623.1	3.7	4 492.7

Table 2: Characteristics and results for the selected benchmark (part 2). Timings and ratios rounded to (nearest) one decimal, except when too large.
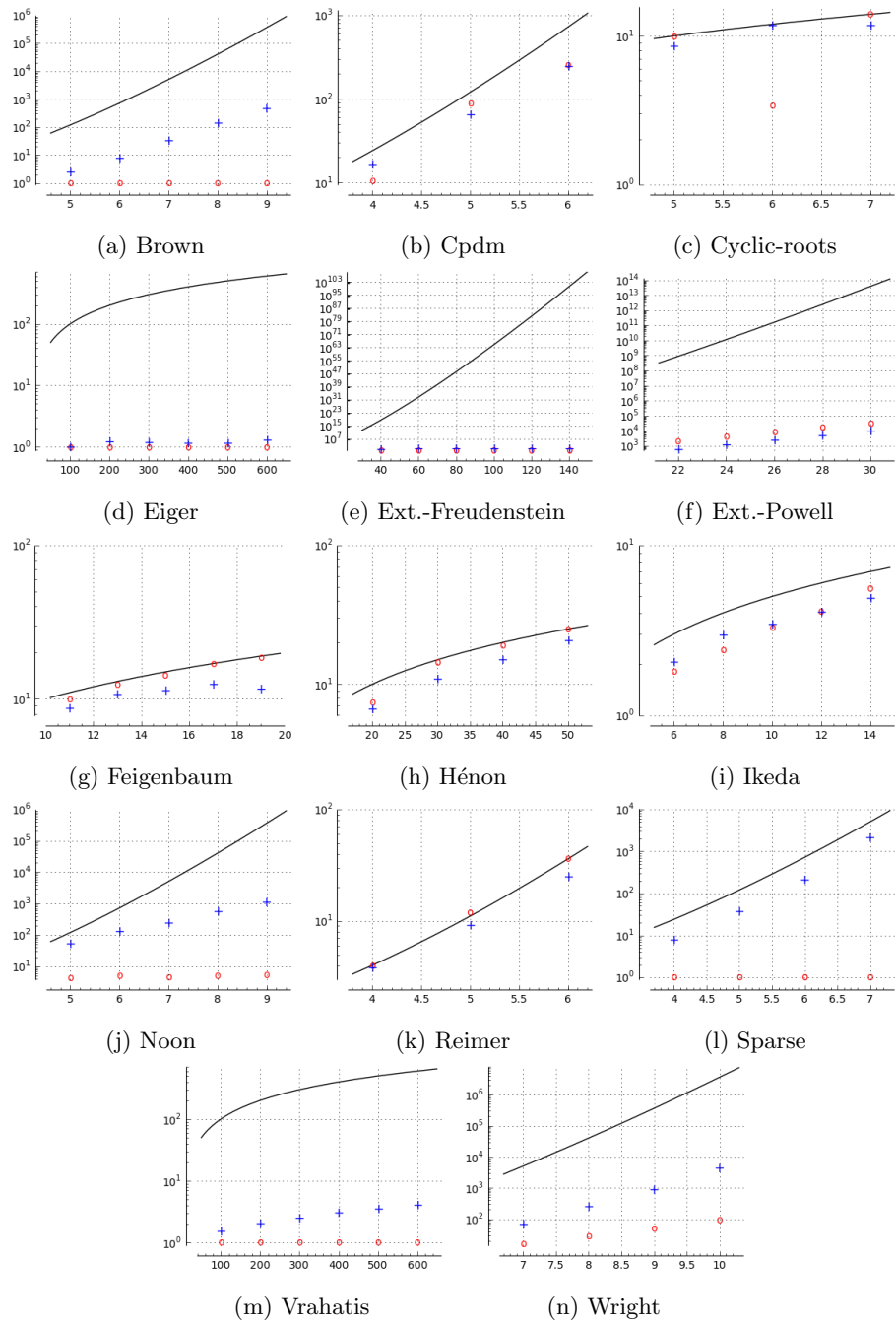
(a) Brown (b) Cpdm (c) Cyclic-roots

(d) Eiger (e) Ext.-Freudenstein (f) Ext.-Powell

(g) Feigenbaum (h) Hénon (i) Ikeda

(j) Noon (k) Reimer (l) Sparse

(m) Vrahatis (n) Wright

Figure 8: Results for scalable problems: ratios in number of solutions (*o*) and in solving time (+); symmetry group order (plain line).

