



HAL
open science

Optimisation du processus de configuration et planification : vers une approche en deux étapes

Paul Pitiot, Michel Aldanondo, Élise Vareilles, Paul Gaborit

► To cite this version:

Paul Pitiot, Michel Aldanondo, Élise Vareilles, Paul Gaborit. Optimisation du processus de configuration et planification : vers une approche en deux étapes. MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nov 2014, Nancy, France. hal-01166669

HAL Id: hal-01166669

<https://hal.science/hal-01166669v1>

Submitted on 23 Jun 2015

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.

OPTIMISATION DU PROCESSUS DE CONFIGURATION ET PLANIFICATION : VERS UNE APPROCHE EN DEUX ÉTAPES

P. PITIOT

Toulouse University, Mines Albi, France
3IL-CCI de l'Aveyron, Rodez, France
p.pitiot@mines-albi.fr

Michel Aldanondo, Elise Vareilles,
Paul Gaborit

Toulouse University, Mines Albi, Albi, France
michel.aldanondo@mines-albi.fr ;
elise.vareilles@mines-albi.fr ;
paul.gaborit@mines-albi.fr

RÉSUMÉ : Dans un contexte de « customization de masse », cet article s'intéresse à l'association de la configuration de produit et de la planification du processus de production associé au produit. Il s'agit d'un problème difficile et contraint où l'on cherche à minimiser à la fois le coût du produit et du processus d'obtention, mais également le temps de cycle du processus d'obtention. Nous proposons des méthodes et outils à base de contraintes pour soutenir cette activité. Dans nos travaux précédents, nous avons proposé une approche combinant interactivité et optimisation appelée CFB-EA (pour constraint filtering based evolutionary algorithm). Dans cet article, nous proposons d'améliorer l'étape d'optimisation pour les problèmes de grandes tailles. Nos expérimentations précédentes ont montrées la capacité de la méthode CFB-EA à trouver rapidement une bonne approximation du front de Pareto recherché. Ceci nous a conduits à proposer une méthode d'optimisation en deux étapes. La première étape fournit rapidement une approximation du front de Pareto à l'utilisateur. Celui-ci peut ainsi choisir la ou les zones qui l'intéressent plus particulièrement. Grâce à une technique de filtrage, la seconde étape d'optimisation permet ensuite de détailler le front de Pareto uniquement dans les zones de préférences de l'utilisateur.

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

1 INTRODUCTION

Cet article s'intéresse à l'optimisation conjointe des problèmes de configuration de produits et de planification de production. Chacun de ces problèmes peut être modélisé comme un problème de satisfaction de contraintes (CSP) et ces deux CSP sont également reliés par des contraintes. Dans nos précédents travaux (Pitiot et al., 2013), nous avons montré ce problème peut être abordé par une approche en deux phases : (i) une phase de configuration/planification interactive où les besoins non-négociables de l'utilisateur peuvent être pris en compte et manipulés grâce à des techniques de filtrage par contraintes, ceci amène une première réduction de l'espace de recherche ; (ii) une phase d'optimisation peut ensuite rechercher les meilleurs choix possibles sur les besoins dits négociables de l'utilisateur dans l'espace restreint issu de la première phase interactive.

Il s'agit également d'un problème multicritères : performance du produit, temps de cycle du processus d'obtention ou coût global (produit et processus) peuvent être optimisés. Le but du processus d'optimisation est alors de proposer à l'utilisateur des solutions appartenant au front de Pareto. Dans cet article, nous considérerons deux critères : le temps de cycle et le coût global. La figure 1 illustre donc notre approche en deux phases : interactivité puis optimisation.

Nos expérimentations précédentes (Pitiot et al., 2012) nous ont permis d'évaluer les performances du processus d'optimisation selon les caractéristiques du problème à résoudre (taille du problème et densité de contraintes).

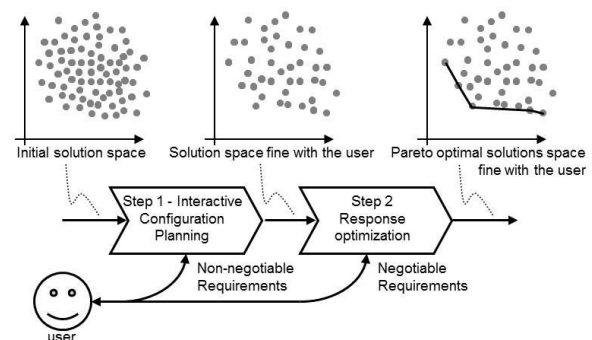


Figure 1 : Approche en deux phases

Afin de pouvoir traiter des problèmes de taille très importantes, nous proposons dans cette étude de séparer la phase d'optimisation multicritères en deux sous-phases : (i) une première approximation rapide qui permet d'avoir une vue d'ensemble du front de Pareto ; (ii) une seconde étape d'optimisation uniquement sur les zones choisies par l'utilisateur à l'étape précédente. L'objectif de ce papier est de présenter quelques résultats expérimentaux qui montrent que ce mécanisme permet de réduire de façon significative le temps d'optimisation en améliorant la qualité de l'optimisation.

Dans les parties suivantes, nous formaliserons le problème de configuration/planification conjointe illustré par un exemple très simple. Le processus d'optimisation sera ensuite décrit en détail avec l'approche proposée précédemment et une présentation du cas d'étude choisi. Finalement, la dernière section présentera les expérimentations menées.

