



HAL
open science

CONFIGURATION / PLANIFICATION : EXPERIMENTATIONS EN OPTIMISATION EVOLUTIONNAIRE

Paul Pitiot, Michel Aldanondo, Élise Vareilles, Paul Gaborit

► **To cite this version:**

Paul Pitiot, Michel Aldanondo, Élise Vareilles, Paul Gaborit. CONFIGURATION / PLANIFICATION : EXPERIMENTATIONS EN OPTIMISATION EVOLUTIONNAIRE. 9th International Conference on Modeling, Optimization & SIMulation, Jun 2012, Bordeaux, France. hal-00728648

HAL Id: hal-00728648

<https://hal.science/hal-00728648v1>

Submitted on 30 Aug 2012

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.

CONFIGURATION / PLANIFICATION : EXPERIMENTATIONS EN OPTIMISATION EVOLUTIONNAIRE SOUS CONTRAINTES

P. PITIOT

3IL Rodez – CCI Aveyron
12000 Rodez – France
Toulouse University – Mines Albi
81013 Albi CEDEX 09 - France
paul.pitiot@mines-albi.fr

M. ALDANONDO, E. VAREILLES, P. GABORIT

Toulouse University – Mines Albi
81013 Albi CEDEX 09 - France
aldanondo@mines-albi.fr, elise.vareilles@mines-albi.fr,
paul.gaborit@mines-albi.fr

RESUME : Dans un contexte de customisation de masse, le système d'aide à la décision est un élément clé du support lors de la phase de négociation avec le client. Nous avons proposé précédemment une approche interactive permettant de traiter de façon concurrente les aspects Produit (configuration) et Projet (planification). Cet aspect correspond à un réel besoin industriel afin de limiter les incohérences entre les décisions prises dans chaque environnement. Dans l'approche proposée, le produit/projet voulu est sélectionné en instanciant successivement un ensemble de variables de décision du modèle selon les exigences de l'utilisateur. Mais ces exigences ne portent généralement que sur un nombre limité de variables. Dans les systèmes existants, ces variables sont instanciées en utilisant des valeurs par défaut. Nous proposons d'ajouter à notre approche une étape d'optimisation multicritère afin de réaliser ce travail. Cette nouvelle étape doit fournir à l'utilisateur un ensemble de solutions appartenant au front de Pareto. Ce problème est intéressant à plusieurs titres : tout d'abord, la taille du problème est très importante du fait de la prise en compte des deux environnements simultanément ; il s'agit ensuite d'un problème contraint et multicritère. Ces éléments nous ont conduits à utiliser un algorithme évolutionnaire modifié pour la prise en compte des contraintes.

MOTS-CLES : aide à la conception, configuration, planification, contraintes, optimisation évolutionnaire.

1 INTRODUCTION

Nous nous intéressons dans cet article à un outil d'aide à la décision utilisé lors de la phase de négociation avec le client pour des produits ou systèmes dits "configurables" tels qu'un avion, un bateau ou une voiture. Lors de cette phase, l'utilisateur doit être capable de proposer rapidement au client une sélection de solutions potentielles permettant de répondre à ses exigences à la fois en termes de délais, de coût et de performances.

La particularité de l'outil proposé tient au fait qu'il traite conjointement des aspects Produit (composants, options, fonctionnalités, performance, coûts, ...) et Projet (ressources, tâches, temps de cycle, coûts, ...). Le fait de prendre en compte les deux environnements simultanément augmente considérablement la taille de l'espace de recherche. Par exemple, rien qu'en configuration, (Amilhastre *et al.*, 2002) utilisent un modèle conduisant à plus de $1,4 \times 10^{12}$ configurations possibles avec de très nombreuses décisions à prendre (choix de composants/fonctions, options, etc.).

Pour répondre à cette problématique d'aide à la décision, la démarche proposée est composée de deux phases :

- i) Une première phase où l'utilisateur (décideur accompagné du client) peut manipuler interactivement le modèle couplé Produit/Projet afin de saisir et évaluer l'impact des exigences du client. Lors de cette phase, l'utilisateur manipule donc un modèle de « ce que sait faire l'entreprise » tant en termes techniques qu'organisationnels. Les exigences du client ne portant que sur un nombre limité de variables (voir section 1.3), il reste à la fin de cette phase de nombreuses variables de décision à instancier. C'est l'objectif de la phase suivante.
- ii) Puis la seconde phase d'optimisation multicritères permet de proposer au client une sélection de solutions performantes parmi lesquelles faire son choix.

Notre but est de définir une méthode correspondant aux besoins industriels (interactivité, modélisation, optimisation), et de fournir les outils logiciels nécessaires pour chacune de ces deux étapes. Lors de précédents travaux, nous avons exposé cette démarche ainsi que les outils nécessaires. Cet article s'intéressera donc plus particulièrement à de nouveaux développements sur l'outil d'optimisation de la seconde phase. Après un rappel de quelques travaux passés, l'objectif de cet article est de présenter des résultats expérimentaux et de montrer l'intérêt de l'approche proposée. Un avion de tourisme et d'affaires est utilisé comme exemple tout au long de cet article.

1.1 Configuration de produit et planification de projet d'obtention associé au produit

Nous entendons par configuration de produit, l'activité de sélection parmi toutes les options et variantes d'un produit, conformément à un modèle de produit générique ou famille de produits, afin de satisfaire au plus juste les exigences du client (Mittal et Frayman, 1989), (Agard *et al.*, 2009). Dans la mesure où le produit est configurable (ou customisable), il doit en être de même pour le processus de fabrication qui lui est associé. Nous entendons en conséquence par planification du processus de fabrication, l'activité visant la détermination du processus de fabrication (ensemble d'opérations de production et de ressources) et des dates de début et de fin de ces opérations. De manière similaire au produit, nous faisons l'hypothèse qu'il est possible de réaliser un modèle de processus de fabrication générique qui, lorsqu'il est mis en regard avec le produit configuré, permet d'élaborer et de planifier le processus de fabrication spécifique au produit configuré comme cela est montré dans (Aldanondo et Vareilles, 2008).

