



HAL
open science

Proposition de compromis pour le calcul de solutions préférées à l'aide d'un algorithme évolutionnaire multiobjectif en optimisation multidisciplinaire

Benoît Guédas

► To cite this version:

Benoît Guédas. Proposition de compromis pour le calcul de solutions préférées à l'aide d'un algorithme évolutionnaire multiobjectif en optimisation multidisciplinaire. Autre [cs.OH]. Ecole Centrale de Nantes (ECN), 2010. Français. NNT: . tel-00563994

HAL Id: tel-00563994

<https://theses.hal.science/tel-00563994>

Submitted on 7 Feb 2011

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.

École Centrale de Nantes

**ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR, GÉOSCIENCES,
ARCHITECTURE**

Année 2010

N° B.U. :

**Thèse de Doctorat
*Diplôme délivré par l'École Centrale de Nantes***

Spécialité : GÉNIE MÉCANIQUE

Présentée et soutenue publiquement par :

BENOÎT GUÉDAS

le 25 octobre 2010

à l'École Centrale de Nantes

**PROPOSITION DE COMPROMIS POUR LE
CALCUL DE SOLUTIONS PRÉFÉRÉES À L'AIDE
D'UN ALGORITHME ÉVOLUTIONNAIRE
MULTIOBJECTIF EN OPTIMISATION
MULTIDISCIPLINAIRE**

JURY

Président :	Jean-Xavier RAMPON, Professeur	<i>Université de Nantes</i>
Rapporteurs :	Piotr BREITKOPF, Docteur, HDR	<i>CNRS – Université de Technologie de Compiègne</i>
	Patrick SIARRY, Professeur	<i>Université Paris-Est Créteil Val-de-Marne</i>
Examineurs :	David BASSIR, Professeur	<i>École Spéciale des Travaux Publics</i>
	Armelle LE GALL, Docteur	<i>Eurodécision</i>
	Jean-Xavier RAMPON, Professeur	<i>Université de Nantes</i>
	Philippe DÉPINCÉ, Professeur	<i>École Centrale de Nantes</i>
	Xavier GANDIBLEUX, Professeur	<i>Université de Nantes</i>

Directeur de thèse : Philippe DÉPINCÉ
Co-directeur de thèse : Xavier GANDIBLEUX
Laboratoire : Institut de Recherche en Communications et Cybernétique de Nantes. UMR CNRS 6597

N° É.D. : 498-xxx

*À mes parents,
À Florence,*

TABLE DES MATIÈRES

TABLE DES MATIÈRES	iii
NOTATIONS	vii
INTRODUCTION	1
1 OPTIMISATION ET ALGORITHMES ÉVOLUTIONNAIRES	5
INTRODUCTION	6
1.1 LES OPTIMUMS D'UN ENSEMBLE	6
1.1.1 Notions de préordres	6
1.1.2 Définition des minimums	9
1.2 PROBLÈME D'OPTIMISATION	10
1.2.1 Minimum d'une fonction	11
1.2.2 Prise en compte des préférences	11
1.2.3 Solutions d'un problème d'optimisation	14
1.3 ALGORITHMES D'OPTIMISATION	17
1.3.1 Métaheuristiques	17
1.3.2 Algorithmes d'optimisation itératifs et à population	18
1.3.3 Principe des algorithmes évolutionnaires	19
1.4 GÉNOTYPE : CODAGE DES INDIVIDUS ET OPÉRATEURS GÉNÉTIQUES	20
1.4.1 Codage des individus	20
1.4.2 Croisements	21
1.4.3 Mutations	23
1.4.4 Structure de l'espace de recherche	24
1.5 PHÉNOTYPE : ÉVALUATION ET SÉLECTION DES INDIVIDUS	26
1.5.1 Évaluation des individus	27
1.5.2 Sélection	30
CONCLUSION	31
2 FORMALISATION ET RÉOLUTION DES PROBLÈMES D'OMD	33
INTRODUCTION	34
2.1 DESCRIPTION DES PROBLÈMES D'OPTIMISATION MULTIDISCIPLINAIRE	34
2.1.1 Analyse disciplinaire	35
2.1.2 Optimisation disciplinaire	37
2.1.3 Analyse multidisciplinaire	38
2.1.4 Problème d'optimisation multidisciplinaire	42
2.2 MÉTHODES DE RÉOLUTION	43
2.2.1 Reformulations de problèmes	44
2.2.2 Théorie des jeux	51

2.2.3	Méthodes basées sur les algorithmes évolutionnaires	53
	CONCLUSION	57
3	LE COMPROMIS EN OMD MULTIOBJECTIF	59
	INTRODUCTION	60
3.1	PROBLÉMATIQUE	60
3.1.1	Définition du problème	60
3.1.2	Traitement du compromis dans les méthodes d'OMD	62
3.1.3	Limites des méthodes actuelles	62
3.2	\mathcal{C}_1 : LE PRODUIT D'ORDRE	64
3.2.1	Principe	64
3.2.2	Formalisation	64
3.3	\mathcal{C}_2 : UNE EXTENSION DU PRODUIT	66
3.3.1	Principe	66
3.3.2	Formalisation	67
3.3.3	Propriétés	72
3.4	\mathcal{C}_3 : LE COMPROMIS COMME LE RANG DES RANGS	74
3.4.1	Principe	74
3.4.2	Formalisation	75
3.4.3	Propriétés	76
3.5	INCONVÉNIENTS DES COMPROMIS \mathcal{C}_2 ET \mathcal{C}_3	76
3.6	\mathcal{C}_4 : UN PRODUIT FAIBLE ENTRE LES DISCIPLINES	79
3.6.1	Principe	79
3.6.2	Formalisme	79
3.6.3	Propriétés	81
	CONCLUSION	81
4	EMO-MDO : UN ALGORITHME ÉVOLUTIONNAIRE POUR L'OMD	83
	INTRODUCTION	84
4.1	ALGORITHME	84
4.1.1	Algorithme général	84
4.1.2	Calcul du rang	85
4.2	EXPÉRIMENTATIONS	90
4.2.1	Protocole expérimental	90
4.2.2	Problème 1	92
4.2.3	Problème 2	94
4.2.4	Problème 3 : problème à quatre barres	102
4.2.5	Problème 4 : le réducteur de vitesse	108
4.2.6	Synthèse des résultats	114
	CONCLUSION	117
	CONCLUSION	119
	A RELATIONS : DÉFINITIONS ET NOTATIONS	123
	B DIAGRAMMES DE HASSE	127
	C APPROXIMATION DES COUPLAGES DANS COSMOS-G	129
C.1	TRAITEMENT DES VARIABLES COUPLÉES	129

C.1.1	Traitement des variables de couplage dans COSMOS	130
C.2	PROBLÈMES DE TEST	131
C.2.1	Problème #1	131
C.3	RÉSULTATS	132
C.3.1	Qualité de l'approximation	133
D	PRISE EN COMPTE DES VARIABLES LOCALES DANS EMO-MDO	139
E	SOLUTIONS POUR LE PROBLÈME DU RÉDUCTEUR DE VITESSE	141
	RÉFÉRENCES BIBLIOGRAPHIQUES	145

NOTATIONS

$:=$	Définition
$\llbracket n \rrbracket$	Ensemble des entiers de 1 à n : $\llbracket n \rrbracket := [1; n] \cap \mathbb{N}$
\mathcal{R}	Relation binaire
\mathcal{R}^1	Fermeture réflexive de \mathcal{R}
\mathcal{R}^+	Fermeture transitive de \mathcal{R}
\mathcal{R}^*	Fermeture réflexo-transitive de \mathcal{R}
(E, \mathcal{R})	Relation binaire de base E
\leq	Relation d'ordre
\preceq	Relation de préordre
\preceq	Relation de dominance de Pareto
\equiv	Relation d'équivalence $\preceq \cap \succeq$
$[n]_{\mathcal{R}}$	Classe d'équivalence de n sur \mathcal{R}
$\downarrow x$	Idéal de x
$\uparrow x$	Filtre de x
\oplus	Somme ordinale de préordres
\mathcal{X}	Espace de décision (ou des variables)
\mathcal{Y}	Espace des critères (ou des objectifs)
\mathcal{Z}	Espace des comparaisons
n	Dimension de l'espace des variables
p	Nombre d'objectifs du problème
p_i	Nombre d'objectifs de la discipline i
q	Nombre de disciplines du problème
f	Fonction coût : $f : \mathcal{X} \rightarrow \mathcal{Y}$
f_i	Fonction coût de la i -ème discipline
$f_{i,j}$	j -ème fonction coût de la i -ème discipline
θ	Fonction de préférence : $\theta : \mathcal{Y} \rightarrow \mathcal{Z}$
θ_i	Fonction de préférence de la i -ème discipline
$\theta_{i,j}$	j -ème fonction de préférence de la i -ème discipline
ϕ	Fonction objectif : $\phi = \theta \circ f$, $\phi : \mathcal{X} \rightarrow \mathcal{Z}$
ϕ_i	Fonction objectif de la i -ème discipline
$\phi_{i,j}$	j -ème fonction objectif de la i -ème discipline
X	Ensemble admissible : $X \subseteq \mathcal{X}$
Y	Ensemble réalisable : $Y := f(X) \subseteq \mathcal{Y}$
Z	Image des solutions réalisables par θ : $Z := \theta(Y) \subseteq \mathcal{Z}$
X^E	Ensemble des solutions efficaces
Y^N	Ensemble des points non-dominés
C_i	Compromis i

INTRODUCTION

A FIN de répondre à la pression croissante de la concurrence, le temps de mise sur le marché des produits manufacturés a été diminué (Stalk 1988, Stalk et Hout 2003). En parallèle, pour résister à la concurrence économique accrue, les petites entreprises se sont regroupées en multinationales, et une organisation sous forme d'entreprise étendue est adoptée grâce notamment au développement des moyens de communication. La disponibilité d'outils plus performants (moyens de calcul, outils de modélisation, gestion de projets...), les plus grandes ressources disponibles (humaines, matérielles, budgétaires), ainsi que les attentes croissantes des clients aboutissent au développement de produits de plus en plus complexes. Cette complexité se retrouve tant sur l'objet technique que sur les moyens de production mis en œuvre pour sa fabrication.