1.1 Configuration et planification

De nombreux auteurs (Mittal and Frayman, 1989), (Soininen et al., 1998) ou encore (Aldanondo et al., 2008) ont défini la configuration comme l'activité permettant d'obtenir un produit spécifique ou "customisé" (c.à.d. un ensemble de propriétés ou caractéristiques permettant de définir un produit donné) à partir d'un modèle générique du produit ou d'une famille de produits en prenant en compte les besoins spécifiques de l'utilisateur. D'autres auteurs (Schierholt 2001), (Bartak et al., 2010) ou (Zhang et al. 2013) ont montré que le même type de raisonnement peut être appliqué pour la planification du processus d'obtention d'un produit. Une planification (c.à.d. un choix de tâches ou de ressources à utiliser, etc...) est alors le résultat de l'instanciation d'un modèle de processus d'obtention générique en prenant en compte les besoins de l'utilisateur et les caractéristique du produit.

Plusieurs études (voir par exemple (Juncker, 2006) ou (Laborie, 2003)) traitant de problèmes de configuration et/ou de planification ont montrées que chacun de ces problèmes peut être représenté par un CSP. Nous proposons de les représenter dans un même CSP enfin de les traiter conjointement.

Ce traitement conjoint et le processus d'aide à la décision associé présentent principalement trois points forts. Premièrement, cela permet de représenter des contraintes liant ces deux environnements (par exemple, exprimer le fait qu'une bonne qualité de finition du produit engendrera un délai de production supplémentaire ou alors qu'un temps d'assemblage donné va interdire l'utilisation de certains composants). Deuxièmement, cela permet de saisir les besoins de l'utilisateur concernant la planification même lorsque le produit n'est pas totalement défini. En effet, la séquence traditionnelle limitait l'utilisateur en lui imposant de choisir un produit avant de pouvoir étudier l'aspect planification du processus d'obtention. Il en résultait parfois de coûteuses itérations lorsque le produit choisi en configuration ne pouvait être obtenu en respectant les contraintes et besoins de l'utilisateur en termes de planification. Troisièmement, la représentation sous forme de CSP est particulièrement bien adaptée pour un processus d'aide à la décision interactif notamment grâce aux techniques de filtrage de contraintes.

Il est à préciser que nous considérons une planification à capacité infinie car pour le type de produits complexes considérés, l'horizon de décision permet d'adapter les capacités de production selon les besoins de l'utilisateur.

Afin d'illustrer ce problème de configuration / planification, nous allons utiliser un exemple très simple déjà proposé dans (Pitiot et al., 2012). Il s'agit de la configuration/planification d'un petit avion de tourisme. Le modèle de contraintes associé est illustré sur la figure 2. L'avion est défini par deux variables produit : le nombre de sièges (Seats : 4 ou 6 sièges possible) et le rayon d'action (Range : 600 ou 900kms). Une contrainte de configuration (Cc1) interdit les avions à 4 sièges et un rayon d'action de 600 kms.

Le processus d'obtention comprend deux opérations : l'approvisionnement et l'assemblage (notées Sourc et Assem). Chaque opération est décrite par deux variables de planification : le type de ressources et la durée. Pour l'approvisionnement, la ressource R-sourc peut être Slow-S ou Fast-S, avec une durée D-sourc de 2, 3, 4 ou 6 semaines ; et pour l'assemblage, la ressource R-assem peut être Norm-A ou Quic-A, avec une durée D-assem de 4, 5, 6 ou 7 semaines.

Deux contraintes relient les variables des deux environnements produit et projet : Cp1 relie le nombre de sièges avec l'opération d'approvisionnement (durée et type de ressources possibles) et Cp2 relie le rayon d'action de l'avion avec l'opération d'assemblage (par exemple le type de réservoir nécessaire fait varier le temps d'assemblage et le type de ressources nécessaires). Les trois tables en bas de la figure 3 illustrent les combinaisons autorisées pour chaque contrainte. Celles-ci mènent à 12 solutions possibles pour le produit/projet.

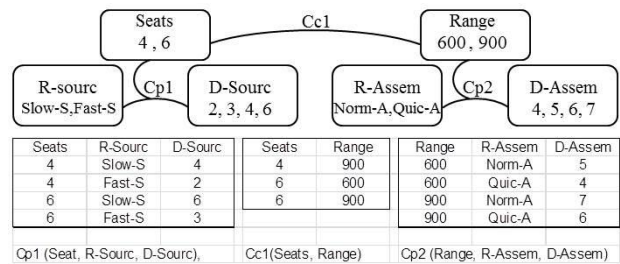


Figure 2 : modèle CSP pour la configuration et la planification conjointe

1.2 Enjeux pour l'optimisation

Par rapport au problème posé, une fois que le client ou l'utilisateur a fournit ses besoins non négociables, il est particulièrement attentif aux notions de coûts et de délais qu'il souhaite obtenir. Nous avons donc complété le modèle précédent avec des variables et des contraintes supplémentaires afin de calculer ces deux critères. Le temps de cycle correspond à la date de fin de la dernière opération de production du produit configuré. Le coût est quant à lui calculé en faisant la somme des coûts de produit et des coûts de production.

Le modèle de la figure 2 est complété dans la figure 3 en tenant compte des notions de coût et de durée. Pour chaque variable côté produit et pour chaque opération à réaliser, une variable coût est ajoutée ainsi que la contrainte de coût associée : (C-Seats, Cs1), (C-Range, Cs2), (C-Sourc, Cs3) et (C-Assem, cs4) détaillées dans les tables de la figure 3.