24

	Easy			Hard			Intractable		
	std	RLEX	gain	std	RLEX	gain	std	RLEX	gain
total	31 353.6	5 062.8	6.19	216 145.3	49 976.2	4.3	7.60E6	1.73E5	43.9
min	0.77	0.13	1.0	4 927.8	0.5	1.1	1.35E5	1 845.3	8.4
max	3 352.5	1 375.0	5 238.7	53 587.4	17 747.4	10 166.2	3.82E6	1.32E5	2 070.1

Table 3: Timings (in seconds) and gains statistics classified in easy (less than an hour), hard (less than a day) and intractable (more than a day) instances.

gains	Easy	Hard	Intractable
all	50	13	5
[0,1)	1	0	0
[1,10)	26	7	0
[10,100)	12	1	4
[100,$+\infty$)	11	5	1

Table 4: Partition of the benchmark with respect to the observed gains, classified easy (less than an hour), hard (less than a day) and intractable (more than a day) instances.

5.2. Detailed analysis of the results

A detailed look at Tables 1 and 2 reveals that RLEX performance does not live up to our expectations in many cases, i.e., the gain is often much lower than, and even not proportional to, the order of the symmetry group of the considered instance. Apart from the cases where RLEX constraints are counter-productive and that we have already explained, one can note that for most of the problems with factorial-order symmetry groups, the gains seem smaller than factorial, apparently exponential for many problems, and even polynomial for Sparse and constant for Extended-Freudenstein (see Figure 8). In fact the observed gain is always lower than the order of the symmetry group, with the exception of Virasoro for which RLEX constraints strangely perform even better than expected.

This leads us to examine more closely the rationale behind our gain expectation, and to suggest that there are two very general factors affecting such gain, which emerge from the ways the rationale can be falsified in our benchmark. Then, we exemplify the dependency of the gain on the features of the solver. Finally, we offer some insights on the potential effect of using different variations of RLEX, like transitive closure application and lex ordering.

5.2.1. Rationale for the gain expectation

From now on, we will call Constrained Space (CS) the space previously denoted by $\mathcal{R}_\Sigma(\mathbf{x})$. The argument for an expected gain given by the ratio between the domain volume and the CS volume, $|\Sigma|$, relies on the following assumptions:

A The time for processing a box inside CS is approximately the same with or without RLEX constraints. The base of this assumption is that RLEX constraints are much simpler than typical nonlinear constraints. For this

reason, the cost of propagating RLEX should be negligible compared to the cost of propagating the constraints of the problem.

B The time for discarding boxes outside CS with RLEX can be neglected with respect to the global solving time. This assumption relies on the fact that a box outside CS violates some RLEX constraint which, when considered during the propagation loop, will discard the box instantly at a very low cost. Then we can consider that, when RLEX constraints are added, the domain of the problem is virtually restricted to CS at no cost.

C The processing time is homogeneous across the domain. That is, the time to process a fixed-size volume is the same everywhere.

If the above assumptions hold, one can then conclude that the time to solve the problem without and with RLEX should respectively be proportional to the initial domain volume and to the CS volume, yielding a gain equal to $|\Sigma|$. The first two assumptions can be easily trusted for most problems of interest (i.e., difficult ones), which are composed of complex nonlinear constraints. The cases when they are evidently not valid are the counter-performing ones we have already discussed. The validity of the third assumption is less obvious as the following paragraphs reveal.

Solution processing prevalence. Assumption (C) gets too far away from reality in many problems. Indeed, the processing time is often concentrated in the areas around the solutions whereas large infeasible portions of the domains are discarded instantly. For that specific kind of problems it is appropriate to substitute Assumption (C) by:

D A distinct box exists for each solution such that the cost of processing such boxes of symmetric solutions is the same and that the time to process the space outside all these boxes is negligible.

