Improving the Computational Efficiency in Symmetrical Numeric Constraint Satisfaction Problems

R.M. Gasca, C. Del Valle, V. Cejudo, and I. Barba

Departmento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. (Spain)
{gasca, carmelo, cejudo, irene}@lsi.us.es

Abstract. Models are used in science and engineering for experimentation, analysis, diagnosis or design. In some cases, they can be considered as numeric constraint satisfaction problems ($NCSP$). Many models are symmetrical $NCSP$. The consideration of symmetries ensures that $NCSP$-solver will find solutions if they exist on a smaller search space. Our work proposes a strategy to perform it. We transform the symmetrical $NCSP$ into a new $NCSP$ by means of addition of symmetry-breaking constraints before the search begins. The specification of a library of possible symmetries for numeric constraints allows an easy choice of these new constraints. The summarized results of the studied cases show the suitability of the symmetry-breaking constraints to improve the solving process of certain types of symmetrical $NCSP$. Their possible speed-up facilitates the application of modelling and solving larger and more realistic problems.

1 Introduction

Symmetries are common in science and engineering applications. It is due to the inherent symmetry of the physical world. Some examples are: electron spin in atoms, inertial changes in mechanics, electromagnetism, some organic chemical compounds, etc... Many problems in these applications exhibit a high degree of symmetry that can be exploited successfully for solving them.

Backtracking and consistency techniques are conventional methods to solve constraint satisfaction problems ($CSPs$). Reducing complexity is the major issue in solving $CSPs$, specially when there is a large number of constraints and/or wide domains of the variables. In first works, the symmetries have been considered in problems where variables have discrete domains and to remove symmetrical solutions by changing the CSP solved [14]. Also, the symmetries are used in the detection and exploitation in planning problems [6], reasoning and optimization [4], the generation of balanced incomplete block designs [13] and applications to low autocorrelation binary sequences [8]. Other approaches consider the design of a new search method that avoids testing of possible symmetrical subsolutions [1] and the addition of constraints during the search [15].

Numeric Constraint satisfaction problems ($NCSP$) are more and more often used to solve engineering problems arisen in different areas of Artificial Intelligence (qualitative reasoning, diagnosis, planning, scheduling, configuration, distributed artificial intelligence, etc...). These problems are formed by a set of constraints among variables whose domains are real interval values. Constraint-solving systems have as a goal to find all solutions, only one solution, or if the model that represents the problem is or not consistent.

The symmetrical $NCSP$ ($SNCSP$) has a set of symmetrical properties or symmetrical constraints. These problems that contain symmetries may be solved most efficiently. The search effort can be reduced specially on hard and large problems. In Numerica [16] the symmetry property of $NCSP$ has been considered by means of soft constraints. These constraints are equivalent to other ones except that they are ignored when the existence of solutions is proved.

Reasoning about the ranges of values of variables is a type of reasoning often used when there are inaccurate data or partially defined parameters. It can be also generalized as a $NCSP$. A natural way of reasoning on the ranges of values is to propagate the domains of the variables through the constraints. It involves assigning values to variables in order to satisfy constraints among subsets of those variables. These problems can be solved efficiently by combining local consistency methods, such as approximations of arc-consistency, together with a backtracking-based search. Different problem solving techniques have been proposed in the bibliography [9] [11] [2] [16] [10]. The search space in numeric constraint problems is usually too wide and a lot of these techniques have a major drawback since they introduce choice points and they are exponential. The efficiency of some previous algorithms was analyzed in a previous work [3]. Another work [16] for improving the accuracy and efficiency in solving $NCSP$ has been proposed by reducing dependencies and variable elimination. Our proposal considers that the symmetry property of $NCSP$ can speed-up significantly its solving process.

Although $NCSP$ modeling is the necessary step preceding CSP solving, little work has been done to help modelers. In this line, our method provides a library of symmetries that allows modelers to remove some symmetries in $SNCSP$. The modeler selects and adds symmetry-breaking constraints of this library, reducing the search space. These additions are performed by hand and require a previous analysis of the modeler. It must not affect the soundness and completeness of the solutions. Nevertheless, the modeler must also consider the increase of the computational cost after adding these new constraints in the original $NCSP$. This paper is interested in reformulating concise models that support more efficient solutions.