Si de très nombreux travaux se sont intéressés à ces deux problèmes, très peu les ont considérés simultanément. Cependant il apparaît clairement qu'une décision de configuration de produit peut avoir des conséquences sur la planification de sa production (par exemple : une finition haut de gamme nécessite deux mois de plus de cycle) et que réciproquement, une décision de planification peut avoir des conséquences sur la configuration du produit (par exemple : cette durée d'assemblage interdit ce type de motorisation). Dans une approche classique dissociée, des itérations entre les deux environnements sont donc nécessaires pour prendre en compte les exigences du client. Par exemple, une fois l'avion souhaité par le client défini, une planification peut faire apparaître un temps de cycle trop long par rapport aux besoins client. Le décideur serait alors obligé de choisir une sélection de composants/fonctions différents puis replanifier pour voir si les délais sont tenus. Afin de prendre en compte les interdépendances produit/projet et donc d'éviter des itérations très consommatrices de temps, nous regroupons les deux problèmes de configuration et de planification en un seul problème. Ceci permettra par exemple de considérer simultanément les besoins « capacité de l'avion : 12 places » et « délai de livraison : 4 mois » et d'évaluer leurs conséquences sur le reste du modèle (coût, autres caractéristiques/fonctions de l'avion, etc.).

1.2 Configuration, planification et approches par contraintes

Dans la communauté configuration, de nombreux auteurs (Junker, 2006), (Mittal et Frayman, 1989) ont montré que l'activité de configuration pouvait être efficacement modélisée et aidée lorsqu'elle était considérée comme un problème de satisfaction de contraintes. De manière similaire, des auteurs travaillant en planification et ordonnancement (Dechter *et al.*, 1991), (Bartak *et al.*,

2010) ont montré que les approches par contraintes pouvaient également aider considérablement la résolution des problèmes de planification. Nous considérons en conséquence pour nos travaux les problèmes de configuration / planification comme des problèmes de satisfaction de contraintes. Nous faisons les deux hypothèses :

- (h1) un modèle de contraintes peut être établi pour représenter d'une part le produit générique et d'autre part son processus de fabrication générique,
- (h2) ces deux modèles peuvent être mis en relation par des contraintes dites de couplage correspondant aux interdépendances mentionnées en fin de section précédente.

Nous restreignons les problèmes de configuration et de planification à l'instanciation de ces deux modèles (Vareilles *et al.*, 2008), (Aldanondo *et al.*, 2010). Recherchant une assistance interactive à la configuration et à la planification, nous n'exploiterons que les principes de filtrage de contraintes. Cela signifie que lors de la saisie de chaque exigence élémentaire, le processus de filtrage de contraintes est déclenché et l'espace de solution est réduit. Nous entendons par exigence élémentaire, la restriction du domaine de définition d'une variable du modèle de configuration (par exemple nombre de siège ≤ 6) ou de planification (par exemple date de fin $< 05/04/2011$). Nous considérons également une planification à capacité infinie de ressource, c'est à dire que les produits ne sont fabriqués qu'à la commande et les capacités de production sont adaptées à chaque nouvelle commande.

1.3 Optimisation multicritère

Lors de la phase précédente, il est interactivement possible de configurer complètement un produit et de planifier son processus de fabrication. Par « complètement » nous sous-entendons que toutes les variables de configuration et de planification ont leur domaine de définition réduit à un singleton. C'est à dire que nous débouchons sur une solution du problème de contraintes associé. Ce fonctionnement ne constitue pas notre objectif.

Nous considérons que l'utilisateur est principalement intéressé par un nombre réduit de caractéristiques produit et processus, sur lesquelles il va imposer des exigences (dites non négociables) et laisser le système rechercher des solutions pour les autres (caractéristiques négociables). Dans les logiciels de configuration industriels, des valeurs « par défaut » données par les experts sont généralement utilisées pour ces caractéristiques négociables, ce qui conduit à des solutions peu performantes. Nous proposons d'utiliser un outil d'optimisation pour sélectionner des valeurs conduisant à des solutions optimisées.

Cette recherche doit par contre minimiser différents critères antagonistes, dans le cas présent, nous considérons le coût global (produit et processus) et le temps de cycle du processus. Afin d'éviter l'emploi de mécanismes d'agrégation de critères, nous recherchons les solutions Pareto-optimales. De plus, le fait de prendre en compte les deux environnements simultanément augmente considérablement la taille de l'espace de recherche. Enfin, l'optimisation doit être réalisée dans un temps d'exécution convenable, c'est-à-dire en relation avec le type de produit configurable. Par exemple, pour une voiture, un temps de réponse d'une heure maximum semble convenir, tandis que pour un avion, un jour de calcul ne semble pas problématique au regard des investissements consentis par le client.

Ces contraintes (espace de recherche important et recherche multicritères) nous ont amené à utiliser un algorithme évolutionnaire que nous avons adapté pour l'optimisation sous contraintes. Nous ne considérons dans cette communication que deux critères, un troisième critère de performance du produit pourrait être ajouté sans problème.

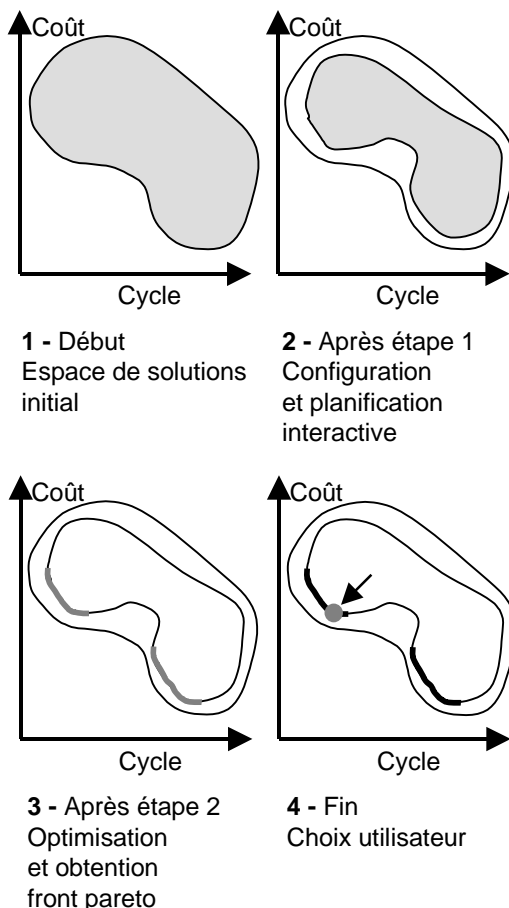


Figure 1. Démarche proposée

1.4 Système résultant, limites de l'approche et organisation de la communication

Nous considérons donc un système travaillant en deux étapes supportées par deux outils. La première étape, avec un outil de propagation de contraintes, assiste interactivement la configuration de produit et la planification du processus de fabrication. Cette première étape considère les exigences non négociables de l'utilisateur et opère une première réduction de l'espace de solution. La seconde étape, à l'aide d'un outil d'optimisation exploitant une approche évolutionnaire, recherche dans l'espace de solutions précédemment restreint, les solutions appartenant au front de Pareto minimisant les critères coût et cycle. L'utilisateur peut alors choisir la solution qui satisfait son propre compromis. Cette démarche est synthétisée en figure 1.

