Parameterized Complexity Results in Symmetry Breaking*

Toby Walsh

NICTA and University of New South Wales, email: toby.walsh@nicta.com.au

Abstract. Symmetry is a common feature of many combinatorial problems. Unfortunately eliminating all symmetry from a problem is often computationally intractable. This paper argues that recent parameterized complexity results provide insight into that intractability and help identify special cases in which symmetry can be dealt with more tractably.

1 Introduction

Symmetry occurs in many constraint satisfaction problems. For example, in scheduling a round robin sports tournament, we may be able to interchange all the matches taking place in two stadia. Similarly, we may be able to interchange two teams throughout the tournament. As a second example, when colouring a graph (or equivalently when timetabling exams), the colours are interchangeable. We can swap red with blue throughout. If we have a proper colouring, any permutation of the colours is itself a proper colouring. Problems may have many symmetries at once. In fact, the symmetries of a problem form a group. Their action is to map solutions (a schedule, a proper colouring, etc.) onto solutions.

Symmetry is problematic when solving constraint satisfaction problems as we may waste much time visiting symmetric solutions. In addition, we may visit many (failing) search states that are symmetric to those that we have already visited. One simple but effective mechanism to deal with symmetry is to add constraints which eliminate symmetric solutions [1]. Unfortunately eliminating all symmetry is NP-hard in general [2]. However, recent results in parameterized complexity give us a good understanding of the source of that complexity. In this survey paper, I summarize results in this area. For more background, see [3–7].

2 An example

To illustrate the ideas, we consider a simple problem from musical composition. The all interval series problem (prob007 in CSPLib.org [8]) asks for a permutation of the

* Supported by the Australian Government's Department of Broadband, Communications and the Digital Economy and the ARC. I wish to thank my co-authors in this area for their many contributions: Christian Bessiere, Brahim Hnich, Emmanuel Hebrard, George Katsirelos, Zeynep Kiziltan, Yat-Chiu Law, Jimmy Lee, Nina Narodytska Claude-Guy Quimper and Justin Yip.

numbers 0 to $n-1$ so that neighbouring differences form a permutation of 1 to $n-1$. For $n = 12$, the problem corresponds to arranging the half-notes of a scale so that all musical intervals (minor second to major seventh) are covered. This is a simple example of a graceful graph problem in which the graph is a path. We can model this as a constraint satisfaction problem in n variables with $X_i = j$ iff the ith number in the series is j. One solution for $n = 11$ is:

$$X_1, X_2, \ldots, X_{11} = 3, 7, 4, 6, 5, 0, 10, 1, 9, 2, 8 \tag{1}$$

The differences form the series: $4, 3, 2, 1, 5, 10, 9, 8, 7, 6$.

The all interval series problem has a number of different symmetries. First, we can reverse any solution and generate a new (but symmetric) solution:

$$X_1, X_2, \ldots, X_{11} = 8, 2, 9, 1, 10, 0, 5, 6, 4, 7, 3 \tag{2}$$

Second, the all interval series problem has a value symmetry as we can invert values. If we subtract all values in (1) from 10, we generate a second (but symmetric) solution:

$$X_1, X_2, \ldots, X_{11} = 7, 3, 6, 4, 5, 10, 0, 9, 1, 8, 2 \tag{3}$$

Third, we can do both and generate a third (but symmetric) solution:

$$X_1, X_2, \ldots, X_{11} = 2, 8, 1, 9, 0, 10, 5, 4, 6, 3, 7 \tag{4}$$

To eliminate such symmetric solutions from the search space, we can post additional constraints which eliminate all but one solution in each symmetry class. To eliminate the reversal of a solution, we can simply post the constraint:

$$X_1 < X_{11} \tag{5}$$

This eliminates solution (2) as it is a reversal of (1). To eliminate the value symmetry which subtracts all values from 10, we can post:

$$X_1 \leq 5, \;\; X_1 = 5 \Rightarrow X_2 < 5 \tag{6}$$

This eliminates solutions (2) and (3). Finally, eliminating the third symmetry where we both reverse the solution and subtract it from 10 is more difficult. We can, for instance, post:

$$[X_1, \ldots, X_{11}] \leq_{\text{lex}} [10 - X_{11}, \ldots, 10 - X_1] \tag{7}$$

Note that of the four symmetric solutions given earlier, only (4) with $X_1 = 2$, $X_2 = 8$ and $X_{11} = 7$ satisfies all three sets of symmetry breaking constraints: (5), (6) and (7). The other three solutions are eliminated.