La conception de systèmes industriels complexes, tels que les avions, voitures ou bateaux, implique aujourd'hui la collaboration de plusieurs équipes spécialisées, distribuées géographiquement sur plusieurs sites et utilisant des outils spécifiques. Le problème de conception est alors décomposé en sous-problèmes traités indépendamment par les différentes entités impliquées dans la conception du produit. La recherche du produit le plus performant au meilleur coût oblige chacune de ces entités à optimiser son sous-système suivant des objectifs souvent antagonistes. La conception d'un avion implique par exemple l'optimisation de l'aérodynamique d'une part, et de la structure d'autre part. Or, l'augmentation des performances aérodynamiques s'accompagne d'une diminution des performances de la structure et inversement. L'optimalité d'une solution, dans ce cadre, n'est pas clairement définie et plusieurs solutions de compromis entre les performances aérodynamiques et structurelles sont possibles. Ces solutions de compromis sont généralement les solutions efficaces (aussi appelées Pareto-optimales) du problème. Ce type de solution n'est cependant pas adapté au cas où chaque discipline dispose de plusieurs objectifs propres (par exemple minimiser la traînée et maximiser la portance pour l'aérodynamique) car il s'agirait dans ce cas d'un compromis entre les objectifs, sans distinction de leur appartenance aux disciplines, plutôt qu'un compromis entre les disciplines en tant que tel. En effet, les solutions de compromis cherchées doivent tenir compte des préférences disciplinaires.

Les méthodes de conception traditionnelles, séquentielles – parfois qualifiées de *conception par-dessus le mur* (*over-the-wall design*, car chaque étape envoie ses ordres à la suivante sans se préoccuper de sa faisabilité (Boothroyd 1992)) – ne sont pas adaptées à une telle organisation décentralisée et concurrente. En effet, en conception séquentielle, les degrés de liberté de chaque sous-système décroissent au fur et à mesure de la conception : les différents sous-systèmes partageant des variables communes, le premier sous-système fixe toutes celles le concernant, réduisant ainsi l'espace de recherche du sous-système suivant... De nouvelles méthodes de conception sont nécessaires pour,

d'une part, décomposer le problème en sous-problèmes plus simples à traiter, d'autre part, synthétiser les solutions locales de chaque sous-problème pour obtenir une ou plusieurs solutions de compromis globales.

Le but de l'optimisation multidisciplinaire est de fournir des méthodes formalisées répondant à ces attentes. En cela, elle est complémentaire d'autres méthodes de conception, telles que l'ingénierie concourante (Winner *et al.* 1988) et la conception collaborative (Klein *et al.* 2006).

L'aérospatial, où le problème de masse est crucial, fut le premier secteur industriel à inclure différentes disciplines dans ses modèles d'optimisation (Agte *et al.* 2009). Les couplages interdisciplinaires ne sont en effet pas négligeables entre la structure, l'aérodynamique, la propulsion. . . Un exemple typique est le problème aéroélastique qui intervient dans la conception d'une aile d'avion (Hodges et Pierce 2002). Dans ce problème, deux disciplines – l'aérodynamique et la structure – sont couplées, c'est-à-dire qu'une modification dans l'une des disciplines implique des modifications dans l'autre. Chacune des disciplines dispose de ses propres objectifs à optimiser : minimisation de la traînée et maximisation de la portance pour l'aérodynamique, et minimisation de la masse et des déformations élastiques pour la structure. Ainsi, une minimisation de la traînée pour l'aérodynamique implique une augmentation de la masse de la structure et inversement. Les deux disciplines ne pouvant être traitées séparément et ayant chacune une influence primordiale sur les performances globales, il convient de les unifier dans une formulation de problème d'optimisation commune. La résolution d'un tel problème requiert l'obtention de solutions de **compromis qui doivent, d'une part, assurer la satisfaction des solutions (résolution des couplages), et, d'autre part, prendre en compte les performances (optimisation)**. Le problème de la satisfaction des couplages fait l'objet de nombreuses recherches, et la majorité des méthodes d'optimisation multidisciplinaire s'intéressent principalement à ce problème. Nous ne considérons pas explicitement la satisfaction des couplages. Pour cela, nous ne considérons que les solutions admissibles du problème, c'est-à-dire celles respectant les couplages entre disciplines. En effet, nous nous intéressons au second problème, qui est à la définition de solutions optimales. Les performances sont exprimées en terme de fonctions objectifs qui peuvent être soit locales aux disciplines (traînée, portance pour l'aérodynamique), soit globales (coût et masse totale. . .). La présence d'un ou plusieurs objectifs locaux et globaux constitue différentes classes de problèmes (voir tableau 1).

		mono-objectif	multiobjectif
1 discipline		1	2
>1 discipline	objectif global	3	4
	objectif local	5	6

TABLEAU 1 – Classes de problèmes d'optimisation suivant le nombre de disciplines et le nombre d'objectifs globaux et disciplinaires. Certaines classes de problèmes pouvant être fusionnées, comme (3,5) et (4,6). Dans cette thèse, nous traiterons du cas n°6.

Les méthodes d'optimisation multidisciplinaire classiques s'intéressent principalement à la résolution des couplages, et le problème d'optimisation traité est un problème d'optimisation global mono-objectif : soit il appartient à la classe 3 (voir tableau 1), soit il appartient à la classe 4, 5 ou 6, mais est reformulé comme un problème de la classe 3. Des méthodes ont été récemment proposées, transformant le problème d'optimisation multidisciplinaire en un problème d'optimisation multiobjectif. Dans ce contexte, les moteurs

d'optimisation reposent sur des algorithmes évolutionnaires multiobjectifs. Ces méthodes sont adaptées aux classes de problèmes 4 et 5, où l'ensemble des solutions de compromis recherché est l'ensemble des solutions efficaces (ou Pareto-optimales). Elles sont cependant également appliquées à la classe 6, où l'ensemble des solutions obtenues ne reflète pas le regroupement des objectifs en disciplines. En effet, **des solutions peuvent être globalement efficaces, alors qu'elles sont localement dominées dans une ou plusieurs disciplines. Les préférences disciplinaires en terme de performance ne sont alors pas respectées.** Cette configuration suppose deux niveaux de compromis : un premier compromis au sein de chaque discipline entre leurs objectifs respectifs, puis un second compromis entre les disciplines. Le compromis au sein de chaque discipline relève de l'optimisation multiobjectif et est usuellement défini comme l'ensemble des solutions efficaces au sens de Pareto. Il n'existe pas, à notre connaissance, de définition du compromis entre les disciplines qui seraient elles-mêmes des problèmes d'optimisation multiobjectif. De plus, la résolution de tels problèmes avec des algorithmes évolutionnaires fait apparaître des problèmes de convergence vers les solutions optimales.

Notre contribution porte sur la définition de compromis adaptés à ce type de problème et prenant en compte les préférences des disciplines. Quatre compromis différents sont proposés. Ils sont étudiés d'un point de vue théorique sur la modélisation des préférences des disciplines, et d'un point de vue pratique sur leur capacité à converger, lorsqu'ils sont utilisés dans un algorithme évolutionnaire multiobjectif. Ces analyses aboutissent à l'implémentation d'un algorithme évolutionnaire multiobjectif adapté aux problèmes d'optimisation multidisciplinaire multiobjectif (classe 6 du tableau 1).

Ces recherches ont été effectuées à l'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN – UMR CNRS 6597) sous la co-direction de Philippe Dépincé, professeur des universités en génie mécanique de l'École Centrale de Nantes, et de Xavier Gandibleux, professeur des universités en informatique de l'Université de Nantes et chercheur au Laboratoire d'Informatique de Nantes Atlantique (LINA – UMR CNRS 6241). Elles s'inscrivent dans la continuité des travaux en optimisation multidisciplinaire et multiobjectif réalisés dans l'équipe MCM¹ de l'IRCCyN, et en particulier les thèses de Giassi (2004) et de Rabeau (2007). Les thèmes abordés par cette thèse rejoignent également certaines problématiques de l'équipe ROOM² du LINA, telles que les métaheuristiques et l'optimisation multiobjectif.

Le manuscrit est organisé en quatre chapitres, dont les deux premiers constituent l'état de l'art et présentent les concepts d'optimisation et en particulier d'optimisation multiobjectif et multidisciplinaire, ainsi que leurs notations et méthodes. Le premier chapitre introduit l'optimisation à l'aide de métaheuristiques et en particulier les algorithmes génétiques multiobjectifs. Dans le second chapitre, le problème d'optimisation multidisciplinaire est défini formellement, et les méthodes de résolution de la littérature y sont présentées.

Les deux chapitres suivants constituent le cœur de cette thèse et détaillent nos propositions sur la définition du compromis en optimisation multidisciplinaire multiobjectif et leur mise en œuvre au sein d'un algorithme évolutionnaire multiobjectif. En examinant les méthodes d'optimisation multidisciplinaire, il apparaît que la notion de compromis

1. Méthodes de Conception en Mécanique.
2. Recherche Opérationnelle et Optimisation Multiobjectif.

en optimisation multidisciplinaire multiobjectif n'est pas clairement définie. Cette notion est développée dans le troisième chapitre, où quatre méthodes de compromis sont présentées. Chaque compromis est défini formellement à l'aide de la théorie des ensembles ordonnés. Le quatrième chapitre présente un algorithme évolutionnaire multiobjectif implémentant les quatre méthodes de compromis proposées. Une analyse expérimentale de cet algorithme est effectuée sur quelques problèmes d'optimisation multidisciplinaire de la littérature. Quatre problèmes, sans fonction de couplage, impliquant deux ou trois disciplines biobjectifs ont été sélectionnés. Différentes configurations de l'algorithme sont testées sur ces problèmes.

CHAPITRE 1

Modélisation des problèmes d'optimisation et algorithmes évolutionnaires

SOMMAIRE

INTRODUCTION	6
1.1 LES OPTIMUMS D'UN ENSEMBLE	6
1.1.1 Notions de préordres	6
1.1.2 Définition des minimums	9
1.2 PROBLÈME D'OPTIMISATION	10
1.2.1 Minimum d'une fonction	11
1.2.2 Prise en compte des préférences	11
1.2.3 Solutions d'un problème d'optimisation	14
1.3 ALGORITHMES D'OPTIMISATION	17
1.3.1 Métaheuristiques	17
1.3.2 Algorithmes d'optimisation itératifs et à population	18
1.3.3 Principe des algorithmes évolutionnaires	19
1.4 GÉNOTYPE : CODAGE DES INDIVIDUS ET OPÉRATEURS GÉNÉTIQUES	20
1.4.1 Codage des individus	20
1.4.2 Croisements	21
1.4.3 Mutations	23
1.4.4 Structure de l'espace de recherche	24
1.5 PHÉNOTYPE : ÉVALUATION ET SÉLECTION DES INDIVIDUS	26
1.5.1 Évaluation des individus	27
1.5.2 Sélection	30
CONCLUSION	31

Introduction

L'OPTIMISATION est un aspect important de la conception. On attend, d'une part, que le produit manufacturé possède des caractéristiques optimales et, d'autre part, que son processus de conception soit lui-même optimal, ce dernier influe sur les objectifs du premier (réduction des coûts, des délais...).

