Symmetry Breaking in Numeric Constraint Problems

Alexandre Goldsztejn[1], Christophe Jermann[1],
Vicente Ruiz de Angulo[2], and Carme Torras[2]

[1] Université de Nantes/CNRS LINA (UMR-6241)
2 rue de la Houssinière, Nantes, F-44300 France.
name.surname@univ-nantes.fr
[2] Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
Llorens i Artigas 4-6, 08028-Barcelona, Spain.
surname@iri.upc.edu

Abstract. Symmetry-breaking constraints in the form of inequalities between variables have been proposed for a few kind of solution symmetries in numeric CSPs. We show that, for the variable symmetries among those, the proposed inequalities are but a specific case of a relaxation of the well-known LEX constraints extensively used for discrete CSPs. We discuss the merits of this relaxation and present experimental evidences of its practical interest.

Keywords: Symmetries, Numeric constraints, Variable symmetries

1 Introduction

Numeric 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 [18], biochemistry, crystallography, robotics [13] and, more generally, in those related to global optimization [9]. 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.

Considerable work on symmetry breaking has been performed for discrete Constraint Satisfaction Problems (CSPs) in the last decades [7, 19]. Two main symmetry-breaking strategies have been pursued: 1) to devise specialized search algorithms that avoid symmetric portions of the search space [14, 8]; and 2) to add *symmetry-breaking constraints* (SBCs) that filter out redundant subspaces [5, 16]. Contrarily to this, 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 [17]. The addition of SBCs has also been proposed, but only for specific problems or specific symmetry classes, as inequalities between variables [6, 11, 3].

In Section 2, we show that such inequalities are but a relaxation of the lexicographic-ordering based SBCs [4] widely used by the discrete CSP community. This relaxation allows us to generalize these previous works to any variable symmetry and can be derived automatically knowing the symmetries of a problem. In Section 3 we discuss its merits with respect to lexicographic-ordering based SBCs. In Section 4 we assess its practical interest. We provide tracks for future developments in Section 5.

2 Symmetry-Breaking Constraints for NCSPs

We are interested in solving the following general *Numeric Constraint Satisfaction Problem* (NCSP) (X, D, C): Find all points $X = (x_1, \ldots, x_n) \in D \subseteq \mathbb{R}^n$ satisfying the constraint $C(X)$, a relation on $\mathbb{R}^n$, typically a conjunction of non-linear equations and inequalities.

A function $s : \mathbb{R}^n \to \mathbb{R}^n$ is a *symmetry* of a NCSP if it maps bijectively solutions to solutions[3], i.e., for all $X \in D$ such that $C(X)$ holds, $s(X) \in D$ and $C(s(X))$ also holds. In this case, we say X and $s(X)$ are *symmetric solutions*, and by extension for any point $Y \in D$, $s(Y)$ is a symmetric point. 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 i^σ, and σ is described by $[1^\sigma, 2^\sigma, \ldots, n^\sigma]$. A symmetry s is a *variable symmetry* iff there is a $\sigma \in \mathcal{S}_n$ such that for any $X \in D$, $s(X) = (x_{1^\sigma}, \ldots, x_{n^\sigma})$. We identify such symmetries with their associated permutations and denote both by σ in the following. Consequently, the set of variable symmetries of a NCSP is isomorphic to a permutation subgroup of $\mathcal{S}_n$, which are both identified and denoted by Σ in the following.

Example 1. The 3-cyclic roots problem is: find all $(x_1, x_2, x_3) \in \mathbb{R}^3$ satisfying $(x_1 + x_2 + x_3 = 0) \wedge (x_1x_2 + x_2x_3 + x_3x_1 = 0) \wedge (x_1x_2x_3 = 1)$. This problem has six variable symmetries including identity, $\Sigma = \{[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]\}$. Hence, all its variables are interchangeable. ◇

