



HAL
open science

CoFiADe Constraints Filtering for Aiding Design

Élise Vareilles, Paul Gaborit, Michel Aldanondo, Sabine Carbonnel, Laurent Steffan

► **To cite this version:**

Élise Vareilles, Paul Gaborit, Michel Aldanondo, Sabine Carbonnel, Laurent Steffan. CoFiADe Constraints Filtering for Aiding Design. JFPC 2012, May 2012, Toulouse, France. paper9. hal-00811512

HAL Id: hal-00811512

<https://hal.science/hal-00811512>

Submitted on 26 Apr 2013

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.

CoFiADe

Constraints Filtering for Aiding Design

Élise Vareilles, Paul Gaborit,
Michel Aldanondo, Sabine Carbonnel, Laurent Steffan

Université de Toulouse - École des Mines d'Albi-Carmaux
Route de Teillet - Campus Jarlard
81000 Albi Cedex 09
{prenom.nom}@mines-albi.fr

Résumé

Les systèmes interactifs d'aide à la conception et à la configuration sont destinés à faciliter les prises de décision en se basant sur des connaissances pertinentes et de différentes natures. En exploitant les *CSP*, ils permettent d'accompagner efficacement cette prise de décision, tout en garantissant leur interactivité, par l'utilisation d'algorithmes de filtrage. *CoFiADe* est un outil interactif dédié à l'aide à la décision en conception ou configuration. Il permet d'une part, aux experts de formaliser les connaissances du problème sous la forme d'un *CSP* via un langage de définition spécifique, et d'autre part, aux utilisateurs de piloter une interface Web, leur permettant via une vision arborescente du modèle, de réduire progressivement le domaines des variables pour converger vers une solution.

Pour cela, *CoFiADe* supporte la définition de variables de différentes natures (discrètes et continues) ainsi que des contraintes de différentes natures (compatibilité, activation) et exploite plusieurs méthodes de filtrage. *CoFiADe* est développé et maintenu en Perl à l'école des mines d'Albi-Carmaux et a d'ores et déjà servi de support à plusieurs applications industrielles. Dans un premier temps, nous exposerons les spécificités (mixité, multi-intervalles, activation) de *CoFiADe*, puis nous illustrerons ses capacités sur un exemple simple d'aide à la décision tiré d'une application industrielle et conclurons sur les améliorations à apporter afin de promouvoir son utilisation.

Abstract

Interactive aiding design or configuration tools support and help decisions making by using relevant knowledge of various origins. Product design and configuration can be efficiently modeled and aided when considered as a *CSP*. Their interactivity can then be guaranteed

through the use of filtering algorithms. *CoFiADe* is an interactive aiding design or configuration tool. It makes it possible, on the first hand for experts to model their knowledge as a *CSP* via a specific language and, on the other hand, for users to reach a solution by progressively reducing the domain of the variables through a Web interface.

In order to achieve this goal, *CoFiADe* supports the definition of various kinds of variables (discrete, continuous), as well as various types of constraints (compatibility, activation) and it uses various filtering algorithms. *CoFiADe* has been developed in Perl and is kept up to date by people at école des mines d'Albi-Carmaux (France). Note that *CoFiADe* has already been used in several industrial projects in order to help users make better decisions. In the second section, the general features of *CoFiADe* (mixed problem, multi-interval, activity) are presented. In the third section, its abilities in term of aiding design are illustrated with a simple but real example. In the last section, we focus on the improvements that can be achieved in order to improve *CoFiADe* and promote its utilization.

1 Introduction

Les systèmes interactifs d'aide à la conception et à la configuration sont destinés à faciliter les prises de décision en se basant sur des connaissances pertinentes [1] et de différentes natures [2]. En exploitant les *CSP*, ils permettent d'accompagner efficacement cette prise de décision [3] [4], tout en garantissant leur interactivité, par l'utilisation d'algorithmes de filtrage. De nombreux travaux en aide à la conception dans différents domaines ont utilisé les approches

par contraintes : en conception d'appareils à pression [5], en conception aéronautique [6], en conception de produits à forte diversité [7], en conception de batteries électriques [8] ou en conception de matériaux multi-couches [9]. À travers cet échantillon d'applications, nous pouvons percevoir la diversité des connaissances à modéliser afin d'aider à la décision en conception ou configuration : connaissances sur les propriétés et fonctionnalités du produit (caractéristiques du produit, discrètes ou numériques comme la consommation ou le poids), sur les composants du produit (liste de composants à intégrer, majoritairement discret) et sur les options du produits (composants à concevoir ou configurer sous certaines conditions) [10].

