



HAL
open science

SYNTHÈSE DES MÉTHODES INTERVALLES D'OPTIMISATION MULTIOBJECTIFS EN CONCEPTION PRÉLIMINAIRE

Jad Matar, Raphaël Chenouard, Alain Bernard

► **To cite this version:**

Jad Matar, Raphaël Chenouard, Alain Bernard. SYNTHÈSE DES MÉTHODES INTERVALLES D'OPTIMISATION MULTIOBJECTIFS EN CONCEPTION PRÉLIMINAIRE. 9th International Conference on Modeling, Optimization & SIMulation, Jun 2012, Bordeaux, France. hal-00728609

HAL Id: hal-00728609

<https://hal.science/hal-00728609>

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.

SYNTHÈSE DES MÉTHODES INTERVALLES D'OPTIMISATION MULTIOBJECTIF EN CONCEPTION PRÉLIMINAIRE

J. Matar, R. Chenouard, A. Bernard

UNAM / École Centrales de Nantes
IRCCyN / Institut de Recherche en Communications et Cybernétique de Nantes
1 rue de la Noë, BP92101
44321 Nantes cedex 03 - France
jad.matar, raphael.chenouard, alain.bernard@irccyn.ec-nantes.fr

RÉSUMÉ : *Ce travail de recherche correspond à un besoin industriel d'optimisation des problèmes multiobjectifs rencontrés en conception préliminaire. En effet, un problème de conception préliminaire peut majoritairement se représenter à l'aide de variables, de contraintes et de critères à optimiser. Ainsi, les problèmes de conception constituent naturellement des problèmes d'optimisation. Les approches de résolution peuvent être classées en deux catégories: les approches méta-heuristiques ou stochastiques et les approches exactes ou déterministes. Dans cet article, après une brève présentation des méthodes méta-heuristiques existantes, différentes méthodes déterministes de résolution de problèmes multiobjectifs basées sur l'analyse par intervalles sont présentées, comparées et validées expérimentalement sur différents problèmes d'optimisation multiobjectif. L'avantage de ces approches déterministes est de trouver avec certitude l'optimum global du problème à résoudre ainsi que d'éviter toute perte de solutions.*

MOTS-CLÉS : *Optimisation globale, optimisation multiobjectif, optimisation sous contraintes, arithmétique par intervalles, Branch And Bound, conception préliminaire.*

1 INTRODUCTION

Les problèmes de conception préliminaire peuvent naturellement être interprétés comme des problèmes d'optimisation. Ces derniers sont rarement mono-objectifs et nécessitent la prise en compte de plusieurs critères conflictuels. La prise en compte dans un problème d'optimisation de plusieurs objectifs contradictoires amplifie encore le degré de difficulté de sa résolution. En effet, l'objectif de la conception préliminaire est de choisir les concepts et composants d'un produit pour le dimensionner (Pahl and Beitz, 1996). Lors de cette phase du processus de conception, les composants d'un produit sont identifiés par des variables de conception correspondant directement à des variables de décision d'un problème d'optimisation. Les critères issus du cahier des charges peuvent généralement être pris en compte comme des objectifs à optimiser ou des inéquations bornant la zone de faisabilité du problème à résoudre. Les autres contraintes restreignant l'espace de faisabilité du problème d'optimisation sont alors

naturellement les relations algébriques traduisant la physique régissant le comportement du produit.

Ce travail d'état de l'art se situe sur la résolution des problèmes d'optimisation multiobjectif à variables continues avec contraintes. Les problèmes de conception comportent généralement des variables discrètes (données de catalogue, choix de dimensions standardisées...), mais par soucis de simplicité nous ne les considérons pas dans cet article, même si les méthodes présentées peuvent très bien les gérer. La résolution de ces problèmes consiste à trouver la ou les meilleures solutions vérifiant un ensemble de contraintes et minimisant un ensemble d'objectifs.

Après une brève présentation des différentes techniques de résolution stochastiques et déterministes, nous montrerons l'intérêt de l'utilisation des méthodes d'optimisation déterministe globale basées sur l'analyse par intervalles.

En effet, l'analyse par intervalles apporte au concepteur durant la phase de conception préliminaire une

vue ensembliste des solutions. De plus, elle permet d'encadrer la totalité des solutions du problème de conception au lieu de se restreindre à un nombre limité de solutions. Ainsi, les méthodes d'optimisation globale étudiées sont basées sur l'analyse par intervalles intégrée au sein des algorithmes de type *Branch And Bound* multiobjectifs. Le principe de ces algorithmes est de découper l'espace initial de recherche en sous espaces de plus en plus restreints afin d'isoler ceux encadrant les optima globaux. Une étude détaillée de ces algorithmes déterministes ainsi qu'une synthèse comparative entre eux sont présentées dans cet article.

Enfin, nous avons mis en œuvre un ensemble de ces méthodes déterministes basées sur l'analyse par intervalles dans un solveur existant de programmation par contraintes (Granvilliers et Benhamou, 2006a) (Benhamou et Granvilliers, 2006b) et nous montrerons des résultats obtenus sur des problèmes présents dans la littérature.

2 CONTEXTE

Un état de l'art effectué sur les méthodes de résolution des problèmes multiobjectifs nous mène à classer ces méthodes en deux catégories: les approches stochastiques principalement basées sur les méta-heuristiques et les approches exactes ou déterministes.

2.1 Problèmes d'optimisation multiobjectif

La formulation, la plus souvent utilisée, d'un problème d'optimisation multiobjectif est la suivante:

$$\begin{array}{lll}
 \text{Minimiser} & \mathbf{F}(\mathbf{X}) & (m \text{ fonctions à optimiser}) \\
 \text{tel que} & \mathbf{G}(\mathbf{X}) \leq 0 & (q \text{ contraintes} \\
 & & \text{à satisfaire}) \\
 & \mathbf{H}(\mathbf{X}) = 0 & (p \text{ contraintes de plus} \\
 & & \text{à satisfaire}) \\
 \text{avec} & \mathbf{X} \in \mathbb{R}^n, \quad \mathbf{F}(\mathbf{X}) \in \mathbb{R}^m, \mathbf{G}(\mathbf{X}) \in \mathbb{R}^q, \\
 & & \mathbf{H}(\mathbf{X}) \in \mathbb{R}^p.
 \end{array}$$

\mathbf{X} représente alors l'ensemble des variables de décision du problème de conception préliminaire et \mathbf{F} l'ensemble des fonctions objectifs à optimiser.