We say that the symmetries of a CSP are completely broken when a single representative in each set of symmetric solutions is retained. To this end, it is possible to add *symmetry-breaking constraints* (SBCs) which will exclude all but a single representative of the symmetric solutions [7, 19]. Crawford *et al.* [4] proposed *lexicographic ordering constraints* (LEX) that completely break any variable symmetry. Recall that given X and Y both in $\mathbb{R}^n$ the lexicographic order is defined inductively as follows:

$$\text{for } n = 1, \quad X \preceq_{lex} Y \;\equiv\; (x_1 \leq y_1)$$
$$\text{for } n > 1, \quad X \preceq_{lex} Y \;\equiv\; (x_1 < y_1) \,\vee\, \Big((x_1 = y_1) \,\wedge\, (X_{2:n} \preceq_{lex} Y_{2:n})\Big)$$

where $X_{2:n} = (x_2, \ldots, x_n)$, and the same for Y. For a given symmetry σ, Crawford *et al.* define the corresponding SBC $\text{LEX}_\sigma(X) \equiv X \preceq_{lex} \sigma(X)$. Intuitively, this constraint imposes a total order on the symmetric solutions, 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. The strength of these constraints is that they reduce the search space by a factor equal to $\#\Sigma$, the order of the symmetry group Σ of the problem. One critical issue however is that the number of SBCs can be exponential with respect to the number of variables.

[3] Nothing is required for non-solution points, i.e., we consider *solution symmetries* [1].

Example 2. Excluding the identity permutation, a symmetry of any problem which is irrelevant to break, the LEX constraints for the symmetries of the 3-cyclic-roots problem are: $(x_1, x_2, x_3) \preceq_{lex} (x_1, x_3, x_2)$, $(x_1, x_2, x_3) \preceq_{lex} (x_2, x_1, x_3)$, $(x_1, x_2, x_3) \preceq_{lex} (x_2, x_3, x_1)$, $(x_1, x_2, x_3) \preceq_{lex} (x_3, x_1, x_2)$, and $(x_1, x_2, x_3) \preceq_{lex} (x_3, x_2, x_1)$. ◇

Since they offer a good trade-off between the solving time reduction they allow, and the difficulty to handle them, *partial SBCs* (PSBCs), that retain *at least* one representative of the symmetric solutions, have often been considered. Especially for NCSPs, several classes of variable symmetries have been broken using PSBCs having the form of inequalities between variables. For instance, Gasca *et al.* [6] 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 ($\Sigma = \mathcal{C}_n$)[4]. Similar PSBCs have been proposed for numeric optimization problems with more peculiar symmetry groups, e.g., $\Sigma = \mathcal{C}_2 \times \mathcal{S}_n$ in [3] and $\Sigma = \prod_i \mathcal{S}_{p_i}$ in [11].

Example 3. Considering again the 3-cyclic-roots problem, Gasca *et al.*'s PSBCs are: $x_1 \leq x_2$ and $x_2 \leq x_3$. Indeed, these inequalities filter out all but a single of the six symmetries of any solution to this problem. ◇

The corner stone of our approach is to note that all the PSBCs mentioned above can be obtained by relaxing Crawford's SBCs as follows: For $\sigma \in \mathcal{S}_n$ different from the identity permutation, and $X = (x_1, \ldots, x_n)$, we define the constraint $\text{RLEX}_\sigma(X) \equiv x_{k_\sigma} \leq x_{{k_\sigma}^\sigma}$, where k_σ is the smallest integer in $\{1, \ldots, n\}$ such that $k_\sigma \neq {k_\sigma}^\sigma$. The following proposition establishes that this constraint is a relaxation of a LEX constraint, i.e., a PSBC: it cannot remove any solution preserved by LEX constraint.

Proposition 1. $\text{LEX}_\sigma(X) \Longrightarrow \text{RLEX}_\sigma(X)$

Proof. Since $i < k_\sigma$ implies $i = i^\sigma$, we have $x_i = x_{i^\sigma}$ for all $i < k_\sigma$. Therefore $\text{LEX}_\sigma(X)$, which is $X \preceq_{lex} \sigma(X)$, is actually equivalent to $X_{k_\sigma:n} \preceq_{lex} \sigma(X)_{k_\sigma:n}$, i.e.,

$$(x_{k_\sigma} < x_{{k_\sigma}^\sigma}) \ \vee \ \big((x_{k_\sigma} = x_{{k_\sigma}^\sigma}) \ \wedge \ (X_{k_\sigma+1:n} \preceq_{lex} \sigma(X)_{k_\sigma+1:n})\big),$$

which logically implies $(x_{k_\sigma} < x_{{k_\sigma}^\sigma}) \ \vee \ (x_{k_\sigma} = x_{{k_\sigma}^\sigma})$, that is $\text{RLEX}_\sigma(X)$. □