CoFiADe est un outil interactif dédié à l'aide à la décision en conception et à la configuration. Il permet d'une part, aux experts de formaliser les connaissances du problème sous forme d'un *CSP* contenant des variables de différents types (discrètes et numériques) et des contraintes de différentes natures (compatibilité et activation) et de différents types (tables de compatibilité et expressions mathématiques). Pour cela, *CoFiADe* possède son propre langage de définition de modèle *CSP*. Il permet d'autre part, aux utilisateurs de piloter une interface Web, <http://cofiade.enstimac.fr/cgi-bin/cofiade.pl>, leur permettant, via une vision arborescente du modèle, de réduire progressivement le domaines des variables pour converger vers une solution. *CoFiADe*, écrit en Perl, est développé et maintenu à l'école des mines d'Albi-Carmaux. Il est issu de recherches menées au sein de notre laboratoire [2], [1], [10] et [11]. Il a servi de support au projet européen *VHT*¹, au projet ANR *ATLAS*² et au projet FUI *Helimaintenance*³.

Dans le cadre de cette communication, nous exposons dans un premier temps, les spécificités de *CoFiADe* (mixité, multi-intervalles, activation) et précisons les méthodes de filtrage utilisées. Dans un deuxième temps, nous illustrons ses capacités sur un exemple simple d'aide à la décision tiré d'une application industrielle [12]. Puis, nous concluons sur les suites à donner afin d'améliorer et promouvoir son utilisation.

2 Spécificités de *CoFiADe*

CoFiADe présente certaines spécificités que n'ont pas les autres outils basés sur les *CSP*, tels qu'*Ilog CP* [13] (<http://www.ilog.com/products/cp/>), *Choco*

[14] (<http://www.emn.fr/z-info/choco-solver/>) ou *ECLiPSe* [15] (<http://87.230.22.228/>). Dans cette section, nous abordons ses spécificités du point de vue du modèle (variables et contraintes de différents types et natures) et du point de vue traitement (intégration de plusieurs méthodes de filtrage de la littérature).

2.1 Différents types de variables

CoFiADe, comme *Ilog CP*, *Choco* et *ECLiPSe*, permet de modéliser plusieurs types de variables mais sans inclusion de bibliothèques spécifiques :

- les variables discrètes dont les domaines de définition sont finis ou dénombrables. Ceux-ci peuvent contenir un ensemble d'éléments soit de type symbolique, soit de type numérique (intervalles de valeurs entières, liste de valeurs entières ou réelles),
- les variables continues dont les domaines de définition sont indénombrables. Ceux-ci contiennent des éléments de type numérique (intervalles de valeurs réelles).

L'une des spécificités de *CoFiADe* est de pouvoir associer aux variables des domaines continus multi-intervalles [2]. Dans ce cas, les domaines ne sont pas constitués d'un unique intervalle de valeurs mais de plusieurs intervalles de valeurs disjoints.

2.2 Natures et types de contraintes

CoFiADe, comme *Ilog CP*, *Choco* et *ECLiPSe*, permet de modéliser plusieurs natures et types de contraintes. Nous distinguons deux natures de contraintes, celles de compatibilité et d'activation et deux types de contraintes, les tables de compatibilité et les expressions mathématiques. La figure ?? synthétise les notions de nature et type de contraintes gérées par *CoFiADe* [2].