2.2 Méthodes de résolution

La résolution d'un problème d'optimisation multiobjectif consiste alors à trouver les solutions minimisant ou maximisant un ensemble d'objectifs et satisfaisant un ensemble de contraintes. Les approches les plus rencontrées dans la littérature pour résoudre les problèmes d'optimisation multiobjectif sont des approches stochastiques évolutionnaires notamment les algorithmes génétiques (Deb, 2001). Un

algorithme génétique est typiquement composé de trois éléments fondamentaux: Une population constituée de plusieurs individus représentant des solutions potentielles du problème donné; Un mécanisme d'évaluation des individus permettant de mesurer l'adaptation de l'individu à son environnement; Un mécanisme d'évolution de la population permettant, grâce à des opérateurs prédéfinis, d'éliminer certains individus et d'en créer de nouveaux. Ces algorithmes évolutionnaires dits stochastiques ou non déterministes, débouchent (si les paramètres sont bien fixés) sur l'obtention de *bonnes* solutions. Néanmoins, celles ci ne sont pas forcément des optimums globaux pour deux raisons:

- L'exploration de l'espace de recherche est basé sur une procédure aléatoire
- L'algorithme s'arrête suivant un critère d'arrêt, qui bien souvent ne garantit pas cette optimalité globale

En effet, le nombre de solutions retournées par les techniques évolutionnaires correspond à la taille de la génération fixée en amont ou au nombre fini d'individus non dominés d'une archive (Zitzler, 1999). Par conséquent, ces approches consistent à calculer un nombre fini de solutions dans un front de Pareto qui, par nature, contient une infinité de solutions. En réalité, la pertinence d'une solution par rapport à une autre dépend d'un ensemble de facteurs incluant le choix du concepteur et le contexte du problème. Pour cela, il serait très avantageux d'encadrer la totalité des solutions du front de Pareto au lieu de se restreindre à un nombre limité de solutions et d'avoir une vue un peu plus globale de l'espace de faisabilité du problème posé.

Cependant, nous n'avons pas toujours besoin d'une représentation ou d'une formulation mathématique des problèmes pour pouvoir les résoudre. Nous pouvons alors utiliser des modèles boîtes noires pour évaluer les contraintes comme les objectifs, par exemple nous pouvons faire appel à des outils de simulation, dont les paramètres sont définis par les caractéristiques des individus. Ainsi, les principes de base d'un algorithme génétique (GA) n'exploitent pas les caractéristiques mathématiques d'un problème, ce qui permet de traiter des problèmes hétérogènes, sans autres difficultés que celles propres au problème.

La deuxième catégorie de méthodes rencontrées dans la littérature pour résoudre les problèmes d'optimisation multiobjectif correspond aux méthodes déterministes souvent basées sur l'analyse par intervalles. Ces méthodes détaillées dans la section 3 sont généralement intégrées au sein d'un algorithme de type *Branch And Bound*. Le principe général de ces algorithmes est de découper (*Branch*)

l'espace initial de recherche en sous-espaces de plus en plus restreints afin d'isoler ceux encadrant les optima globaux. L'algorithme de recherche forme ainsi un arbre dont chaque nœud représente une partie de l'espace de recherche dont l'union forme l'espace initial. Ensuite chaque nœud est évalué de façon à déterminer sa borne (*Bound*) inférieure (inférieure dans le cas de minimisation et supérieure dans le cas de maximisation) ou de l'éliminer en fonction des fonctions objectifs.

2.3 Discussion

Dans le cadre du traitement d'un problème de conception, une méthode de résolution adaptée doit permettre de répondre à la question suivante:

- Comment offrir au concepteur un bon aperçu de l'espace de faisabilité en encadrant l'ensemble des solutions optimales?

Les méthodes déterministes d'optimisation basées sur l'analyse par intervalles nous semblent répondre à ces questions. En effet, l'exploration exhaustive de l'espace de recherche nous garantit d'obtenir une totalité de solutions optimales et globales du problème. De plus, les erreurs engendrées par le calcul numérique sont contrôlées par l'arithmétique par intervalles intégrée au sein des méthodes déterministes étudiées. Cette arithmétique par intervalles, contrairement à la norme IEEE-754 de l'arithmétique flottante, a pour but de ne plus manipuler des nombres qui approchent plus ou moins fidèlement une valeur, mais des intervalles encadrant cette valeur. Ainsi, on peut remplacer une valeur non exactement représentable, telle que π , par un intervalle la contenant. Par ailleurs, les critères de convergence (ou d'arrêt) à définir à un GA ainsi que son paramétrage ne sont souvent pas intuitifs et sont spécifiques à chaque problème.

Par conséquent, nous nous sommes concentrés sur les méthodes déterministes intégrant l'arithmétique des intervalles en sachant que de telles garanties se payent en temps de calcul. Pour cela, nous commençons par une brève présentation de l'arithmétique des intervalles. Ensuite nous détaillerons dans la section 3 les méthodes déterministes multiobjectifs classées en deux catégories; méthodes directes et méthodes inverses.

2.4 Arithmétique des intervalles

L'arithmétique par intervalles a été développée par Ramon Moore (Moore 1966). En effet, tout calcul par intervalles est un calcul garanti, c'est-à-dire que le résultat calculé est un intervalle qui garantit de contenir la valeur ou l'ensemble de valeurs cherché.

Intervalles

L'arithmétique par intervalles, contrairement à la norme IEEE-754 de l'arithmétique flottante, a pour but de ne plus manipuler des nombres qui approchent plus ou moins fidèlement une valeur, mais des intervalles encadrant cette valeur. Par exemple, on peut tenir compte d'une erreur de mesure en remplaçant une valeur mesurée x avec une incertitude ϵ par l'intervalle $[x - \epsilon, x + \epsilon]$. De plus, on peut remplacer une valeur non exactement représentable, telle que π , par un intervalle la contenant ; si l'on dispose d'un ordinateur représentant les nombres en base 10 avec 3 chiffres, π sera remplacée par $[3.14, 3.15]$. Enfin, si l'on désire obtenir un résultat valide pour tout un ensemble de valeurs, on utilise un intervalle contenant ces valeurs.