Le cout total et le temps de cycle sont alors obtenu grâce aux contraintes numériques suivantes :

$$\text{Cout total} = C\text{-Seats} + C\text{-Range} + C\text{-Sourc} + C\text{-Assem.}$$

$$\text{Temps de cycle} = D\text{-Sourc} + D\text{-Assem}$$

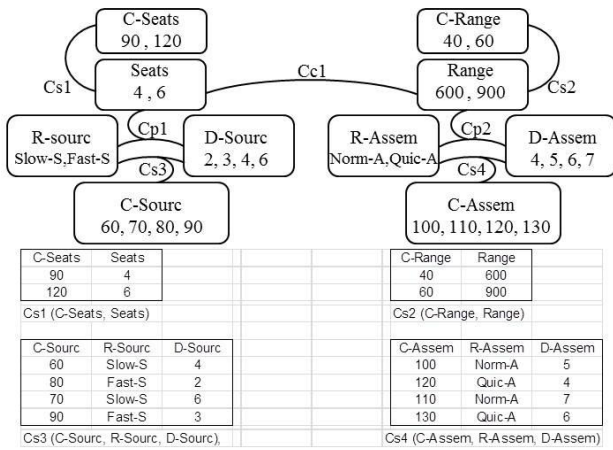


Figure 3 : Modèle CSP à optimiser

Les douze solutions existantes sont illustrées sur la figure 4 avec le front de Pareto qui correspond aux solutions optimales en termes de compromis coût total / temps de cycle. Le but de notre étude est d'améliorer la recherche de ce front de Pareto selon les préférences de l'utilisateur.

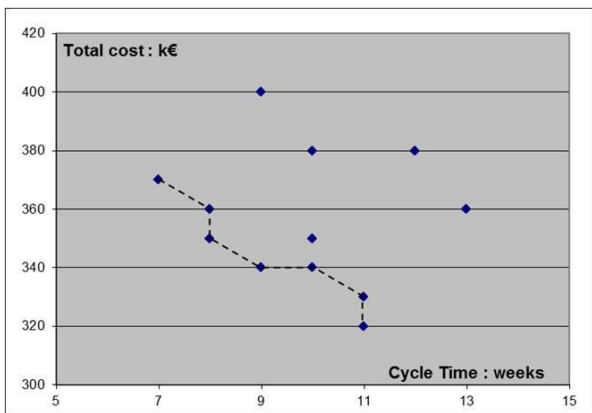


Figure 4 : Ensemble des solutions et front de Pareto

2 PROBLÈME D'OPTIMISATION

Nous définissons dans cette section le problème d'optimisation puis l'algorithme d'optimisation utilisé. Le processus expérimental est ensuite présenté.

2.1 Problème d'optimisation

Le problème d'optimisation à résoudre peut être généralisé tel qu'exposé sur la figure 5. Il s'agit d'un problème d'optimisation contraint (O-CSP) défini par le quadruplet $\langle V, D, C, f \rangle$ où V est l'ensemble des variables de décision dans le modèle (choix possibles pour l'utilisateur), D est l'ensemble des domaines associés aux variables de V , C est l'ensemble des contraintes portant sur les variables de V et enfin f la fonction d'évaluation multivaluées.

L'ensemble V comprend les variables de décisions côté produit et les variables de choix de ressources côté projet (la durée d'une activité découle du choix de ressources selon le type de produit sélectionné).

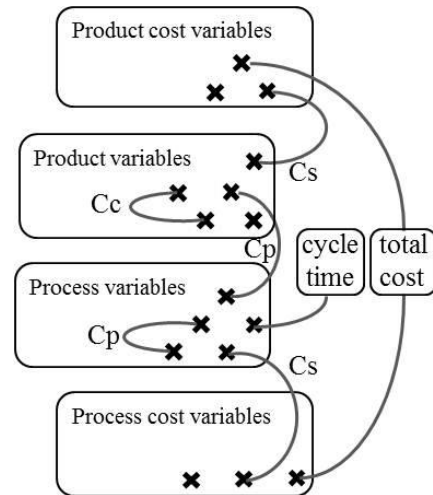


Figure 5 : Ensemble des solutions et front de Pareto

L'ensemble C comprend uniquement les contraintes de configuration (C_c) et les contraintes projet (C_p). Les durées des opérations et le temps de cycle total sont liés par des contraintes numériques qui n'impacte pas la définition d'une solution et donc ne sont pas inclus dans V et C . De même, le coût global et les variables de coût aussi bien côté produit que projet sont reliés par des contraintes de coût (C_s). Le système de filtrage permet la mise à jour des domaines de toutes ces variables en tenant compte de l'ensemble des contraintes du modèle. Toutes les variables de V sont discrètes ou symboliques. Les durées et les coûts sont des variables numériques et continues. En conséquence, les contraintes sont : discrètes (C_c), numériques (temps de cycle et coût total) ou mixtes (C_p et C_s). Le filtrage de contraintes discrètes est réalisé par une technique d'arc-consistance conventionnelle (Bessière, 2006) tandis que les contraintes numériques et mixtes sont traitées par une technique de consistance de bornes (Lhomme, 1993).

2.2 Algorithme d'optimisation

La principale spécificité de ce type de problème est un espace de recherche extrêmement important. (Amilhatre et al., 2002) a montré que l'espace de solution en configuration est supérieur à $1.4 \cdot 10^{12}$ pour un problème de configuration d'une voiture. En ajoutant les variables de décision côté projet, l'espace de recherche devient alors encore plus important.