Les algorithmes d'optimisation en conception doivent répondre à plusieurs exigences. Ils doivent être robustes (fonctionner sur des fonctions bruitées, pouvoir être arrêtés en cours d'optimisation...), facilement adaptables à d'autres problèmes, et proposer des solutions de bonne qualité dans un temps borné. La solution optimale du problème n'est pas toujours recherchée ou accessible, on recherche alors la meilleure solution possible dans un temps donné d'exécution, voire, plus simplement, une solution améliorant une solution existante. De plus, la complexité des problèmes ne permet pas toujours de définir formellement l'ensemble des contraintes et objectifs. Dans ce contexte, il est alors intéressant de pouvoir disposer d'un ensemble varié de solutions de bonne qualité dans lequel le concepteur pourra faire son choix en y intégrant éventuellement des données externes (d'après son expérience ou de nouvelles données). Les algorithmes évolutionnaires sont un type d'algorithme d'optimisation adaptés à ces exigences.

Dans ce chapitre, nous donnons quelques définitions sur la théorie des ordres qui nous permettent d'introduire le formalisme utilisé pour modéliser les problèmes d'optimisation. À partir de ce formalisme, nous décrivons, dans un cadre commun, différentes interprétations d'optimalité définies à partir des structures ordonnées sous-jacentes. Les algorithmes évolutionnaires multiobjectifs sont présentés dans ce cadre.

1.1 Les optimums d'un ensemble

La notion d'optimalité suppose l'existence d'une structure d'ordre. Dans cette section, nous présentons les prérequis nécessaires sur les ensembles ordonnés et préordonnés (qui sont plus généraux) à la définition des solutions optimales d'un problème d'optimisation.

1.1.1 Notions de préordres

Pour une introduction plus complète aux ordres, le lecteur intéressé pourra se référer aux ouvrages de Schröder (2002) ou de Caspard, Leclerc, et Monjardet (2007). Les définitions suivantes se basent sur les définitions et notations de relations détaillées dans l'annexe A.

Les relations d'ordre sont des relations binaires vérifiant des propriétés particulières. On appellera respectivement préordre et ordre le couple formé d'un ensemble de base et d'une relation de préordre ou d'ordre.

Définition 1.1 (Préordre) *Une relation binaire \preceq sur un ensemble de base E est une relation de préordre si et seulement si elle est réflexive et transitive. (E, \preceq) est alors appelé préordre ou ensemble préordonné.*

Définition 1.2 (Ordre) *Une relation binaire \leq sur un ensemble de base E est une relation d'ordre si et seulement si elle est une relation de préordre antisymétrique. (E, \leq) est alors appelé ordre ou ensemble ordonné.*

On peut donc avoir une relation de préordre telle que $a \preceq b$ et $b \preceq a$, bien qu' a et b

soient distincts. Les éléments a et b sont alors équivalents et on note $a \equiv b$ cette relation d'équivalence. À partir de tout préordre, on peut donc obtenir un ordre, appelé ordre quotient, en regroupant chaque élément dans sa classe d'équivalence par la relation \equiv .

Définition 1.3 (Ordre quotient) *Soit un préordre $P = (E, \preceq)$. L'ordre quotient Q de P est l'ordre défini comme*

$$Q := (E / \equiv, \preceq / \equiv)$$

avec respectivement (E / \equiv) l'ensemble quotient de E , et (\preceq / \equiv) la relation quotient de \preceq par \equiv (voir en annexe, les définitions A.16 et A.17).

Remarque 1.1 *Le quotient d'un préordre par \equiv est toujours un ordre.*

Les relations d'ordre et de préordre ne sont pas nécessairement totales. Il peut en effet exister des éléments incomparables entre eux. Par exemple \mathbb{R} est totalement ordonné puisque $\forall x, y \in \mathbb{R}$, on a soit $x \leq y$ soit $y \leq x$. En revanche, il n'y a pas d'ordre trivial ordonnant totalement \mathbb{R}^2 . Par exemple, soient x et y deux éléments de \mathbb{R} , comment comparer (x, y) et (y, x) ?

L'ensemble \mathbb{R}^2 étant un ensemble produit, l'ordre le plus naturel est l'ordre produit, correspondant à cet ensemble. Cet ordre est construit par le produit direct, qui est une opération associative et commutative.

Définition 1.4 (Produit direct) *Soient $O_1 = (E_1, \preceq_{E_1})$ et $O_2 = (E_2, \preceq_{E_2})$ deux préordres. On définit leur produit $(E, \preceq_E) = O_1 \times O_2$ tel que :*

- $E = E_1 \times E_2$
- $(a_1, a_2) \preceq_E (b_1, b_2) \Leftrightarrow a_1 \preceq_{E_1} b_1 \wedge a_2 \preceq_{E_2} b_2$

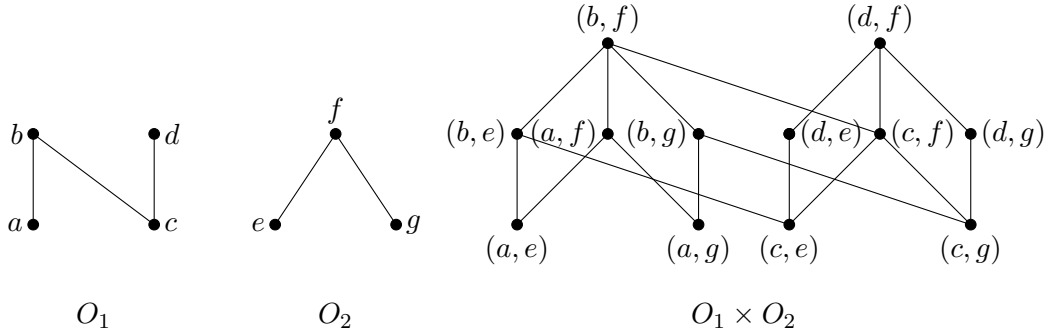


FIGURE 1.1 – Diagramme de Hasse (définition B.3) du produit de deux ordres O_1 et O_2 .

Exemple 1.1 *Soient $O_1 = (\{a, b, c, d\}, \{(a, a), (a, b), (b, b), (c, b), (c, c), (c, d), (d, d)\})$ et $O_2 = (\{e, f, g\}, \{(e, e), (e, f), (f, f), (g, f), (g, g)\})$ deux ordres. Leur produit direct $O = (E, \preceq)$ est l'ensemble obtenu par le produit cartésien de leur ensemble de base ordonné tel qu'un élément est inférieur à un autre si les deux composantes du premier sont respectivement inférieures aux deux composantes du second. Le résultat est illustré figure 1.1.*

Compléter un ordre pour qu'il soit totalement ordonné est une *extension linéaire* de l'ordre. Il peut exister plusieurs extensions linéaires d'un même ordre.

Définition 1.5 Soit (E, \leq_1) un ensemble ordonné. L'ordre (E, \leq_2) est une extension linéaire de (E, \leq_1) si et seulement si (E, \leq_2) est totalement ordonné et $\leq_1 \subseteq \leq_2$.

L'extension linéaire va garder les relations de comparaison entre les éléments. L'application permettant de passer d'un ordre à une de ses extensions linéaires est un *morphisme*. D'autres morphismes peuvent également être définis.

Définition 1.6 (Morphisme) Soient deux préordres (E_1, \preceq_1) et (E_2, \preceq_2) . Une application θ de E_1 dans E_2 est un morphisme de (E_1, \preceq_1) dans (E_2, \preceq_2) si et seulement si $\forall a, b \in E_1, a \preceq_1 b \Rightarrow \theta(a) \preceq_2 \theta(b)$.

Un morphisme est une application préservant la relation de départ. Si deux préordres de même cardinalité partagent la même structure (ils sont identiques à un étiquetage des éléments près), ils sont en relation d'isomorphisme (il y a un morphisme dans les deux sens).

Définition 1.7 (Isomorphisme) Soient deux préordres (E_1, \preceq_1) et (E_2, \preceq_2) . Une application θ de E_1 dans E_2 est un isomorphisme de (E_1, \preceq_1) dans (E_2, \preceq_2) si et seulement si θ est bijective et $\forall a, b \in E_1, a \preceq_1 b \Leftrightarrow \theta(a) \preceq_2 \theta(b)$.

Exemple 1.2 Soit l'ordre produit $(\mathbb{R}^2, \leq_{\mathbb{R}^2})$ et θ l'application de \mathbb{R}^2 dans \mathbb{R} : $(x_1, x_2) \mapsto x_1 + x_2$. θ est un morphisme de $(\mathbb{R}^2, \leq_{\mathbb{R}^2})$ dans $(\mathbb{R}, \leq_{\mathbb{R}})$.

Une autre opération élémentaire sur les préordres est la somme ordinale qui est ainsi définie : étant donné deux préordres A et B , la somme ordinale entre A et B conserve la relation interne de chacun des préordres et ajoute que tous les éléments de A sont inférieurs aux éléments de B (Figure 1.2).

Définition 1.8 Soient deux préordres $O_1 = (E_1, \preceq_{E_1})$ et $O_2 = (E_2, \preceq_{E_2})$. Leur somme ordinale $O = (E, \preceq_E)$, notée

$$O = O_1 \oplus O_2$$

est définie par :

$$E = E_1 \cup E_2$$

et

$$\forall i, j \in \{1, 2\}, \forall a \in O_i, \forall b \in O_j, a \preceq_E b \iff i \leq j$$

Remarque 1.2 La somme ordinale est associative mais pas commutative.

Nous souhaitons également définir des opérations internes aux ordres définies sur leurs sous-ordres. Pour cela, nous définissons la notion de *partition d'ordre*.

Définition 1.9 (Partition d'ordre) Soit un ordre (E, \leq) . Nous appellerons partition de (E, \leq) toute famille d'ensemble $\{(E, \leq)[E_i]\}_{E_i \in P}$ telle que $((E, \leq)[E_i])$ est un sous-ordre de (E, \leq) restreint à E_i (définition A.20) et P est une partition de E .

Définition 1.10 (Partition ordonnée d'ordre) Soit $\{(E_1, \leq_1), \dots, (E_n, \leq_n)\}$ une partition de (E, \leq) . $((E_1, \leq_1), \dots, (E_n, \leq_n))$ est une partition ordonnée de (E, \leq) si

- $\forall i \in \llbracket n \rrbracket, \forall a, b \in E_i, a \leq_i b \Rightarrow a \leq b$
- $\forall i, j \in \llbracket n \rrbracket, \forall a_i \in E_i, \forall b_j \in E_j, i \leq j \Rightarrow a \leq b$

Propriété 1.1 Soient un ordre O et deux sous-ordres O_1 et O_2 de O tels que (O_1, O_2) soit une partition ordonnée de O . Alors l'application définie par \oplus est un morphisme.