La démarche de configuration/planification étant posée, les propositions et résultats sont organisés de la manière suivante. Dans une seconde section le modèle de contraintes supportant la configuration et la planification est proposé. En section trois, l'approche évolutionnaire utilisée est explicitée. Elle consiste en fait en l'adaptation d'un algorithme performant, SPEA2 (Zitzler *et al.*, 2001) aux problèmes mettant en œuvre des contraintes. La section quatre est consacrée à la présentation de résultats d'expérimentations.

2 MODELE SUPPORT POUR LA CONFIGURATION/PLANIFICATION

2.1 Modèle de configuration de produit

Le modèle de configuration représente une famille de produits avec toutes ses options et variantes. Il est représenté dans la partie gauche de la figure 2. Il fait apparaître des variables descriptives (par exemple pour un avion : rayon d'action, nombre de sièges...) et des variables de coût (par exemple pour un avion : coût du moteur, coût de la finition...) qui sont des variables numériques ou symboliques. Les contraintes de configuration relient les variables descriptives (lignes pleines en noir sur la figure 2 - par exemple entre rayon d'action et type de moteur). Les contraintes de coût relient certaines variables descriptives à des variables de coût (lignes pleines en gris sur la figure 2 - par exemple type de moteur et coût du moteur). Ces contraintes correspondent à des tables de compatibilité discrète indiquant les combinaisons autorisées de valeurs de variables ou d'intervalles de valeurs de variables. Pour ces contraintes discrètes les principes de filtrage par arc cohérence (Bessiere, 2006) permettent la configuration interactive du produit et l'obtention de son coût.

2.2 Modèle de planification du processus d'obtention

Le modèle de planification est composé d'un ensemble d'opérations génériques (approvisionnement, fabrication, assemblage...) reliées par des contraintes d'antériorité. Il est représenté dans la partie droite de la figure 2. Chaque opération est définie par trois variables temporelles, deux variables ressources et une variable coût. Les trois variables temporelles sont : date de début, date de fin et durée. Ce sont des variables numériques définies par des intervalles. Les deux variables ressources indiquent pour le besoin en ressource critique de l'opération la liste des ressources utilisables et les quantités de ressources envisageables. Le besoin en ressource est une variable symbolique, la quantité de ressource et la variable coût sont des variables numériques. En ce qui concerne les contraintes : deux contraintes temporelles (lignes pleines en noir) expriment : (i) pour une opération, que sa date de fin égale sa date de début plus sa durée (contrainte de durée), (ii) pour deux opérations successives, que la date de début de la seconde est supérieure ou égale que la date de fin de la précédente (contrainte d'antériorité). Une contrainte de ressource et de coût met en relation, la durée de l'opération, la liste des ressources utilisables, les quantités envisageables et le coût de l'opération (contrainte d'arité quatre, lignes pleines en gris). Etant donné que les contraintes temporelles sont des contraintes numériques calculatoires simples (+, -, *, /, =, >, <), elles respectent les hypothèses de la cohérence aux bornes des intervalles proposée par (Lhomme, 1993). Basée sur l'arithmétique des intervalles, la cohérence aux bornes associée à l'arc cohérence des contraintes discrètes permet de filtrer les contraintes du problème et en conséquence de planifier le processus de fabrication et d'obtenir son coût.

2.3 Modèle du problème complet

Les contraintes reliant les deux modèles (en lignes pointillées noires) associent les variables descriptives du produit et les variables décrivant les opérations. Elles permettent de propager les décisions de la configuration vers la planification et inversement de la planification vers la configuration. En termes d'optimisation, le critère coût est obtenu en ajoutant une variable coût total et une contrainte exprimant que cette variable est égale à la somme de toutes les variables coût (produit et processus). Le cycle équivaut à la date de fin au plus tôt (borne inférieure de l'intervalle) de la dernière opération du processus de fabrication. Le modèle et le processus de configuration/planification étant posés, la section suivante va documenter son optimisation.

3 OPTIMISATION EVOLUTIONNAIRE PROPOSEE

3.1 Le problème d'optimisation

Le problème d'optimisation est en fait contraint et peut se définir comme un quadruplet $\langle V, D, C, f \rangle$ avec : V ensemble des variables et ensemble de domaines associés D , C un ensemble de contraintes et f une fonction multicritère à minimiser. L'ensemble V comprend toutes les variables descriptives du problème de configuration et uniquement les variables ressource (liste des ressources utilisables, et quantités de ressources envisageables) de toutes les opérations.

Chaque variable de V est un gène et l'ensemble des gènes constitue un individu. L'ensemble C rassemble les contraintes identifiées dans la section précédente. La fonction multicritère regroupe le coût produit/processus et le cycle du processus.

Les algorithmes évolutionnaire, initialement définis pour des problèmes non contraints ont fait l'objet de nombreux travaux pour prendre en compte les contraintes. Quatre courants de travaux existent, pour plus de détail voir (Coello Coello, 2002) : (i) Les fonctions de pénalités qui dégradent le résultat de la fonction d'évaluation de l'individu lorsque celui-ci ne respecte pas les contraintes ; (ii) les méthodes de réparation qui redirigent un individu ne respectant pas les contraintes vers la zone cohérente en modifiant l'individu ; (iii) les méthodes dites multi-objectifs qui séparent les critères d'optimisation du critère de cohérence des individus ; et (iv) les opérateurs spécifiques qui forcent la génération d'individus cohérents avec les contraintes. Nous nous inscrivons dans ce dernier courant de travaux et notamment l'idée de (Kowalczyk, 1997) proposant d'utiliser les contraintes pour éviter l'incohérence des individus. Cette idée peu outillée et peu développée est la base de l'approche que nous proposons, voir pour plus de détail (Pitiot *et al.*, 2010).

