



HAL
open science

ICONIC: Interactive CONstraint-based Configuration

Élise Vareilles, Hélène Fargier, Michel Aldanondo, Paul Gaborit

► **To cite this version:**

Élise Vareilles, Hélène Fargier, Michel Aldanondo, Paul Gaborit. ICONIC: Interactive CONstraint-based Configuration. 19 th International Configuration Workshop, Sep 2017, Paris, France. p.28-32. hal-01618426

HAL Id: hal-01618426

<https://hal.science/hal-01618426v1>

Submitted on 17 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ICONIC

Interactive CONstraint-based Configuration

Élise Vareilles¹ and H el ene Fargier² and Michel Aldanondo¹ and Paul Gaborit¹

Abstract. Constraint satisfaction problems or CSP are very often used to formalize product configuration problems in both research and industry. CSP formalize relevant knowledge through variables, each one associated to a definition domain, linked by constraints, limiting the combinations of their permissible values. Thus, CSP makes it possible to describe exhaustively the solution space, corresponding to a set of all possible products. Two different methods of processing CSP allow to exploit the generic models in an interactive way: problem filtering methods (reasoning directly on the CSP network and removing inconsistent values) and solution filtering methods (reasoning on a representation of the solution space in the form of a compiled graph). Both of the methods have advantages and drawbacks in online product configuration. This paper aims at putting the first ideas on the joint use of these two methods in the same interactive configuration problem.

1 Introduction

Who has never wanted to own a particular product, such as shoes, smart-phone, cosmetic, car, etc., specially designed for him/her, perfectly suited to his/her desires, and affordable ? For several decades now, customers want to bring a personal touch to their products to make them special and unique. To meet this demand of personalization, companies nowadays no longer only offer standard products, but more and more personalizable ones. Thanks to the Web technologies, this personalization is done directly and interactively online. Customers can play with the range of choices and options offered by companies: they can assemble, cut, color, choose, ... visualize the result of their desires and ultimately order it, in a few minutes with a few clicks.

Enabling consumers or customers to personalize their products (glasses, shoes, computers, cars, etc.) is one of the current concerns of companies, whatever their size or activity sector. From the consumers' point of view, this customization has to be simple and fast (a few clicks on a web-page) while allowing them to obtain the product corresponding to their desires and their budget. From a business perspective, this customization is based on the definition of configurable products, represented by catalogs of predefined components and their relationships, as well as the implementation of interactive configuration systems or configurators.

Consequently, configuration systems have to cope with high combinatorial problems. To do so, they exploit a generic model [37] [34] [35] gathering knowledge about:

- customers requirements and desires on product definition,
- product components including their compatibility and / or incompatibilities (defining the generic bill-of-material for a product family),
- product production or manufacturing process.

Constraint satisfaction problems or CSP are very often used to formalize product configuration problems in both research and industry [7] [19]. CSP formalize relevant knowledge through variables, each one associated to a definition domain, linked by constraints, limiting the combinations of their permissible values [29]. Thus, CSP makes it possible to describe exhaustively the configuration problem and its solution space, corresponding to a set of all possible products. The users interact with the configuration system by progressively giving a value to or limiting the domain of the variables of their choice until all the variables have a unique value and the product is completely configured. The job of the configuration system is to guarantee that:

1. all choices are consistent with each other, at each step of the configuration process,
2. they can lead to the configuration of the desirable product and
3. the relevant indicators, such as price, delivery time and so on, are maintained up-to-date.

Two different methods of processing CSP allow to exploit the generic models in an interactive way:

- problem filtering methods which reason directly on the CSP network and remove inconsistent values. These methods use the constraints to make deductions on the problem by detecting locally inconsistent values. They guarantee interaction with the users but not the withdrawal of all values leading to non-solutions (they are therefore "incomplete").
- solution filtering methods which reason on a representation of the solution space in a form of a compiled graph. This compilation takes place off-line before the query phase, which relaxes the constraints on their temporal complexity but repels the difficulty in space: the compiled form can have an exponential theoretical spatial complexity.