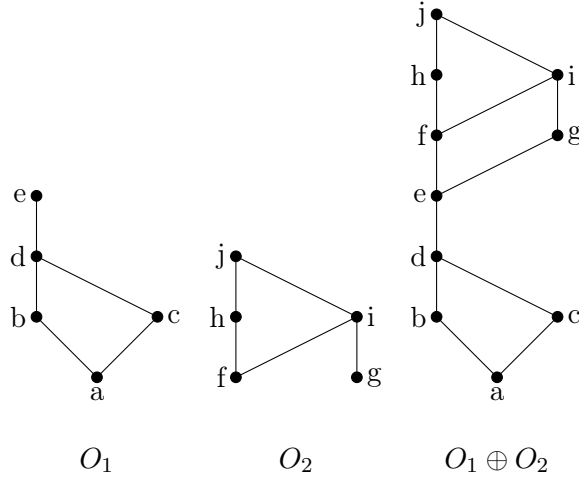


FIGURE 1.2 – Le diagramme de Hasse de la somme ordinale de deux ordres O_1 et O_2 .

Le dual est également un morphisme (il s’agit plus précisément d’un isomorphisme).

Définition 1.11 (Préordre dual) *Soit un préordre (E, \preceq) . (E, \succeq) est le préordre dual de (E, \preceq) si et seulement si \succeq est la relation duale de \preceq (définition A.21).*

À toute relation d’ordre ou de préordre on peut associer une relation dite *stricte*. Il ne s’agit cependant pas d’une relation d’ordre ou de préordre au sens formel du terme car elle n’est pas réflexive.

Définition 1.12 (Relation stricte) *Soit une relation de préordre \preceq . On appelle relation de préordre stricte associée à \preceq la relation \prec telle que $\prec = (\preceq \setminus =)$.*

Remarque 1.3 *Si \leq est une relation d’ordre, nous noterons $<$ la relation d’ordre stricte associée.*

1.1.2 Définition des minimums

Le terme *optimum* recouvre les définitions de *minimum* et de *maximum*. Le maximum d’un préordre étant équivalent au minimum de son dual, nous ne donnerons que les définitions du minimum.

Définition 1.13 (Minimums d’un ordre) *Soit un ordre (E, \leq) . L’ensemble des minimums de (E, \leq) , noté $\min(E, \leq)$, est défini par*

$$\min(E, \leq) := \{e^* \in E : \forall e \in E, e \leq e^* \Rightarrow e = e^*\}$$

Ce qui peut s’interpréter comme l’ensemble des éléments tels que tous les éléments qui leur sont inférieurs ne sont autres qu’eux-mêmes. Autrement dit, il n’existe pas d’autres éléments qui leurs soient inférieurs. On a donc

$$\min(E, \leq) = \{e \in E : \downarrow_{(E, \leq)} e = \{e\}\} \tag{1.1}$$

avec $\downarrow_{(E, \leq)} e$ l’idéal de e dans (E, \leq) (définition A.18).

Remarque 1.4 *La définition 1.13 n’est valable que si (E, \leq) est un ordre.*

En effet, une telle définition appliquée à un préordre pourrait donner des configurations où il n'y aurait pas de minimum.

Exemple 1.3 Soit un préordre $P = (\{a, b\}, \{(a, b), (b, a)\})$. L'application du minimum d'un ordre donnerait $\min P = \emptyset$ puisqu'on a $b \preceq a$, donc a n'est pas un minimum, mais $a \preceq b$, donc b n'est pas minimum non plus. Or ce sont les seuls éléments de P .

Dans l'exemple 1.3, on peut considérer que a et b sont tous deux des minimums car $a \equiv b$ et il n'existe pas d'autre élément. On généralise alors la définition 1.13 en changeant le critère d'égalité par l'équivalence.

Définition 1.14 (Minimums d'un préordre) Soit un préordre (E, \preceq) . L'ensemble des minimums de (E, \preceq) est défini par

$$\min(E, \preceq) := \{e^* \in E : \forall e \in E, e \preceq e^* \Rightarrow e \equiv e^*\}$$

Tout comme les minimums d'un ordre, tout élément inférieur à un minimum appartient à sa classe d'équivalence. Soit

$$\min(E, \preceq) = \{e \in E : \downarrow_{(E, \preceq)} e = [e]_{\equiv}\} \quad (1.2)$$

Une autre interprétation équivalente est de voir les minimums d'un préordre comme les minimums de son quotient. Soit P un préordre et (P/\equiv) son ordre quotient :

$$\min P = \bigcup \min(P/\equiv) \quad (1.3)$$

Remarque 1.5 $\min P/\equiv$ étant un ordre, on peut lui appliquer le min de la définition 1.13. Ceci peut être intéressant d'un point de vue algorithmique.

À partir du minimum, pour un ensemble dénombrable, nous pouvons définir le rang d'un élément : un élément minimum a un rang de 1, puis les minimums de l'ensemble privé des éléments de rang 1 ont un rang 2, etc.

Définition 1.15 (Rang d'un élément) Soit un préordre (E, \preceq) tel que E soit dénombrable. Le rang de a dans (E, \preceq) est une application r à valeurs dans \mathbb{N}^* :

$$r(a) := \begin{cases} 1 & \text{si } a \in \min(E, \preceq) \\ \max\{r(b) : b \prec a\} + 1 & \text{sinon} \end{cases}$$

Remarque 1.6 Le rang d'un élément dans un préordre est le rang de sa classe d'équivalence dans son ordre quotient.

Remarque 1.7 Le rang permet d'obtenir une partition ordonnée du préordre considérée.

1.2 Problème d'optimisation

Un problème d'optimisation est la recherche des minimums d'une fonction. En recherche opérationnelle, les problèmes d'optimisation doivent prendre en compte les préférences du décideur sur les objectifs ou sur les solutions du problème. Ainsi, nous présentons, dans un second temps, une modélisation des problèmes d'optimisation permettant l'intégration de ces préférences.

1.2.1 Minimum d'une fonction

Un problème d'optimisation est la minimisation d'une fonction appelée *fonction coût* sur un ensemble donné.

Dans un espace \mathcal{X} appelé *espace des variables*, nous nous intéressons aux solutions admissibles du problème, c'est-à-dire l'ensemble des solutions respectant les contraintes du problème. Nous notons $X \subseteq \mathcal{X}$ cet ensemble. Soient ϕ la fonction coût, et $Z \subseteq \mathcal{Z}$ l'image de X par ϕ . Minimiser ϕ sur X consiste à trouver, d'une part, l'ensemble des minimaux de (Z, \preceq_Z) , noté

$$\min_{x \in X} \phi(x) := \min(Z, \preceq_Z) \quad (1.4)$$

et, d'autre part, l'ensemble des éléments de X dont l'image est un minimum de (Z, \preceq_Z) noté

$$\arg \min_{x \in X} \phi(x) := \{x \in X : \phi(x) \in \min(Z, \preceq_Z)\} \quad (1.5)$$

Remarque 1.8 *Maximiser ϕ est équivalent à minimiser $-\phi$, nous nous placerons toujours dans le cas de la minimisation.*

En pratique, nous formulerons un problème d'optimisation \mathcal{P} sous la forme

$$\min_{x \in X} \phi(x) \quad (1.6)$$

Les images des variables pouvant être déduites du problème par application de la fonction coût, nous appellerons seulement *solution* les éléments de \mathcal{X} . Nous distinguons cependant trois sous-ensembles de solutions : les *solutions*, les *solutions admissibles* et les *solutions optimales*.

Définition 1.16 (Solution) *Nous appelons solution de \mathcal{P} tout élément x de \mathcal{X} .*

Définition 1.17 (Solution admissible) *Nous appelons solution admissible de \mathcal{P} tout élément x de l'ensemble admissible X .*

Définition 1.18 (Solution optimale) *Nous appelons solution optimale de \mathcal{P} tout élément x de l'ensemble $\arg \min_{x \in X} \phi(x)$.*

Les solutions de \mathcal{P} ne vont dépendre que de la définition de l'espace \mathcal{X} . Les solutions admissibles vont, de plus, dépendre des contraintes du problème. Les solutions optimales vont, quant à elles, dépendre de la relation \preceq_Z appliquée à Z , induite par ϕ . Dans un problème d'aide à la décision, la relation \preceq_Z peut intégrer les préférences d'un décideur.

1.2.2 Prise en compte des préférences

Les problèmes de recherche opérationnelle font intervenir un décideur qui choisit les solutions du problème en fonction de ses préférences. Ces préférences peuvent être intégrées au problème d'optimisation *a priori*, *a posteriori* ou au cours du processus d'optimisation (on parle dans ce cas de méthode interactive). Nous ne traiterons pas ce dernier cas. Les préférences *a priori* sont prises en compte dans la fonction objectif ϕ et influencent la relation \preceq_Z . Ce n'est pas le cas des préférences *a posteriori* qui, par définition, ne sont prises en compte qu'après l'obtention des solutions optimales. Afin

de séparer les critères objectifs (intrinsèques au problème) et subjectifs (du décideur), nous distinguons deux composantes de ϕ qui sont f , les critères objectifs et θ les critères subjectifs, avec $\phi = \theta \circ f$. Nous distinguons alors trois espaces¹ (Figure 1.3) :

L'espace de décision ou espace des variables, que nous noterons \mathcal{X} . C'est l'espace contenant les variables du problème d'optimisation notées x , c'est-à-dire les alternatives parmi lesquelles le décideur doit sélectionner les meilleures. Il est également appelé espace des solutions, car c'est l'instanciation des variables du problème dans cet espace qui définit une solution.

L'espace des objectifs que nous noterons \mathcal{Y} . C'est l'espace dans lequel les *performances* des solutions sont représentées.

L'espace de comparaison ou espace ordonné, que nous noterons \mathcal{Z} . C'est l'espace dans lequel les solutions sont comparées entre elles en fonction de leurs performances, afin de pouvoir sélectionner les « meilleures ». Pour cela, il est muni d'une structure d'ordre \preceq . Cet espace est souvent confondu avec l'espace des objectifs, en particulier en optimisation mono-objectif.