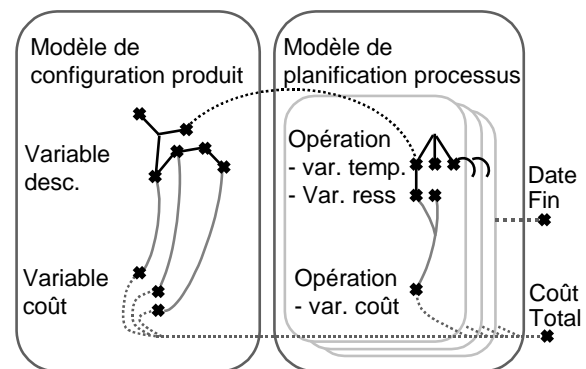


Figure 2. Architecture du modèle

3.2 Algorithme proposé

Les idées principales de notre approche sont les suivantes. Nous considérons à la base l'algorithme SPEA2 auquel nous allons associer le système de filtrage de contrainte empêchant les individus de devenir incohérents. SPEA2 comprend les étapes suivantes :

1. Initialisation ou génération d'une population initiale d'individus
2. Evaluation des individus et détermination du fitness de chaque individu
3. Sélection des meilleurs individus et constitution de l'archive les mémorisant, (fin si critère d'arrêt atteint)
4. Identification d'individus pour l'évolution, par tournoi mettant en jeu des couples d'individus
5. Evolution par mutations ou croisements des individus identifiés
6. Retour à l'étape 2 avec ajouts des nouveaux individus.

Les individus sont générés lors de l'initialisation en étape 1 puis lors de l'évolution de l'étape 5 à l'aide des opérateurs de croisement et de mutation. Pour ces deux étapes et lorsqu'un nouvel individu est généré le fonctionnement est modifié de la manière suivante. Lors de l'initialisation, chaque gène est tiré aléatoirement. La modification consiste après chaque tirage à filtrer les contraintes afin de restreindre l'espace des possibles et de « forcer » le tirage des valeurs des gènes dans un espace de solutions respectant les contraintes. Le filtrage n'étant pas infiniment puissant un mécanisme de retour arrière permet de résoudre les éventuelles incohérences. Il en va de même pour les opérateurs de mutation et de croisement. Chaque fois qu'il y a modification d'un gène, la modification est propagée pour contraindre les autres gènes à rester dans un espace de solution cohérent. Le fonctionnement de l'algorithme modifié est représenté en figure 3.

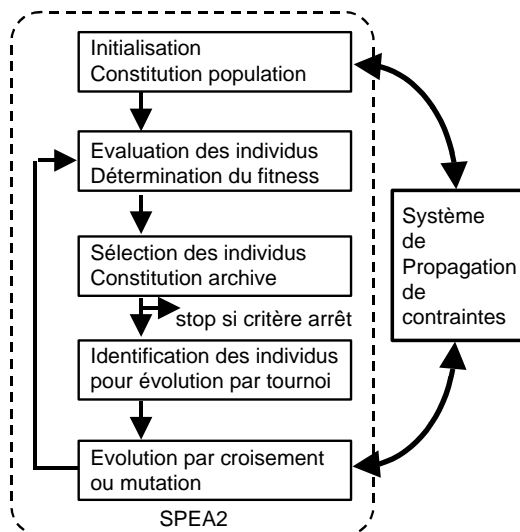


Figure 3. Algorithme SPEA2 modifié

Les modèles et processus de configuration/planification ainsi que leurs optimisations étant définis, la section suivante va s'intéresser à différentes expérimentations.

4 EXPERIMENTATIONS ET PROPOSITIONS

Ce chapitre présente des résultats d'expérimentation. Les algorithmes d'optimisation présentés ont été implémentés en langage C++ et interagissent avec le module de filtrage codé en langage Perl. Le système complet a été réalisé pour pouvoir faire fonctionner plusieurs moteurs de filtrage en parallèle (pour bénéficier de l'utilisation de population de solutions dans l'algorithme évolutionnaire). Mais pour évaluer clairement les performances de notre système, un seul système de filtrage est utilisé pour les résultats exposés dans cet article. Tous les tests ont été réalisés sur un Intel core i5 (2.27Ghz, un seul cœur utilisé) avec 2.8 Go de RAM.

Une première sous section détaille l'exemple de base. Ensuite l'approche proposée est comparée à une approche *branch and bound*. Pour terminer, nous avons commencé l'évaluation de notre approche pour de grandes tailles de problème avec une évaluation des performances de l'algorithme lorsque la taille du modèle puis la proposition d'une démarche d'optimisation en deux étapes.

4.1 Exemple

L'exemple est une version simplifiée d'un modèle permettant la configuration et planification d'avion de tourisme et d'affaires. Il respecte exactement l'architecture représentée en figure 2 et est représenté en figure 4.

Pour la partie configuration, l'avion est caractérisé par sept variables descriptives : SN ("Seat Number"), EN ("ENgine"), CS ("Crusing Speed"), FR (Flight Range), FI ("FINish"), ES ("Engine Setting") et ST ("Supplementary Tank"). Trois contraintes de configuration décrivent les combinaisons de variables autorisées. Le nombre de siège est lié au moteur, contrainte (SN, EN). Le type de moteur et la vitesse sont liés au réglage moteur, contrainte (EN, CS, ES). La vitesse et le rayon d'action peuvent nécessiter un réservoir supplémentaire, contrainte (CS, FR, ST). Cinq variables de coût sont associées par des contraintes aux variables descriptives : SNC (associé au nombre de sièges), ENC (associé au type de moteur), FIC (associé à la finition), ESC (associé au réglage moteur) et STC (associé au réservoir supplémentaire). Cette partie configuration du modèle apparaît dans la partie supérieure de la figure 4.

Pour la partie la partie planification, le processus met en œuvre six opérations. Les deux premières sans contrainte d'antériorité: approvisionnement (Op_{11}) et fabrication (Op_{12}) sont suivies par une opération d'assemblage (Op_{20}). Il y a ensuite des opérations successives de montage de réservoir supplémentaire (Op_{30}), de finition (Op_{40}) et de conditionnement/livraison (Op_{50}).

Chaque opération est caractérisée par les six variables : date de début (.pst), date de fin (.pft), durée (.pdt), res-

sources utilisables (.rrs), quantités envisageables (.qrr), coût de l'opération.

Chaque opération fait apparaître la contrainte temporelle de durée et les contraintes reliant durée, ressources utilisables, quantités envisageables et coût. Une contrainte temporelle d'antériorité est de même présente pour chaque enchaînement d'opération. Cette partie planification du modèle apparaît dans la partie inférieure de la figure 4.