3 Formal background

We will need some formal notation to present some of the more technical results. A *constraint satisfaction problem* (CSP) consists of a set of variables, each with a finite domain of values, and a set of constraints [9]. Each *constraint* is specified by the allowed combinations of values for some subset of variables. For example, $X \neq Y$ is a binary constraint which ensures X and Y do not take the same values. A *global constraint* is one in which the number of variables is not fixed. For instance, the global constraint $\text{NVALUE}([X_1, \ldots, X_n], N)$ ensures that n variables, X_1 to X_n, take N different values [10]. That is, $N = |\{X_i \mid 1 \leq i \leq n\}|$.

Constraint solvers typically use backtracking search to explore the space of partial assignments. After each assignment, constraint propagation algorithms prune the search space by enforcing local consistency properties like domain or bound consistency. A constraint is *domain consistent* (*DC*) iff when a variable is assigned any of the values in its domain, there exist compatible values in the domains of all the other variables of the constraint. Such values are called a *support*. A CSP is domain consistent iff every constraint is domain consistent.

Recently, Bessiere *et al.* have shown that a number of common global constraints are intractable to propagate [11, 12]. For instance, enforcing domain consistency on the NVALUE constraint is NP-hard [13, 14]. Parameterized complexity can provide a more fine-grained view of such results, identifying more precisely what makes a global constraint (in)tractable. We will say that a problem is *fixed-parameter tractable* (*FPT*) if it can be solved in $O(f(k)n^c)$ time where f is *any* computable function, k is some parameter, c is a constant, and n is the size of the input. For example, vertex cover ("Given a graph with n vertices, is there a subset of vertices of size k or less that cover each edge in the graph") is NP-hard in general, but fixed-parameter tractable with respect to k since it can be solved in $O(1.31951^k k^2 + kn)$ time [15]. Hence, provided k is small, vertex cover can be solved effectively.

4 Symmetry breaking

As we have argued, symmetry is a common feature of many real-world problems that dramatically increases the size of the search space if it is not factored out. Symmetry can be defined as a bijection on assignments that preserves solutions. The set of symmetries form a group under the action of composition. We focus on two special types of symmetry. A *value symmetry* is a bijective mapping, σ of the values such that if $X_1 = d_1, \ldots, X_n = d_n$ is a solution then $X_1 = \sigma(d_1), \ldots, X_n = \sigma(d_n)$ is also. For example, in our all interval series example, there is a value symmetry σ that maps the value i onto $n - i$. A *variable symmetry*, on the other hand, is a bijective mapping, θ of the indices of variables such that if $X_1 = d_1, \ldots, X_n = d_n$ is a solution then $X_{\theta(1)} = d_1, \ldots, X_{\theta(n)} = d_n$ is also. For example, in our all interval series example, there is a variable symmetry θ that maps the index i onto $n + 1 - i$. This swaps the variable X_i with X_{n+1-i}.

A simple and effective mechanism to deal with symmetry is to add constraints to eliminate symmetric solutions [1, 2, 16–19]. The basic idea is very simple. We pick an ordering on the variables, and then post symmetry breaking constraints to ensure that the final solution is lexicographically less than any symmetric re-ordering of the variables. That is, we select the "lex leader" assignment. For example, to break the variable symmetry θ, we post the constraint:

$$[X_1, \ldots, X_n] \leq_{\text{lex}} [X_{\theta(1)}, \ldots, X_{\theta(n)}]$$

Efficient inference methods exist for propagating such constraints [20, 21]. The symmetry breaking constraints in our all interval series example can all be derived from such lex leader constraints.