If we further assume that the processing cost of such a box is the same for all classes, assumptions (A), (B) and (D) together yield the conclusion that the solving time gain should be the ratio of the number of solutions without RLEX over that with RLEX. This ratio depends on the cardinality and the number of elements satisfying RLEX for each solution class:

$$\frac{\sum_{x\in\text{LEX}_\Sigma(\chi)}|x^\Sigma|}{\sum_{x\in\text{LEX}_\Sigma(\chi)}|\mathcal{R}_\Sigma(x^\Sigma)|}. \tag{19}$$

For an interior solution x, we have $|x^\Sigma| = |\Sigma|$ and $|\mathcal{R}_\Sigma(x^\Sigma)| = 1$. Thus, if a problem has only interior solutions the above ratio is $|\Sigma|$. Therefore, in this kind of problems, homogeneity and solution processing prevalence assumptions yield the same gain expectation. In our test set, only problems Cyclic-roots in dimensions $n = 5, 7$ and Reimer are of this kind.

The gains observed for problems Feigenbaum, Henon and Ikeda are also clearly influenced by this factor and they follow quite closely (19) though these problems do not have only interior solutions. In fact, it turns out they have a
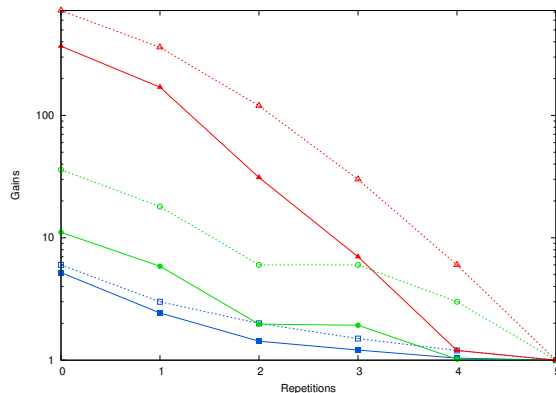
Figure 9: Experimental time ratio (plain) and number of solutions ratio (19) (dashed) for generated problems (20) of dimension $n = 6$ under Cyclic symmetry (order n, squares), 2-Cyclic+Mirror symmetry (order n^2, circles) and Full permutation symmetry (order $n!$, triangles).

large majority of interior solutions and only a few frontier solutions, explaining the observed behavior.

Although the set of frontier points has zero volume, in practice it is not so rare that symmetric problems have frontier solutions. The existence of frontier solutions makes (19) smaller than $|\Sigma|$. This is particularly striking for problems Brown, Extended-Freudenstein, Eiger and Vrahatis which have only fully-frontier solutions (i.e., the same set of solutions is found with and without RLEX). The two latter problems have already been discussed as they present inefficient gains. The two former ones have factorial order symmetry groups but only exponential and constant gains, respectively.

Problems with continuous solution sets, like Cyclic-roots of dimension $n = 4$ and Cpdm of dimension $n = 3$, are another case of the solution prevalence type. Moreover, in most cases they have a gain not too far from $|\Sigma|$. Although their solution manifolds may intersect with the frontier of CS, it is sufficient that the dimensionality of the overlap is lower than that of their solution set to have a majority of boxes enclosing interior solutions. Therefore gains of the order of $|\Sigma|$ are generally obtained, sometimes with significant accuracy.

In order to confirm the impact of solution prevalence on the expected gain, we have prepared a specific problem generator parameterized by the dimension of the problem n, the number of repetitions r of the smallest coordinate in the single solution class representative s to the problem, and the symmetry group Σ. The generated problems have a single constraint of the form

$$\prod_{\sigma \in \Sigma} \left(\sum_{i \in [1,n]} (x_i - s_{i^\sigma})^2 \right) = 0, \tag{20}$$

where the solution class representative s satisfies $s_i = \max(0, i - r - 1)^2$, i.e., the solutions are all symmetries of $s = (\underbrace{0, \ldots, 0}_{\text{r+1 times}}, 1, 4, 9, \ldots)$. When $r = 0$, the generated problems have only interior solutions, all members of the same symmetry class. In this case, we expect the gains to be proportional to the order of the considered symmetry group at the considered dimension. When r increases, the size of the solution class decreases, eventually reaching 1 when $r = n - 1$, in which case the solution is $s = (0, \ldots, 0)$. The gains should also decrease according to (19). Figure 9 displays the evolution of the gain in dimension $n = 6$ for three different symmetry groups when the number of repetitions r varies. In all cases, it is clear that the gain decreases with the increase in the number of repetitions, i.e., the degree of frontierness of the solutions, which in turn determines the numbers of solutions with and without RLEX, having an important impact on the gain. The shapes of gain evolution and ratios of solution numbers are indeed strikingly proportional. Still they are not identical and the gains are always worse than the expected ones according to (19). We explain this phenomenon with another factor: the *frontier effect*.