Une autre spécificité de ce problème correspond à l'allure de l'espace de recherche qui n'est pas continue et comporte la plupart du temps de nombreuses singularités. Pour terminer, il s'agit d'un problème d'optimisation multicritères pour lequel un front de Pareto optimal est recherché.

Ces contraintes fortes ont conduit la plupart des études associées vers des approches génétiques ou évolutives : (Hong et al., 2010) ou (Li et al., 2006). Dans cet article, nous reprenons l'algorithme CFB-EA (Constraint Filtering Based Evolutionary Algorithm) proposé lors de nos travaux précédents spécialement pour ce type de problème.

Cet algorithme est basé sur la méthode SPEA2 proposé par (Zizler et al., 2001). Il s'agit d'une des méthodes les plus connues et utilisées, reposant la préservation d'une sélection des meilleurs solutions trouvées dans une archive séparée. Elle utilise une stratégie d'évaluation performante permettant l'obtention d'une population bien répartie sur l'ensemble de l'espace de recherche, ainsi qu'une technique de « clusterisation » de l'archive qui préserve la diversité des solutions du front de Pareto proposées à l'utilisateur. Elle permet donc une convergence rapide tout en préservant la diversité.

Nous avons repris et complété cette méthode afin de pouvoir traiter notre problème contraint. Des opérateurs évolutionnaires spécifiques (initialisation, mutation et croisement) préservant la faisabilité des solutions ont été proposés. Il en résulte une méthode en six étapes :

1. Initialisation de la population en respectant les contraintes du problème (filtrage)
2. Calcul de la performance des solutions de la population (compromis entre la performance au sens de Pareto et de la densité de solutions)
3. Sélection des individus et mise à jour de l'archive
4. Test de la condition d'arrêt
5. Sélection des individus pour la génération suivante (tournoi binaire)
6. Croisement et mutation des individus en respectant les contraintes (par filtrage)
7. Retour à l'étape 2

Pour l'initialisation, le croisement ou la mutation, un gène (variable de décision de V) est choisi et instancié aléatoirement dans son domaine de valeurs. Pour éviter de générer des individus infaisables (c.à.d. ne respectant pas l'ensemble des contraintes), l'ensemble des domaines des variables restantes est mis à jour par filtrage des contraintes. Étant donné que le processus de filtrage n'est pas parfait (c.à.d. qu'un filtrage ne garanti pas la consistance totalement), des solutions non conformes peuvent toute fois être générés. Un processus de retour en arrière limité (backtrack) permet alors de revenir sur les choix précédents afin d'obtenir uniquement des individus faisables dans la population courante. Il est important de noter que nous n'avons pas ajouté de paramètre supplémentaire à régler dans la méthode originale qui en comporte déjà plusieurs. Dans la suite de cette section, nous allons rappeler brièvement les principes et opérateurs utilisés par notre méthode CFB-EA.

De nombreuses recherches sur l'intégration de contraintes dans un processus évolutionnaire ont déjà été menées. C. Coello-Coello propose un état de l'art synthétique dans (Mezura-Montes and Coello Coello, 2011). Les méthodes les plus courantes sont les fonctions de pénalités, le classement stochastique, la ε -contrainte, les concepts multiobjectif, les règles de faisabilité, et les opérateurs spécifiques. CFB-EA appartient à cette dernière catégorie qui comprend : les méthodes de réparation où les individus non faisables sont réparés, les méthodes de préservation de la faisabilité pour lesquelles la consistance est préservée au fur et à mesure de la cons-

truction des individus. Cette classe de méthodes est généralement considéré comme performante pour les problèmes non sur-contraint (c.à.d. lorsque une solution faisable peut être obtenue en un temps raisonnable afin de pouvoir constituer une population de solutions faisables), ce que nous avons confirmé avec nos propres expérimentations.

CFB-EA est basé sur la préservation de la faisabilité des individus lors de leurs constructions ou leurs modifications. Les opérateurs proposés permettent de limiter la recherche aux solutions faisables uniquement. La principale différence entre notre approche et celles existantes est que nous n'avons aucune solution infaisable durant tout le processus d'optimisation. A chaque modification d'un individu, un filtrage permet de vérifier la préservation de la consistance de l'individu.

Nos expérimentations précédentes (Pitiot et al., 2012) nous ont permis de vérifier que les approches exactes type « branch and bound » sont limitées aux cas de petits tailles tandis que CFB-EA est complètement compétitif pour le type de modèle qui nous intéresse (taille et niveau de contraintes). Dans cet étude, nous proposons une nouvelle approche d'optimisation en deux sous-étapes. Cette approche est basée sur trois caractéristiques du problème à résoudre : (i) Les algorithmes évolutionnaires (EA) fournissent un ensemble de solutions (front de Pareto) dès l'initialisation achevée (l'ensemble est ensuite bonifié au fur et à mesure de l'optimisation) ; (ii) nous « disposons » d'un utilisateur qui peut ajuster sa demande selon l'ensemble de solutions proposé en cours d'optimisation (allure du front de Pareto) ; (iii) CFB-EA est adapté au type de problème à résoudre en termes de taille et de niveau de contraintes et plus particulièrement, il est capable de fournir une bonne approximation du front de Pareto en un temps raisonnable permettant ainsi à l'utilisateur de choisir son propre compromis coût/temps de cycle.

2.3 Méthode d'optimisation en deux sous-étapes

Tel qu'exposé lors de l'introduction, le but de notre étude est d'évaluer, pour des problèmes de grandes tailles, l'intérêt de séparer l'étape d'optimisation en deux sous-étapes successives : (i) une première approximation rapide du front de Pareto afin d'avoir une vision globale des compromis coût/temps de cycle possibles, (ii) une deuxième étape d'optimisation restreinte à la ou les zones choisies par l'utilisateur.