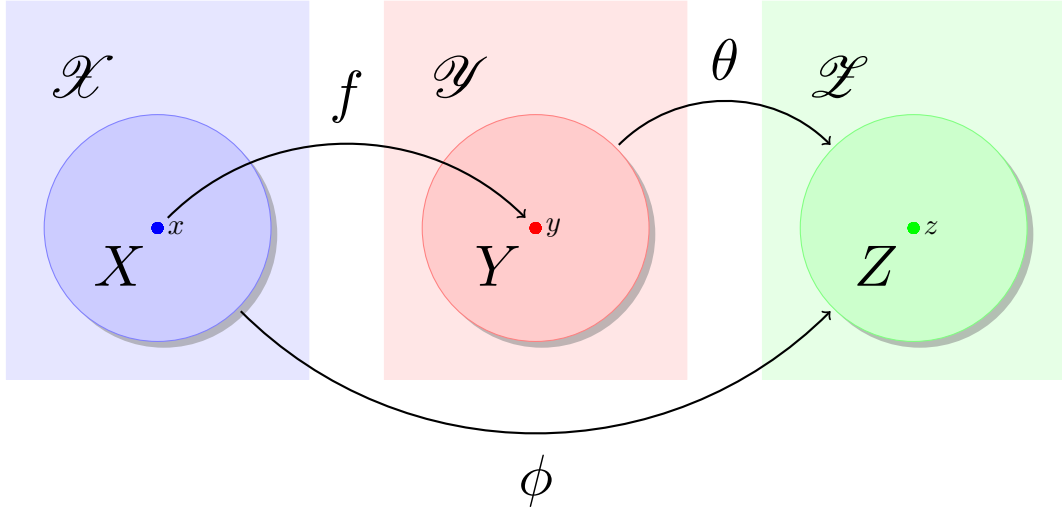


FIGURE 1.3 – Trois espaces sont à considérer dans un problème d'optimisation. L'espace des variables \mathcal{X} , l'espace des objectifs \mathcal{Y} et l'espace de comparaison \mathcal{Z} .

Nous appellerons *ensemble réalisable* l'image des solutions admissibles dans l'espace des objectifs. Nous noterons $Y \subseteq \mathcal{Y}$ cet ensemble défini ainsi :

$$Y := f(X) \tag{1.7}$$

Enfin, nous noterons $Z \subseteq \mathcal{Z}$ l'image de Y par θ :

$$Z := \theta(Y) \tag{1.8}$$

Cet ensemble est muni d'une structure d'ordre \preceq_Z qui peut être différente de l'ordre de l'espace des objectifs \preceq_Y . On souhaite cependant que les préférences de (Y, \preceq_Y) soient conservées dans (Z, \preceq_Z) . On impose donc que θ soit un morphisme de (Y, \preceq_Y) dans (Z, \preceq_Z) .

1. Cette distinction en trois espaces et deux fonctions est inspirée de (Ehrgott 2005).

Remarque 1.9 Y est un espace produit. (Y, \leq_Y) est donc le produit des ordres de chaque objectif. Il s'agit donc d'un ordre.

Remarque 1.10 En pratique, on a souvent $\mathcal{X} = \mathbb{R}^n$, $\mathcal{Y} = \mathbb{R}^p$ et $\mathcal{Z} = \mathbb{R}^q$ où n est le nombre de variables du problème, p le nombre d'objectifs et q un entier strictement positif tel que $q \leq p$.

En posant $\phi = \theta \circ f$, on retrouve le problème 1.6. Le problème d'optimisation est alors

$$\min_{x \in X} \theta \circ f(x) \tag{1.9}$$

Remarque 1.11 En optimisation mono-objectif – c'est-à-dire les problèmes d'optimisation pour lesquels f est à valeurs dans \mathbb{R} – l'espace des critères et l'espace de comparaison sont confondus. En effet, les performances des solutions sont comparées dans l'ordre usuel de \mathbb{R} , on a ainsi $\theta = \text{id}$ (la fonction identité) et le problème d'optimisation est alors

$$\min_{x \in X} f(x) \tag{1.10}$$

Cependant, dans le cas de l'optimisation multiobjectif, la définition d'optimalité n'est pas unique. Plusieurs définitions de θ différentes peuvent être appliquées sur le même espace des objectifs, aboutissant ainsi à plusieurs définitions de l'optimalité.

Exemple 1.4 Imaginons une classe de vingt élèves participant à un concours dont chaque élève est jugé par une note sur 20 dans deux épreuves : mathématiques et anglais. Une note inférieure à 10/20 dans au moins une des épreuves est éliminatoire. Seuls cinq élèves ont obtenu des résultats les qualifiant : Arthur, Bénédicte, Camille, Denis, et Émilie. Leurs notes sont données dans le tableau ci-dessous.

X	Mathématiques	Anglais
Arthur	12	13
Bénédicte	18	10
Camille	13	17
Denis	18	13
Émilie	10	17

Nous pouvons modéliser le problème comme un problème d'optimisation en posant \mathcal{X} l'ensemble des élèves de la classe. Les applications $m : \mathcal{X} \rightarrow [0; 20]$ et $a : \mathcal{X} \rightarrow [0; 20]$ correspondent respectivement aux notes de mathématiques et d'anglais. L'ensemble admissible $X = \{x \in E : m(x) \geq 10, a(x) \geq 10\}$ représente l'ensemble des élèves ayant une note supérieure à 10 dans chacune des matières, soit $X = \{\text{Arthur, Bénédicte, Camille, Denis, Émilie}\}$. Le problème consiste à sélectionner les élèves x parmi X maximisant $m(x)$ et $a(x)$. Il s'agit d'un problème biobjectif à valeurs dans $[0; 20]^2 \subset \mathbb{R}^2$. La fonction objectif est $f := (m, a)$ et l'ensemble réalisable $Y = \{(12, 13), (18, 10), (13, 17), (18, 13), (10, 17)\}$. Mais comment décider lequel des cinq est le meilleur élève ? La méthode la plus souvent utilisée dans ce cas est la moyenne pondérée des notes en choisissant a priori un jeu de pondération $(\alpha, 1 - \alpha)$ avec $\alpha \in [0; 1]$. Pour chaque x , on cherche alors la valeur maximale de $\alpha m(x) + (1 - \alpha)a(x)$. L'application θ est alors

$$\begin{aligned} \theta : [0; 20]^2 &\rightarrow \mathbb{R} \\ (y_1, y_2) &\mapsto \alpha y_1 + (1 - \alpha)y_2 \end{aligned}$$

Le problème d'optimisation est alors

$$\max_{x \in X} \theta \circ f(x) := \alpha m(x) + (1 - \alpha)a(x)$$

L'espace réalisable est un sous-ensemble de \mathbb{R}^2 , mais les solutions sont comparées dans \mathbb{R} . On a ici $(Z, \preceq) = (\mathbb{R}, \leq)$. Pour $\alpha = 0,5$, c'est Denis qui obtient les meilleures performances, alors qu'avec $\alpha = 0,3$ c'est Camille.

1.2.3 Solutions d'un problème d'optimisation

Lorsque la fonction objectif f est à valeurs dans \mathbb{R} , l'ensemble des solutions d'un problème d'optimisation est usuellement défini par

$$\arg \min_{x \in X} f(x) := \{x^* \in X : \forall x \in X, f(x^*) \leq f(x)\} \quad (1.11)$$

où \leq est l'ordre usuel des nombres réels. Ces définitions du minimum ne fonctionnent cependant plus dans le cas où f est à valeurs dans \mathbb{R}^p avec $p \in \mathbb{N} \setminus \{0, 1\}$ car l'ordre considéré est alors partiel. Une première approche consiste à se ramener à un ordre total. On doit donc trouver une application θ telle que (Z, \leq_Z) soit une extension linéaire de (Y, \leq_Y) .

La manière la plus courante d'y procéder est d'utiliser une somme pondérée comme illustré dans l'exemple 1.4. On cherche alors les solutions du problème \mathcal{P}_ω suivant :

$$\min_{x \in X} \sum_{i=1}^p \omega_i f_i(x) \quad (1.12)$$

où $\omega \in [0; 1]^p$ est le vecteur des pondérations tel que $\sum_{i=1}^p \omega_i = 1$. Le minimum est alors

$$\arg \min_{x \in X} f(x) := \{x^* \in X : \forall x \in X, \theta(f(x^*)) \leq \theta(f(x))\} \quad (1.13)$$

où \leq est l'ordre usuel des nombres réels, et l'application θ :

$$\theta : y \mapsto \sum_{i=1}^p \omega_i y_i \quad (1.14)$$

Remarque 1.12 On peut chercher plusieurs solutions en faisant varier le vecteur ω .

On peut également se ramener à un ordre total si l'on ajoute un ordre total sur les composantes des éléments de Y . On obtient alors un ordre appelé *ordre lexicographique*. L'optimisation lexicographique consiste alors à se donner un ordre total de préférence sur les fonctions objectifs, puis à commencer à optimiser suivant le critère le plus important, puis suivant le second critère tel que la solution soit au moins toujours aussi bonne pour le premier critère, et ainsi de suite.

Définition 1.19 (Ordre lexicographique) Soient a et b deux points de \mathbb{R}^p , nous définissons la relation d'ordre $(\mathbb{R}^p, \leq_{lex})$ telle que

- i) $a <_{lex} b \Leftrightarrow \exists k \in \llbracket p \rrbracket$ t.q. $\forall i < k, a_i = b_i, a_k < b_k$
- ii) $(a \leq_{lex} b) \Leftrightarrow (a <_{lex} b) \vee (a = b)$

Exemple 1.5 Dans l'exemple 1.4, on a deux objectifs : la note de mathématiques m , et la note d'anglais a . En notant $f_1 := m$ et $f_2 := a$ les deux objectifs, nous avons deux permutations possibles et donc deux ordres lexicographiques associés :

1. $\pi = (1, 2)$: les mathématiques sont plus importants que l'anglais. L'ordre lexicographique correspondant est $\{(10, 17), (12, 13), (13, 17), (18, 10), (18, 13)\}$. C'est Denis qui maximise cet ordre.
2. $\pi = (2, 1)$: l'anglais est plus important que les mathématiques. L'ordre lexicographique correspondant est $\{(18, 10), (12, 13), (18, 13), (10, 17), (13, 17)\}$. C'est Camille qui maximise cet ordre.

L'obtention d'un ordre total à partir d'un ordre partiel suppose que l'on ait ajouté des préférences à l'espace des objectifs. Une autre approche consiste à ne pas faire de choix *a priori* sur l'importance relative des objectifs. On a donc $\theta = id$. On considère alors qu'une solution est meilleure qu'une autre si elle a au moins d'aussi bonnes performances sur tous les objectifs et qu'elle est strictement plus performante sur au moins un objectif. C'est le principe de la dominance de Pareto.