Frontier effect. There are problems with no solution prevalence or without solutions at all in which gains are very different from $|\Sigma|$. Something besides homogeneous processing is failing in the premises of Subsection 5.2.1. One symptom is box classification in inside and outside CS; frontier boxes, partially inside and partially outside CS are neglected. And they turn out to be a significant fraction, or even the predominant type of boxes in some problems, e.g., Ext.-Freudenstein, Wright and, in minor degree, Feigenbaum.

The reason is that the assumption of constant time cost for a fixed volume size can be falsified not only by the position but also by the shape of the volume when using a box-oriented solver. Suppose we have to pave the domain and CS with boxes. If only one box is used, the tighest enclosing one, the box is the same in both cases and coincides with the domain, itself a frontier box. Thus, there is no difference in the paving despite CS is $|\Sigma|$ times smaller. If we allow a greater number of boxes, they become smaller, the volumes covered in the CS and the domain case differ progressively. Asymptotically, the ratio of volumes covered tend to $|\Sigma|$ and the fraction of frontier boxes in CS tends to zero.

Another facet of the same frontier effect phenomenon, is considering the size of the leaf boxes in the search tree when RLEX are added. It can happen that the size of a leaf is too big to be completely inside CS (because of its shape, the leaf can be at the same time much smaller than the domain), but small enough to be discarded or to converge to a solution in the given problem. It is, thus, a frontier box, eventually one scarcely overlapping CS. This would mean that the pruning power of RLEX has not been exploited effectively. The smaller the CS size with respect to the domain size, the more easily this may occur. On the opposite side, given a fixed a symmetry and its associated CS, the smaller the leaf boxes, the more immune the problem to the frontier effect.

The frontier effect thus depends on two factors: the relative volumes of CS and initial domain, and the precision at which the problem is solved.

To sum up, solution processing prevalence overrides the importance of the volume in the time cost. When there is no solution prevalence, the ratio of volume processing cost is mainly influenced by the frontier effect.

5.2.2. Impact of RLEX variants

The set of RLEX constraints for a given symmetry group is in general not unique. We study here the impact of various possible sets on the observed gains.

Transitive closure. The set of RLEX constraints of a given symmetry group can be minimized, keeping only inequalities which are not transitively implied by other inequalities in the set. Such a minimal set comprises at most n inequalities for a problem in dimension n. Conversely, it can be maximized considering the transitive closure of all inequalities, yielding at most n^2 inequalities for a problem in dimension n. Of course, all intermediate sets between these two extremes also constitute a valid set of PSBCs.

In theory, the maximal set could be preferred because it may accelerate interval propagation. For instance, the reduction of the domain of x_1 in a problem with full permutation symmetry can be propagated instantly to the domain of x_n if the RLEX inequality $x_1 \leq x_n$ is imposed, while it will require $n-1$ propagation loops if only RLEX inequalities of the form $x_i \leq x_{i+1}$ are considered. On the other hand, the minimal set could be preferred since it may comprise significantly less constraints, possibly reducing importantly the number of applied contracting operators to achieve a fix-point of interval propagation. For example, in order to realize that no reduction at all is possible on the domain of any variable, each contracting operator must be applied once, i.e., only n for the minimal set in contrast with n^2 for the maximal set.

In practice, however, we have observed that considering either the minimal set or the maximal set results in very similar timings for most problems. This is in particular the case for problems Brown, Cyclohexane, Extended-Powell, Noon, Sparse and Wright, i.e., problems with very different minimal and maximal sets (typically n and n^2 inequalities, respectively) and for which the gain with maximal PSBCs over that with minimal PSBCs vary between 0% and 8%, sometimes in favor of the maximal set, sometimes in favor of the minimal set. The only exception is for problem Extended-Freudenstein for which solving times are approximately 3 times faster with the maximal set than with the minimal set. This problem, like the Extended-Powell problem, has a very specific structure: it is made of the repetition $\frac{n}{2}$ times of a subsystem of 2 equations. But unlike Extended-Powell, this one has a single solution. Hence, the contraction of the domain of the variables in any subproblem of 2 equations can be immediately propagated to the domains of the variables of all other duplicated subproblems. It appears that the maximal set of RLEX, with its redundant inequalities, exploits this property to its best, allowing a much faster convergence of the solver towards the unique solution.