Les intervalles sont des sous-ensembles fermés connexes de \mathbb{R} . On notera \mathbb{IR} l'ensemble des intervalles de \mathbb{R} .

On peut les généraliser en vecteur intervalle ou boîte $\mathbf{X} \in \mathbb{IR}^n$ qui est un vecteur dont les n composantes sont des intervalles. Une représentation graphique de vecteur de \mathbb{IR} , \mathbb{IR}^2 et \mathbb{IR}^3 est donnée dans la figure 1. Elle illustre le fait qu'un vecteur intervalle ou boîte est un ensemble parallélépipédique de vecteurs aux côtés parallèles aux axes du repère ; cela justifie l'indifférence entre les termes de vecteur intervalle et de boîte.

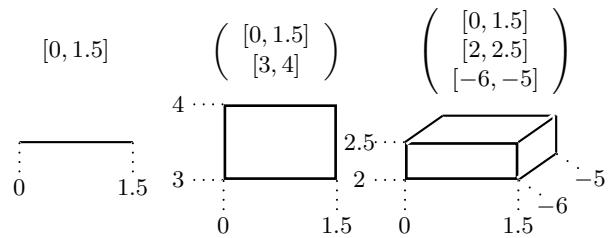


Figure 1: Exemples de vecteurs d'intervalles de dimensions 1, 2 et 3

Notations. Les boîtes sont représentées par des caractères gras et majuscules: \mathbf{X} . Les intervalles sont désignés en majuscules: X . On notera \underline{X} le minimum de X et \overline{X} son maximum. Alors on a $X = [\underline{X}, \overline{X}]$. Enfin, le centre $mid(\mathbf{x})$ est défini par $mid(\mathbf{x}) = (\underline{x} + \overline{x})/2$.

Une fonction d'inclusion $(F)(\mathbf{X})$ est une fonction de réels $f(x)$ étendue sur les intervalles. Ainsi $(F)(\mathbf{X})$ permet d'effectuer une évaluation intervalle de \mathbf{X} pour obtenir un encadrement de la fonction sur les réels $f(x)$, avec $x \in \mathbf{X}$.

Opérations arithmétiques. Le résultat d'une opération arithmétique entre deux intervalles $X \diamond Y$

est le plus petit intervalle au sens de l'inclusion contenant

$$X \diamond Y = \{x \diamond y | x \in X, y \in Y\},$$

c'est-à-dire le plus petit intervalle contenant tous les résultats possibles de l'opération appliquée à tous les éléments x de X et tous les éléments y de Y . En appliquant cette définition aux opérations arithmétiques $=, -, \times, ^2, /$ ou $\sqrt{\quad}$, on obtient les formules suivantes:

$$\begin{aligned} [\underline{X}, \overline{X}] + [\underline{Y}, \overline{Y}] &= [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}] \\ [\underline{X}, \overline{X}] - [\underline{Y}, \overline{Y}] &= [\underline{X} - \overline{Y}, \overline{X} + \underline{Y}] \\ [\underline{X}, \overline{X}] \cdot [\underline{Y}, \overline{Y}] &= [\min(\underline{X}\underline{Y}, \underline{X}\overline{Y}, \overline{X}\underline{Y}, \overline{X}\overline{Y}), \\ &\quad \max(\underline{X}\underline{Y}, \underline{X}\overline{Y}, \overline{X}\underline{Y}, \overline{X}\overline{Y})] \\ [\underline{X}, \overline{X}]^2 &= [\min(\underline{X}^2, \overline{X}^2), \max(\underline{X}^2, \overline{X}^2)] \\ &\quad \text{si } 0 \notin [\underline{X}, \overline{X}] \\ &\quad \text{et } [0, \max(\underline{X}^2, \overline{X}^2)] \text{ sinon} \\ 1/[\underline{Y}, \overline{Y}] &= [\min(1/\underline{Y}, 1/\overline{Y}), \max(1/\underline{Y}, 1/\overline{Y})] \\ &\quad \text{si } 0 \notin [\underline{Y}, \overline{Y}] \\ [\underline{X}, \overline{X}]/[\underline{Y}, \overline{Y}] &= [\underline{X}, \overline{X}] \times (1/[\underline{Y}, \overline{Y}]) \text{ si } 0 \notin [\underline{Y}, \overline{Y}] \\ \sqrt{[\underline{X}, \overline{X}]} &= [\sqrt{\underline{X}}, \sqrt{\overline{X}}] \text{ si } 0 \leq \overline{X} \end{aligned}$$

Propriétés algébriques. On peut constater que les opérations définies ci-dessus ne présentent pas les propriétés algébriques de leurs contreparties sur les réels. Tout d'abord, la soustraction n'est pas la réciproque de l'addition. Par exemple, si $X = [2, 3]$, $X - X = [2, 3] - [2, 3] = [-1, 1] \neq 0$ même si 0 appartient à l'intervalle obtenu. En effet, $X - X = \{x - y | x \in X, y \in X\} \supset \{x - x | x \in X\} = \{0\}$ et l'inclusion est stricte. Idem pour la division qui n'est pas la réciproque de la multiplication. De plus, la multiplication d'un intervalle par lui-même n'est pas égale à l'élevation au carré : si $X = [-3, 2]$, $X \times X = [-3, 2] \times [-3, 2] = [-6, 9]$ alors que $X^2 = \{x^2 | x \in X\} = [0, 9]$.