Knowing this context, the aim of this article is therefore to give the first idea on a joint use of filtering and compilation methods in the same configuration problem to exploit the best of both approaches and mitigate their identified limits in the interactive product configuration problems.

The paper is divided as follows. In section 2, the motivation of our proposal, the constraint background and a simple illustrative example are presented. In sections 3 and 4, a focus is made respectively on problem filtering methods and solution filtering methods as well as

¹ Universit e de Toulouse, Mines d'Albi, Route de Teillet Campus Jarlard, 81013 Albi Cedex 09, France, email: firstname.lastname@mines-albi.fr

² Universit e de Toulouse, IRIT, avenue de l' tudiant, 31400 Toulouse, France, email: fargier@irit.fr

their their advantages and disadvantages in product configuration. In section 5, the main idea of the proposed hybrid method is exposed as well as some discussions.

2 Motivation, Background and Example

In interactive configuration problems, it is the user and not the machine that solves a combinatorial problem of optimizing preferences. Product configuration problems are ones of the typical examples (interactive configuration of a car, of a computer, etc.). By allowing the customers to explore the solution space, the online configuration system allows them to maximize their satisfaction.

Constraint satisfaction problems or CSP are very often used in product configuration problems, both in research and in industry [7] [19]. Many authors such as [40], [34], [27] have shown that configuration could be efficiently modeled and aided when considered as a CSP (Constraints Satisfaction Problem). A CSP is a triplet $\{\mathcal{X}, \mathcal{D}, \mathcal{C}\}$ where \mathcal{X} is a set of variables, \mathcal{D} is a set of finite domains (one for each variable) and \mathcal{C} a set of constraints linking the variables [29]. The variables can be either discrete or continuous. The constraints can either be of compatibility, when defining the possible or forbidden combinations of values for a set of variables (lists of compatible values, mathematical expressions), or of activity, when allowing the activation of a subset of variables and constraints [27] [20] [37].

The constraint-based modeling makes it possible to easily formalize the generic product family by a set of variables, each one associated to its definition domain and linked to the others by constraints that limit the combinations of allowed values. Constraints make it possible to describe exhaustively the solution space, i.e. the set of possible products. CSP have several advantages:

- a great freedom of knowledge modeling: compatibility between components, mathematical formulas for product evaluation, activation of optional components, etc.,
- non-orientation of reasoning: any variable present in the problem is both an input variable (which may be restricted by the user) or an output variable (resulting from a calculation, for example). It is therefore quite possible to constrain the price and then identify all the corresponding products,
- a clear separation between knowledge models and their exploitation (algorithmic processing),
- the possibility of combinations with other knowledge based approaches, and more particularly with case-based reasoning or CBR [26] [22] [1] [41], data-mining [18] [21] [2] and ontologies [38] [30] [39].

Constraint-based configuration systems allow to the users to browse the CSP solution space (the set of all possible products) by offering them the ability to:

- visualize the current solution being configured in a relevant way by presenting only the components, variants and options which are actually part of the solution,
- express preferences on components, variants, and options, such as selecting a single value, choosing a set of values, excluding a set of values, or expressing explicitly preferences explicitly between values,
- estimate the current solution being configured according to several criteria, sometimes antagonistic, such as its cost, performance or delivery time,
- express constraints on evaluation criteria, such as limiting the cost / performance / delivery time of the current solution, which limit

the choice of options and variants, or optimize solutions on one of these criteria [32] [33].

Technically, product configuration is an iterative process of removing solutions (products or components) from the solution space that are no longer consistent with the choices made by a user (typically, the potential customer) and the generic model. Through an iterative process, the user gradually specifies his/her needs and gradually converges towards a solution or a set of solutions satisfying his/her needs and desires.

Two concurrent methods of CSP processing allow us to reason on the generic model interactively: the methods reasoning on the problem described by a network of constrained variables (object of section 3), and those reasoning on a solution space represented as a compiled graph (object of section 4).

We illustrate our proposal on a very simple example of car configuration, coming from [4] and presented in Fig. 1.