The rest of the article is organized as follows. In Section 2, we start presenting some examples of $SNCSPs$ to introduce the problem domain. Section 3 presents some definitions and preliminaries. Section 4 exposes a simple library of symmetries to be taken into account in symmetrical $NCSPs$ and proposes the selection of modelling schemas and their solving process. The experimental results are presented in Section 5. Finally, in the last section we present our conclusions and future work.

2 Illustrative Examples

In order to clarify the aim of this work, the following geometrical problem is very illustrative:

$$Model \equiv \begin{cases} X = \{x, y\} \\ D = \{x, y \in (-\infty, +\infty), \\ C = \{x^2 + y^2 = 4, x * y = 1\} \\ G = AllSolutions(X) \end{cases}$$

For this problem, the search space is $\mathbb{R}^2$, but if we consider the symmetries, this space is reduced four times. In fact, this reduction improves significantly the computational complexity of the search process.

Also we have selected two examples from different Artificial Intelligence areas, the first problem is commonly used in order of magnitude reasoning and the second one is an example of a configuration task. Both examples are specified by means of a four-tuple:

- **X**, the set of variables of the model,
- **D**, the set of domains of the variables,
- **C**, the set of relations of the variables of the model, and
- **G** the goal of the model analysis.

This representation allows an easy mapping from this specification to a $NCSP$.

2.1 A Countercurrent Heat-Exchanger

This problem is studied in order of magnitude reasoning [12] and [5]. A schema of a countercurrent heat-exchanger is shown in Figure 1. The important variables in the analysis of the device are the molar-heat KH and the molar-flowrate FH of the hot stream, and the molar-heat KC and the molar-flowrate FC of the cold stream. Also, the temperature differences have been named $DTH = Th_1 - Th_2$, $DTC = Tc_1 - Tc_2$, $DT_1 = Th_1 - Tc_1$, $DT_2 = Th_2 - Tc_2$. The temperature drop of the hot stream is DTH, the temperature rise of the cold stream is DTC and the driving force at the left and right ends of the device are DT_1 and DT_2

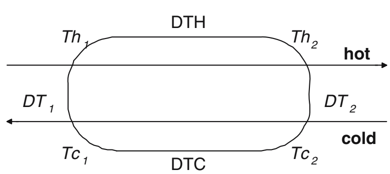

Fig. 1. A countercurrent heat-exchanger

respectively. The numeric constraints of the problem are the energy balance of the exchanger and the result from the definition of the temperature differences: $DTH * KH * FH = DTC * KC * FC$ and $DTH - DT_1 - DTC + DT_2 = 0$. In a particular case, the following order of magnitude relations may be known: DT_2 is *moderately smaller than* DT_1 and DT_1 is *much smaller than* DTH. A possible analysis of the model could consists in obtaining the qualitative relation between FC and FH.

$$Model \equiv \begin{cases} \mathbf{X} = \{DTH, KH, FH, DTC, KC, FC, DT_1, DT_2\} \\ \mathbf{D} = \{DTH, KH, FH, DTC, KC, FC, DT_1, DT_2 \in (-\infty, +\infty), \\ \qquad r_1 \in [0.1, 0.9]\ r_2 \in [0, 0.1]\ r_3 \in [0, +\infty)\} \\ \mathbf{C} = \{-DTC *\ FC *\ KC + DTH *\ FH *\ KH = 0, \\ \qquad -DTC + DTH - DT_1 + DT_2 = 0, DT_2 - DT_1\, r_1 = 0, \\ \qquad DT_1 - DTH\, r_2 = 0, FC - FH\, r_3 = 0\} \\ \mathbf{G} = Solution(r_3) \end{cases}$$

where r_1, r_2 are the corresponding intervals for the order of magnitude relations and r_3 is the interval for the unknown order of magnitude relation. Due to the symmetrical properties of this model it may be considered as a $SNCSP$.

In the last section, the modeler specifies that he would like to know one solution of the unknown order of magnitude relation.

2.2 Structural Configuration of Resistors

This example presents a given configuration of resistors as shows Figure 2. The modeler has the aim of selecting from two types of resistors whose values may be $[9.8, 10.1]\,\Omega$ or $[99.6, 100.4]\,\Omega$ in order to obtain an equivalent total resistor whose value is $[149.5, 150.5]\ \Omega$ according to the previous configuration. This problem can be modelled by means of the four-tuple below. The goal of the modeler is to search for only one possible solution.