In theory, the lex leader method solves the problem of symmetries, eliminating all symmetric solutions and pruning many symmetric states. Unfortunately, the set of symmetries might be exponentially large (for example, in a graph k-colouring, there are $k!$ symmetries). There may therefore be too many symmetry breaking constraints to post. In addition, decomposing symmetry breaking into many lex leader constraints typically hinders propagation. We focus on three special but commonly occurring cases where symmetry breaking is more tractable and propagation can be more powerful: value symmetry, interchangeable values, and row and column symmetry. In each case, we identify islands of tractability but show that the quick-sands of intractability remain close to hand.

5 Value symmetry

Value symmetries are a commonly occurring symmetry that are more tractable to break [6]. For instance, Puget has proved that a linear number of symmetry breaking constraints will eliminate any number of value symmetries in polynomial time [22]. Given a set of value symmetries Σ, we can eliminate all value symmetry by posting the global constraint $\textsc{LexLeader}(\Sigma, [X_1, \ldots, X_n])$ [16]. This is a conjunction of lex leader constraints, ensuring that, for each $\sigma \in \Sigma$:

$$\langle X_1, \ldots, X_n \rangle \leq_{\text{lex}} \langle \sigma(X_1), \ldots, \sigma(X_n) \rangle$$

Unfortunately, enforcing domain consistency on this global constraint is NP-hard. However, this complexity depends on the number of symmetries. Breaking all value symmetry is fixed-parameter tractable in the number of symmetries.

Theorem 1 *Enforcing domain consistency on* $\textsc{LexLeader}(\Sigma, [X_1, \ldots, X_n])$ *is NP-hard in general but fixed-parameter tractable in* $k = |\Sigma|$.

Proof: NP-hardness is proved by Theorem 1 in [23], and fixed-parameter tractability by Theorem 7 in [24]. □

One situation where we may have only a small number of symmetries is when we focus on just the *generators* of the symmetry group [2, 25]. This is attractive as the size of the generator set is logarithmic in the size of the group, many algorithms in computational group theory work on generators, and breaking just the generator symmetries has been shown to be effective on many benchmarks [25]. In general, breaking just the generators may leave some symmetry. However, on certain symmetry groups (like that for interchangeable values considered in the next section), all symmetry is eliminated (Theorem 3 in [23]).

6 Interchangeable values

By exploiting special properties of the value symmetry group, we can identify even more tractable cases. A common type of value symmetry with such properties is that due to interchangeable values. We can break all such symmetry using the idea of value *precedence* [26]. In particular, we can post the global symmetry breaking constraint $\text{PRECEDENCE}([X_1, \ldots, X_n])$. This ensures that for all $j < k$:

$$\min\{i \mid X_i = j \vee i = n+1\} < \min\{i \mid X_i = k \vee i = n+2\}$$

That is, the first time we use j is before the first time we use k for all $j < k$. For example, consider the assignment:

$$X_1, X_2, X_3, \ldots, X_n = 1, 1, 1, 2, 1, 3, \ldots, 2 \tag{8}$$

This satisfies value precedence as 1 first occurs before 2, 2 first occurs before 3, etc. Now consider the symmetric assignment in which we swap 2 with 3:

$$X_1, X_2, X_3, \ldots, X_n = 1, 1, 1, 3, 1, 2, \ldots, 3 \tag{9}$$

This does not satisfy value precedence as 3 first occurs before 2. A PRECEDENCE constraint eliminates all symmetry due to interchangeable values. In [27], we give a linear time propagator for enforcing domain consistency on the PRECEDENCE constraint. In [23], we argue that PRECEDENCE can be derived from the lex leader method (but offers more propagation by being a global constraint).