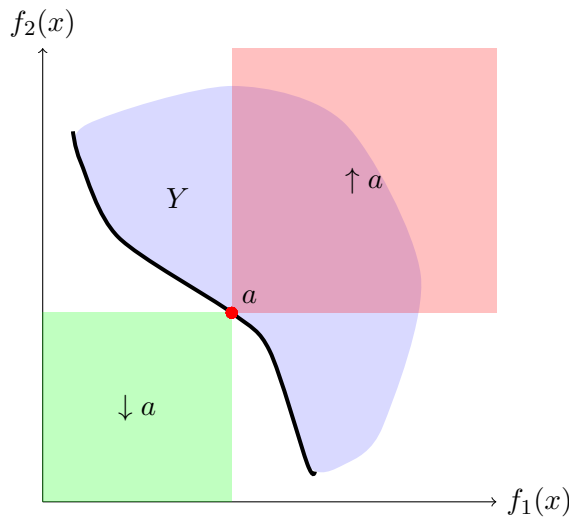


FIGURE 1.4 – La ligne en gras représente le front de Pareto d'un problème de minimisation biobjectif dont l'ensemble réalisable Y . $\downarrow a$ représente le sous-ensemble de l'espace des objectifs qui domine a , tandis que $\uparrow a$ représente le sous-ensemble de l'espace des objectifs dominé par a . L'élément a appartient au front de Pareto car il n'existe pas de solution qui le domine.

Définition 1.20 (Dominance de Pareto) Soient $a, b \in Y \subseteq \mathbb{R}^p$, on dit que a domine b au sens de Pareto pour un problème de minimisation si et seulement si a est inférieur à b sur chaque composante et s'il existe au moins une composante pour laquelle l'infériorité est stricte :

$$a \lesssim_{\mathbb{R}^p} b \iff \begin{cases} \forall i \in [p] & a_i \leq_{\mathbb{R}} b_i \\ \exists j \in [p] & a_j <_{\mathbb{R}} b_j \end{cases}$$

Les meilleures solutions seront alors celles qui ne sont pas dominées. Soit Y^N l'ensemble des points non dominés de Y :

$$Y^N := \{y^* \in Y : \nexists y \in Y \text{ t.q. } y \lesssim_{\mathbb{R}^p} y^*\} \quad (1.15)$$

L'ensemble des points non dominés est également appelé *front de Pareto*. Nous appelons *solution efficace* l'antécédent par f d'un point *non dominé*. Soit X^E l'ensemble des

solutions efficaces du problème :

$$X^E := \{x \in X : f(x) \in Y^N\} \quad (1.16)$$

L'ensemble des solutions efficaces est également appelé *ensemble de Pareto*. L'ensemble de Pareto est donc l'ensemble des solutions tel qu'il n'existe aucune solution qui soit meilleure sur tous les objectifs à la fois, ou autrement dit, on ne peut pas améliorer une solution de l'ensemble de Pareto sur un critère sans la dégrader sur un autre.

Exemple 1.6 *Dans l'exemple 1.4, les résultats d'Arthur, Bénédicte et Émilie sont dominés. Nous avons $X^E = \{\text{Camille}, \text{Denis}\}$ et $Y^N = \{(13, 17), (18, 13)\}$.*

\mathbb{R}^p est un espace produit. L'ordre le plus naturel à appliquer sur un tel espace est donc le produit direct de ses composantes, soit

$$(\mathbb{R}^p, \leq_{\mathbb{R}^p}) := \prod_{i=1}^p (\mathbb{R}, \leq_{\mathbb{R}}) \quad (1.17)$$

On peut remarquer que la relation d'ordre produit $\leq_{\mathbb{R}^p}$ est l'ordre large associé à la relation de dominance de Pareto \preceq qui est un ordre strict². Autrement dit :

$$\leq_{\mathbb{R}^p} = (\preceq_{\mathbb{R}^p} \cup =) \quad (1.18)$$

La figure 1.4 illustre l'ensemble réalisable ainsi que le front de Pareto. Le point a est non dominé car l'ensemble $\downarrow a$ ne contient pas d'autre élément que a .

Remarque 1.13 *Nous notons $\downarrow a$ le cône de dominance de a qu'Ehrgott (2005) note $a - \mathbb{R}_{\geq}^p$ où p est la dimension de l'espace des objectifs et \geq le dual de la dominance (que nous notons \succeq). Ce cône correspond cependant à l'idéal de a (définition A.18) en théorie des ordres.*

Une autre définition de l'ensemble des points non dominés de Y est donc

$$Y^N := \{y \in Y : \downarrow y = \{y\}\} \quad (1.19)$$

Autrement dit, le minimum au sens de Pareto n'est autre que le minimum classique de l'ordre produit. Nous pouvons ainsi définir les minimums d'un problème d'optimisation dans le cas général. Nous distinguons les minimums globaux des minimums locaux.

Définition 1.21 (Minimum global d'une fonction) *Soient ϕ une fonction de \mathcal{X} dans \mathcal{Z} , X l'ensemble réalisable, Z l'image de X par ϕ et \preceq_Z la relation de préordre sur Z . On dit que x^* réalise un minimum global de ϕ sur X si et seulement si*

$$\forall x \in X, \phi(x) \preceq_Z \phi(x^*) \Rightarrow \phi(x) \equiv \phi(x^*)$$

Définition 1.22 (Minimum local d'une fonction) *Soient ϕ une fonction de \mathcal{X} dans \mathcal{Z} , X l'ensemble réalisable, Z l'image de X par ϕ et \preceq_Z la relation de préordre sur Z . On dit que x^* réalise un minimum local de ϕ sur X si et seulement si il existe une boule ouverte $\mathcal{B}(x^*, \rho)$ de centre x^* et de rayon $\rho > 0$ telle que*

$$\forall x \in X \cap \mathcal{B}(x^*, \rho), \phi(x) \preceq_Z \phi(x^*) \Rightarrow \phi(x) \equiv \phi(x^*)$$

Cette distinction est importante, car certains algorithmes échouent à trouver les minimums globaux et convergent vers des minimums locaux.

2. C'est pourquoi nous utilisons le symbole \preceq : il symbolise la non-réflexivité de la relation, tout en la dissociant de la relation $<$, que nous réservons pour la relation d'ordre stricte suivante : $a < b \Leftrightarrow \forall i \in \llbracket p \rrbracket, a_i < b_i$.

1.3 Algorithmes d'optimisation

Il existe plusieurs types d'algorithmes d'optimisation suivant le type de problème traité et de solution que l'on souhaite obtenir. Nous nous intéressons particulièrement aux algorithmes évolutionnaires, qui sont des métaheuristiques à population. Dans cette section, nous situons les algorithmes évolutionnaires parmi les autres types d'algorithmes existants et nous en présentons brièvement les principaux, qui sont les algorithmes génétiques et les stratégies d'évolution.

1.3.1 Métaheuristiques

Nous classons les algorithmes d'optimisation suivant trois grands critères :

Constructif vs Itératif Certains algorithmes d'optimisation aboutissent à une solution par construction en affectant progressivement des valeurs à toutes les variables du problème jusqu'à obtenir une solution totalement définie. D'autres convergent vers une solution par améliorations successives d'une solution. C'est le cas des algorithmes d'optimisation numérique. Nous appellerons ces derniers algorithmes d'optimisation *itératifs*, par opposition aux premiers, que nous nommerons *constructifs*.

Exact vs Approché Pour certains problèmes difficiles³ une solution approchée est parfois suffisante. On utilise alors dans ce cas un algorithme *approché*, par opposition à un algorithme *exact*, donnant de façon certaine une solution optimale.

Déterministe vs Stochastique Le résultat d'un algorithme d'optimisation peut être *déterministe*, c'est-à-dire que chaque exécution de l'algorithme donnera la même solution. Les algorithmes *non déterministes* sont dits *stochastiques*, leur évolution au cours du temps est déterminée par une fonction aléatoire. L'algorithme du gradient est par exemple déterministe, tandis que le recuit simulé est stochastique.

Les métaheuristiques constituent un type d'algorithme d'optimisation approché stochastique. Elles permettent de résoudre des problèmes d'optimisation dits *difficiles*. Il s'agit généralement de méthodes inspirées de phénomènes naturels (physiques ou biologiques) ayant un caractère stochastique. Elles ont la particularité de pouvoir s'extraire d'optimums locaux en acceptant temporairement une dégradation de la solution. Les métaheuristiques fonctionnent en deux temps : une phase de diversification, où on explore l'espace de recherche ; une phase d'intensification, où l'algorithme va se concentrer sur les solutions prometteuses.

Il existe de nombreuses métaheuristiques. Elles ont l'avantage de pouvoir aisément s'adapter à l'optimisation multiobjectif et multimodale⁴. Elles sont également naturellement parallélisables sur plusieurs unités de calcul. Cependant, s'il existe un algorithme exact et efficace pour résoudre le problème considéré, il est alors judicieux de l'utiliser plutôt qu'une métaheuristique.

3. On entend par *problèmes d'optimisation difficiles* les problèmes d'optimisation combinatoires pour lesquels il n'existe pas de méthode de résolution exacte en temps polynomial, ou les problèmes d'optimisation à variables continues pour lesquels nous ne connaissons pas de méthode d'optimisation trouvant l'optimum global en un temps donné.

4. Un problème d'optimisation est multimodal quand il admet plusieurs solutions.

1.3.2 Algorithmes d'optimisation itératifs et à population

Idéalement, un algorithme d'optimisation doit donner un ensemble $X^* \subseteq X$ correspondant à l'ensemble des solutions optimales du problème. Pour cela, il doit être capable d'atteindre l'intégralité de l'ensemble admissible. Cependant, l'espace de recherche de l'algorithme, du fait du codage et des fonctions d'exploration, diffère de X . Nous noterons \mathcal{X} cet espace.

En nous inspirant de la définition d'algorithme d'optimisation de Bergounioux (2001), nous définissons les algorithmes d'optimisation itératifs de la manière suivante :

Définition 1.23 (Algorithme d'optimisation itératif) *Un algorithme d'optimisation itératif est défini par une application \mathcal{A} de \mathcal{X} dans \mathcal{X} permettant de générer une suite d'éléments de \mathcal{X} par la formule :*

$$\begin{cases} x_0 \in \mathcal{X} \\ x_{k+1} = \mathcal{A}(x_k) \end{cases}$$

Il s'agit donc de se donner une suite $(x_k)_{k \in \mathbb{N}}$ de \mathcal{X} .

Définition 1.24 (Convergence d'un algorithme d'optimisation) *On dit que l'algorithme d'optimisation \mathcal{A} converge si la suite $(x_k)_{k \in \mathbb{N}}$ engendrée par l'algorithme converge vers une limite \hat{x} .*

La valeur de \hat{x} peut être différente de x^* pour plusieurs raisons. Premièrement, il peut s'agir d'un minimum local et non global. De plus, \hat{x} est un élément de \mathcal{X} et non de X .

Exemple 1.7 *Les algorithmes d'optimisation itératifs les plus répandus sont des algorithmes de descente, dont fait partie l'algorithme du gradient. Dans le cadre d'une minimisation, les méthodes de descente sont des algorithmes d'optimisation dont le principe est de minimiser une fonction objectif f en partant d'un point initial x_0 , puis pour chaque point x_k , de déterminer le point suivant x_{k+1} tel que $f(x_{k+1}) < f(x_k)$. En pratique, on construit une suite*

$$\begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = x_k + \rho_k d_k \end{cases}$$

Dans le cas général (fonction non convexe), les algorithmes de descente convergent vers un minimum local et non global.