Except for such peculiar cases, we should thus conclude that using a minimal set of RLEX or a maximal set of RLEX has almost no impact, and so we

recommend using the minimal, non-redundant, set as it involves a lot fewer constraints in general.

Lexicographic order. The variable ordering underlying the considered lexicographic order also impacts the set of generated RLEX, not only in the name of the involved variables but even in cardinality. For instance, consider the symmetry group generated by compositions of $\sigma = [2, 1, 4, 5, 6, 3]$. It is composed of two synchronized cyclic symmetries, one on the first two variables, the other one on the last four variables. The two corresponding RLEX constraints obtained with the standard variable order are $x_1 \leq x_2$ and $x_3 \leq x_5$. If the variable order chosen to build RLEX begins by x_3 instead of x_1, we obtain the RLEX constraints $x_3 \leq x_4$, $x_3 \leq x_5$ and $x_3 \leq x_6$, i.e., a set of three PSBCs instead of two.

Besides the variables involved and cardinality of the generated RLEX, the variable order can influence also the number of symmetries broken for the solutions and, thus, the number of symmetric solutions found with RLEX. Indeed, a variable order can confer a frontier character to a solution while another one can make it an interior point. However, it is difficult to take advantage from this in practice because it requires the previous knowledge of the solutions.

For the problems in our benchmark more prone to be impacted by a change in variable ordering (we have excluded those with full permutation of all variables and those with full cycle of all variables), we have compared runs with several variable orderings and it appears this factor does not have a very important impact on the observed gains: 0% to 10% difference in solving time.

We conclude the variable ordering does not have a very important role in the performance of RLEX constraints in general.

5.2.3. Interest of RLEX in other solving contexts

In the whole previous section, we have considered only the problem of finding all solutions (up to a given precision) of the considered problems, whether they have continuous solution sets, discrete solution sets or empty solution sets. Another perspective on RLEX efficiency consists in searching for a single solution of the considered NCSPs, i.e., solving them as decision problems.

Part of the answer is already included in Tables 1 and 2 : for problems having no solution (Sparse), a single solution (Extended Freudenstein), or an identical (and small) solution set with or without RLEX constraints (Brown, Eiger, Vrahatis), the obtained gains validate their interest in such settings too.

Considering the other problems in our test set, those with large, or largely varying with and without RLEX constraints, solution sets, the figures are very unstable : for problems with continuous solution sets (CPDM at $n = 3$, Cyclic-roots at $n = 4$), the decision problem is answered identically fast (< 0.05s) with and without RLEX ; For other problems, either the decision problem is solved faster without RLEX (CPDM at $n = 4, 5, 6$, Cyclic-roots at $n = 5, 7$, Extended Powell at $n = 22, 24, 26$), or it is solved almost as fast with and without RLEX (Feigenbaum at $n = 11, 13$, Ikeda at all considered n, Noon at $n = 5, 6, 7$), or else it is solved faster with RLEX constraints (all other 48 instances).

The fact no predictable behavior is observed is easily interpreted: the search with and without RLEX constraints proceeds in a totally different way since, due to contracting boxes with RLEX constraints, the split points are generally different in both cases yielding a totally unrelated search tree, unlike what happens in the discrete case where the search-tree with PSBCs is a subpart of that without PSBCs.

The overall statistics are however in favor of RLEX constraints since when they are counter-productive the solving time are typically small[17] and the gain remains above 0.2 in all cases, while solving time vary from small to significant (> 100s) and gains grow up to 816[18] when RLEX constraints are used. The most favorable gains are in fact observed when the first solution found is identical with and without RLEX, indicating the search strategy operates similarly in both cases, just avoiding unnecessary branches that are pruned by our PSBCs.