Another way to ensure value precedence is to map onto dual variables, Z_j which record the first index using each value j [22]. This transforms value symmetry into variable symmetry on the Z_j. We can then eliminate this variable symmetry with some ordering constraints:

$$Z_1 < Z_2 < Z_3 < \ldots < Z_m \tag{10}$$

In fact, Puget proves that we can eliminate *all* value symmetry (and not just that due to value interchangeability) with a linear number of such ordering constraints. Unfortunately, this decomposition into ordering constraints hinders propagation even for the tractable case of interchangeable values (Theorem 5 in [23]). Indeed, even with *just* two value symmetries, mapping into variable symmetry can hinder propagation. This is supported by the experiments in [23] where we see faster and more effective symmetry breaking with the global PRECEDENCE constraint. This global constraint thus appears to be a promising method to eliminate the symmetry due to interchangeable values.

A generalization of the symmetry due to interchangeable values is when values partition into sets, and values within each set (but not between sets) are interchangeable. The idea of value precedence can be generalized to this case [27]. The global GENPRECEDENCE constraint ensures that values in each interchangeable set occur in order. More precisely, if the values are divided into s equivalence classes, and the jth equivalence class contains the values $v_{j,1}$ to v_{j,m_j} then GENPRECEDENCE ensures $\min\{i \mid X_i = v_{j,k} \vee i = n+1\} < \min\{i \mid X_i = v_{j,k+1} \vee i = n+2\}$ for all $1 \leq j \leq s$ and $1 \leq k < m_j$. Enforcing domain consistency on GENPRECEDENCE is NP-hard in general but fixed-parameter tractable in $k = s$ [23, 24].

7 Row and column symmetry

Another common type of symmetry where we can exploit special properties of the symmetry group is row and column symmetry [28]. Many problems can be modelled by a matrix model involving a matrix of decision variables [29–31]. Often the rows and columns of such matrices are fully or partially interchangeable [28]. For example, the Equidistant Frequency Permutation Array (EFPA) problem is a challenging combinatorial problem in coding theory. The aim is to find a set of v code words, each of length $q\lambda$ such that each word contains λ copies of the symbols 1 to q, and each pair of code words is at a Hamming distance of d apart. For example, for $v = 4$, $\lambda = 2$, $q = 3$, $d = 4$, one solution is:

$$\begin{array}{cccccc} 0&2&1&2&0&1\\ 0&2&2&1&1&0\\ 0&1&0&2&1&2\\ 0&0&1&1&2&2 \end{array} \tag{11}$$

This problem has applications in communications, and is closely related to other combinatorial problems like finding orthogonal Latin squares. Huczynska *et al.* [32] consider a simple matrix model for this problem with a v by $q\lambda$ array of variables, each with domains 1 to q. This model has row and column symmetry since we can permute the rows and the columns of any solution. Although breaking all row and column symmetry is intractable in general, it is fixed-parameter tractable in the number of columns (or rows).

Theorem 2 *With a n by m matrix, checking lex leader constraints that break all row and column symmetry is NP-hard in general but fixed-parameter tractable in $k = m$.*

Proof: NP-hardness is proved by Theorem 3.2 in [2], and fixed-parameter tractability by Theorem 1 in [33]. □

Note that the above result only talks about *checking* a constraint which breaks all row and column symmetry. That is, we only consider the computational cost of deciding if a complete assignment satisfies the constraint. Propagation of such a global constraint is computationally more difficult.

Just row or column symmetry on their own are tractable to break. To eliminate all row symmetry we can post lexicographical ordering constraints on the rows. Similarly, to eliminate all column symmetry we can post lexicographical ordering constraints on the columns. When we have both row and column symmetry, we can post a DOUBLELEX constraint that lexicographically orders both the rows and columns [28]. This does not eliminate all symmetry since it may not break symmetries which permute both rows and columns. Nevertheless, it is more tractable to propagate and is often highly effective in practice. Note that DOUBLELEX can be derived from a *strict* subset of the LEXLEADER constraints. Unfortunately propagating such a DOUBLELEX constraint completely is already NP-hard.