Ce processus est illustré par la figure 6. La partie de gauche montre notre précédente approche avec le front de Pareto obtenu en une seule étape d'optimisation longue. La partie de droite montre notre nouvelle méthode en deux étapes : un premier front approximatif obtenu rapidement, suivi d'une seconde étape uniquement sur une zone choisie par l'utilisateur (limitée par *max cost* et *max time*). La zone restreinte pour la deuxième étape est obtenue en ajoutant des contraintes sur les deux critères coût et temps de cycle puis en filtrant ces contraintes supplémentaires dans le modèle complet.

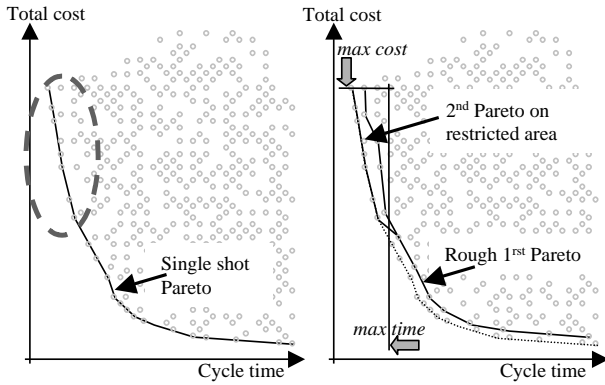


Figure 6 : Principe de l'approche proposée

La seconde étape d'optimisation réutilise les solutions trouvées lors de l'étape précédente lorsqu'elles appartiennent à la zone désirée par l'utilisateur. L'étape d'initialisation de la seconde étape est donc remplacée par la sélection d'un ensemble des meilleures solutions obtenues lors de la première étape.

La méthode complète proposée est donc :

1. Configuration et planification interactive permettant de saisir les besoins non-négociables de l'utilisateur
- 2.1 1^{ère} optimisation globale pour les besoins négociables de l'utilisateur fournissant une approximation du front de Pareto
- 2.2 2^{ème} optimisation sur la ou les zones d'intérêt de l'utilisateur initialisée avec les individus de la première étape d'optimisation

3 EXPÉRIMENTATIONS

3.1 Modèle utilisé et mesure de la performance

L'objectif de ses expérimentations est de comparer les deux approches (une seule ou deux étapes d'optimisation) en termes de qualité du résultat fourni et de temps de calcul nécessaire. En termes de qualité de résultat, il est nécessaire de pouvoir comparer différents front de Pareto. Nous avons opté pour la mesure de l'hypervolume proposé par zitzler dans (Zitzler et Thiele, 1998) illustrée sur la figure 7.

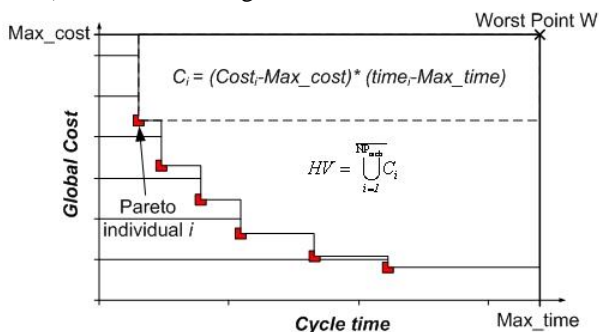


Figure 7 : Définition et illustration de l'hypervolume

L'hypervolume est défini comme l'espace dominé par un ensemble de solutions. Cela permet de mesurer à la fois la convergence et la diversité des solutions en une

seule mesure (le front le plus performant et diversifié va maximiser l'hypervolume qu'il domine). En termes de temps de calcul, nous voulons mesurer, pour un hypervolume donné, le gain de temps permis par la nouvelle approche.

Concernant la taille du problème à résoudre, nous avons utilisé un modèle inspiré de cas réels appelé « full_aircraft ». Il comprend 92 variables (symbolique, entier ou réel) reliées par 67 contraintes (table de compatibilité, équation ou inégalité). Parmi ces variables, 21 correspondent aux variables de décision qui seront manipulées par l'algorithme évolutionnaire (gènes du chromosome) :

- 12 variables décrivant les possibilités de configuration du produit (avec 6 valeurs possibles par variable),
- 9 variables qui vont décrire les possibilités de planification (3 choix de ressources et 3 quantité de ressources soit 9 combinaisons possibles par opération et il y a 9 opérations).

Sans prendre en compte les contraintes, cela représente une combinatoire de l'ordre de 10^{18} combinaisons possibles pour le chromosome d'un individu. Par rapport au niveau de contrainte, dans notre modèle 93 % de ces combinaisons ne sont pas valides, ce qui laisse une combinatoire de $7.3 \cdot 10^{16}$ solutions faisables. Les résultats de nos expérimentations précédentes avec différents niveaux de contraintes et tailles de modèle sont référencés dans (Pitiot et al., 2012).

La figure 8 montre les fronts de Pareto obtenus après 3 et 24 heures de calcul. Le front de Pareto approximatif obtenu en 3 heures permet à l'utilisateur d'avoir une première idée du résultat. Il peut ainsi choisir une zone qui l'intéresse plus particulièrement.