Cependant, les problèmes d'optimisation n'ont pas nécessairement de solution unique, et il peut être intéressant de connaître toutes les solutions ou un sous-ensemble de solutions. Un problème d'optimisation admettant plusieurs solutions est qualifié de *multimodal*.

Définition 1.25 (Problème d'optimisation multimodal) *Un problème d'optimisation est dit multimodal s'il admet plus d'une solution optimale.*

Une des méthodes pour trouver plusieurs solutions d'un problème d'optimisation multimodal est de lancer plusieurs algorithmes d'optimisation en partant de différents points de départ (différentes valeurs de x_0). Une autre méthode est non plus de considérer un unique point, mais un ensemble de points appelé population. Un algorithme d'optimisation à population fait alors évoluer un ensemble de solutions par itérations successives.

Définition 1.26 (Algorithme d'optimisation à population) *Un algorithme d'optimisation à population est défini par une application \mathcal{A}_p de $\mathcal{P}(\mathcal{X})$ dans $\mathcal{P}(\mathcal{X})$ permettant de générer une suite de sous-ensembles de \mathcal{X} par la formule :*

$$\begin{cases} x_0 \subseteq \mathcal{X} \\ x_{k+1} = \mathcal{A}_p(x_k) \end{cases}$$

L'ensemble x_k obtenu à l'itération k est alors la k -ième génération de la population.

Un algorithme d'optimisation à population est dit *élitiste* si des solutions optimales obtenues lors d'une itération sont conservées à l'itération suivante.

Définition 1.27 (Algorithme d'optimisation à population élitiste) *Soit un algorithme d'optimisation à population \mathcal{A}_p . Il est dit élitiste si pour toute génération $k \geq 1$ on a $\min x_{k-1} \subseteq x_k$.*

1.3.3 Principe des algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des métaheuristiques à population s'inspirant de la théorie de l'évolution de Darwin (1869). Il en existe plusieurs variantes suivant le schéma général suivant :

1. Initialisation : création d'une population aléatoire ou suivant l'expérience du concepteur,
2. Sélection d'une population de parents,
3. Évolution des parents donnant une population d'enfants,
4. Sélection de la nouvelle génération à partir de la génération précédente et des enfants,
5. Revenir à l'étape 2 tant que le critère d'arrêt n'est pas atteint.

Les différences entre les algorithmes évolutionnaires vont apparaître dans le choix des opérations d'initialisation, de sélection et d'évolution, ainsi que dans le codage des individus. Deux des grandes catégories d'algorithmes évolutionnaires furent développées indépendamment : les algorithmes génétiques et les stratégies d'évolution.

Les algorithmes génétiques (*Genetic Algorithms* - GA) sont apparus dans les années 70 et leur invention est attribuée à Holland (1962; 1992) popularisée par Goldberg (1989). Les individus d'un algorithme génétique sont historiquement codés sous la forme d'une chaîne de bits. Deux opérations d'évolution sur ces individus sont effectuées à chaque génération : le croisement, qui à partir de deux parents, crée un ou plusieurs enfants partageant une partie du code des deux parents (une sous-chaîne de bits); la mutation, qui consiste en la modification aléatoire d'un ou plusieurs bits sur un individu.

Les stratégies d'évolution (*Evolution Strategies* - ES) ont été développées parallèlement aux algorithmes génétiques. Leur apparition est attribuée à Rechenberg (1965).

Les principales différences entre les premières stratégies d'évolution et les premiers algorithmes génétiques sont, d'une part, que les ES utilisent un codage réel tandis que les GA utilisent un codage binaire, et d'autre part que les ES n'ont pas d'opérateur de croisement : ils n'effectuent que des mutations. Ces différences tendent cependant à s'estomper puisqu'il existe maintenant des ES avec des opérateurs de croisement et des GA à codage réel.

Les stratégies d'évolution sont décrites à l'aide de deux entiers μ et λ et deux opérateurs (+) et (,) composant ces symboles. Les symboles μ et λ représentent respectivement

les tailles de la population de parents et le nombre d'enfants créés. La différence entre les stratégies $(\mu + \lambda)$ et (μ, λ) vient de la population utilisée pour la sélection : la première utilise la population de parents à laquelle on ajoute les enfants, tandis que la seconde n'utilise que la population d'enfants.

Les stratégies d'évolution utilisant des croisements sont notées $(\mu/\rho + \lambda)$ et $(\mu/\rho, \lambda)$, où ρ indique le nombre d'individus à recombinaison parmi μ parents avant de les muter. Il s'agit d'un croisement à ρ individus.

1.4 Génotype : codage des individus et opérateurs génétiques

Les algorithmes évolutionnaires utilisent l'analogie avec les lois de la génomique. On distingue le codage d'une solution (le gène - le génotype) avec sa valeur (l'expression du gène - le phénotype). Les différentes opérations sont réalisées sur le génotype et les individus sont évalués sur leur phénotype. Dans notre modélisation des problèmes d'optimisation, le génotype appartient à l'espace de décision \mathcal{X} tandis que le phénotype est dans l'espace des objectifs \mathcal{Y} .

Un algorithme d'optimisation à population doit être capable d'atteindre l'intégralité de l'ensemble admissible sous peine d'omettre des solutions. Nous noterons \mathcal{X} l'ensemble des solutions atteignables par l'algorithme que nous appellerons espace de recherche. Dans l'idéal, si l'on ne considère pas l'approximation des solutions réelles par des nombres flottants, on a $\mathcal{X} = X$ ou $\mathcal{X} \supseteq X$. En effet, l'espace de recherche est souvent plus grand que l'ensemble admissible car on peut s'autoriser à explorer des solutions ne vérifiant pas les contraintes.

La structure de voisinage de \mathcal{X} doit être adaptée au problème traité. Cette structure est définie par le codage des individus et par les opérateurs de croisement et de mutation.

Il existe une grande variété de codages et d'opérateurs de croisement et mutation. Dans cette section, nous ne présentons que les principaux, afin d'illustrer leur influence sur la définition de l'espace de recherche et la différence entre cet espace et l'espace de décision théorique.

1.4.1 Codage des individus

Historiquement, les variables des algorithmes génétiques sont représentées par des chaînes de bits. Ces chaînes peuvent être interprétées de différentes manières. Nous prenons l'exemple de leur interprétation comme nombres binaires et comme code de Gray. Une alternative est de simplement les coder dans leur représentation réelle.

Codage binaire Le codage binaire est le plus simple et le plus répandu. Il simplifie les opérateurs de croisement et de mutation. Chaque variable est transformée en une suite de 0 et de 1. Le nombre de bits utilisés dépend évidemment du nombre de valeurs que l'on doit représenter et donc il faut fournir des bornes aux variables ainsi qu'une précision souhaitée.

Codage de Gray Un code de Gray (1953) est un type de codage binaire tel que le passage d'un nombre au suivant ne se fasse qu'en modifiant un seul bit (Tableau 1.1).

base 10	base 2	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

TABLEAU 1.1 – Comparaison de l'interprétation d'une chaîne de bits en nombre binaire et en code de Gray sur des nombres entiers de 0 à 7.

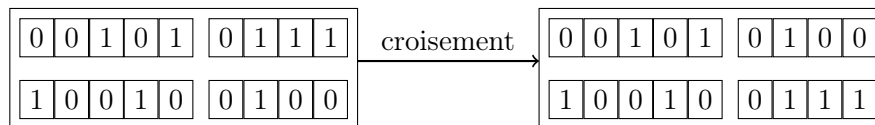


FIGURE 1.5 – Exemple de croisement binaire en un point entre deux parents.

Codage réel Le codage réel est le plus simple puisque chaque variable est directement représentée par sa valeur. Un réel est alors représenté par un nombre flottant, un entier par un entier, etc. Le codage réel est mieux adapté aux problèmes d'optimisation continus. Cependant, les opérateurs de croisement et de mutation sont plus délicats à écrire.

Codage spécifique Dans les problèmes d'optimisation combinatoires, il convient souvent d'utiliser un codage spécifique à la structure du problème. Par exemple, dans le problème de voyageur de commerce, un codage naturel d'une solution est d'utiliser une liste de villes qui représentera l'ordre de parcours. Dans ce cas, le codage est une *permutation*.

Suivant le codage utilisé, différents opérateurs de croisement et de mutation sont définis.

1.4.2 Croisements

L'opérateur de croisement consiste à partir de deux solutions appelées *parents*, à créer de nouvelles solutions (généralement deux) appelées *enfants*. Les nouvelles solutions ainsi créées partagent, en théorie, des caractéristiques des deux parents. Les parents ayant été sélectionnés sur des critères de performance, on espère ainsi créer des enfants combinant les bonnes caractéristiques de chacun des parents.

Nous ne présentons ici que les opérateurs de croisement les plus utilisés, à savoir le croisement en un point pour un codage binaire, et le croisement binaire simulé pour un codage réel.

Croisement en un point Le croisement en un point est l'opérateur de croisement le plus courant et le plus simple. Il consiste à croiser deux parents qui sont représentés par une chaîne binaire en choisissant un point de croisement entre le début et la fin de la chaîne puis en échangeant, d'un parent à l'autre, la partie située au-delà de ce point.

Le point de croisement est choisi aléatoirement. Évidemment, on peut généraliser ce croisement à plusieurs points.

Croisement binaire simulé Deb et Agrawal (1995) ont proposé un croisement pour des variables réelles simulant le croisement binaire. Ils l'ont nommé *SBX* (*Simulated Binary Crossover*). Ils observent qu'un enfant est proche de l'un des deux parents. Pour deux parents éloignés, trouver une valeur pour l'enfant située au milieu des deux n'aurait pas beaucoup de sens. Cet opérateur va privilégier les valeurs autour des parents (figure 1.6).