Theorem 3 *With a n by m matrix, enforcing domain consistency on* DOUBLELEX *is NP-hard in general.*

Proof: See Threorem 3 in [33]. □

There are two special cases of matrix models where row and column symmetry is more tractable to break. The first case is with an all-different matrix, a matrix model in which every value is different. If an all-different matrix has row and column symmetry then the lex-leader method ensures that the top left entry is the smallest value, and the first row and column are ordered [28]. Domain consistency can be enforced on such a global constraint in polynomial time [33]. The second more tractable case is with a matrix model of a function. In such a model, all entries are 0/1 and each row sum is 1. If a matrix model of a function has row and column symmetry then the lex-leader method ensures the rows and columns are lexicographically ordered, the row sums are 1, and the sums of the columns are in decreasing order [34, 35, 28]. Domain consistency can also be enforced on such a global constraint in polynomial time [33].

8 Related work

The study of computational complexity in constraint programming has tended to focus on the structure of the constraint graph (e.g. especially measures like tree width [36, 37]) or on the semantics of the constraints (e.g. [38]). However, these lines of research are mostly concerned with constraint satisfaction problems as a whole, and do not say much about individual (global) constraints. For global constraints of bounded arity, asymptotic analysis has been used to characterize the complexity of propagation both in general and for constraints with a particular semantics. For example, the generic domain consistency algorithm of [39] has an $O(d^n)$ time complexity on constraints of arity n and domains of size d, whilst the domain consistency algorithm of [40] for the n-ary ALLDIFFERENT constraint has $O(n^{\frac{3}{2}}d)$ time complexity. Bessiere *et al.* showed that many global constraints like NVALUE are also intractable to propagate [11]. More recently, Samer and Szeider have studied the parameterized complexity of the EGCC constraint [41]. Szeider has also studied the complexity of symmetry in a propositional resolution calculus [42]. See Chapter 10 in [43] for more about symmetry of propositional systems.

9 Conclusions

We have argued that parameterized complexity is a useful tool with which to study symmetry breaking. In particular, we have shown that whilst it is intractable to break all symmetry completely, there are special types of symmetry like value symmetry and row and column symmetry which are more tractable to break. In these case, fixed-parameter tractability comes from natural parameters like the number of generators which tend to be small. In future, we hope that insights provided by such analysis will inform the design of new search methods. For example, we might build a propagator that propagates completely when the parameter is small, but only partially when it is large. In the longer term, we hope that other aspects of parameterized complexity like kernels will find application in the domain of symmetry breaking.