This problem can be solved using the previous mentioned $NCSP$ techniques, but the computational effort is exponential. We propose to exploit the symmetries of this problem in order to achieve better computational costs.

$$Model \equiv \begin{cases} \mathbf{X} = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R\} \\ \mathbf{D} = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, \\ \qquad R_{10} \in [9.8, 10.1], [99.6, 100.4], R \in [149.5, 150.5]\} \\ \mathbf{C} = \{(R_1 * R_2)/(R_1 + R_2) + \\ \qquad (R_3 + R_4) * R_5 * R_6/(R_3 + R_4 + R_5 + R_6) + \\ \qquad (R_7 + R_8) * (R_9 + R_{10})/(R_7 + R_8 + R_9 + R_{10}) = R \\ \mathbf{G} = OneSolution(X) \end{cases}$$

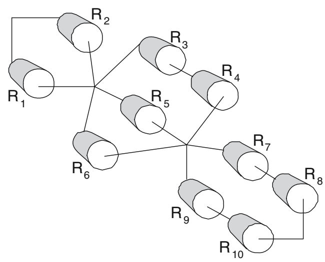

Fig. 2. Configuration problem with resistors

3 Definitions and Notation

We have presented informally *SNCSP* in terms of examples. This section presents a number of definitions for the formalization of symmetrical NCSP. The variable domains in these problems are real intervals.

Definition 1 (Interval). Let $\mathbb{F}$ denote a finite subset of $\mathbb{R}$ augmented with the symbols $\{-\infty, \infty\}$. If $a, b \in \mathbb{F}$ then an interval $[a, b]$ represents the set of real numbers $\{r \in \mathbb{R} \mid a \leq r \leq b\}$.

Definition 2 (Numeric Variable). A variable of the model whose domain is a real interval. The set of numeric variables of the problem is denoted by X.

Definition 3 (Numeric Constraint). It is a relation (equations) involving a finite subset of numeric variables.

Definition 4 (Goal). A predicate that denotes the users' preferences to search for only one solution, all solutions or the consistent domain of certain variables.

Definition 5 (Numeric Satisfaction Problem). A four-tuple $P=(X,D,C,G)$ where $X = \{x_1, ..., x_n\}$ is a set of variables, whose continuous domains are respectively $D = \{d_1, ..., d_n\}(n \geq 1)$, $C = \{c_1, ..., c_m\}(m \geq 1)$ is a set of numeric constraints and G are the goals.

Definition 6 (Symmetrical NCSP). A *NCSP* that has symmetry constraints.

Definition 7 (Solution). An instantiation of the numeric variables such that all constraints are satisfied. They correspond to n-dimensional cubes, that are named *hypercubes*.

We use Ψ to denote a $SNCSP(X, D, C, G)$. Ψ has a set of transformations $T_\Psi = \{T_1, ...T_m\}$. Every transformation $T_i \in T_\Psi$ has associated a finite set of expressions over the variables of the problem $\alpha_i = \{\alpha_i^1(X), \alpha_i^2(X), ..., \alpha_i^k(X)\}$ and a set of real regions associated to the previous expressions $\Omega_i = \{\Omega_i^1(X), \Omega_i^2(X), ..., \Omega_i^k(X)\}$. For example in the following *SNCSP*:

$$Model \equiv \begin{cases} X = \{x, y\} \\ D = \{x, y \in (-\infty, +\infty), \\ C = \{x^2 + y^2 = 4, x * y = 1\} \\ G = AllSolutions(X) \end{cases}$$

some transformations are shown in Table 1.

Table 1. Symmetrical transformations of a previous example

Transformations	α	Ω
$T_1 \equiv Permutations(x, y)$	$\alpha_1^1(x, y) \equiv x = y \wedge y = x$	$\Omega_1^1(x, y) \equiv y \leq x$
		$\Omega_1^2(x, y) \equiv y \geq x$
$T_2 \equiv Symmetry(x, y)$	$\alpha_2^1(x, y) \equiv x = -x \wedge y = -y$	$\Omega_2^1(x, y) \equiv x + y \geq 0$
		$\Omega_2^2(x, y) \equiv x + y \leq 0$