2.5 Relations de dominance

Par la suite, nous ferons la distinction entre un espace de recherche \mathbf{X} qui est le produit cartésien des intervalles correspondant aux variables de décision du problème et un espace des objectifs \mathbf{Y} qui correspond à l'image d'un espace de recherche au regard des fonctions objectifs d'inclusion. Une fonction objectif d'inclusion, comme vue précédemment, est une fonction objectif des réels étendue sur les intervalles. L'intérêt alors d'une fonction objectif d'inclusion $[f](\mathbf{X})$, avec $[f] = ([f_1], [f_2], \dots, [f_n])$, est d'effectuer une évaluation intervalle pour obtenir un encadrement des fonctions objectifs réelles $f_i(x)$ du problème, avec $x \in \mathbf{X}$ et $i = 1, \dots, n$.

Par ailleurs, une solution optimale obtenue durant le processus d'optimisation est présentée sous forme

d'une boîte dans l'espace de décision (englobant un ensemble de paramètres optimaux) et sa projection dans l'espace des objectifs. Cette boîte-solution fait partie d'un ensemble de boîtes-solutions retenues et fournies à la fin du processus de résolution. Les solutions non retenues (ou non optimales) quant à elle ont été écartées grâce aux relations de dominance. Il est alors indispensable de définir ces relations de dominance qui sont établies dans l'espace des objectifs. En multiobjectifs, plusieurs relations de dominance ont été proposées comme la dominance au sens de Geoffrion (Ehrgott, 2000) et la cône dominance (Collette et Siarry, 2002). En revanche, la relation dominance la plus célèbre est la dominance au sens de Pareto définie sur les points.

L'arithmétique des intervalles consiste à manipuler des boîtes étant un ensemble d'intervalles. Vue que nous cherchons à définir les relations de dominance entre ces boîtes, il est vital tout d'abord de définir la relation d'ordre entre les intervalles eux-même. Alors, soient U et V deux intervalles:

$$\begin{aligned} U = V &\quad \text{ssi} \quad \underline{U} = \underline{V} \text{ et } \overline{U} = \overline{V} \\ U < V &\quad \text{ssi} \quad \overline{U} < \underline{V} \end{aligned}$$

Une relation de dominance au sens de Pareto étendue aux intervalles peut être définie comme suit: Considérons un problème multiobjectif de minimisation. Soient $\mathbf{U} = \{U_1, \dots, U_n\}$ et $\mathbf{V} = \{V_1, \dots, V_n\}$ deux boîtes ou deux vecteurs d'intervalles dans l'espace des objectifs;

$$\begin{aligned} \mathbf{U} \prec \mathbf{V} &\quad \text{ssi} \quad U_i < V_i, \\ (\mathbf{U} \text{ domine certainement } \mathbf{V}) &\quad \text{ssi} \quad \text{avec } i = 1, \dots, n \\ \mathbf{U} \approx \mathbf{V} &\quad \text{ssi} \quad \exists U_i \not< V_i, \\ (\mathbf{U} \text{ incomparable avec } \mathbf{V}) &\quad \text{ssi} \quad \text{avec } i = 1, \dots, n \end{aligned}$$

Illustrons ces relations de dominance entre boîtes par des exemples en 2 dimensions (figures 2 et 3). La boîte \mathbf{U} (figure 2), par exemple, domine toutes les boîtes incluses dans sa zone de préférence et est dominée par toutes les boîtes incluses dans sa zone de dominance ($\mathbf{U} \prec \mathbf{V}$ et $\mathbf{W} \prec \mathbf{U}$). Sur la figure 3, nous représentons les zones où une boîte est incomparable avec une autre boîte intersectant ces zones. Prenons à titre d'exemple la boîte \mathbf{U} de la figure 3, \mathbf{U} est incomparable à \mathbf{V} et à \mathbf{W} ($\mathbf{U} \approx \mathbf{V}$ et $\mathbf{U} \approx \mathbf{W}$). En effet, il se peut que la boîte \mathbf{W} englobe une solution p_w non dominée au sens de Pareto par rapport à la solution potentielle p_u de la boîte \mathbf{U} . De ce fait, toute boîte intersectant les zones "incomparables" (figure 3) d'une boîte \mathbf{U} ne peut être considérée comme dominée par cette boîte.

3 APPROCHES DÉTERMINISTES PAR INTERVALLES

La plupart des méthodes déterministes repérées dans la littérature sont basées sur l'analyse par intervalles

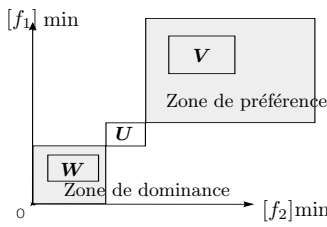


Figure 2: Zones de préférence et de dominance d'une boîte U donnée

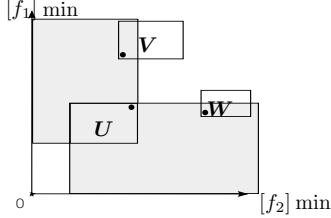


Figure 3: En gris les zones "incomparables" d'une boîte U donnée

et généralement intégrées au sein d'un algorithme de type *Branch And Bound*. Nous classons ces méthodes déterministes multiobjectifs en deux catégories; méthodes directes et méthodes inverses.

3.1 Méthodes directes

Le principe général des algorithmes des méthodes directes est comme suit; un espace de recherche (ou de décision) est découpé en plusieurs sous espaces pour obtenir leurs images dans l'espace des objectifs. L'image, comme vue ci-dessus, est la projection d'un sous espace de recherche vers l'espace des objectifs aux regard des fonctions objectifs d'inclusion du problème.