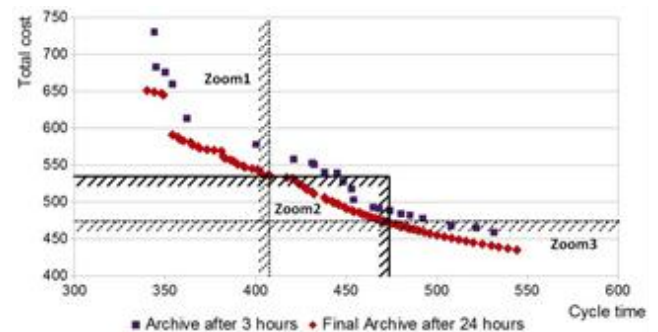


Figure 8 : Front de Pareto obtenus sur le « full_aircraft » après 3h et 24h, ainsi que le découpage en trois zooms

Dans les prochaines sections, nous allons étudier une division en trois zones restreintes :

- « Aircraft_zoom1 » : zone correspondant aux solutions dont le temps de cycle est inférieur à 410 (c.à.d. les solutions avec le temps de cycle les plus courts),
- « Aircraft_zoom2 » : zone correspondant aux solutions dont le temps de cycle est inférieur à 470 et le coût total inférieur à 535 (c.à.d. les solutions de compromis),

- « Aircraft_zoom3 » : zone correspondant aux solutions dont le coût total inférieur à 475 (c.à.d. les solutions de faible coût),

Ces trois zones correspondent à la division du front de Pareto obtenu après 24h de calcul en trois parts égales. Ces zones, choisies pour évaluer la performance de la nouvelle approche en deux étapes d'optimisation, correspondent également à des préférences classiques d'un utilisateur : (i) un avion de faible coût, (ii) un avion disponible rapidement ou (iii) un compromis entre le coût et le temps de cycle. Nous reviendrons sur cet aspect dans la section 3.3.

L'algorithme d'optimisation est codé en langage C++ et le système de filtrage est quant à lui codé en langage Perl. Tous les tests ont été réalisés sur un ordinateur portable disposant du Intel i5 (2.27Ghz, un seul cœur utilisé) et de 2.8 Go de mémoire vive.

3.2 Paramétrage de l'approche évolutionnaire

Pour cette première évaluation de l'approche en deux étapes, nous avons utilisé nos réglages classiques (c.à.d. ceux utilisés pour l'approche en une seule passe : taille de population : 80, taille de l'archive : 100, probabilité de mutation d'un individu : 0.3, probabilité de mutation d'un gène : 0.2, probabilité de croisement : 0.8). La principale différence avec l'approche en une seule étape est la limite de « backtrack » autorisée (c.à.d. le nombre de retours en arrière autorisé lors des croisements et mutations). Cette limite fixée à 100 pour l'approche en une étape, a été fixée à 30 dans l'approche en deux étapes. En effet, la restriction à une zone spécifique entraîne un plus grand nombre de solutions rejetées. Dans l'approche en une seule étape, seuls 2.5% des individus filtrés étaient infaisable, tandis que pour l'approche en deux étapes ce taux monte à près de 7% et 0.3% des individus sont abandonnés (limite de backtrack atteinte) ce qui n'arrivait jamais avec l'autre approche. L'abaissement de la limite de backtrack permet de limiter le temps de calcul dépensé pour tenter de réparer les individus.

L'autre grande différence entre les deux approches est le critère d'arrêt. Dans l'approche en une étape, nous utilisons une limite de temps de calcul fixe (24h) correspondant au temps disponible (décideur). L'approche en deux étapes utilise une condition à deux éléments : soit un arrêt si aucune amélioration n'a été apportée durant deux heures, soit un arrêt selon une limite de temps de 12 heures (auquel il faut ajouter les 3 heures de la première étape pour obtenir le front de Pareto approximatif).

3.3 Résultats expérimentaux

Le but de cette section est d'évaluer l'optimisation en deux étapes sur les trois zones choisies sur la figure 8. (zoom1, zoom2 et zoom3) par rapport à l'optimisation en une seule passe.

3.3.1 Une première illustration

La figure 9 montre un exemple des fronts de Pareto obtenu sur la zone du zoom1 :

- le front de Pareto « approximatif » obtenu en 3 heures de calcul (fig. 9, carrés)
- le résultat de l'approche en deux étapes après 3 + 9 heures de calcul (fig. 9, triangles)
- le résultat de l'approche en une seule étape de 24h (fig. 9, diamants).

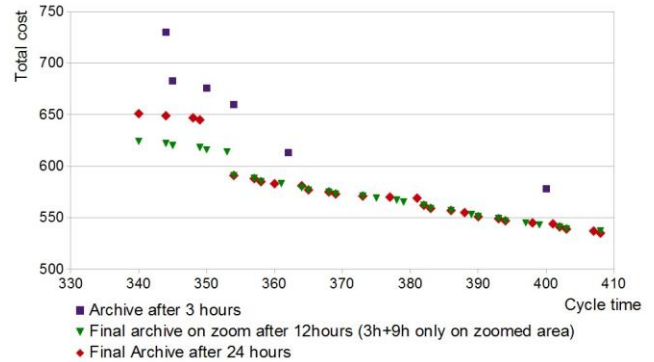


Figure 9 : Exemple de résultats sur le zoom1

Les fronts de Pareto finaux obtenus par les deux approches sont très proches quand le temps de cycle est supérieur à 355. Pour les temps de cycle les plus courts, la méthode en deux étapes est même légèrement meilleure. Ces courbes illustrent bien le résultat attendu, néanmoins elles ne correspondent qu'à un seul calcul. Pour obtenir des conclusions plus significatives, nous allons analyser les résultats obtenus avec dix exécutions différentes.