Every symmetrical transformation belonging to the set T_Ψ must be invariant. Let $T_i \in T_\Psi$ be a transformation that has associated an expression $X = \alpha_i(X)$. A transformation is invariant if $T_i(\Psi) \equiv \Psi$ where $\equiv$ denotes symbolic equality. The transformation of Ψ is obtained by means of the symbolic substitution of the variables X by expressions $\alpha_i(X)$,

$$T_i(\Psi) = \Psi[\alpha_i(X)/X] \quad i \in \{1, ..., m\}$$

Let $\Omega_i(X)$ be a finite set of associated regions for a transformation $T_i(\Psi)$, where

$$\Omega_i(X) = \{\Omega_i^1(X), \Omega_i^2(X), ..., \Omega_i^k(X)\}$$

The elements of this set must satisfy the following properties:

- Minimum overlapping of regions $\forall s, t : s, t \in 1..k \wedge s \neq t \mid \Omega_i^s(X) \cap \Omega_i^t(X)$ may be a hyperplane of a lower dimension with respect to the dimension of the search space.
- To fill in the search space $\bigcup_j \Omega_i^j(X) = I\!R^n$ where n is the cardinality of the set X and $j \in \{1, 2, ..., k\}$.
- Obtaining the next regions

  $$\forall s \in 1..k - 1 \mid T_i(\Omega_i^s(X)) = \Omega_i^{s+1}(X)$$

  and if k is finite, then $T_i(\Omega_i^k(X)) = \Omega_i^1(X)$

4 Symmetry Analysis in Symmetrical NCSP

The aim of symmetry analysis is the identification of symmetries and the generation of the corresponding symmetry-breaking constraints. The search for symmetries in the initial $SNCSP$ are performed by hand and requires a previous symmetry analysis. This analysis always must ensure the soundness and completeness of the search. The symmetry analysis has the following steps:

1. Identification of possible symmetries of a $SNCSP$. These problems may have an infinite number of symmetries and also some symmetries are hard to calculate. It determines that in this work we only identify the simplest symmetries. There are several approaches to identify these symmetries:
   - the symmetry-breaking constraints are derived from the model.
   - the symmetry-breaking constraints are derived from a visual representation of the model
   - Some tools calculate the symmetry-breaking constraints.

   The first item of the above list may be adequate to carry out the identification of the symmetries in symmetrical $NCSP$.
2. Reduction of the domain variables or generation of symmetry-breaking constraints, that are related to the different regions of the transformations.
3. Determination of the expressions to calculate symmetrical solutions and transformation of the variables' domains in order to ensure the completeness of the search. It depends on the users' goals. For example, if the goal is to obtain all the solutions of a $SNCSP$ then we determine these expressions.

4.1 Numeric Symmetries Library

The consideration of symmetries in symmetrical $NCSP$ can be a hard process. There is no known polynomial algorithm to detect all the possible symmetries. This work only considers a partial analysis of the symmetries which covers the most elemental numeric symmetries. It is named Numeric Symmetry Library (NSL) and includes the following symmetries:

- **Permutations**
  1. Transformation: Given a set $X' \subseteq X$ whose elements are $X' = \{x_1, ..., x_m\}$, the number of permutations in this set is $k = m!$.
  2. The regions of the transformation *Permutation* are

$$\Omega^1(X) \equiv x_1 \leq x_2 \leq ... \leq x_m$$

$$\Omega^2(X) \equiv x_2 \leq x_1 \leq ... \leq x_m$$

$$...$$

$$\Omega^k(X) \equiv x_m \leq ... \leq x_2 \leq x_1$$

     One of these regions can be added to break symmetries.
  3. The symmetric solutions are obtained as the permutations of the previous solutions. According to the notation used before, if s_1 is a solution of Ψ then:

$$s_2 = P^2(s_1), \quad s_3 = P^3(s_2), ..., s_k = P^k(s(k-1))$$

- **Symmetries with respect to a hyperplane with all variables of the SNCSP**
  1. One possible transformation T_i could be $T_i \equiv \alpha_i(X) = -X$
  2. The regions could be

$$\Omega_i^1 \equiv x_1 + ... + x_n \geq 0$$

$$\Omega_i^2 \equiv x_1 + ... + x_n \leq 0$$

  3. The symmetric solutions have the same absolute values than solutions obtained in solving the $SNCSP$ but the sign of these values must be changed.
- **Symmetries with respect to a hyperplane with a subset of variables of the SNCSP**
  Given a subset $X' \subset X$ whose variables are $\{x_1, ..., x_m\}$
  1. The transformation in this case could be $T_i \equiv \alpha_i(X') = -X'$
  2. The regions could be $\Omega_i^1 \equiv x_1 + x_2 + ... + x_m \geq 0$ and $\Omega_i^2 \equiv x_1 + x_2 + ... + x_m \leq 0$.
  3. The symmetric solutions of the $SNCSP$ are obtained changing the sign of the following variables $\{x_1, .., x_m\}$ in the solutions of the $SNCSP$.