Contraintes de différentes natures : Les contraintes de compatibilité permettent de restreindre l'espace de recherche en délimitant les combinaisons de valeurs que les variables peuvent prendre simultanément [16]. Elles peuvent être exprimées en *extension* ou en *intension*. Les contraintes de compatibilité exprimées en extension sont représentées par des listes de n -uplets indiquant quelles sont les valeurs compatibles entre elles. Les contraintes de compatibilité exprimées en intension concentrent les informations pour plusieurs raisons. Tout d'abord, il peut être plus facile d'écrire la contrainte en intension [17] (pour éviter de lister toutes les possibilités autorisées) ou parce qu'il peut être tout à fait impossible de lister de manière complète les combinaisons de valeurs [18].

Les contraintes d'activation permettent d'activer (inclure) ou de supprimer (exclure) des variables du

1. Virtual Heat Treatment, projet n° G1RD-CT-2002-00835
 2. Aides et assistances pour la conception, la conduite et leur couplage par les connaissances, projet n° ANR-07-TLOG-002
 3. projet pôle de compétitivité Aerospace Valley financé par FUI-DGE

problème courant [19]. C'est la détection d'événements singuliers qui déclenche l'ajout ou l'exclusion d'un ensemble d'éléments. Les contraintes d'activation sont définies par deux parties : une prémisse (contrainte de compatibilité) et un conséquent (ajout ou retrait d'éléments) liées par un opérateur d'implication \rightarrow . Si la prémisse est vraie, il y a application du conséquent. Sinon, le problème reste inchangé.

CoFiADe permet de définir et traiter dans un même problème, des contraintes de compatibilité décrites en extension ou en intension. En ce qui concerne les contraintes d'activation, il intègre une extension du concept d'activation aux contraintes et aux sous-problèmes, tels qu'introduit par [3]. Les contraintes d'activation manipulées par *CoFiADe* autorisent explicitement l'ajout d'ensembles de variables, de contraintes et de sous-problèmes selon le modèle des *RequireVariable* de Mittal [19]. Les variables, les contraintes et les groupes possèdent donc un état *actif* ou *inactif* indiquant leur participation au problème courant. Dans *CoFiADe*, de manière identique aux *DCSP*, les variables *inactives* ne sont pas considérées dans le problème courant. Par contre, les contraintes *actives* sont toujours prises en compte et filtrées quelque soit l'état (*actif* ou *inactif*) des variables liées par les contraintes alors que les contraintes *inactives* ne sont jamais prises en compte et filtrées quelque soit l'état (*actif* ou *inactif*) des variables liées par celle-ci.

Contraintes de différents types : Les tables de compatibilité permettent de représenter sous forme tabulaire des listes de n -uplets de valeurs autorisées. Chaque combinaison de n -uplet est mise sous forme de tuple autorisé. La combinatoire exprimée sous forme de table de compatibilité correspond à un sous-ensemble du produit cartésien des domaines des variables participant à la contrainte. Les tables de compatibilité peuvent être soit discrètes, soit continues, soit mixtes si elles lient à la fois des variables discrètes et continues [12].

Les expressions mathématiques sont essentielles pour calculer des paramètres de conception ou pour représenter des lois physiques et des résultats d'expérimentations. Certaines relations peuvent se mettre sous la forme d'une fonction numérique de \mathbb{R}^n dans \mathbb{R} où n équivaut au moins à l'arité de la contrainte associée. Mais il est aussi courant, en conception, de prendre en compte des résultats expérimentaux sous forme d'abaques [6] [2]. L'intégration de ce type de connaissances dans un *CSP* peut se faire à l'aide d'approximation par des fonctions numériques continues et définies par morceaux [25].

CoFiADe permet la définition et le traitement de tables de compatibilité discrètes, continues et mixtes. Il autorise aussi la définition d'expressions mathématiques

telles que les fonctions numériques et les contraintes par morceaux.

2.3 Moteur de filtrage

CoFiADe est un outil interactif d'aide à la décision en conception et à la configuration. Contrairement à *Ilog CP*, *Choco* et *ECLiPSe*, *Cofiade* n'intègre que des méthodes de filtrage : il ne fait pas de résolution de problème. Le moteur de propagation de *CoFiADe*, présenté par l'algorithme 1, permet de répercuter sur l'ensemble des variables du modèle, *via* les contraintes, chaque choix de conception. Chaque choix correspond à une réduction du domaine d'une variable fournie par l'utilisateur. Tout d'abord, grâce aux contraintes d'activation, nous incluons dans le modèle courant les variables, les contraintes et les groupes qui ne l'étaient pas auparavant. La fonction *CA* retourne l'ensemble des variables nouvellement actives qui sont ajoutées à l'ensemble des variables réduites sur lesquelles le filtrage a lieu.