The ad-hoc inequalities proposed so far to partially break specific classes of variable symmetries in NCSPs are just special cases of the RLEX constraints. For instance, when $\Sigma = \mathcal{S}_n$, Gasca *et al.*'s PSBCs are $x_i \leq x_{i+1}$ $(i \in \{1, \ldots, n-1\})$ [6]. In this case, k_σ takes all possible values in $\{1, \ldots, n-1\}$ and ${k_\sigma}^\sigma$ all possible values in $\{k_\sigma + 1, \ldots, n\}$. Hence the corresponding RLEX constraints are $x_i \leq x_j$ $(i < j)$. Since all the inequalities $x_i \leq x_j$ with $i + 1 < j$ among them are redundant, they can be eliminated, yielding the inequalities proposed by Gasca *et al.*. A similar verification is easily carried out for the other specific variables symmetries tackled in [6, 3, 11]. Hence, RLEX constraints generalize these PSBCs to any variable symmetries.

Example 4. Continuing Example 2, the corresponding RLEX constraints are respectively: $x_2 \leq x_3$, $x_1 \leq x_2$, $x_1 \leq x_2$, $x_1 \leq x_3$ and $x_1 \leq x_3$. This set of inequalities can be simplified to $x_1 \leq x_2$ and $x_2 \leq x_3$, i.e., that presented in Example 3. ◇

[4] $\mathcal{C}_n = \{[k, \ldots, n, 1, \ldots, k-1] : k \in \{1, \ldots, n\}\}$

3 RLEX vs LEX

Advantages. First, we draw the reader's attention to the simplicity of the relaxed constraints w.r.t. the original ones: RLEX constraints are just binary inequalities while LEX constraints involve all the variables of the symmetries in a large combination of logical operations. Hence, we expect it is much more efficient to prune RLEX constraints (no specific algorithm is required) and to propagate the obtained reductions (successful reductions trigger only constraints depending on two variables), than LEX constraints.

Second, and more prominently, the number of RLEX constraint is always smaller than the number of LEX constraints, and it is bounded upward by $\frac{n(n-1)}{2}$ (number of different pairs (x_i, x_j) with $i < j$), or only $n - 1$ if one considers a non-redundant subset of inequalities as we explained previously. In contrast, there can be exponentially many LEX constraints, one for each permutation in $\mathcal{S}_n$. As remarked by Crawford *et al.*, this makes the use of LEX constraints impractical in general and has yielded research towards simplifying and relaxing them [4]. Oppositely, adding $O(n)$ RLEX constraints to a CSP model should never be a problem for its practical treatment by a solver.

Similar constraints $x_{k_\sigma} < x_{{k_\sigma}^\sigma}$ were proposed by Puget in [15] as SBCs for (discrete) problems where the variables are subject to an all different constraint. It is thus possible to obtain the RLEX constraints without having to compute all LEX constraints by applying the group theory results already used by Puget: From a generating set of the symmetries Σ of a problem, it is possible to derive a *stabilizer chain*, i.e., a sequence of permutation subgroups such that each is contained in the preceding and the permutations in the i^{th} subgroup map all integers in $\{1, \ldots, i\}$ to themselves. The orbit of the integer $i + 1$ in the i^{th} subgroup, i.e., all the integers it can be mapped to by any permutation in this subgroup, thus define exactly the pairs for which we must impose an inequality. These pairs can be obtained with the Shreier-Sims algorithm which runs in $O(n^2 \log^3(\#\Sigma) + tn\log(\#\Sigma))$, where t is the cardinality of the input generating set[5]. Since $\#\Sigma$ is at most $n!$ (when $\Sigma = \mathcal{S}_n$), this algorithm runs in polynomial time in n and t.

Hence, RLEX constraints constitute a generalization of the inequalities proposed so far for NCSPs that remains of tractable size and can be computed in polynomial time for any variable symmetries.

Drawbacks. The RLEX constraints break only partially the symmetries that LEX constraints break completely. Let us describe more precisely symmetric solutions which are discarded by LEX but not by RLEX.

Given a symmetry σ and a solution $X = (x_1, \ldots, x_n)$, if $\sigma(X)$ is discarded by the corresponding LEX constraint, it means that there exists i such that $x_i < x_{i^\sigma}$ and $\forall j \in \{1, \ldots, i-1\}$, $x_j = x_{j^\sigma}$. If $\sigma(X)$ is not discarded by the corresponding RLEX constraint $x_{k_\sigma} \leq x_{{k_\sigma}^\sigma}$, it means that $k_\sigma < i$. Thus, $x_{k_\sigma} = x_{{k_\sigma}^\sigma}$ while $k_\sigma \neq {k_\sigma}^\sigma$ by definition, i.e., X must lie on a given hyperplane $H_{uv} = \{X | x_u = x_v\}$.