3.1.1 Approche naïve (MOIBBA)

Une approche naïve MOIBBA *Multi-Objective Interval Branch And Bound* basée sur l'analyse par intervalles pour résoudre les problèmes d'optimisation multiobjectif consiste tout d'abord à découper l'espace de recherche en deux sous espaces (ou boîtes) et stocker ces deux boîtes dans une liste L . À chaque itération, une boîte de \mathbf{X} est choisie, les contraintes sont propagées et la nouvelle boîte contractée (ou non) est évaluée par rapport aux objectifs. Ce processus est répété à chaque itération jusqu'à ce que la boîte choisie valide un test d'obtention de solution. Dans notre cas, le test d'obtention d'une boîte solution consiste à vérifier si une boîte \mathbf{X}_ϵ atteint une précision ϵ définie par l'utilisateur. Une fois \mathbf{X}_ϵ obtenue, un test de dominance est effectué sur l'image $\mathbf{F}(\mathbf{X}_\epsilon)$ de cette boîte solution: si $\mathbf{F}(\mathbf{X}_\epsilon)$ n'est pas strictement dominée au sens de Pareto par une des

autres boîtes déjà traitées alors elle est insérée dans une liste S puis une mise à jour est effectuée sur les listes S et L en retirant toutes les boîtes dominées par cette solution, sinon \mathbf{X}_ϵ est éliminée. Ce processus est répété à chaque exploration et tant qu'il existe des boîtes dans la liste L n'ayant pas atteint la précision ϵ . Après ce traitement, nous sommes assurés que l'union des boîtes présentes dans la liste S contient l'intégralité du front de Pareto optimal grâce aux bonnes propriétés de l'arithmétique par intervalles. Pour être rigoureux, il faudra faire une preuve d'existence de solution dans les boîtes finales obtenues. Or, en conception préliminaire cette preuve n'est pas nécessaire vue l'approximation déjà effectuée en modélisant le système réel.

L'algorithme suivant reprend toutes les étapes de l'approche MOIBBA décrite ci-dessus:

MOIBBA ($\mathbf{X}_{initial}, C, [f]$).

$L \leftarrow \mathbf{X}_{initial}$
Tant que $L \neq \emptyset$ **faire**
 $\mathbf{X} \leftarrow \text{extract}(L)$
 Propager les contraintes C sur \mathbf{X}
 Si \mathbf{X} "contractée" n'a pas atteint ϵ **alors**
 $(\mathbf{X}_1, \mathbf{X}_2) \leftarrow \text{split}(\mathbf{X})$
 $L \leftarrow L + \{\mathbf{X}_1, \mathbf{X}_2\}$
 Sinon
 $\mathbf{Y} = [f](\mathbf{X})$
 Si une boîte de $S \prec \mathbf{Y}$ **alors**
 Rejeter \mathbf{X}
 Sinon
 Mettre à jour S et L en retirant les boîtes dominées (\prec) par \mathbf{Y}
 $S = S + \mathbf{X}$
 FinSi
FinSi
FinTantQue

La partie la plus coûteuse d'un tel algorithme est la mise à jour effectuée sur la liste de recherche L . Pour cela, MOIBBA peut être amélioré en ajoutant une nouvelle heuristique qui consiste à ne plus effectuer cette mise à jour sur la totalité des boîtes présentes dans la liste L mais seulement sur les boîtes non testées depuis la dernière mise à jour.

Une autre approche ayant comme but d'améliorer l'exploration de boîtes dans l'espace de recherche nommé *Direct-comparison Method* a été proposée dans (Ruetsch, 2005). Cette nouvelle technique d'exploration s'effectue d'une façon similaire à un *Branch And Bound* standard. En effet, elle consiste à représenter l'espace de recherche sous forme d'un arbre binaire puis à parcourir cet arbre en largeur d'abord. La construction de cet arbre s'effectue en commençant par une boîte de recherche initiale. Cette boîte est découpée pour obtenir deux boîtes-filles formant le premier niveau de l'arbre. Ensuite,

chaque boîte d'un niveau donné est découpée en deux boîtes-filles qui formeront le niveau suivant de l'arbre. L'élimination de boîtes en se basant sur le critère de dominance de Pareto se passe à chaque niveau généré; si une boîte est dominée par une autre boîte du même niveau alors la boîte dominée ne produit plus de fils autrement dit ne sera plus découpée. Ce processus est répété à chaque génération de niveau jusqu'à ce que toutes les boîtes d'un niveau aient atteint une précision ϵ donnée. Les feuilles de l'arbre ayant toujours le droit de produire des fils et ayant atteint une précision ϵ engloberont l'intégralité des solutions sur le front de Pareto. Enfin, le test de dominance d'une boîte par rapport à une autre boîte d'un même niveau est optimisée et s'effectue comme suit: une boîte \mathbf{X}_1 est comparée à une autre boîte \mathbf{X}_2 d'un même niveau si et seulement si la boîte \mathbf{X}_2 a un père qui intersecte le père de la boîte \mathbf{X}_1 ou qui intersecte le champs de dominance du père de la boîte \mathbf{X}_1 . Le but de cette gestion est de réduire le test de comparaison coûteux entre toutes les boîtes du même niveau.

Afin d'accélérer ces algorithmes déterministes, nous retrouvons dans la littérature différentes méthodes basées sur les gradients des fonctions objectifs. Ces méthodes ont pour but de rejeter des sous espace de recherche, ne contenant pas d'optimum global, le plus tôt possible. La méthode la plus simple, appelée test de monotonie, consiste à supprimer une boîte ou un sous-espaces où toutes les fonctions objectifs sont monotones. De plus, elles doivent être toutes décroissantes ou croissantes. Si les fonctions objectives sur une boîte dans l'espace de recherche donnée sont monotones mais les unes croissantes et des autres décroissantes alors cette boîte peut contenir un optimum global au sens de Pareto et ne doit pas être supprimée. Par ailleurs, Reutsch a intégré à son algorithme déterministe *Direct-comparison Method*, vu précédemment, une nouvelle approche appelée formulation différentielle par intervalles (*Interval Differential Formulation*) décrite dans le paragraphe suivant.

3.1.2 Formulation différentielle par intervalles

Ruetsch propose une nouvelle méthode *Interval Differential Formulation* (Ruetsch, 2010) consistant à accélérer les algorithmes déterministes multiobjectifs intégrant l'analyse par intervalles. Cette méthode, basée sur les gradients des fonctions objectifs, permet de déterminer une direction de recherche où toutes les fonctions objectifs sont améliorées. L'idée est la suivante: pour chaque point x de l'espace de recherche, le gradient ∇f d'une fonction objective f définit une région où la fonction f diminue en partant de ce point (à gauche de la figure 4). Cette région est appelée la région de gradient négatif et indiquée par N_k . Un point n'est pas alors optimal au sens de Pareto si

les régions de gradient négatif de toutes les fonctions objectifs ont une intersection non vide (une intersection entre elles et entre l'espace de recherche \mathbf{X}): $\bigcap_{k=1}^N N_k \cap \mathbf{X} \neq \emptyset$.