Ces deux modèles sont ensuite reliés par des contraintes de couplage reliant : le moteur à l'approvisionnement (EN, Op₁₁), le nombre de siège à la fabrication (SN, Op₁₂), les deux variables précédentes à l'assemblage (EN, SN, Op₂₀), le réservoir à son opération de montage (ST, Op₃₀) et le niveau de finition à son opération de finition (FI, Op₄₀). Les caractéristiques de la dernière opération sont indépendantes de l'avion configuré. La variable coût global ("Global cost" sur la figure 4) est l'addition des cinq variables coût du modèle de configuration et des variables coût des six opérations. La variable cycle est égal à la borne inférieure de la date de fin de l'opération conditionnement livraison (Op₅₀).

Les variables descriptives du modèle de configuration comportent de 4 à 6 valeurs. La combinatoire des variables ressources utilisables et quantités possibles vaut de 3 à 25. Sans prendre en compte les contraintes, la combinatoire s'élève à 497.664.000 solutions. En prenant en compte les contraintes l'espace de solution se réduit à 10% et comporte alors 47 600 000 solutions. Ce modèle peut être consulté et utilisé interactivement, pour la partie configuration/ planification à l'adresse (<http://cofiade.enstimac.fr/cgi-bin/cofiade.pl>) en sélectionnant le modèle "Aircraft-CSP-EA-10". Ce modèle a été utilisé pour les premières expérimentations exposées dans la section suivante. Pour terminer, le modèle est « agrandi » par ajout de variables et de valeurs dans le domaine des variables existantes pour confronter notre algorithme à des problèmes de plus grandes tailles dans la dernière section.

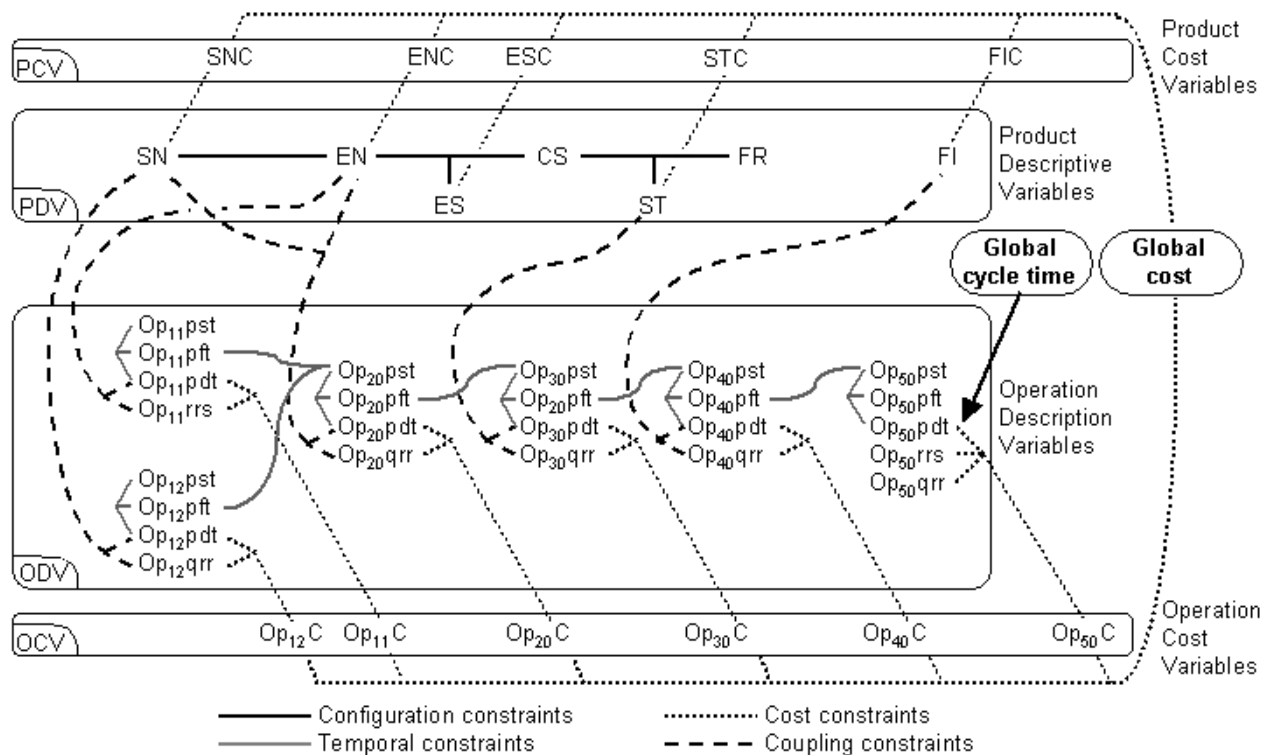


Figure 4. Exemple considéré pour les expérimentations

4.2 Premières expérimentations

La première expérimentation consiste à lancer une optimisation évolutionnaire sans passer par la phase de réduction du modèle, c'est à dire sur le problème complet. Après quelques essais, les paramètres de l'algorithme sont réglés de la manière suivante : taille de la population : 150, taille de l'archive : 100, probabilités de mutation : 0.4 et de croisement : 0.8. En moins d'une heure on obtient le front de Pareto inférieur gauche de la figure 5. Sur cette même figure, un ensemble de solutions représentant l'espace de solutions apparaît sous forme de points.

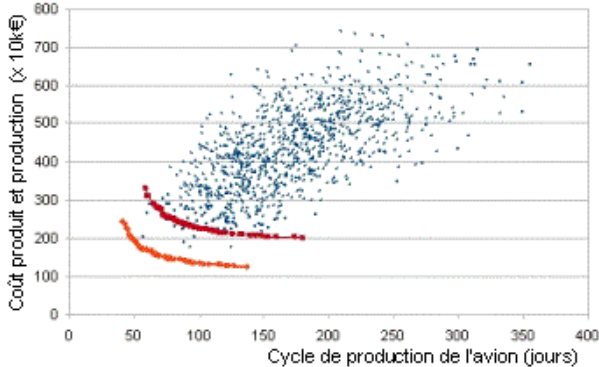


Figure 5. Espace de solutions : Front de Pareto avec et sans réduction

Une seconde expérimentation avec une étape de configuration /planification suivie d'une optimisation est considérée. Lors de la première étape le nombre de siège de l'avion (SN) est réduit de {2, 4, 6, 8, 10, 12} à {6, 8} et la finition (FI) est réduite de {"standard", "luxury", "comfort", "custom"} à {"luxury", "comfort"}. Cela correspond à une configuration que l'on peut qualifier de "moyenne". En conséquence les avions les moins onéreux et les plus rapides à produire ne devraient plus être présents dans les solutions. De même en une heure, le front de Pareto est obtenu et est de même représenté en figure 5. Ce front confirme bien le décalage vers des solutions plus coûteuses et plus longues à produire. Il apparaît donc que l'approche proposée pour l'optimisation permet d'obtenir un front de Pareto en un temps raisonnable.