Hence, all the symmetric solutions that are discarded by LEX constraints (w.r.t. all the symmetries of the problem) but not by RLEX constraints belong to such hyperplanes. Because the volume of these hyperplanes is null in $\mathbb{R}^n$, the set of points filtered out by

[5] A minimal generating set is $O(n)$ for any subgroup of variable symmetries.

LEX constraints and preserved by RLEX constraints represents a null volume of the search space. We conclude that RLEX constraints reduce the search space volume by a factor $\#\Sigma$ identical to that achieved with LEX constraints.

Moreover, numerical constraint solvers cannot eliminate these *singular* symmetric solutions even with LEX constraints since they do not distinguish strict and non-strict inequalities. Indeed, they perform computations using intervals and thus cannot approximate open sets differently from closed ones.

In conclusion, since the aim of PSBCs is essentially to enhance the solvers performances by allowing quick and easy reduction of the search space, it appears RLEX constraints are a very good trade-off between simplicity and efficiency: they are easy to derive, simple to handle, and still filter out most of the symmetric search space.

4 Experimental results

We provide experimental evidences of the important performance gains RLEX constraints can bring when solving symmetric NCSPs. Indeed, the solving time of a given NCSP is in general proportional to its search space. We expect RLEX constraints allow to quickly eliminate large portions of the search space, isolating an asymmetric subsearch space whose volume is divided by $\#\Sigma$ w.r.t. the initial search space. As a result, we expect to observe computation time gains proportional to $\#\Sigma$.

All experiments are conducted on a dual-core equipped machine (2.5GHz, 4Gb RAM) using the Realpaver [10] constraint solver with default settings.

Preliminar analysis: We first consider homemade scalable problems whose solutions either lie outside any hyperplane H_{uv} (problems P_1, P_2), all lie on such hyperplanes (problems P_3, P_4), or lie at the intersection of all these hyperplanes (problems P_5, P_6). In all cases, we consider problems with only cyclic permutations (P_1, P_3, P_5) and others with full permutations (P_2, P_4, P_6), i.e., problems for which the volume of the asymmetric search space is $\frac{1}{n}$ and $\frac{1}{n!}$ of that of the initial search-space respectively :

$$\mathbf{P_1} : \; X \in [-n,n]^n, \prod_{\sigma\in\mathcal{C}_n} ||\sigma(X) - X^*|| = 0$$

$$\mathbf{P_2} : \; X \in [-n,n]^n, \prod_{\sigma\in\mathcal{S}_n} ||\sigma(X) - X^*|| = 0$$

$$\mathbf{P_3} : \; X \in [-2,2]^n, \prod_{j=1}^{n}(\sum_{i=1}^{n}(x_{((i+j)\ mod\ n)} + (-1)^i)^2) = 0$$

$$\mathbf{P_4} : \; X =\in [0,1]^n, \forall i \in 1..n \; \sum_{j\neq i} x_j^2 + x_i \cos(\sum_{j=1}^{n} x_j) = 0$$

$$\mathbf{P_5} : \; X \in [-2,2]^n, \sum_{i=1}^{n}(x_i^2 - 1)^2 = 0$$

$$\mathbf{P_6} : \; X \in [0,1]^n, \forall i \in 1..n \; \sum_{j=1}^{n-1}(\prod_{l=1}^{n-1} x_{(i+j+l) mod n}) = 1$$

where X^* is the point $(1,\ldots,n) \in \mathbb{R}^n$. The solutions of P_1 are all cyclic permutations of X^* while that of P_2 are all permutations of X^*. The solutions of P_3 are the cyclic permutations[6] of $(-1,1,\ldots,-1,1) \in \mathbb{R}^n$; that of P_4 are all points of the form $\{-1,1\}^n$. P_5 and P_6 both have a single (very symmetric) solution: 0^n.

[6] Note there are only 2 different solutions when n is even, n solutions when it is odd.