References

1. Puget, J.F.: On the satisfiability of symmetrical constrained satisfaction problems. In Komorowski, J., Ras, Z., eds.: Proceedings of ISMIS'93. LNAI 689, Springer-Verlag (1993) 350–361
2. Crawford, J., Ginsberg, M., Luks, G., Roy, A.: Symmetry breaking predicates for search problems. In: Proceedings of the 5th International Conference on Knowledge Representation and Reasoning, (KR '96). (1996) 148–159
3. Walsh, T.: Symmetry within and between solutions. In Zhang, B.T., Orgun, M.A., eds.: PRICAI 2010: Trends in Artificial Intelligence, 11th Pacific Rim International Conference on Artificial Intelligence, Proceedings. Volume 6230 of Lecture Notes in Computer Science., Springer (2010) 11–13
4. Katsirelos, G., Walsh, T.: Symmetries of symmetry breaking constraints. In Coelho, H., Studer, R., Wooldridge, M., eds.: ECAI 2010 - 19th European Conference on Artificial Intelligence, Proceedings. Volume 215 of Frontiers in Artificial Intelligence and Applications., IOS Press (2010) 861–866
5. Heule, M., Walsh, T.: Symmetry in solutions. In Fox, M., Poole, D., eds.: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, AAAI Press (2010)
6. Walsh, T.: Breaking value symmetry. In Fox, D., Gomes, C., eds.: Proceedings of the 23rd National Conference on AI, Association for Advancement of Artificial Intelligence (2008) 1585–1588
7. Walsh, T.: Symmetry breaking. In Sattar, A., Kang, B., eds.: Australian Conference on Artificial Intelligence. Volume 4304 of Lecture Notes in Computer Science., Springer (2006) 7–8
8. Gent, I., Walsh, T.: CSPLib: a benchmark library for constraints. Technical report, Technical report APES-09-1999 (1999) A shorter version appears in the Proceedings of the 5th International Conference on Principles and Practices of Constraint Programming (CP-99).
9. Rossi, F., van Beek, P., Walsh, T., eds.: Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
10. Pachet, F., Roy, P.: Automatic generation of music programs. In Jaffar, J., ed.: Proceedings of Fifth International Conference on Principles and Practice of Constraint Programming (CP99), Springer (1999) 331–345
11. Bessiere, C., Hebrard, E., Hnich, B., Walsh, T.: The complexity of global constraints. In: Proceedings of the 19th National Conference on AI, Association for Advancement of Artificial Intelligence (2004)
12. Bessière, C., Hebrard, E., Hnich, B., Walsh, T.: The tractability of global constraints. In: 10th International Conference on Principles and Practices of Constraint Programming (CP-2004). Volume 3258 of Lecture Notes in Computer Science., Springer (2004) 716–720
13. Bessière, C., Hebrard, E., Hnich, B., Kiziltan, Z., Walsh, T.: Filtering algorithms for the NVALUE constraint. Constraints **11**(4) (2006) 271–293
14. Bessière, C., Hebrard, E., Hnich, B., Kiziltan, Z., Walsh, T.: Filtering algorithms for the NVALUE constraint. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 2nd International Conference (CPAIOR-2005). (2005)
15. Downey, R.G., Fellows, M.R., Stege, U.: Parameterized complexity: A framework for systematically confronting computational intractability. In: Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future. Volume 49 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. (1999) 49–99
16. Walsh, T.: General symmetry breaking constraints. In Benhamou, F., ed.: Principles and Practice of Constraint Programming - CP 2006, 12th International Conference, CP 2006,

Nantes, France, September 25-29, 2006, Proceedings. Volume 4204 of Lecture Notes in Computer Science., Springer (2006)