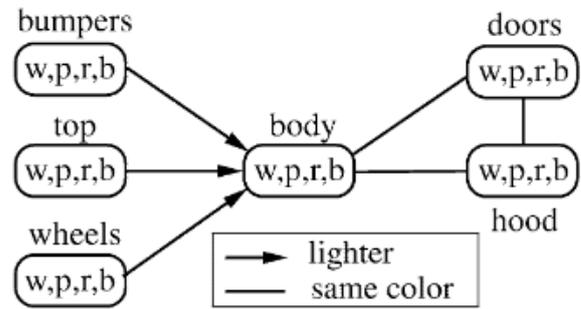


Figure 1. Car configuration problem [4]

This very simple example is composed of:

- six components: $\mathcal{X} = \{\text{bumpers}, \text{top}, \text{wheels}, \text{wheels}, \text{body}, \text{hood}, \text{doors}\}$ with all the variable sharing the same initial domain: $\mathcal{D} = \{\text{white}, \text{pink}, \text{red}, \text{black}\}$
- six constraints of \mathcal{C} :
 - $V(C_1) = \{\text{body}, \text{doors}\}$ with $\{(white, white), (pink, pink), (red, red), (black, black)\}$ as allowed combinations,
 - $V(C_2) = \{\text{hood}, \text{doors}\}$ with $\{(white, white), (pink, pink), (red, red), (black, black)\}$ as allowed combinations,
 - $V(C_3) = \{\text{body}, \text{hood}\}$ with $\{(white, white), (pink, pink), (red, red), (black, black)\}$ as allowed combinations,
 - $V(C_4) = \{\text{bumpers}, \text{body}\}$ with $\{(white, pink), (white, red), (white, black), (pink, red), (pink, black), (red, black)\}$ as allowed combinations,
 - $V(C_5) = \{\text{top}, \text{body}\}$ with $\{(white, pink), (white, red), (white, black), (pink, red), (pink, black), (red, black)\}$ as allowed combinations,
 - $V(C_6) = \{\text{wheels}, \text{body}\}$ with $\{(white, pink), (white, red), (white, black), (pink, red), (pink, black), (red, black)\}$ as allowed combinations.

3 Problem Filtering Methods

The problem filtering methods use the constraints locally to detect the values which are no more consistent with the current problem.

One of the most widely used methods is the one of arc consistency [29]. Arc consistency verifies that any value d of a domain \mathcal{D} of a variable v of \mathcal{V} is compatible with each constraint taken one by one. Dedicated filtering methods based on the arc consistency exist for each type of CSP: k consistency techniques (arc consistency and path consistency) used mainly for discrete or mixed CSP [24] [11] [13], arc continuous consistency [16], 2B-consistency [23] or Box-consistency [10] [9] for continuous CSP.

All the difficulty of the problem filtering methods lies in the fact that a perfect filtering is generally an NP-complete problem. Therefore, CSP filtering algorithms are limited to local, approximate, but polynomial reasoning. In interactive configuration, the use of problem filtering methods ensures the interactivity with users but does not guarantee the pruning of all values leading to non-solutions.

- **Advantages of problem filtering methods:**
Reasoning on any type of configuration problem (discrete, continuous or mixed with or optional variables, etc.).
- **Limitations of problem filtering methods:**
Probable conservation of not realizable solutions and use of the backtrack mechanism to restore coherence.

Let's have a look at the use of problem filtering methods on our simple example. The corresponding constraints network is presented in Fig. 2.

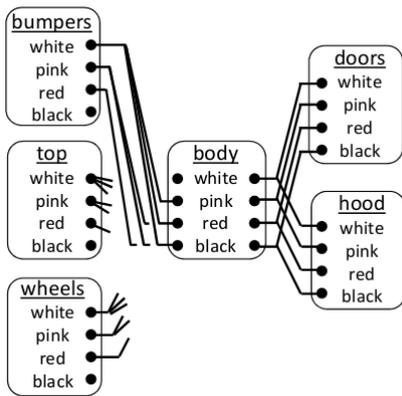


Figure 2. Car configuration problem Network

On such an example, there is no problem of filtering: all the inconsistent values are easily removed as the CSP is discrete and quite simple. But on much more complex configuration problem with discrete and continuous variables, the pruning of all values leading to non-solutions cannot be guaranty [12].