6. Conclusion and Future Prospects

LEX constraints are known to break exactly every variable symmetry retaining only a representative member of each class. They are valid for all kinds of problems, independently of the nature of the domains. However, the number of LEX constraints is potentially factorial in the number of variables. Moreover, we have shown they are impractical for numerical solvers.

To overcome these drawbacks, we have proposed the RLEX PSBCs, a set of simple inequalities between two variables. The maximum number of non-redundant constraints in RLEX is linear in the number of variables. Besides they are easily handled by numerical solvers, which are able to quickly propagate them. They can be considered a generalization of PSBCs proposed for particular symmetry groups in some previous works.

We have shown that RLEX are a relaxation of LEX with optimal properties. In fact, they enclose a space which is the closure of that enclosed by LEX SBCs, and we have proved both have the same volume. A detailed analysis of RLEX PSBCs also shows that their filtering power is almost identical to complete SBCs in the context of numerical CSPs.

The experiments we have conducted on a benchmark of standard problems with various characteristics and sizes demonstrate the practical efficacy of RLEX constraints: (almost) never counterproductive, they induce gains in solving time which can be factorial for problems with symmetry groups of large orders. Still, the presented results also demonstrate these gains are not always directly proportional to the symmetry group order, and we have thoroughly analyzed the reasons why.

All the arguments we have used are also valid for continuous (constrained) optimization. RLEX can also be interesting in discrete domains, though Jefferson and Petrie [22] have demonstrated there exist superior PSBCs in this context.

---

[17]Except for Cyclic-roots at $n = 7$ solved in 380s without RLEX and 1320s with RLEX.

[18]Noon at $n = 8$ is solved in 490s without RLEX and in 0.6s with RLEX

Still, in discrete domains with many values for each variable, the frontier (in which reside the only symmetry classes not completely broken by RLEX) will be small with respect to the domain. Hence, we should also consider Mixed-Integer Nonlinear Programming and Integer Linear Programming for which some of the PSBCs we have generalized have been proposed [17, 36].

Acknowledgements

References

[1] A. Goldsztejn, C. Jermann, V. Ruiz de Angulo, C. Torras, Symmetry breaking in numeric constraint problems, in: CP 2011, Vol. 6876 of Springer LNCS, 2011, pp. 317–324.

[2] X.-H. Vu, H. Schichl, D. Sam-Haroud, Interval propagation and search on directed acyclic graphs for numerical constraint solving, J. Global Optimization 45 (4) (2009) 499–531.

[3] J.-P. Merlet, Interval analysis for certified numerical solution of problems in robotics, Applied Mathematics and Computer Science 19 (3) (2009) 399–412.

[4] A. Goldsztejn, Y. Lebbah, C. Michel, M. Rueher, Capabilities of constraint programming in safe global optimization, in: International Symposium on Nonlinear Theory and its Applications, 2008, pp. 601–604.

[5] I. Gent, K. Petrie, J.-F. Puget, Handbook of Constraint Programming, Elsevier, 2006, Ch. Symmetry in constraint programming, pp. 329–376.

[6] T. Walsh, Breaking value symmetry, in: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, 2008, pp. 1585–1588.

[7] T. Walsh, Parameterized complexity results in symmetry breaking, in: Parameterized and Exact Computation, proc. 5th International Symposium IPEC, Vol. 6478 of Springer LNCS, 2010, pp. 4–14.

[8] P. Meseguer, C. Torras, Exploiting symmetries within constraint satisfaction search, Artificial Intelligence 129 (1-2) (2001) 133–163.

[9] I. Gent, Groups and constraints: Symmetry breaking during search, in: CP 2002, Vol. 2470 of Springer LNCS, 2002, pp. 415–430.

[10] P. Flener, A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, Breaking row and column symmetries in matrix models, in: CP '02: Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming, Springer, 2002, pp. 462–476.

[11] J.-F. Puget, Symmetry breaking revisited, Constraints 10 (1) (2005) 23–46.

[12] V. Ruiz de Angulo, C. Torras, Exploiting single-cycle symmetries in continuous constraint problems, Journal of Artificial Intelligence Research 34 (2009) 499–520.

[13] L. Liberti, Reformulations in mathematical programming: automatic symmetry detection and exploitation, Mathematical Programming 131 (1-2) (2012) 273–304.