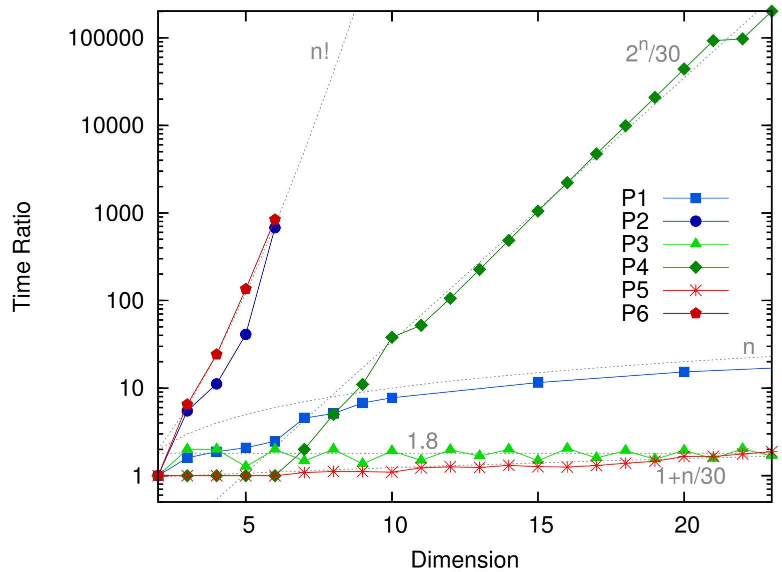

Fig. 1. Time ratios for homemade problems

Figure 1 presents the variation of the ratio between the computation time without RLEX and the computation time with it (called *gain* in the following) when the dimension n varies. In addition to the measured gains, the figure displays (in dotted gray) the functions of n that best approximate them.

The gains for P_1 and P_2 follow very closely the reduction factor of their search space volume, hence confirming our expectations. Note that although the gains are not as impressive for P_1 as for P_2, they are already significant: E.g., for $n = 50$ the computation time is 1124s ($>$ 18min) without RLEX and 29s with RLEX. For P_2 they are really outstanding: E.g., for $n = 6$, the computations time is 12863s ($>$ 3.5h) without RLEX but only 19s with RLEX.

For the other problems, the results are more varied: P_3 presents only an (almost) constant gain; P_4 shows a gain closer to the reduction factor of the size of its solution set than to its search space volume reduction factor; P_5 offers a (quite flat) linear gain, i.e., proportional to its search space volume reduction factor; the gain for P_6 follows closely its search space volume reduction factor[7]. The factors that could explain this diversity of behaviors are numerous (e.g., relative pruning power of the original constraints w.r.t. the added PSBCs, proportion of symmetric solutions with and without RLEX, ...). Further experiments will be necessary to distinguish the exact effects of all these factors.

The conclusion we draw from these results is that one cannot always expect as much gain as the search space volume reduction factor, especially when the problem

[7] Computations for P_6 could not be performed further because the timings were becoming too large, e.g., 41751s ($>$ 11.5h) for $n = 6$ without RLEX, as compared to 49.5s with it.

Problem	n	Sol	$\#\Sigma$	Time w/RLEX	Time w RLEX	gain
Brown	5	S	$n!$	0.95	0.24	3.9
	8			1218	5.32	229.0
Cyclic roots	4	GS*	$2n$	260	32.1	8.1
	5	S		46.6	4.7	9.7
	6	S		2017	183	10.9
Cyclohexane	3	S	$n!$	0.24	0.16	1.5
Extended Freudenstein	20	S	$\frac{n}{2}!$	0.41	0.26	1.6
	140			422	315	1.3
Extended Powell	16	S	$n\frac{n}{2}!$	1.42	0.03	47.3
	30			844	0.1	8442.0
Feigenbaum	11	GS	n	7.30	0.81	9.0
	23			10924	1027	10.6

Table 1. Results for various problems from the literature

has *singular* solutions; still, the gains can be outstanding, and adding RLEX constraints did not induce any uncompensated overhead in any of the settings we have considered.

Standard benchmark: We also consider a benchmark composed of standard problems picked from [2]. Their characteristics and the results obtained are reported in Table 1. For scalable problems we report timings for the smallest and largest dimension n we tested, allowing one to imagine the gain variation with the dimension. Column "Sol" indicates the type of solutions of the problem: G=Generic (i.e., out of any hyperplane H_{uv}) and S=Singular. Note that most of these standard problems are of type S. Problem *4-cyclic-roots* is marked GS^* because this problem has a continuous solution set which intersects some H_{uv} hyperplanes. For this problem, timings correspond to paving its solution manifold with 10^{-2}-wide boxes.