4 Solution Filtering Methods

The solution filtering methods are based on the transformation, by compilation, of a formalized configuration problem such as a CSP into a finite state automaton which therefore exhaustively represents the space of solutions [42]. This type of method has the advantage of avoiding subsequent backtracks after the compilation of the automaton and solves the filtering quality problem [5].

The compilation of a discrete problem into a finite state automaton is NP-complete, but it is done off-line, not online as it is for problem filtering methods. It has already proved its relevance and efficiency on actual industrial applications [5]. The online use of the compiled

automaton guarantees that the users' choices lead to a solution in linear time which is completely consistent with interactivity in configuration.

- **Advantages of solution filtering methods:**
Guarantee of achieving a solution without backtracks.
- **Limitations of solution filtering methods:**
Inability to efficiently compile certain types of problems (especially with continuous variables) and spatial explosion on problems with optional variables.

Let's have a look at the use of solution filtering methods on our simple example. The corresponding automaton is presented in Fig. 3.

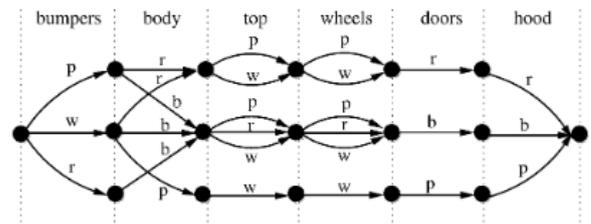


Figure 3. Car configuration problem Network

On such an example, there is no problem of compilation: the automaton is very quickly generated. But on much more complex configuration problem with discrete and continuous variables, it can be very difficult, and sometimes impossible, to build the finite state automaton.

5 Hybrid Proposed Method & Discussions

Product configuration is a topic that emerged some twenty years ago and which is developing significantly at the present time, driven by industrial applications and scientific results [7] [19].

With respect to constraint-based configuration problems, problem filtering methods [29] [12] have already proved their worth for discrete problems (core of many configuration systems, including those of ILOG, Pros); The problem filtering methods have been recently enriched (global inverse consistency [12], filtering and alternative values [8]). Nowadays, the main difficulty is to efficiently integrate continuous variables without discretizing their domain (essentially filtering by bound-consistency [23] [9] [16]) and secondly, to effectively take into account optional components without adding a specific value in domains [3] [43] [25].

The solution filtering methods [5] have also been extended by taking into account price information, which the problem filtering methods have difficulty to handle [6], by developing particularly efficient data structures [15], and finally by the definition of methods of learning user's preferences on compiled structures [14]). Some works have extended the solution filtering methods to continuous or mixed CSP [28] [31] [17], but rather on scheduling problems than on product configuration ones.

The objective of the article is to discuss on the design, development and testing of an interactive configuration algorithm that combines the problem filtering and solution filtering methods. The development and testing of this new hybrid filtering method would allow the best of both approaches to be exploited by mitigating their limits identified in interactive product configuration problems.

Scientifically, the first idea is to compile the sub-components of the product family and to use problem filtering algorithms to propagate users' choices from one automaton to another. This first joint use could avoid the pitfall of the spatial explosion due to optional components. The second working line deals with a proposed data structures more suited to continuous variables than the conventional automatons and exceeding the bound consistency filtering approaches. Rather than adapting the structures designed for discrete variables to continuous ones (which is possible but ineffective by discretizing the continuous domains), the approaches resulting from the work on continuous domains could be used, such as Q-trees and R-trees [36].

Compared to existing work, the proposed hybrid method is therefore both logical and innovative: how to combine problem filtering and solution filtering methods together in the same configuration problem. The future results seem very promising for interactive configuration problems.

A PhD subject is actually waiting for a good candidate and future PhD student. The PhD thesis is conducted between two teams: the **ORKID research team** of the Industrial Engineering Lab of Mines Albi France and the **ADRIA research team** of IRIT Toulouse France. All the proposals will be validated on several industrial cases mainly from the automotive sector.