La seconde expérimentation vise à évaluer la qualité de ce front de Pareto ainsi que sa vitesse de convergence. Pour évaluer la qualité du front de Pareto, nous utilisons le critère de l'hypervolume, HV, proposé par (Zitler et Thiele, 1998). Ce critère mesure une surface union des surfaces de chaque rectangle de diagonale délimitée par chaque solution dominante et un point fixe de l'espace de solution (de valeur cycle et coût les plus élevés) comme cela est représenté en figure 6. Ce critère prend en compte simultanément la convergence de la solution et la bonne répartition des points sur l'ensemble du front.

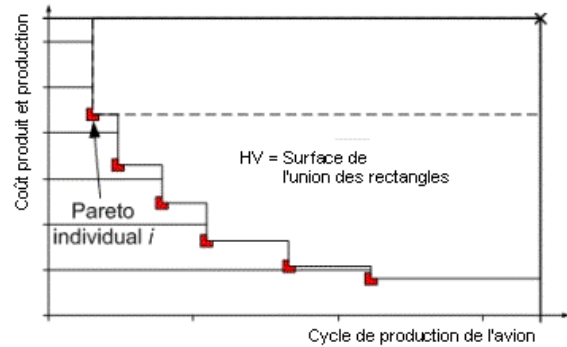


Figure 6. Critère de qualité du front : Hypervolume

En ce qui concerne la vitesse de convergence, nous comparons l'approche proposée à un mécanisme de *Branch and Bound* qui parcourt l'espace de solutions. La procédure de séparation consiste à choisir l'une des variables du problème et à l'instancier. A chaque instanciation, le filtrage est lancé et fournit une borne inférieure pour chaque critère. Si une de ces bornes est dominée par une solution du front déjà trouvée, la branche est abandonnée. La figure 7 montre l'évolution de l'hypervolume (moyenne de vingt exécutions) avec le second jeu de données, c'est à dire avec les variables nombre de sièges (SN) et finition (FI) réduites avant optimisation. Il apparaît clairement que l'approche évolutionnaire proposée est beaucoup plus rapide que le *Branch and Bound*. Ce dernier commence à fonctionner tardivement (6000 secondes) du fait de la difficulté à trouver les premiers individus cohérents. La rapidité d'obtention d'un Pareto de bonne qualité par l'approche évolutionnaire proposée peut être également soulignée, au bout de 2000 secondes on est à moins de 1% de l'hypervolume optimal. Cette remarque va conduire à une seconde expérimentation proposée pour les problèmes de taille plus conséquente.

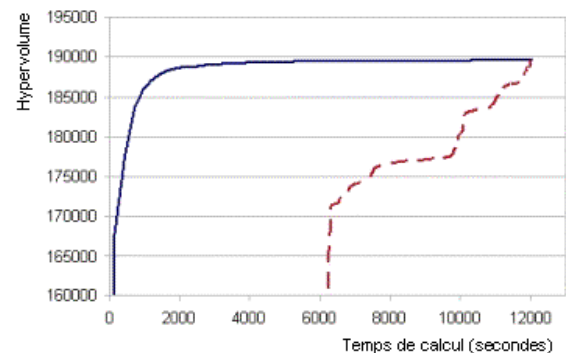


Figure 7. Évolution de l'hypervolume moyen obtenu par l'algorithme évolutionnaire proposé en bleu et par le *Branch and Bound* en rouge discontinu

4.3 Temps de calcul et taille de l'espace de recherche

Comme nous l'avons évoqué en introduction, les modèles utilisés par les industriels, rien qu'en configuration ont une taille très importante (de l'ordre de 10^{12} solutions possibles). Nous en déduisons que l'outil que nous proposons devra être capable de faire face à une combinatoire très importante ; même si il n'est pas très utile d'utiliser une modélisation trop détaillée en phase de négociation tant du côté produit que du côté projet.

Nous avons voulu savoir dans un premier temps la limite de taille à laquelle notre algorithme prend le pas sur l'algorithme B&B et jusqu'à quelle limite celui-ci est utilisable. La figure 8 illustre les résultats obtenus par les deux algorithmes lorsque le nombre de variables de décision du modèle complet augmente (ici entre 13 et 16 variables comportant chacune trois valeurs possibles). L'axe verticale correspond au temps nécessaire (en secondes) pour que i) l'algorithme B&B trouve le front de Pareto optimal ou ii) l'algorithme évolutionnaire atteigne le front optimal dans 90% des cas (ici, temps nécessaire pour que neuf exécution sur dix de l'algorithme trouvent l'optimal). Les tests montrent que l'algorithme B&B est à réserver pour les cas de petites tailles où il est plus performant que l'AE mais il est très vite dépassé puis inutilisable au-delà de 16 variables à trois valeurs (soit une combinatoire de l'ordre de 43 milliards de solutions possibles sans prendre en compte les contraintes).

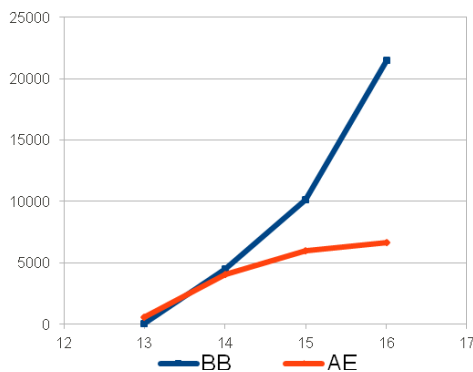


Figure 8. Temps de calcul pour trouver le front optimal selon le nombre de variables de décision du modèle

Nous avons ensuite confronté l'algorithme évolutionnaire modifié à un très grand modèle comprenant 30 variables de décision avec six ou trois valeurs possibles pour chacune d'elles, soit une combinatoire totale de 8.4×10^{17} solutions potentielles sans prise en compte des contraintes. N'ayant pas le repaire que constitue le front de Pareto optimal obtenu par l'algorithme B&B comme pour les cas de petites tailles, nous avons utilisé un autre critère d'arrêt lié à la performance obtenue : L'exécution s'arrête si l'algorithme n'a pas amélioré le front trouvé depuis plus de deux heures. La figure 9 illustre l'évolution de l'hypervolume au cours de l'exécution de l'algorithme (paramètres de l'AE : taille de la popula-

tion : 100, taille de l'archive : 150). L'algorithme s'est arrêté après 47,35h de calcul (170 000 secondes environs). Néanmoins, en analysant le comportement de l'algorithme en cours d'exécution, on s'aperçoit que celui-ci progresse très rapidement vers une valeur proche du front de Pareto final. Il faut 2,8 heures pour obtenir 90% de l'hypervolume final, 10,1 heures pour obtenir 99% de cette valeur et 21,8h pour obtenir 99.9% de cette valeur.