3.3.2 Comparaison détaillée

Les résultats détaillés obtenus sur les trois zones sont présentés dans le tableau 1 et illustrés sur la figure 10. Sur les graphes de la figure 10, l'axe vertical correspond à l'hypervolume obtenu (moyenne des dix exécutions) et l'axe horizontal représente le temps de calcul en secondes. L'algorithme classique (lignes en pointillés) commence à la première seconde puis après trois heures de calcul (10800 secondes) :

- l'approche en une seule étape continue (lignes en pointillés),
- la seconde étape de l'optimisation en deux étapes est lancée sur la zone restreinte (lignes continues).

Le tableau 1 précise les résultats numériques associés pour chaque zone. Les colonnes correspondent respectivement aux résultats de l'approche en une étape, de l'approche en deux étapes et la différence entre les deux en pourcentage. Sont comparés :

- l'hypervolume final moyen,
- l'écart-type moyen de l'hypervolume final,
- le temps de calcul moyen,
- l'écart-type moyen du temps de calcul,
- la valeur maximale de l'hypervolume parmi les dix exécutions.

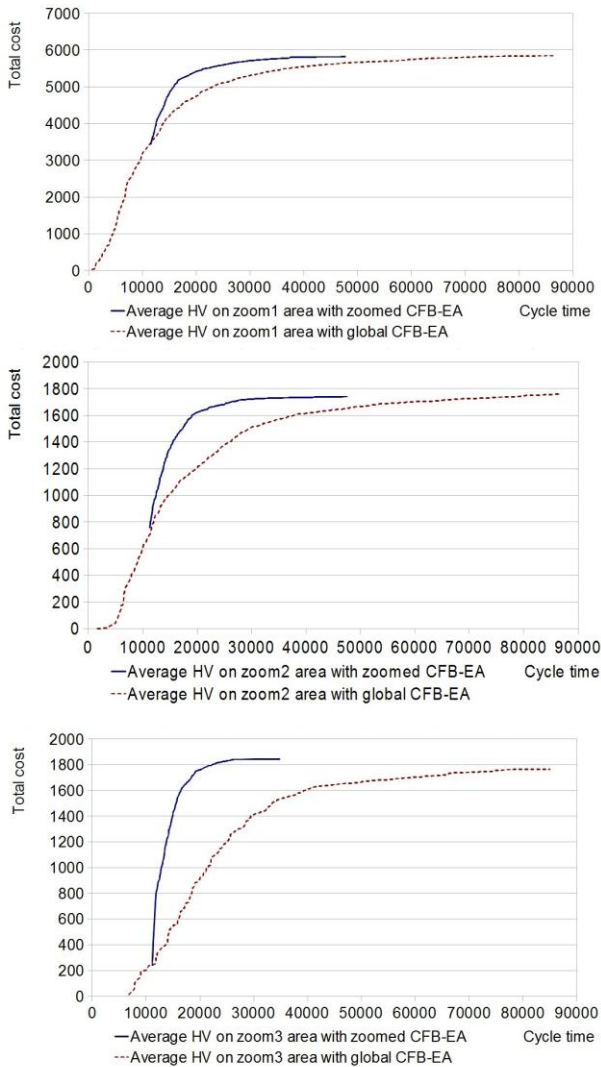


Figure 10 : évolution de l'hypervolume

En termes de qualité de front, la nouvelle approche proposée obtient des résultats similaires à l'approche initiale en une seule étape :

- de 0.4 points inférieur sur le zoom1,
- de 1 point inférieur sur le zoom2,
- de 4 points supérieurs sur le zoom3.

Mais ces résultats similaires sont obtenus en, approximativement, moitié moins de temps :

- 13h contre 24h sur le zoom1,
- 13.5h contre 24h sur le zoom2,
- 10.5h contre 24h sur le zoom3.

Même avec la même consigne (12h ou 24h), les temps de calcul ne sont pas identiques car le critère d'arrêt n'est évalué qu'à la fin de chaque génération.

Ces résultats sont d'autant plus concluants que, dans l'approche en deux étapes, la performance finale est obtenue avant l'arrêt de l'algorithme (cf. critère des deux heures de calcul sans amélioration).

Nous constatons également que l'approche en deux étapes est plus performante sur les zones où l'approche en une seule étape n'a pas obtenu de bons résultats lors des trois premières heures de calcul.

		CFB-EA en 1 étape	CFB-EA en 2 étapes	Écart en %
Zoom1	HV final moyen	5849	5823	-0.4
	ETR moyen	3.8%	5.1%	
	Temps de calcul	86660(24h)	47996 (≈13h)	-44.6
	ETR moyen	0	15%	
	Max HV	6043	6057	0.2
Zoom2		CFB-EA en 1 étape	CFB-EA en 2 étapes	Écart en %
	HV final moyen	1758	1740	-1.
	ETR moyen	2.1%	2.3%	
	Temps de calcul	86660(24h)	48501 (≈13.5h)	-44
	ETR moyen	0	16%	
Zoom3		CFB-EA en 1 étape	CFB-EA en 2 étapes	Écart en %
	HV final moyen	1765	1844	4.4
	ETR moyen	3.16%	0.07%	
	Temps de calcul	86660(24h)	38185 (≈10.5h)	-55.9
	ETR moyen	0	26%	
	Max HV	1831	1845	0,7

Tableau 1 : Comparatif des deux approches pour chaque zones (ETR : écart-type relatif, HV : hypervolume)

Dans la zone du zoom1, l'approche en une étape atteint relativement rapidement un résultat proche du front de Pareto final ; tandis que pour le zoom3, elle en est beaucoup plus loin.