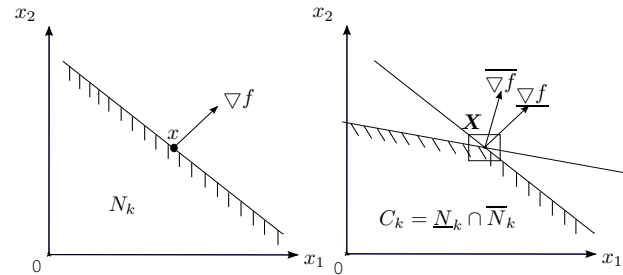


Figure 4: Région de gradient négatif. À gauche l'approche classique, à droite l'approche intervalle

La formulation différentielle en intervalles consiste à calculer pour chaque intervalle d'une boîte (ou d'un sous-espace) donnée la borne inférieure et supérieure du gradient $[\underline{\nabla f}, \overline{\nabla f}]$. Ainsi, la région de gradient négatif est $C_k = \underline{N}_k \cap \overline{N}_k$ représentée sur la droite de la figure 4. Si toutes les régions C_k des différentes fonctions objectifs ont une intersection non nulle alors nous concluons qu'il n'existe pas de solutions Pareto-optimales dans la boîte en question et elle peut être rejetée en cours de résolution. Cette méthode, comme la méthode de monotonie basée sur les gradients, permet d'accélérer l'algorithme de résolution en rejetant au plus tôt les boîtes ne contenant pas d'optima globaux.

3.1.3 Problème de recouvrement

La dominance au sens de Pareto citée ci-dessus (section 2.5) d'une boîte par rapport à une autre est définie dans l'espace des objectifs. Prenons l'exemple de la figure 5 comme illustration du problème de recouvrement. La boîte \mathbf{U} domine au sens de Pareto (\prec) la boîte \mathbf{O} parce que les intervalles de la boîte \mathbf{U} correspondant ici aux objectifs f_1 et f_2 du problème sont strictement plus petits que ceux de la boîte \mathbf{O} . De la même manière, la boîte $\mathbf{V} \prec \mathbf{P}$.

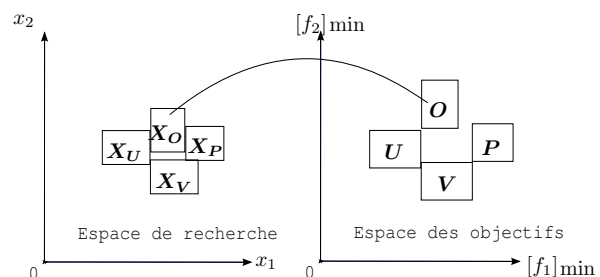


Figure 5: Exemple de dominance au sens de Pareto

Par conséquent, les solutions Pareto optimales sont

encadrées par les boîtes U et V qui ne sont pas dominées l'une par l'autre au sens de Pareto. Les boîtes O et P seront alors éliminées ainsi que leurs images inverses respectives X_O et X_P . En réalité, les boîtes obtenues dans l'espace des objectifs sont issues des fonctions d'inclusion. Ces fonctions d'inclusion fournissent un encadrement de l'image directe d'une boîte de l'espace de recherche. Cet encadrement génère souvent un chevauchement ou recouvrement des boîtes dans l'espace des objectifs (figure 6). Ce chevauchement ne permet pas de déterminer la dominance dans certains cas.

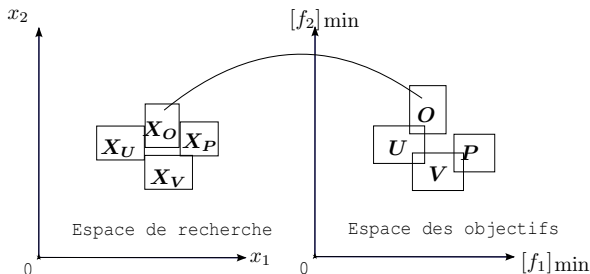


Figure 6: Exemple de chevauchement

Prenons, par exemple, le chevauchement entre les deux boîtes U et O de l'espace des objectifs; $U \approx O$. Ainsi, nous ne pouvons pas éliminer la boîte O en n'ayant aucune garantie qu'une solution optimale globale ne fasse pas partie de cette dernière et ne soit pas issue de X_O . De ce fait, nous nous retrouvons avec un large front de Pareto contenant un nombre irréductible de boîtes qui se chevauchent. Kubica et Wozniak (2007) se sont intéressés à ce problème en proposant une nouvelle approche de résolution consistant à explorer (ou découper) dans l'espace des objectifs au lieu de l'espace de recherche. Cette approche nommée IMMT fera l'objet de la section suivante et rentrera dans le cadre des méthodes inverses de résolution.

3.2 Méthodes inverses

La différence entre les méthodes inverses et les méthodes classiques (ou directes) de résolution est dans l'espace d'exploration. Les méthodes directes, comme vu précédemment, consistent à explorer les boîtes dans l'espace de recherche pour obtenir leurs images dans l'espace des objectifs. Les méthodes inverses, à partir d'une boîte dans l'espace des objectifs, définissent son image inverse dans l'espace de recherche (ou l'espace de décision). Seul Kubica, à notre connaissance, propose une telle méthode. En effet, son algorithme consiste dans un premier temps à effectuer une première évaluation de l'espace initial de recherche X pour obtenir un espace initial des objectifs Y . Ensuite il s'agit de découper l'espace des objectifs Y en deux sous espaces Y_1 et Y_2 , puis de choisir

une des deux boîtes et trouver son évaluation inverse dans l'espace de recherche. Cette évaluation inverse fait appel à l'algorithme SIVIA (Jaulin et al., 2001). SIVIA prend en entrée la boîte Y_i et un espace initial de recherche à explorer et renvoie en sortie deux listes de boîtes de recherche L_{in} et L_{bound} correspondant à l'évaluation inverse de Y_i . L_{in} ; une liste contenant les boîtes de recherche ayant une évaluation dans l'espace des objectifs incluse dans Y_i et L_{bound} ; une liste contenant les boîtes de recherche de précision ϵ_x ayant une évaluation dans l'espace des objectifs qui intersecte Y_i . Enfin, l'opérateur *Componentwise Newton Operator* (Kubica et Malinowski, 2005) et un test de monotonie (Hansen 1992) sont utilisés pour accélérer la résolution. Ils consistent à éliminer au plus tôt les boîtes où en améliorant une fonction objectif, on peut au moins améliorer une des autres fonctions objectifs.