  The modeler must check if the hyperplane $x_1 + x_2 + ... + x_m = 0$ in $\mathbb{R}^m$ is a symmetry hyperplane for all the surfaces that represent the $SNCSP$.
- **Translations**
  These transformations are convenient in $SNCSPs$ with trigonometric or periodic functions.
  1. The transformation in this case is $T_i \equiv \alpha_i(x) = x + \tau$ where τ is a real number.
  2. The regions are $\Omega_i^1 \equiv 0 \leq x \leq \tau$, $\Omega_i^2 \equiv k \leq x \leq 2 * \tau$, ... The number of regions is determined by the domain of the variable x.
  3. The symmetric solutions are obtained from the original $SNCSP$ such that if s_1 is a solution then $s_2 = s_1 + \tau, s_3 = s_2 + \tau, ...$

Therefore, the symmetrical analysis must determine from the previous NSL which type of symmetries are most convenient in a particular $SNCSP$. Modeler must also analyze the pruning of the search space resulted from the addition of symmetry-breaking constraints and the complexity of the treatment of these symmetry-breaking constraints. The efficiency of the numeric constraint solver must be improved in any case.

The modelling of symmetrical $NCSPs$ is oriented to a search process of solutions. A broad variety of approaches have been focused to solve $NCSPs$, essentially, exhaustive or/and local search techniques. Our work considers exclusively the exhaustive approach. Modeler must determine a modelling schema of $SNCSPs$ that considers the symmetries before the search begins.

The key idea is to obtain from a $SNCSP$ an equivalent $NCSP$ by the addition of symmetry-breaking constraints and/or the update domains. This idea can be represented as

$$SNCSP(X, D, C, G) \overset{\text{SymmetryAnalysis}}{\longrightarrow} NCSP(X, D, C', G, T)$$

The initial Symmetrical $NCSP$, denoted as Ψ, is transformed by means of a previous symmetry analysis into a new $NCSP$ where $C \subset C'$ and T are the set of transformations identified in the symmetry analysis. The conversion is shown in the following expression:

$$\Psi \equiv \{X, D, C \wedge \Omega^1_{T_1} \wedge \Omega^1_{T_2} ... \wedge \Omega^1_{T_k}, G\}$$

This modeling methodology proposes that the modeler must perform the symmetry analysis according to the particular syntax of a symmetrical $NCSP$ and *NSL*. It allows the identification of the symmetry-breaking constraints that participate in the modelling schema of this problem. The transformations T, which allows the subsequent calculation of the symmetrical solutions are obtained from *NSL*.

The specification of a $SNCSP$ must consider the symmetries belonging to the *NSL*. Illustrative examples, that we presented in the section 2 of this article, have some symmetries. Table 2 shows the symmetries that we propose in the different example models.

Table 2. Symmetry-breaking constraints of the example models

Problem	Symmetry-breaking constraints
Geometrical	$x \leq y, x + y \leq 0$
Heat-exchanger	$DTH + KH + FH + DTC + KC + FC + DT1 + DT2 \geq 0$
Resistors	$R_1 \geq R_2,\ R_5 \geq R_6, R_7 + R_8 \geq R_9 + R_{10},\ R_7 \geq R_8,\ R_9 \geq R_{10}$

These previous schemas must be solved by means of the adequate algorithm. If the goal is to search for all solutions, then we will apply the corresponding transformations to obtain the symmetric solutions.

5 Experimental Results

We have used the previous examples in the experimentation with some extensions. The heat-exchanger example considers one and two heat exchangers in series. The resistors example considers two configurations with ten and thirteen resistors respectively. The results are a mean value considering different variable domains and goals. All the experiments have been performed on the same machine. The program with schema modelling has been run five times on each problem instance, and the results displayed are an average of these five runs. The application to the different proposed problems is shown in Table 3. These results show solutions calculation without considering the CPU computing time of obtaining symmetry-breaking constraints and symmetrical solutions. In the underconstrained problem (heat-exchanger), the computational time is the same in both cases. It indicates that the reduction of the search space is not compensated with a reduction of the computational time. In the other problems, the