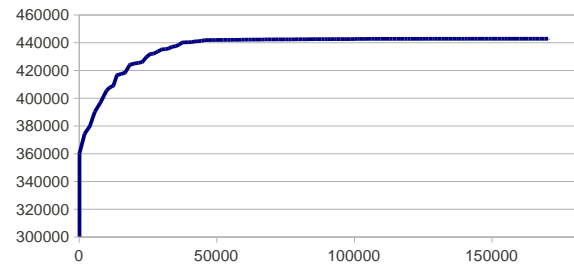


Figure 9. Evolution de l'hypervolume pour un problème de grande taille avec l'algorithme évolutionnaire proposé

Nous en concluons que notre approche reste valide au moins jusqu'à cette taille de modèle, puisque nous nous sommes fixé une durée maximale de l'ordre d'un jour pour fournir à l'utilisateur le front de Pareto optimisé (voir section 1.2). Dans le cas où les modèles industriels seraient encore plus conséquents que ce dernier exemple, nous avons mis au point une méthode d'optimisation itérative présentée dans la section suivante. De plus, nous travaillons actuellement pour compléter et détailler ces résultats dans de grands espaces de recherche avec notamment un comparatif avec une méthode dite par pénalités pour la prise en compte des contraintes.

4.4 Optimisation itérative pour les problèmes conséquents

Les premières expériences ont montré qu'une bonne approximation du front de Pareto pouvait être obtenue relativement rapidement. Ceci nous conduit à proposer une démarche itérative pour les problèmes de taille plus grande.

En reprenant la démarche en deux étapes de la figure 1 : configuration/planification puis optimisation, l'idée est de conserver la première étape en l'état (premier et second graphe de la figure 10) mais de décomposer l'étape d'optimisation en deux temps comme cela est schématisé en figure 10. Tout d'abord une première approximation grossière du front de Pareto est rapidement recherchée et présentée à l'utilisateur (troisième graphe de la figure 10). Ce dernier peut alors indiquer la zone du compromis l'intéressant (quatrième graphe de la figure 10). Cette zone correspondant à deux valeurs maximums pour les deux critères coût et cycle, ces contraintes sont alors saisies dans le système de configuration/ planification et un filtrage restreignant l'espace de solutions est effectué. Une fois cet espace de solutions restreint, une seconde

optimisation évolutionnaire est lancée mais uniquement sur l'espace de solutions restreint (cinquième graphe de la figure 10).