REFERENCES

- [1] A. Aamodt and E. Plaza, 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AI Commun.*, **7**(1), 39–59, (March 1994).
- [2] R. Agrawal, T. Imieliński, and A. Swami, 'Mining association rules between sets of items in large databases', *SIGMOD Rec.*, **22**(2), 207–216, (June 1993).
- [3] J. Amilhastre, *Représentation par automate d'ensemble de solutions de problèmes de satisfaction de contraintes*, Ph.D. dissertation, Université de Montpellier, 1999.
- [4] J. Amilhastre, H. Fargier, and P. Marquis, 'Consistency restoration and explanations in dynamic CSPs—Application to configuration', *Artificial Intelligence*, **135**(1-2), 199–234, (février 2002). bb modif 28/02/02.
- [5] J. Amilhastre, H. Fargier, and P. Marquis, 'Consistency restoration and explanations in dynamic cps application to configuration', *Artificial Intelligence*, **135**(1 2), 199 – 234, (2002).
- [6] H.R. Andersen, T. Hadzic, and D. Pisinger, 'Interactive cost configuration over decision diagrams', *CoRR*, **abs/1401.3830**, (2014).
- [7] J.M. Astesana, L. Cosserat, and H. Fargier, 'Constraint-based vehicle configuration : a case study (regular paper)', in *International Conference on Tools with Artificial Intelligence (ICTAI)*, Arras, 27/10/2010-29/10/2010, pp. 0–1, <http://www.computer.org>, (octobre 2010). IEEE Computer Society.
- [8] C. Becker and H. Fargier, 'Maintaining alternative values in constraint-based configuration', in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 454–460, (2013).
- [9] F. Benhamou, 'Heterogeneous constraint programming', in *5th international conference on algebraic and logic programming*, ed., Springer Verlag, pp. 62–76, Aachen, Allemagne, (Mars 1996).
- [10] F. Benhamou, D. Mc Allester, and P. Van Hentenryck, 'Clip(intervals) revisited', in *ILPS'94*, pp. 1–21, (1994).
- [11] C. Bessière and M. Cordier, 'Arc-consistency and arc-consistency again', in *AAAI*, pp. 108–113, Cambridge MA, (1993).
- [12] C. Bessière, H. Fargier, and C. Lecoutre, 'Computing and restoring global inverse consistency in interactive constraint satisfaction', *Artificial Intelligence*, **vol. 241**, pp. 153–169, (September).
- [13] C. Bessière and J. Régim, 'An arc-consistency algorithm optimal in the number of constraint checks', in *ECAI Workshop on Constraint Processing*, (1994).
- [14] A. Choi, G. Van den Broeck, and A. Darwiche, 'Tractable learning for structured probability spaces: A case study in learning preference distributions.', in *IJCAI*, volume 15, pp. 2861–2868, (2015).
- [15] A. Darwiche, 'Sdd: A new canonical representation of propositional knowledge bases.', in *IJCAI*, ed., Toby Walsh, pp. 819–826. IJ-CAI/AAAI, (2011).
- [16] B. Faltings, 'Arc consistency for continuous variables', in *Artificial Intelligence*, volume 65, pp. 363–376, (1994).
- [17] H. Fargier, F. Maris, and V. Roger, 'Temporal Constraint Satisfaction Problems and Difference Decision Diagrams: A Compilation Map. (regular paper)', in *IEEE International Conference on Tools with Artificial Intelligence, Vietry sul mare, Italie, 09/11/2015-11/11/2015*, pp. 429–436, <http://www.ieee.org/>, (novembre 2015). IEEE.
- [18] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
- [19] A. Felber, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edn., 2014.
- [20] E. Gelle and R. Weigel, 'Interactive configuration based on incremental constraint satisfaction', in *IFIP*, pp. 117–126, (1995).
- [21] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edn., 2011.
- [22] J. Kolodner, *Case-based Reasoning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [23] O. Lhomme, 'Consistency techniques for numeric CSP', in *International Joint Conference on Artificial Intelligence*, pp. 232–238, Chambéry, France, (8 1993).