Alg. 1 MOTEUR(CSP : P)

```

- .: - Cet algorithme présente le fonctionnement général du moteur de propagation.
- .: -  $P = (\mathbb{V}, \mathbb{D}, \mathbb{C}_C, \mathbb{C}_A)$ 
- .: - Varfilt : ensemble des variables à considérer pour filtrer les contraintes
Varfilt  $\leftarrow \emptyset$ 
- .: - Varval : ensemble des variables actives nouvellement évaluées (domaine réduit à un singleton)
Varval  $\leftarrow \emptyset$ 
Répéter
| - .: - Présentation de l'ensemble des variables actives à valuer à l'utilisateur
| - .: - Réduction d'une variable v par l'utilisateur D'_v
|   Dv  $\leftarrow D_v \cap D'_v$ 
|   Si (Dv est réduit à un singleton) Alors
|     | - .: - v est nouvellement évalué
|     |   Varval  $\leftarrow v$ 
|   Fin Si
|   - .: - La variable doit être considérée pour le filtrage
|   Varfilt  $\leftarrow v$ 
|   Répéter
|     | - .: - Propagation sur les contraintes d'activation actives pour inclure les nouveaux éléments
|     | - .: - Prise en compte dans le problème courant des variables nouvellement activées par ajout de celles-ci à l'ensemble Varfilt
|     |   Varfilt  $\leftarrow (Var_{filt} \cup CA(Var_{val}))$ 
|     | - .: - Filtrage sur toutes les contraintes de compatibilité actives pour toutes les variables réduites
|     |   Varval  $\leftarrow CC(Var_{filt})$ 
|   Jusqu'à (variables actives réduites  $\neq \emptyset$ )
Jusqu'à (il ne reste plus de variables actives à valuer)

```

Puis, différentes méthodes de filtrage sont appliquées suivant les types de contraintes de compatibilité. *CoFiADE* intègre donc plusieurs méthodes de filtrage de faible degré :

- un algorithme de filtrage par arc-cohérence pour les tables de compatibilité discrètes [20], continues [2] et mixtes [21],
- un algorithme de filtrage par $2B$ -cohérence [22], basé sur l’arithmétique des intervalles [23] pour les fonctions numériques et étendu aux multi-intervalles [2],
- un algorithme de filtrage par arbre quaternaire [24] et arbre quaternaire étendu pour les contraintes par morceaux [25].

La fonction CC retourne l’ensemble des variables actives nouvellement réduites.

3 Exemple illustratif

Nous illustrons dans cet exemple tiré de [12] et [1], différentes spécificités de *CoFiADE* : les variables continues avec des domaines de définition multi-intervalles, les tables de compatibilités mixtes et l’activation de variables et contraintes. Puis, nous déroulerons deux scénarii de test illustrant les possibilités de filtrage du moteur de *CoFiADE*. Le modèle présenté ici est disponible à l’url suivante <http://cofiade.enstimac.fr/cgi-bin/cofiade.pl> en choisissant le modèle COFIADÉ.

3.1 Modèle

Notre exemple illustratif correspond à la configuration et au dimensionnement d’une cuve de mixeur. Il comporte

- quatre variables :
 - une variable discrète *Géométrie_Cuve* caractérisant la géométrie de la cuve du mixeur, toujours présente dans le problème :
variable symbolic Vessel in {"cylindric", "cubic", "spheric"};
 - trois variables continues :
 - la variable *Volume* caractérisant le volume de la cuve, toujours présente dans le problème et ayant un domaine multi-intervalle :
variable float Volume in {[0, 27], [28, 150]};
 - la variable *Hauteur* caractérisant la hauteur de la cuve, toujours présente dans le problème :
variable float Hauteur in {[1, 3]};
 - la variable *Rayon*, présente uniquement si la cuve est cylindrique et donc *inactive* au début du problème :
inactive variable float Rayon in {[3, 4]};