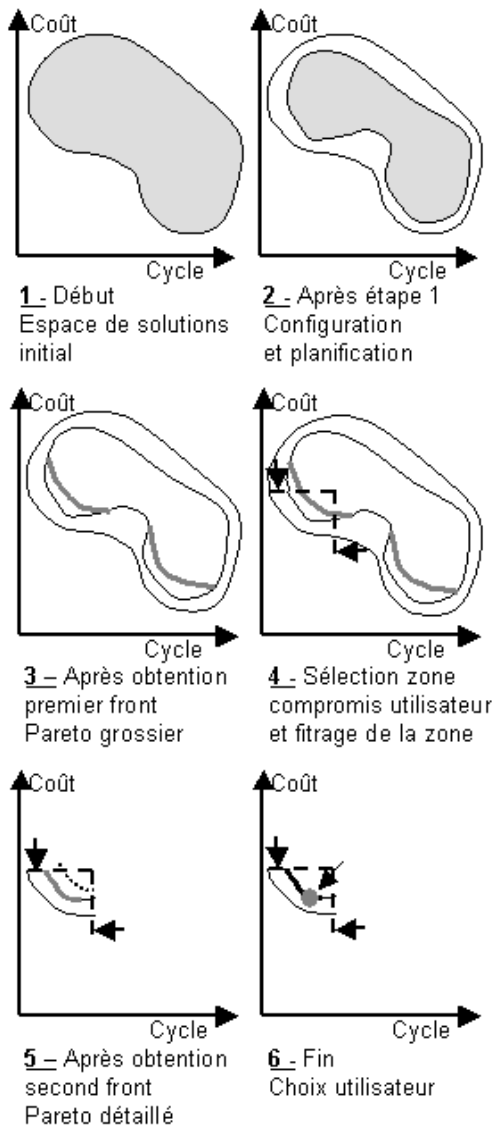


Figure 10. Démarche avec optimisation itérative

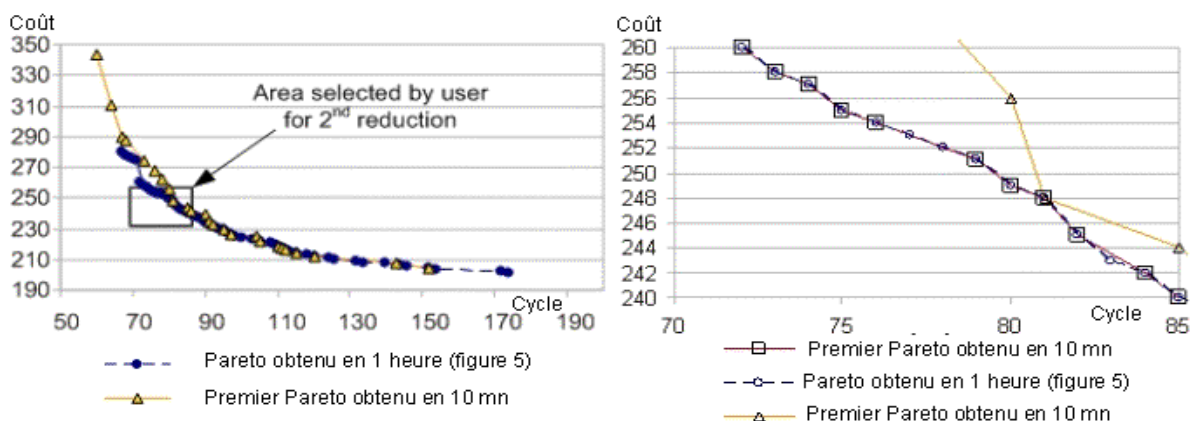


Figure 11. Optimisation itérative

Cette démarche a été testée sur le modèle de base (section 4.1). La partie gauche de la figure 11 représente le Pareto de la figure 5 obtenu en une heure. Il montre également un Pareto grossier obtenu en 10 minutes et la zone de compromis sélectionnée par l'utilisateur. La partie droite de cette même figure fait apparaître un zoom sur la zone de compromis sélectionnée, les deux Pareto précédents ainsi que le second front de Pareto détaillé obtenu après filtrage en 10 minutes. Cette seconde vue montre que la démarche itérative conduit en deux optimisations de 10 minutes à un résultat identique à l'optimisation lancée sur une heure. Ces résultats prometteurs nous conduisent à développer actuellement un modèle d'expérimentation paramétrable de plus grande taille pour valider définitivement cette proposition.

5 CONCLUSIONS

Nous avons présenté dans cette communication un outil d'aide permettant d'assister la configuration de produit et la planification de sa fabrication en exploitant un outil de propagation de contraintes couplé à un outil d'optimisation évolutionnaire. La configuration/planification est réalisée en filtrant un ensemble d'exigences non négociables qui restreint l'espace de solutions. L'optimisation évolutionnaire, en exploitant les caractéristiques négociables, propose ensuite des solutions Pareto optimales minimisant coût et cycle.

L'originalité et l'intérêt du premier outil réside dans le couplage des deux problèmes de configuration et de planification qui permet de considérer de manière simultanée et en parfaite cohérence les exigences et contraintes produit et les exigences et contraintes de fabrication. En ce qui concerne l'optimisation, l'adaptation de l'algorithme SPEA-2 proposée permet sans ambiguïté d'optimiser les problèmes d'optimisation évolutionnaire contraints et surtout de laisser l'utilisateur déterminer son propre compromis coût/cycle.

En terme de performance, les expérimentations proposées ont montré qu'il était tout à fait envisageable d'obtenir un front de Pareto de bonne qualité en un temps raisonnable. La comparaison avec un mécanisme de *branch and bound* a également confirmé ce résultat. De plus, pour des problèmes de taille plus conséquente,

une démarche en deux itérations a été proposée et expérimentée.

Ces résultats prometteurs nous ont conduits à initialiser différents travaux. Tout d'abord, élaborer des modèles paramétrables plus conséquents pour valider définitivement les propositions. Nous pensons ensuite travailler sur la comparaison de notre approche avec les approches d'optimisation sous contraintes exploitant les pénalités. Dernier aspect qui retient notre attention, ajouter un troisième critère de performance produit.

6 REMERCIEMENTS

Les auteurs souhaitent remercier l'ensemble des partenaires du projet ATLAS, l'Agence Nationale de la Recherche pour son financement et le pôle de compétitivité mondial *Aerospace valley* pour leur implication et soutien.

7 REFERENCES

- Agard, B., Da Cunha, C., Cheung, B., 2009. Composition of Module Stock for Final Assembly Using an Enhanced Genetic Algorithm. *International Journal of Production Research*, 47(20), p. 5829-5842
- Aldanondo M., Vareilles E., 2008. Configuration for mass customization: how to extend product configuration towards requirements and process configuration, *Journal of Intelligent Manufacturing*, vol. 19 n° 5, p. 521-535A.
- Aldanondo M., Vareilles E., Djefel M., 2008. Towards an association of product configuration with production planning, *International Journal of Mass Customisation*, vol. 3 n°4, 2010, p. 316-332.,
- Amilhastre J., Fargier H., Marquis P., 2002. Consistency restoration and explanations in dynamic cpsps - application to configuration, in: *Artificial Intelligence*, vol.135, p. 199-234.
- Barták R., Salido M., Rossi F., 2010. Constraint satisfaction techniques in planning and scheduling, in: *Journal of Intelligent Manufacturing*, vol. 21, n°1, p. 5-15.
- Bessiere C. 2006. Handbook of Constraint Programming, Eds. Elsevier, chap. 3, p. 29-70
- Chelouah R., Baron C., Zholghadri M., Gutierrez C. 2009. Meta-heuristics for System Design Engineering, in: *Studies in Computational Intelligence*, vol. 203, p. 387-423.
- Coello Coello C. 2002., Theoretical and numerical constraint-handling techniques used with EAs : A survey of the state of art, *Computer Methods in Applied Mechanics and Engineering*, vol. 191, n°11-12, p. 1245-1287
- Dechter R., Meiri I., Pearl J. 1991. Temporal Constraint Satisfaction Problems, *Artificial Intelligence*, n°49, p. 61-95.
- Junker U., 2006. *Handbook of Constraint Programming*, Eds. Elsevier, chap. 24, p. 835-875.
- Kowalczyk R. 1997. Constraint Consistent Genetic Algorithms, proc. of *IEEE conf. on evolutionary computation*, 1997, p. 343-348.
- Lhomme O. 1993. Consistency techniques for numerical CSPs, in: *IJCAI 93*, Chambéry, France, p. 232-238
- Li L., Chen L., Huang Z., Zhong Y. 2006. Product configuration optimization using a multiobjective GA, *I.J. of Adv. Manufacturing Technology*, vol. 30, p. 20-29.
- Mittal S., Frayman F., 1989. Towards a generic model of configuration tasks, *proc of IJCAI*, p. 1395-1401,
- Pitiot P., Aldanondo M., Djefel M., Vareilles E., Gaborit P., Coudert T., 2010. Using constraints filtering and evolutionary algorithms for interactive configuration and planning. IEEE press, *IEEE Industrial Engineering and Engineering Management*, p.1921-1925, Macao China.
- Vareilles E., Aldanondo M., Djefel M., Gaborit P. 2008. Coupling interactively Product and Project Configuration: a Proposal using Constraints Programming. *International Mass Customization and International Conference on Economic, Technical and Organizational Aspects of Product Configuration Systems*, Copenhagen, Denmark, ISBN: 978-87-90855-12-3.
- Zitzler E., Laumanns M., Thiele L. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *Technical Report 103, Computer Engineering and Communication Networks Lab (TIK)*, Swiss Federal Institute of Technology (ETH), Zurich.
- Zitzler E., Thiele L. 1998. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study, f *5th Conference on Parallel Problem Solving from Nature*, Eds. Springer Verlag, p. 292-301.