For problems *Brown*, *Cyclic-roots* and *Extended-Powell*, the gain closely follows the search space volume reduction factor (column $\#\Sigma$). Still, for problems *Extended-Freudenstein* and *Feigenbaum* the gains remain almost constant as the dimension grows. These experiments support the preliminary analysis we have performed: We can achieve important gains for highly symmetric problems and the introduction of RLEX constraints at least does not appear counterproductive.

5 Conclusion and Future Prospects

We have presented a generalization of the PSBCs proposed so far for variable symmetries in NCSPs. It corresponds to a relaxation of the famous LEX constraints used for breaking symmetries essentially for discrete CSPs so far. We have discussed the merits of this relaxation w.r.t. LEX constraints and illustrated its practical interest for NCSPs.

All the arguments we have used are also valid for continuous optimization and constrained optimization problems. Considering that many of them are not specific to numeric problems or solvers, it would also be interesting to consider this relaxation in discrete domains. Hence, we should also consider Mixed-Integer Linear/Nonlinear Programming and Integer Linear Programming where some of the PSBCs we have generalized have been proposed [3, 12].

References

1. Cohen, D., Jeavons, P., Jefferson, C., Petrie, K., Smith, B.: Symmetry definitions for constraint satisfaction problems. Constraints 11(2-3), 115–137 (2006)
2. COPRIN: The inria project COPRIN examples webpage. `http://www-sop.inria.fr/coprin/logiciels/ALIAS/Benches/` (2011)
3. Costa, A., Liberti, L., Hansen, P.: Formulation symmetries in circle packing. Electronic Notes in Discrete Mathematics 36, 1303 – 1310 (2010)
4. Crawford, J., Ginsberg, M., Luks, E., Roy, A.: Symmetry-breaking predicates for search problems. In: KR. pp. 148–159 (1996)
5. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I.: Breaking row and column symmetries in matrix models. In: CP '02: Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming. pp. 462–476. Springer (2002)
6. Gasca, R., Valle, C.D., Cejudo, V., Barba, I.: Improving the computational efficiency in symmetrical numeric constraint satisfaction problems. In: 11th Conference of the Spanish Association for Artificial Intelligence (CAEPIA). pp. 269–279 (2005)
7. Gent, I., Petrie, K., Puget, J.F.: Handbook of Constraint Programming, chap. Symmetry in constraint programming, pp. 329–376. Elsevier (2006)
8. Gent, I.P.: Groups and constraints: Symmetry breaking during search. In: In Proceedings of CP-02, LNCS 2470. pp. 415–430. Springer (2002)
9. Goldsztejn, A., Lebbah, Y., Michel, C., Rueher, M.: Capabilities of constraint programming in safe global optimization. In: International Symposium on Nonlinear Theory and its Applications. pp. 601–604 (2008)
10. Granvilliers, L., Benhamou, F.: Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques. ACM T. on Mathematical Software 32, 138–156 (2006)
11. Ji, X., Ma, F., Zhang, J.: Solving global unconstrained optimization problems by symmetry-breaking. In: 8^{th} IEEE/ACIS International Conference on Computer and Information Science. pp. 107–111 (2009)
12. Margot, F.: Symmetry in integer linear programming. In: 50 Years of Integer Programming 1958-2008, pp. 647–686. Springer (2010)
13. Merlet, J.P.: Interval analysis for certified numerical solution of problems in robotics. Applied Mathematics and Computer Science 19(3), 399–412 (2009)
14. Meseguer, P., Torras, C.: Exploiting symmetries within constraint satisfaction search. Artif. Intell. 129(1-2), 133–163 (2001)
15. Puget, J.F.: Breaking symmetries in all different problems. In: Proc. 19th international joint conference on Artificial intelligence (IJCAI). pp. 272–277 (2005)
16. Puget, J.F.: Symmetry breaking revisited. Constraints 10(1), 23–46 (2005)
17. Ruiz de Angulo, V., Torras, C.: Exploiting single-cycle symmetries in continuous constraint problems. Journal of Artificial Intelligence Research 34, 499–520 (2009)
18. Vu, X.H., Schichl, H., Sam-Haroud, D.: Interval propagation and search on directed acyclic graphs for numerical constraint solving. J. Global Optimization 45(4), 499–531 (2009)
19. Walsh, T.: Parameterized complexity results in symmetry breaking. In: Parameterized and Exact Computation, proc. 5^{th} International Symposium IPEC. Springer LNCS, vol. 6478, pp. 4–14 (2010)