Nous avons apporté quelques améliorations à l'algorithme de Kubica en intégrant une nouvelle heuristique de choix de Y_i à explorer et une gestion différente des listes des sous espaces déjà explorés. L'heuristique de choix consiste à choisir, à chaque itération, la boîte Y_i la plus prometteuse, i.e. celle minimisant au moins un des objectifs. Cette heuristique nous permet de nous diriger plus rapidement vers une solution Pareto optimale et donc de créer moins de branches dans notre arbre de recherche. Concernant, la gestion des espaces de recherche déjà explorés, nous avons ajouté, en plus de la liste des sous-espaces à explorer, une liste de sous-espace-solutions, c'est-à-dire les boîtes solutions ayant atteint une précision ϵ_y précisée par l'utilisateur. Cette nouvelle gestion de sous-espace nous permet d'effectuer moins de parcours sur les sous-espaces déjà explorés et moins de mises à jour sur ces derniers.

4 EXPÉRIMENTATION

Nous présentons les résultats des approches MOIBBA (*Multi-Objective Interval Branch And Bound*), ID-CDF (*Interval Differential Formulation and Direct Comparison Method*) et IMMT (*Inverse Method and Monotonicity Test*) sur deux problèmes souvent utilisés dans l'optimisation multiobjectif.

Le premier problème SRN issu de (Deb et al., 2002) :

$$\begin{aligned} & \text{Min } (f_1, f_2) ; \\ & f_1 = (x_1 - 2)^2 + (x_2 - 1)^2 + 2, \\ & f_2 = 9 * x_1 - (x_2 - 1)^2; \\ & \text{tel que : } x_1^2 + x_2^2 \leq 225 \text{ et } x_1 - (3 * x_2) \leq -10; \\ & \text{avec } x_1, x_2 \in [-20, 20]. \end{aligned}$$

Le deuxième problème RUE issu de (Ruetsch, 2005):

$$\text{Min } (f_1, f_2) ;$$

$$f_1 = (((x_1 + 0.5)^2 + x_2^2)^{0.5})$$

$$(0.1 * (\sin(8.0 * \pi * x_1) + \sin(8.0 * \pi * x_2) - 2.0))$$

$$f_2 = (((x_1 - 0.5)^2 + x_2^2)^{0.5})$$

avec $x_1 \in [-1, 1]$ et $x_2 \in [-0.5, 0.5]$.

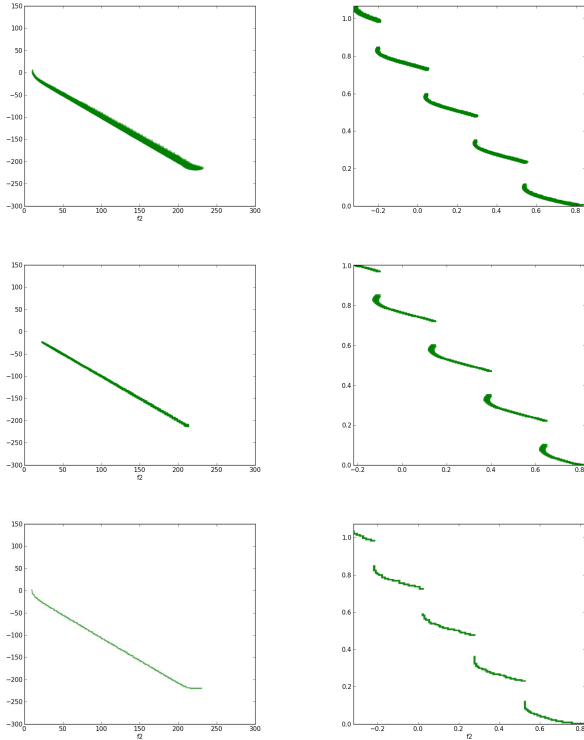


Figure 7: Les figures de gauche sont les résultats du problème SRN et de droite sont les résultats du problème RUE. Du haut vers le bas sont représentés les résultats obtenus par: MOIBBA, IDCDF et IMMT.

	MOIBBA	IDCDF	IMMT
Nbr de solutions	4261	2341	137
Nbr d'itérations	8018	7547	34256
Temps de calcul (s)	36.45	6.55	17.45
Nbr sols où $L_{in} \neq \emptyset$	-	-	21

Tableau 1: Les résultats numériques du premier problème traité (SRN), $\epsilon_x = 0.2$ et $\epsilon_f = 8$

Les résultats illustrés dans la figure 7 montrent que MOIBBA est capable de calculer un encadrement total du front de Pareto même ceux non convexes. Par contre, deux problèmes se présentent; un large front de Pareto obtenu dû à la problématique de recouvrement de boîtes dans l'espace des objectifs(section 3.1.3) et un grand temps de réponse (tableaux 1 et 2) dû à

	MOIBBA	IDCDF	IMMT
Nbr de solutions	1918	272	248
Nbr d'itérations	112425	1079	23054
Temps de calcul (s)	49.50	0.61	121.89
Nbr sols où $L_{in} \neq \emptyset$	-	-	0

Tableau 2: Les résultats numériques du deuxième problème traité (RUE), $\epsilon_x = 0.005$ et $\epsilon_f = 0.005$

l'exploration exhaustive de l'espace de recherche.