17. Katsirelos, G., Narodytska, N., Walsh, T.: Combining symmetry breaking and global constraints. In Oddi, A., Fages, F., Rossi, F., eds.: Recent Advances in Constraints, 13th Annual ERCIM International Workshop on Constraint Solving and Constraint Logic Programming, CSCLP 2008. Volume 5655 of Lecture Notes in Computer Science., Springer (2008) 84–98
18. Law, Y.C., Lee, J., Walsh, T., Yip, J.: Breaking symmetry of interchangeable variables and values. In: 13th International Conference on Principles and Practices of Constraint Programming (CP-2007), Springer-Verlag (2007)
19. Hnich, B., Kiziltan, Z., Walsh, T.: Combining symmetry breaking with other constraints: lexicographic ordering with sums. In: Proceedings of the 8th International Symposium on the Artificial Intelligence and Mathematics. (2004)
20. Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Global constraints for lexicographic orderings. In: 8th International Conference on Principles and Practices of Constraint Programming (CP-2002), Springer (2002)
21. Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Propagation algorithms for lexicographic ordering constraints. Artificial Intelligence **170**(10) (2006) 803–908
22. Puget, J.F.: Breaking all value symmetries in surjection problems. In van Beek, P., ed.: Proceedings of 11th International Conference on Principles and Practice of Constraint Programming (CP2005), Springer (2005)
23. Walsh, T.: Breaking value symmetry. In: 13th International Conference on Principles and Practices of Constraint Programming (CP-2007), Springer-Verlag (2007)
24. Bessière, C., Hebrard, E., Hnich, B., Kiziltan, Z., Quimper, C.G., Walsh, T.: The parameterized complexity of global constraints. In Fox, D., Gomes, C., eds.: Proceedings of the 23rd National Conference on AI, Association for Advancement of Artificial Intelligence (2008) 235–240
25. Aloul, F., Ramani, A., Markov, I., Sakallah, K.: Solving difficult SAT instances in the presence of symmetries. In: Proceedings of the Design Automation Conference. (2002) 731–736
26. Law, Y., Lee, J.: Global constraints for integer and set value precedence. In: Proceedings of 10th International Conference on Principles and Practice of Constraint Programming (CP2004), Springer (2004) 362–376
27. Walsh, T.: Symmetry breaking using value precedence. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: ECAI 2006, 17th European Conference on Artificial Intelligence, August 29 - September 1, 2006, Riva del Garda, Italy, Including Prestigious Applications of Intelligent Systems (PAIS 2006), Proceedings. Volume 141 of Frontiers in Artificial Intelligence and Applications., IOS Press (2006) 168–172
28. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Breaking row and column symmetry in matrix models. In: 8th International Conference on Principles and Practices of Constraint Programming (CP-2002), Springer (2002)
29. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Matrix modelling. Technical Report APES-36-2001, APES group (2001) Presented at Formul'01 Workshop on Modelling and Problem Formulation, CP2001 post-conference workshop.
30. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Matrix modelling: Exploiting common patterns in constraint programming. In: Proceedings of the International Workshop on Reformulating Constraint Satisfaction Problems. (2002) held alongside CP-2002.
31. Walsh, T.: Constraint patterns. In Rossi, F., ed.: 9th International Conference on Principles and Practices of Constraint Programming (CP-2003), Springer (2003)
32. Huczynska, S., McKay, P., Miguel, I., Nightingale, P.: Modelling equidistant frequency permutation arrays: An application of constraints to mathematics. In Gent, I., ed.: Principles

and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20-24, 2009, Proceedings. Volume 5732 of Lecture Notes in Computer Science., Springer (2009) 50–64

33. Katsirelos, G., Narodytska, N., Walsh, T.: On the complexity and completeness of static constraints for breaking row and column symmetry. In Cohen, D., ed.: Principles and Practice of Constraint Programming - CP 2010, 16th International Conference, Proceedings. Lecture Notes in Computer Science, Springer (2010)
34. Shlyakhter, I.: Generating effective symmetry-breaking predicates for search problems. Electronic Notes in Discrete Mathematics **9** (2001) 19–35
35. Flener, P., Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Symmetry in matrix models. Technical Report APES-30-2001, APES group (2001) Presented at SymCon'01 (Symmetry in Constraints), CP2001 post-conference workshop.
36. Freuder, E.: A sufficient condition for backtrack-free search. Journal of the Association for Computing Machinery **29**(1) (1982) 24–32
37. Dechter, R., Pearl, J.: Tree clustering for constraint networks. Artificial Intelligence **38** (1989) 353–366
38. Cooper, M., Cohen, D., Jeavons, P.: Characterizing tractable constraints. Artificial Intelligence **65** (1994) 347–361
39. Bessiere, C., Régin, J.: Arc consistency for general constraint networks: Preliminary results. In: Proceedings of the 15th International Conference on AI, International Joint Conference on Artificial Intelligence (1997) 398–404
40. Régin, J.C.: A filtering algorithm for constraints of difference in CSPs. In: Proceedings of the 12th National Conference on AI, Association for Advancement of Artificial Intelligence (1994) 362–367
41. Samer, M., Szeider, S.: Tractable cases of the extended global cardinality constraint. In: Proceedings of CATS 2008, Computing: The Australasian Theory Symposium. (2008)
42. Szeider, S.: The complexity of resolution with generalized symmetry rules. Theory of Computing Systems **38**(2) (2005) 171–188
43. Biere, A., Heule, M., van Maaren, H., Walsh, T., eds.: Handbook of Satisfiability. IOS Press, Amsterdam (2009)