4 CONCLUSIONS

L'objectif de cette étude était d'évaluer les possibilités d'un nouveau principe d'optimisation par étapes successives sur notre problème de configuration / planification conjointes. Nous avons rappelé le contexte de la configuration et planification conjointes et sa modélisation possible par un modèle de type CSP ; puis les principes de notre approche initiale avec notre algorithme d'optimisation appelé CFB-EA. Nous avons ensuite exposé notre proposition pour une optimisation par étapes en utilisant les préférences de l'utilisateur pour restreindre le processus d'optimisation sur sa ou ses zones de prédilection.

Cette approche est basée sur le principe suivant : calculer rapidement un front de Pareto approximatif afin de permettre à l'utilisateur de préciser ses attentes selon le compromis coût/temps de cycle qui se dessine, puis relance l'optimisation uniquement sur les zones d'intérêt sélectionnées.

Selon nos premiers résultats sur un modèle proche de la réalité en termes de taille et de niveau de contraintes, l'approche proposée permet d'économiser environ la moitié du temps de calcul pour un résultat similaire voir meilleur dans certains cas.

Nous travaillons actuellement sur un ensemble de cas de test plus large avec différentes tailles de modèle et différents niveaux de contraintes tel que nous l'avons fait précédemment pour CFB-EA dans (Pitiot et al., 2012). Un autre aspect important restant à évaluer est la façon d'obtenir rapidement le front de Pareto approximatif qu'utilise l'utilisateur pour restreindre l'optimisation lors de la seconde étape.

REFERENCES

- (Aldanondo *et al.*, 2008) M. Aldanondo, E. Vareilles. 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 (2008)
- (Amilhastre et al, 2002) J. Amilhastre, H. Fargier, P. Marquis, Consistency restoration and explanations in dynamic cps - application to configuration, in: *Artificial Intelligence* vol.135, 2002, pp. 199-234.
- (Bartak *et al.*, 2010) R. Barták, M. Salido, F. Rossi. Constraint satisfaction techniques in planning and scheduling, in: *Journal of Intelligent Manufacturing*, vol. 21, n°1, p. 5-15 (2010)
- (Bessiere, 2006) C. Bessiere, *Handbook of Constraint Programming*, Eds. Elsevier, chap. 3 Constraint propagation, 2006, pp. 29-70.
- (Hong *et al.*, 2010) G. Hong, D. Xue, Y. Tu., Rapid identification of the optimal product configuration and its parameters based on customer-centric product modeling for one-of-a-kind production, in: *Computers in Industry* Vol.61 n°3, 2010, pp. 270-279.
- (Junker, 2006) U. Junker. *Handbook of Constraint Programming*, Elsevier, chap. 24 Configuration, p. 835-875 (2006)
- (Laborie, 2003) P. Laborie. Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results, in: *Artificial Intelligence* vol 143, 2003, pp 151-188.
- (Lhomme, 1993) O. Lhomme. Consistency techniques for numerical CSPs, in: *proc. of IJCAI 1993*, pp. 232-238.
- (Li *et al.*, 2006) L. Li, L. Chen, Z. Huang, Y. Zhong, Product configuration optimization using a multiobjective GA, in: *I.J. of Adv. Manufacturing Technology* vol. 30, 2006, pp. 20-29.
- (Mezura-Montes and Coello Coello 2011) E. Mezura-Montes, C. Coello Coello, *Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future*, in: *Swarm and Evolutionary Computation*, Vol. 1 n°4, 2011, pp. 173-194
- (Mittal and Frayman, 1989) S. Mittal, F. Frayman. Towards a generic model of configuration tasks, *proc of IJCAI*, p. 1395-1401(1989).
- (Pitiot *et al.*, 2012) P. Pitiot, M. Aldanondo, E. Vareilles, L. Zhang, T. Coudert. Some Experimental Results Relevant to the Optimization of Configuration and Planning Problems, in : *Lecture Notes in Computer Science* Volume 7661, 2012, pp 301-310
- (Pitiot *et al.*, 2013) P. Pitiot, M. Aldanondo, E. Vareilles, P. Gaborit, M. Djefel, S. Carbonnel, Concurrent product configuration and process planning, towards an approach combining interactivity and optimality, in: *I.J. of Production Research* Vol. 51 n°2, 2013 , pp. 524-541.
- (Schierholt 2001) K. Schierholt. Process configuration: combining the principles of product configuration and process planning *AI EDAM / Volume 15 / Issue 05 / novembre 2001* , pp 411-424
- (Soininen *et al.*, 1998) T. Soininen, J. Tiihonen, T. Männistö, and R. Sulonen, Towards a General Ontology of Configuration., in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* vol 12 n°4, 1998, pp. 357-372.
- (Zhang *et al.*, 2013) L. Zhang, E. Vareilles, M. Aldanondo. Generic bill of functions, materials, and operations for SAP2 configuration, in: *I.J. of Production Research* Vol. 51 n°2, 2013 , pp. 465-478.
- (Zitzler and Thiele 1998) E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms - a comparative case study, in: *proc. of 5th Int. Conf. on parallel problem solving from nature*, Eds. Springer Verlag, 1998, pp. 292-301.
- (Zitzler *et al.*, 2001) E. Zitzler, M. Laumanns, L. Thiele., SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Swiss Fed. Inst. of Technology (ETH), Zurich (2001)