Table 3. Computational results of the symmetry and no symmetry excluding in schema models with bounded solutions

Model	Symmetry-excluding		No symmetry exclusion	
	Fails/Choice Points	Cpu sec.	Fails/Choice Points	Cpu sec.
Heat-exchanger	7/0	0.015	7/0	0.016
Heat-exchanger2	10/0	0.018	10/0	0.019
Resistors10	232/285	0.083	1343/2009	0.631
Resistors13	1746/2535	1.590	15852/26824	25.839

computational time is reduced. Then we can conclude that the computational efficiency of symmetries depends on the type of symmetrical $NCSP$ that the modeler specifies.

6 Conclusions and Future Work

In this work we propose a strategy to consider symmetry-breaking in symmetrical numeric constraint satisfaction problems. The addition of constrains to break symmetries in these problems reduces the search space. The modeler must consider the tradeoff between the increase of computational treatment of these constraints and the previous reduction what is important in certain underconstrained problems. The use of breaking-symmetries constraints provides significant computational savings in a lot of problems. Their speed-up facilitates the application of modeling and solves larger and more realistic problems.

In future work, we also would like to eliminate the possible redundancies in the calculation of symmetry-breaking constraints to reduce the computational complexity of the $SNCSP$. Another interesting research area is the automatic insertion of symmetry-breaking constraints in runtime, when a new symmetry appears during the search. A future application of the symmetrical reduction of $NCSP$ will be the efficient modelling in engineering projects.

Acknowledgements

This work has been partially supported by the Spanish *Ministerio de Ciencia y Tecnología* through a coordinated research project(grant DIP2003-0666-02-2) and Feder (ERDF).

References

1. Benhamou F. and Sais L. *Theoretical study of symmetries in propositional calculus and applications* in Proceedings of CADE92. (1992).
2. Benhamou F. and Older W. *Applying Interval Arithmetic to Real, Integer and Boolean Constraints.* In *The Journal of Logic Programming* pp. 1-24, (1997).
3. Collavizza H., Delobel F. and Rueher M. *Extending consistent domains of numeric CSP.* In *Proceedings of Sixteenth IJCAI'99*, Stockholm, pp. 406-411, (1999).

4. Crawford J., Ginsberg M., Luks E. and Roy A. *Symmetry-breaking Predicates for search problems.* In *Proc. of KR-96*, pp. 148-159, (1996).
5. Dague P. *Numeric Reasoning with relative orders of magnitude.* In *Proc. of the Thirteenth IJCAI*, Cambery, pp. 541-547, (1993).
6. Fox M. and Long D. *The Detection and Explotation of Symmetry in Planning Problems.* In *Proceedings IJCAI'99* pp. 956-961, (1999).
7. Gent I.P. and Smith B. M. *Symmetry Breaking During Search in Constraint Programming.* In *Report 99.02 University of Leeds*, (1999).
8. Gent I.P. and Smith B. M. *Symmetry Breaking in Constraint Programming.* In *Proc. ECAI 2000*, (2000).
9. Hyvönen E. *Constraint reasoning based on interval arithmetic: the tolerance propagation.* In *Artificial Intelligence* 58, pp. 1-112, (1992).
10. Jussien N. and Lhomme O. *Dynamic domain splitting for numeric CSPs.* In *Proceedings ECAI98*, pp. 224-228, (1998).
11. Lhomme O. *Contribution à la résolution de constraintes sur les réels par propagation d'intervalles.* Ph. D. Nice-Sophia University. Antipolis. (1994).
12. Mavovrouniotis M. L. and Stephanopoulos G. *Formal Order of Magnitude Reasoning in process engineering.* Comput. Chem Engineering 12(9-10), pp. 67-880, (1988).
13. Meseguer P. and Torras C. *Solving Strategies for Highly Symmetric CSPs.* In *Proceedings IJCAI'99* pp. 400-411, (1999).
14. Puget J. F. *On the satisfiability of symmetrical constrained satisfaction problems.* In *Proceedings of ISMIS'93*, pp. 350-361, (1993).
15. Puget J. F. *Symmetry Breaking using stabilizers.* In *Principles and Practice of Constraint Programming- CP 2003 LNCS 2833* pp. 585-589, (2003).
16. Van Hentenryck P., Michel L. and Deville Y. *Numerica. A modeling language for global optimization. The MIT Press*, (1997).