Afin de réduire la largeur de l'encadrement du front de Pareto dû au problème de recouvrement des boîtes d'objectifs, nous avons implémenté une méthode inverse qui consiste à explorer dans l'espace des objectifs proposée par Kubica. Nous concluons que sa méthode est efficace en termes de temps de résolution. Par contre, en plus de la complexité de l'implémentation, son algorithme nécessite une très bonne connaissance du problème en risquant de retourner des solutions inexploitables. En effet, nous avons vu qu'une solution Pareto optimale retournée par cette méthode inverse est formée de trois éléments ; Une boîte \mathbf{Y}_{ϵ_Y} présente dans le front de Pareto de l'espace des objectifs et deux listes (L_{in} , L_{bound}) associées à \mathbf{Y}_{ϵ_Y} . Un mauvais paramétrage de précision d'une des variables du problème (ϵ_x) ou d'une des fonctions objectifs (ϵ_f) conduira à obtenir des solutions inexploitables ayant un L_{in} vide, c'est-à-dire une boîte objective n'ayant aucune image inverse dans l'espace de recherche. Prenons par exemple les résultats du problème SRN(tableau 1), 21 solutions fournies par Kubica sur 137 ayant un $L_{in} \neq \emptyset$ sont exploitables.

Reutsch, propose un nouveau test d'optimalité *Interval Differential Formulation* (Ruetsch, 2010) consistant à accélérer d'une façon considérable les algorithmes déterministes multiobjectifs et à proposer au concepteur moins de solutions inutiles. En revanche, nous remarquons une perte de solutions optimales en adoptant un tel test (figure 7 problème SRN) fortement liée à la formulation mathématique des fonctions objectifs.

5 CONCLUSION ET FUTURS TRAVAUX

Nous avons vu que les approches déterministes d'optimisation globale des problèmes multiobjectifs, abordées dans ce papier et validées expérimentalement, consistent à explorer exhaustivement l'espace de recherche afin d'obtenir la totalité des optima globaux du problème.

Nous montrons que les approches déterministes naïves de résolution (section 3.1.1) sont capables de résoudre les problèmes d'optimisation multiobjectifs. Par contre, le front de Pareto obtenu et encadrant les solutions optimales du problème, est assez large. Ce problème est dû aux chevauchements de boîtes développés dans la section 3.1.3). Une solution à ce problème de chevauchement serait la méthode de résolution inverse, c'est à dire explorer et découper les boîtes dans l'espace des objectifs au lieu de l'espace de recherche. Nous avons apporté quelques améliorations à ces méthodes inverses présentées dans la section 3.2. En revanche, nos résultats montrent leur limitation en terme de résolution. En effet, en plus de leur complexité d'implémentation, ces méthodes de résolution inverse sont susceptibles de retourner des solutions inexploitable. Une solution inexploitable est une solution (ou une boîte de taille ϵ) dans l'espace des objectifs qui ne possède aucune évaluation inverse dans l'espace de recherche. Une autre approche *Interval Differential Formulation* (Ruetsch, 2010) permet de réduire la largeur du front de Pareto en proposant au concepteur moins de solutions inutiles. L'implémentation de cette approche et les résultats obtenus nous ont permis de détecter une perte de solutions fortement liée à la formulation mathématique des fonctions objectifs du problème. En revanche, cette approche a permis d'accélérer d'une façon remarquable les algorithmes déterministes basés sur l'analyse par intervalle.

Par ailleurs, sur des problèmes de grandes tailles, la résolution est très coûteuse en temps du fait de l'exploration exhaustive de l'espace de recherche. Ainsi, une perspective serait d'adopter une exploration interactive permettrait de limiter les temps de calcul. De plus, dans un contexte d'aide à la décision pour la conception (*set-based design*) ou le pilotage d'un système, la programmation par contraintes offre des pistes intéressantes pour aborder ce problème telles que le calcul d'enveloppes de l'espace de faisabilité, une heuristique d'exploration diversifiée voire une manipulation dynamique des contraintes. Finalement, un problème d'optimisation réel de propulsion de navire est en cours d'élaboration et les premiers résultats sont encourageants pour la suite malgré l'obstacle des temps de calculs encore relativement longs (supérieurs à 30 minutes).

REFERENCES

- Benhamou F. and L. Granvilliers, 2006b. Continuous and Interval Constraints. *In Handbook of Constraint Programming*, Elsevier, pages 571-604.
- Collette Y. and P. Siarry, 2002. Optimisation multiobjectif. *Eyrolles*.
- Deb K., 2001. Multi-Objective Optimization using

Evolutionary Algorithms. *John Wiley & Sons*, Chichester.

- Deb K., A. Pratap, S. Agarwal and T. Meyarivan, 2002. A Fast an Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on evolutionary computation*, vol. 6, no. 2.
- Ehrgott M., 2000. Multicriteria optimization. *In Lecture Notes in Economics and Mathematical Systems*, volume 491. Springer.
- Granvilliers L. and F. Benhamou, 2006a. Realpaver: an Interval Solver using Constraint Satisfaction Techniques. *ACM TOMS*, 32(1):138-156.
- Hansen E., 1992. Global Optimization Using Interval Analysis. *Marcel Dekker*, New York.
- Jaulin L., M. Kieffer, O. Didrit and E. Walter, 2001. Applied Interval Analysis. *Springer*, London.
- Kubica B.J. and K. Malinowski, 2005. An Interval Global Optimization Algorithm Combining Symbolic Rewriting and Componentwise Newton Method Applied to Control a Class of Queueing Systems. *Reliable Computing 11*, pages 393-411.
- Kubica B.J. and A. Wozniak, 2007. Interval methods for computing the Pareto-front of a multicriterial problem. *presented at PPAM 2007 Conference*, Gdansk.
- Moore R.E., 1966. Interval analysis. *Prentice Hall*.
- Pahl G. and W. Beitz, 1996. Engineering Design: A systematic approach. *Springer-Verlag*, ISBN 3-540-19917-9.
- Ruetsch G.R., 2005. An interval algorithm for multi-objective optimization. *Structural and Multidisciplinary Optimization 30*, pages 27-37.
- Ruetsch G.R., 2010. Using interval techniques of direct comparaison and differential formulation to solve a multi-objective optimization problem. *Brevet US7742902B1*.
- Zitzler E., 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. *PhD. Thesis*, ETH Zurich, Switzerland.