[14] L. Liberti, J. Ostrowski, Stabilizer-based symmetry breaking constraints for mathematical programs, Journal of Global Optimization 60 (2) (2014) 183–194.

[15] R. Gasca, C. D. Valle, V. Cejudo, I. Barba, Improving the computational efficiency in symmetrical numeric constraint satisfaction problems, in: 11th Conference of the Spanish Association for Artificial Intelligence (CAEPIA), 2005, pp. 269–279.

[16] X. Ji, F. Ma, J. Zhang, Solving global unconstrained optimization problems by symmetry-breaking, in: 8th IEEE/ACIS International Conference on Computer and Information Science, 2009, pp. 107–111.

[17] A. Costa, L. Liberti, P. Hansen, Formulation symmetries in circle packing, Electronic Notes in Discrete Mathematics 36 (2010) 1303 – 1310.

[18] J. Crawford, M. Ginsberg, E. Luks, A. Roy, Symmetry-breaking predicates for search problems, in: KR, 1996, pp. 148–159.

[19] D. Cohen, P. Jeavons, C. Jefferson, K. Petrie, B. Smith, Symmetry definitions for constraint satisfaction problems, Constraints 11 (2-3) (2006) 115–137.

[20] N. Narodytska, T. Walsh, Breaking symmetry with different orderings, in: Principles and Practice of Constraint Programming, Vol. 8124 of LNCS, 2013, pp. 545–561.

[21] A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, T. Walsh, Propagation algorithms for lexicographic ordering constraints, Artificial Intelligence 170 (2006) 803–834.

[22] C. Jefferson, K. Petrie, Automatic generation of constraints for partial symmetry breaking, in: CP 2011, Vol. 6876 of Springer LNCS, 2011, pp. 729–743.

[23] A. V. Aho, M. R. Garey, J. D. Ullman, The transitive reduction of a directed graph, SIAM Journal on Computing 1 (2) (1972) 131–137.

[24] J.-F. Puget, Breaking symmetries in all different problems, in: Proc. 19th international joint conference on Artificial intelligence (IJCAI), 2005, pp. 272–277.

[25] C. Sims, Computational Problems in Abstract Algebra, Pergamon Press, 1970, Ch. Computational methods in the study of permutation groups, pp. 169–183.

[26] B. D. McKay, A. Piperno, Practical graph isomorphism, {II}, Journal of Symbolic Computation 60 (0) (2014) 94 – 112.

[27] E. J. Friedman, Fundamental domains for integer programs with symmetries, in: First International Conference on Combinatorial Optimization and Applications, Vol. 4616 of LNCS, Springer, 2007, pp. 146–153.

[28] F. Diaz y Diaz, E. Friedman, Signed fundamental domains for totally real number fields, Proceedings of the London Mathematical Society.

[29] A. M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, T. Walsh, Propagation algorithms for lexicographic ordering constraints, Artificial Intelligence 170 (10) (2006) 803 – 834.

[30] H. A. Elgabou, A. M. Frisch, Encoding the lexicographic ordering constraint in sat modulo theories, in: Thirteenth International Workshop on Constraint Modelling and Reformulation (ModRef 2014), 2014, pp. 85–96.
URL http://web.udl.es/usuaris/q4374304/modref2014.pdf

[31] L. Granvilliers, F. Benhamou, Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques, ACM T. on Mathematical Software 32 (2006) 138–156.

[32] COPRIN, The COPRIN examples page (2011).
URL http://www-sop.inria.fr/coprin/logiciels/ALIAS/Benches/

[33] COCONUT European Project, The COCONUT benchmark (2003).
URL http://www.mat.univie.ac.at/∼neum/glopt/coconut/Benchmark/Benchmark.html

[34] PHCPack, The database of polynomial systems (2003).
URL http://homepages.math.uic.edu/∼jan/demo.html

[35] A. Goldsztejn, L. Granvilliers, C. Jermann, Constraint based computation of periodic orbits of chaotic dynamical systems, in: 19th International Conference on Principles and Practice of Constraint Programming (CP'13), Vol. 8124 of LNCS, 2013, pp. 774–789.

[36] F. Margot, Symmetry in integer linear programming, in: 50 Years of Integer Programming 1958-2008, Springer, 2010, pp. 647–686.