- [24] A.K. Mackworth, 'Consistency in networks of relations', in *Artificial Intelligence*, volume 8(1), pp. 99–118, (1977).
- [25] K. McDonald and P. Prosser, 'A case study of constraint programming for configuration problems', Technical report, APES Research Group, (2002). APES-45-2002.
- [26] M. Minsky, 'A framework for representing knowledge', Technical report, Cambridge, MA, USA, (1974).
- [27] S. Mittal and B. Falkenhainer, 'Dynamic constraint satisfaction problems', in *AAAI*, pp. 25–32, Boston, US, (1990).
- [28] J.B. Møller, Jakob Lichtenberg, H.R. Andersen, and H. Hulgaard, 'Difference decision diagrams', in *Computer Science Logic, 13th International Workshop, CSL '99, 8th Annual Conference of the EACSL, Madrid, Spain, September 20-25, 1999, Proceedings*, pp. 111–125, (1999).
- [29] U. Montanari, 'Networks of constraints: fundamental properties and application to picture processing', in *Information sciences*, volume 7, pp. 95–132, (1974).
- [30] B. Neumann, 'Configuration expert systems: a case study and tutorial', in *SGAICO Conference on Artificial Intelligence in Manufacturing, Assembly and Robotics*, pp. 27–68, Munich, Germany, (1988).
- [31] A. Niveau, H. Fargier, C. Pralet, and G. Verfaillie, 'Knowledge Compilation Using Interval Automata and Applications to Planning (regular paper)', in *European Conference on Artificial Intelligence (ECAI), Lisboa, 16/08/2010-20/08/2010*, pp. 459–464, <http://www.iospress.nl/>, (aot 2010). IOS Press.
- [32] P. Pitiot, M. Aldanondo, and E. Vareilles, 'Concurrent product configuration and process planning: Some optimization experimental results', *Computers in Industry*, **65**(4), 610–621, (2014). WoS*.
- [33] P. Pitiot, M. Aldanondo, E. Vareilles, P. Gaborit, M. Djefel, and S. Carboneel, 'Concurrent product configuration and process planning, towards an approach combining interactivity and optimality', *International Journal of Production Research*, **51**(2), 524–541, (2013). WoS*.
- [34] D. Sabin and E.C. Freuder, 'Configuration as composite constraint satisfaction', in *Artificial Intelligence and Manufacturing Research Planning Workshop*, pp. 153–161, (1996).
- [35] D. Sabin and R. Weigel, 'Product configuration frameworks - a survey', *IEEE Intelligent System and their Applications*, (1998).
- [36] J. Sam-Haroud, *Constraint consistency techniques for continuous domains*, Ph.D. dissertation, Ecole Polytechnique Fdrale de Lausanne, 1995.
- [37] T. Soinen and E. Gelle, 'Dynamic constraint satisfaction in configuration', in *American Association for Artificial Intelligence, Workshop on Configuration*, Orlando, US, (1999).
- [38] T. Soinen, J. Tiihonen, T. Mannisto, and R. Sulonene, 'Towards a general ontology of configuration', *AI EDAM-ARTIFICIAL INTELLIGENCE FOR ENGINEERING DESIGN ANALYSIS AND MANUFACTURING*, **12**(4), 357–372, (1998).
- [39] S. Staab and R. Studer, *Handbook on Ontologies*, Springer Publishing

Company, Incorporated, 2nd edn., 2009.

- [40] E. Tsang, 'Foundations of constraints satisfaction', in *Academic Press*, London, (1993).
- [41] E. Vareilles, M. Aldanondo, A. Codet De Boisse, T. Coudert, P. Gaborit, and L. Geneste, 'How to take into account general and contextual knowledge for interactive aiding design: Towards the coupling of CSP and CBR approaches', *Engineering Applications of Artificial Intelligence*, vol. **25**, pp. 31 – 47, (2012).
- [42] N.R. Vempaty, 'Solving constraint satisfaction problems using finite state automata', in *Swartout*, pp. 453–458, (1992).
- [43] M. Veron, *Modélisation et résolution du problème de configuration industrielle : utilisation des techniques de satisfaction de contraintes*, Thèse de doctorat, Institut National Polytechnique de Toulouse, France, 2001.