- une contrainte de compatibilité mixte *ccm* de type table liant la géométrie de la cuve à son volume, toujours présente dans le problème :
constraint ccm using (Geometrie_cuve, Volume)
table {
{"cylindric", [28, 150]},
{"spheric", [0.6, 15]},
{"cubic", [1, 27]},};
- trois contraintes de compatibilité continue de type fonction numérique liant le volume, la hauteur et pour l’une d’entre elles, le rayon. Ces contraintes sont inactives au début du problème de configuration et ne sont donc pas filtrées :
 - *freeze constraint ccc1 using (Hauteur, Volume) condition {Volume = Hauteur * Hauteur * Hauteur * 4 / 3 * 3.14 / 8};*
 - *freeze constraint ccc2 using (Rayon, Hauteur, Volume) condition {Volume = Rayon * Rayon * 3.14 * Hauteur};*
 - *freeze constraint ccc3 using (Hauteur, Volume) condition {Volume = Hauteur * Hauteur * Hauteur};*
- trois contraintes d’activation de type table liant la géométrie de la cuve à la fonction de calcul de son volume et pour l’une d’entre elles, au rayon :
 - *constraint cca1 using (Geometrie_cuve) activate (ccc1) table {"spheric"};*
 - *constraint cca2 using (Geometrie_cuve) activate (ccc2, Rayon) table {"cylindric"};*
 - *constraint cca3 using (Geometrie_cuve) activate (ccc3) table {"cubic"};*

3.2 Scénarii de test

L’aide à la décision interactive consiste à valuer ou réduire progressivement le domaine des variables. Lorsqu’une variable v est réduite, la modification de son domaine est propagée aux autres variables via les contraintes. Les valeurs qui ne sont plus compatibles avec le nouveau domaine de v sont alors retirées du domaine des variables. Les contraintes n’étant pas orientées, l’utilisateur peut soit commencer par donner la géométrie de la cuve et en déduire le volume, soit fixer un volume maximal pour en déduire une géométrie de cuve. Nous illustrons ici ces deux modes de fonctionnement.

Déduction du volume Dans le premier scénario, l’utilisateur va tout d’abord fixer la géométrie de sa cuve à {cylindric}. Ce choix a pour conséquence :

- l’activation de la variable *Rayon* qui est ajoutée au problème courant,
- l’activation de la contrainte de calcul du volume *ccc2* liant le volume, la hauteur et le rayon,
- le filtrage de la contrainte *ccm* par arc-cohérence réduisant le domaine de la variable *Volume* à {[28,

150}},

- le filtrage de la contrainte *ccc2* par 2B-cohérence réduisant le domaine de la variable *Volume* à {[28.26, 150]}.

Si l'utilisateur limite la hauteur de cuve à 1.5, *Hauteur* < 1.5, le volume de la cuve est réduit via la contrainte *ccc2* à {[28.26, 75.36]}. S'il fixe maintenant le rayon de sa cuve à 3.75, le volume possible est réduit à {[44.15, 66.23]}. Pour une hauteur de 1.25, le volume final de la cuve est de 55.19.

Déduction de la géométrie Dans le second scénario, l'utilisateur va tout d'abord limiter son volume de cuve à {> 30}. Ce choix a pour conséquence :

- la valuation de la variable *Géométrie_cuve* à {cylindric} via la contrainte *ccm*,
- l'activation de la variable *Rayon* qui est ajoutée au problème courant,
- l'activation de la contrainte de calcul du volume *ccc2* liant le volume, la hauteur et le rayon,
- le filtrage de la contrainte *ccc2* par 2B-cohérence réduisant le domaine de la variable *Volume* à {[30, 150]}.

Si l'utilisateur limite la hauteur de cuve à 1.25, *Hauteur* < 1.5, le volume de la cuve est resté inchangé et égal à {[30, 75.36]}. S'il fixe maintenant le rayon de sa cuve à 3.75, le volume possible est réduit à {[44.15, 66.23]}. Pour une hauteur de 1.25, le volume final de la cuve est de 55.19.

4 Intérêts et améliorations prochaines

CoFiADe a servi de support au projet européen *VHT*⁴, au projet ANR *ATLAS*⁵ et au projet FUI *Helimaintenance*⁶ et a donné entière satisfaction en termes :

- de couverture des besoins de modélisation de problèmes de conception (définition de domaines multi-intervalles, de contraintes mixtes et par morceaux, de contraintes d'activation de variables, contraintes et sous-problèmes),
- de facilité de modélisation du problème sous forme de *CSP*,
- de propagation des choix utilisateur (rapidité de traitement, réduction des domaines et de l'espace de solutions).

Grâce à l'interfaçage avec un outil d'optimisation évolutionnaire [26], *CoFiADe* intègre maintenant la possibilité d'optimiser une pré-solution après avoir renseigné un sous-ensemble de variables non négociables

4. Virtual Heat Treatment, projet n° G1RD-CT-2002-00835

5. Aides et assistances pour la conception, la conduite et leur couplage par les connaissances, projet n° ANR-07-TLOG-002

6. projet pôle de compétitivité Aerospace Valley financé par FUI-DGE

et fixé des seuils sur certains critères (tels que le coût ou le délai de fabrication). Cette fonctionnalité récente n'est pas encore accessible en ligne.

À court terme, *CoFiADe* va offrir la possibilité de saisir et mettre en ligne de modèles *CSP*, via une interface Web de saisie de modèles, avec accès limité par mot de passe. *CoFiADe*, manipulant des variables continues, est confronté à des problèmes de stabilité numérique. L'implémentation d'un système de « pas numérique » permettant d'arrêter les itérations lorsque la différence entre les valeurs d'entrée et les valeurs de filtrage sont inférieures à un certain seuil sera bientôt réalisée [27]. Une fonctionnalité de conseil est actuellement en cours de définition et développement dans le cadre des travaux de thèse d'A. Codet de Boisse [28]. Celle-ci permettra de conseiller l'utilisateur sur ses choix à partir d'une base de cas stockant les configurations passées et de lui indiquer les valeurs les plus souvent choisies au vue de la solution courante.

À plus long terme, *CoFiADe* intégrera des contraintes globales telle que *AllDif* [29] [30] et des quantificateurs \exists et \forall afin d'offrir de plus large possibilité de modélisation. Afin d'améliorer le filtrage de *CoFiADe* sur certaines parties spécifiques du problème, son couplage avec des moteurs externes est aussi envisagé. Enfin, afin d'offrir de nouvelles fonctionnalités à l'utilisateur, *CoFiADe* intégrera la notion de préférence [31].

Références

- [1] T. van Oudenhove de Saint-Géry. Contribution à l'élaboration d'un formalisme gérant la pertinence pour les problèmes d'aide à la conception à base de contraintes. *Mémoire de doctorat*, Institut National Polytechnique de Toulouse, 2006.
- [2] É. Vareilles. Conception et approches par contraintes : contribution à la mise en oeuvre d'un outil d'aide interactif. *Mémoire de doctorat*, Institut National Polytechnique de Toulouse, 2005.
- [3] D. Sabin and E.C. Freuder. Configuration as Composite Constraint Satisfaction, *Artificial Intelligence and Manufacturing Research Planning Workshop*, pp. 153-161, 1996.
- [4] T. Soininen, T. Tiihonen, T. Männistö and Sulonen R. Towards a General Ontology of Configuration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 12, n°4, pp. 357-372, 1998.
- [5] X. Fischer. Stratégie de conduite du calcul pour l'aide à la décision en conception mécanique intégrée : application aux appareils à pression *Thèse*

de doctorat, École Nationale Supérieure d'Arts et Métiers 2000.

- [6] T. Mulyanto. Utilisation des techniques de programmation par contraintes pour la conception d'avions *Thèse de doctorat* Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, 2002.
- [7] K. Hadj-Hamou. Contribution à la conception de produits à forte diversité et de leur chaîne logistique : une approche par contraintes *Thèse de doctorat*, Institut National Polytechnique de Toulouse, 2002.
- [8] Y. Vernat. Formalisation et qualification de modèles par contraintes en conception préliminaire *Thèse de doctorat*, École Nationale des Arts et Métier, Bordeaux, 2004.
- [9] S. Giaccobi. Méthode de conception de multimatériaux à architecture multicouche : application à la conception d'une canalisation sous-marine *Thèse de doctorat*, Université de Bordeaux, 2009.
- [10] M. Djefel. Couplage de la configuration de produit et de projet de réalisation : exploitation des approches par contraintes et des algorithmes évolutionnaires *Thèse de doctorat*, Institut National Polytechnique de Toulouse, 2011.
- [11] J. Abeille. Vers un couplage des processus de conception de systèmes et de planification de projets : formalisation de connaissances méthodologiques et de connaissances métier *Thèse de doctorat*, Institut National Polytechnique de Toulouse, 2011.
- [12] E. Gelle. On the generation of locally consistent solution spaces in mixed dynamic constraint problems *Thèse de doctorat* École Polytechnique Fédérale de Lausanne, Suisse, 1998.
- [13] J-F. Puget. A C++ Implementation of CLP, *Proceedings of SPICIS 94*, November 1994, Singapore.
- [14] CHOCO Team. Choco : an Open Source Java Constraint Programming Library, *Research report*, Ecole des Mines de Nantes, <http://www.emn.fr/z-info/choco-solver/pdf/choco-presentation.pdf>, 2010.
- [15] M. Meier and J. Schimpf, *An Architecture for Prolog Extensions*, Proceedings of the 3rd International Workshop on Extensions of Logic Programming, Bologna, 1992.
- [16] U. Montanari. Networks of constraints : fundamental properties and application to picture processing *Information sciences*, Vol(7), pp. 95-132, 1974.
- [17] J.C. Régim. Amélioration de l'expressivité des contraintes de table. *Journées francophones de programmation par contraintes JFPC 2011*, pp 281-287, 2011.
- [18] E. Tsang. Foundations of Constraints Satisfaction. *Academic Press*, London, 1993.
- [19] S. Mittal and B. Falkenhainer. Dynamic Constraint Satisfaction Problems *AAAI Boston*, US, pp. 25-32, 1990.
- [20] A.K. Mackworth, *Consistency in Networks of Relations*, Artificial Intelligence, pp. 99-118, 1977.
- [21] B. Faltings. Arc Consistency for Continuous Variables, *Artificial Intelligence*, vol(65), pp. 363-376, 1994.
- [22] O. Lhomme. Consistency techniques for numeric CSP, *International Joint Conference on Artificial Intelligence*, pp. 232-238, 1993.
- [23] R.E. Moore. Interval Analysis, *Prentice-Hall*, 1966.
- [24] D. Sam-Haroud and B. Faltings. Consistency Techniques for Continuous Constraints, *Constraints Journal*, pp. 85-118, 1996.
- [25] É. Vareilles, M. Aldanondo and P. Gaborit. How to take into account piecewise constraints in constraint satisfaction problem, *Engineering Applications of Artificial Intelligence*, Elsevier, 2009.
- [26] P. Pitiot, M. Aldanondo, É. Vareilles, P. Gaborit, M. Djefel and S. Carbonnel. Concurrent product configuration and process planning, towards an approach combining interactivity and optimality, *International Journal of Production Research*, Accepted, 2011.
- [27] O. Lhomme and M. Rueher. Application des techniques CSP au raisonnement sur les intervalles, *revue d'Intelligence Artificielle*, 11(3) :283-311, 1997.
- [28] É. 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*, EAAI, DOI : 10.1016/j.engappai.2011.09.002, 2011.
- [29] J.C. Régim. A Filtering algorithm for constraints of difference in csps. *AAAI*, pages 362 :367, 1994.
- [30] N. Beldiceanu and S. Demassey. Global constraint catalog, <http://www.emn.fr/x-info/sdemasse/gccat/index.html>, 2009.
- [31] H. Fargier, J. Lang and T. Schiex. Selecting preferred solutions in Fuzzi Constraint Satisfaction Problems. *Proc. of the 1st European Congress on Fuzzy and Intelligent Technologies*, 1993.