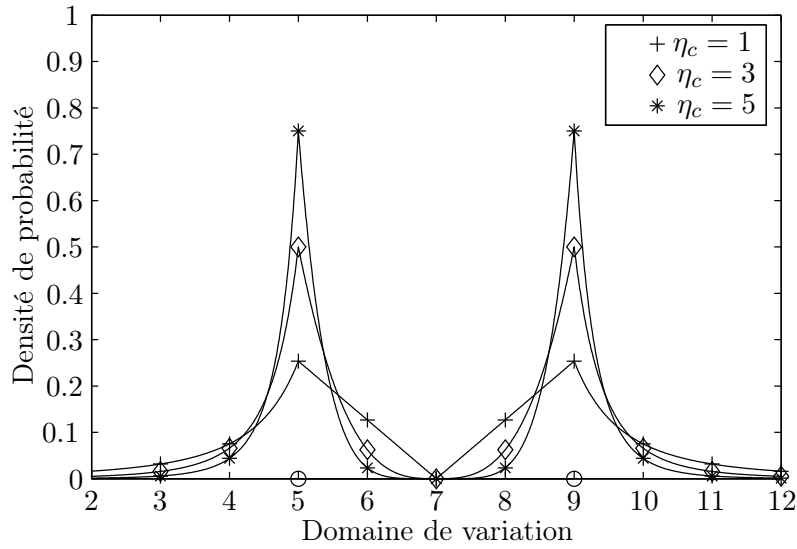


FIGURE 1.6 – Densité de probabilité de création d'enfants avec l'opérateur de croisement $SBX-\eta_c$. Les parents sont représentés par 2 cercles sur l'axe des abscisses : $x_{p1} = 5$, $x_{p2} = 9$ (Jacquetot 2010).

Pour chaque paire de variables (x_1, x_2) issue des parents, les nouvelles valeurs sont données par :

$$x'_1 = 0,5 \times ((1 + \beta_q)x_1 + (1 - \beta_q)x_2) \quad (1.20)$$

$$x'_2 = 0,5 \times ((1 - \beta_q)x_1 + (1 + \beta_q)x_2) \quad (1.21)$$

avec :

$$\beta_q = \begin{cases} (2\mu)^{\frac{1}{\eta_c+1}} & \text{si } \mu \leq 0,5 \\ \left(\frac{1}{2(1-\mu)}\right)^{\frac{1}{\eta_c+1}} & \text{sinon} \end{cases} \quad (1.22)$$

où μ est un nombre aléatoire entre 0 et 1, et η_c est l'indice de distribution qui est un entier positif. Plus η_c est grand et plus la probabilité d'obtenir un enfant proche des parents est importante (figure 1.6).

Dans le cas où les variables sont bornées ($x \in [x^l; x^u]$), l'opérateur SBX a été modifié pour éviter de créer des enfants en dehors des bornes (Deb et Agrawal 1999) :

$$\beta_q = \begin{cases} (\alpha\mu)^{\frac{1}{\eta_c+1}} & \text{si } \mu_i \leq \frac{1}{\alpha} \\ \left(\frac{1}{2-\alpha\mu}\right)^{\frac{1}{\eta_c+1}} & \text{sinon} \end{cases} \quad (1.23)$$

où $\alpha = 2 - \beta^{-(\eta_c+1)}$ et $\beta = 1 + \frac{2}{|x_2-x_1|} \min\{\min\{x_1, x_2\} - x^l, x^u - \max\{x_1, x_2\}\}$.

1.4.3 Mutations

L'opérateur de mutation consiste à modifier une solution pour créer une nouvelle solution qui lui soit proche. Cela permet, d'une part, de diversifier la population et, d'autre part, d'optimiser localement la population (si la solution mutée est meilleure que la solution d'origine).

Bascule L'idée est simple et consiste simplement, pour une probabilité p de mutation donnée⁵, à parcourir la chaîne binaire codant l'individu et à changer le 0 en 1 ou le 1 en 0, si le nombre tiré aléatoirement est inférieur à p .

Mutation distribuée normalement Pour un codage réel, un opérateur de mutation simple à mettre en œuvre est, pour chaque variable, de tirer une nouvelle valeur x' suivant une loi normale de probabilité distribuée autour de la valeur de la variable x .

$$x' = x + \mathcal{N}(0, \sigma) \quad (1.24)$$

où $\mathcal{N}(0, \sigma)$ est la loi normale de moyenne 0 et de variance σ déterminée par l'utilisateur.

Mutation polynomiale La mutation polynomiale, proposée par Deb et Agrawal (1995), est très proche de la mutation normale. On obtient une nouvelle solution x' par mutation de la solution x , en faisant muter chacune de ses composantes par la formule suivante :

$$x'_i = x_i + \Delta_{\max} \delta_q \quad (1.25)$$

où Δ_{\max} est la variation utilisée et δ_q le coefficient de variation, calculé de la manière suivante :

$$\delta_q = \begin{cases} (2\mu)^{\frac{1}{\eta_m+1}} - 1 & \text{si } \mu \leq 0,5 \\ 1 - (2(1-\mu))^{\frac{1}{\eta_m+1}} & \text{sinon} \end{cases} \quad (1.26)$$

où μ est un nombre aléatoire tiré uniformément entre 0 et 1, et η_m l'indice de distribution. La figure 1.7 illustre la mutation polynomiale pour différentes valeurs de η_m .

Si x est borné dans $[x^l; x^u]$, on a $\Delta_{\max} = (x^u - x^l)$ et δ_q est alors donné par la formule

$$\delta_q = \begin{cases} (2\mu + (1-2\mu)(1-\delta)^{\eta_m+1})^{\frac{1}{\eta_m+1}} - 1 & \text{si } \mu \leq 0,5 \\ 1 - (2(1-\mu) + 2(\mu-0,5)(1-\delta)^{\eta_m+1})^{\frac{1}{\eta_m+1}} & \text{sinon} \end{cases} \quad (1.27)$$

avec

$$\delta = \frac{\min\{x - x^l, x^u - x\}}{x^u - x^l} \quad (1.28)$$

5. généralement $p = \frac{1}{\text{nombre de bits de la chaîne}}$ afin d'obtenir un changement de bits dans la chaîne.

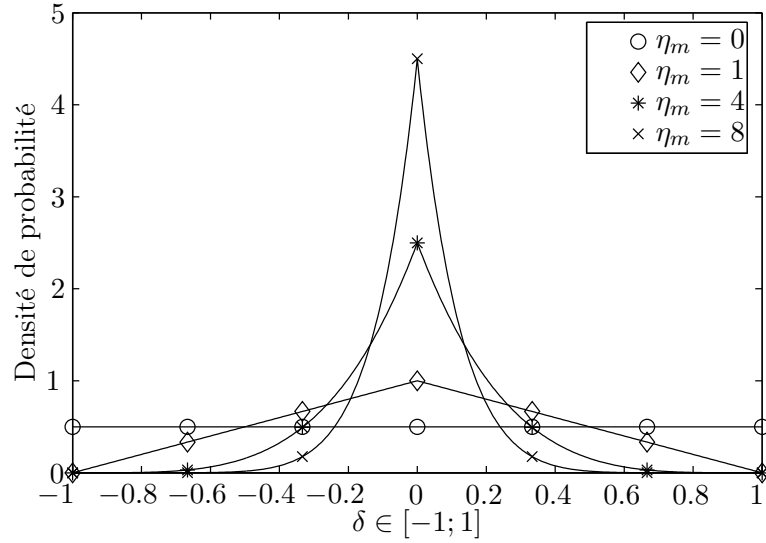


FIGURE 1.7 – Densité de probabilité pour l'opérateur de mutation polynomiale (Jacquenot 2010).

Cependant, si x se trouve sur une de ses bornes, la valeur x' engendrée sera identique à x . L'opérateur de mutation a été modifié pour pallier ce problème (Deb et Tiwari 2008). Dans ce cas, on a

$$\delta_q = \begin{cases} (2\mu + (1 - 2\mu)\delta_2^{\eta_m+1})^{\frac{1}{\eta_m+1}} - 1 & \text{si } \mu \leq 0,5 \\ 1 - (2(1 - \mu) + 2(\mu - 0,5)\delta_1^{\eta_m+1})^{\frac{1}{\eta_m+1}} & \text{sinon} \end{cases} \quad (1.29)$$

avec $\delta_1 = \frac{x-x^l}{x^u-x^l}$ et $\delta_2 = 1 - \delta_1$.

1.4.4 Structure de l'espace de recherche

Le codage des variables, ainsi que les différents opérateurs s'y appliquant, vont définir l'espace de recherche. L'espace de recherche peut être différent de l'espace des solutions ou de l'espace admissible qu'il représente.

De la structure de l'espace de recherche va dépendre la définition du voisinage d'une solution, et ainsi la capacité de l'algorithme d'optimisation à converger vers l'optimum global du problème.

Exemple 1.8 *Considérons un espace admissible $X = \{0, 1, 2\}^2$. X contient 9 solutions.*

Nous commençons par exposer une représentation du codage réel de l'exemple 1.8 car ce codage est plus naturel, puis nous le comparons au codage binaire.

Espace de recherche réel

Le codage réel paraît plus naturel, car on rapproche ainsi \mathcal{X} de X . Cependant, les opérateurs génétiques sont plus complexes. Considérons, pour simplifier, un cas sans opérateur de croisement et avec l'opérateur de mutation $m : X \times [0; 1]^2 \rightarrow X$:

$$m((x_1, x_2), (\rho_1, \rho_2)) \mapsto \begin{cases} (x_1 - 1, x_2) & \text{si } \rho_1 \leq 0,5 \wedge \rho_2 \leq 0,5 \\ (x_1, x_2 - 1) & \text{si } \rho_1 \leq 0,5 \wedge \rho_2 > 0,5 \\ (x_1 + 1, x_2) & \text{si } \rho_1 > 0,5 \wedge \rho_2 \leq 0,5 \\ (x_1, x_2 + 1) & \text{si } \rho_1 > 0,5 \wedge \rho_2 > 0,5 \end{cases} \quad (1.30)$$

où (ρ_1, ρ_2) représentent les probabilités de mutation.

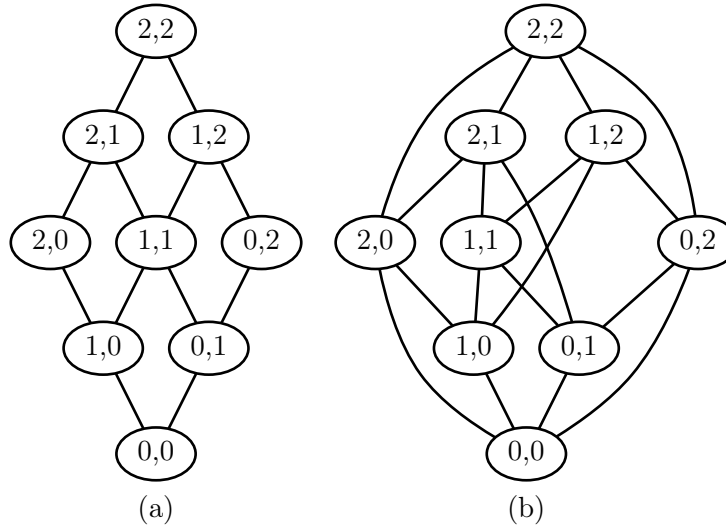


FIGURE 1.8 – L’espace admissible $X = \{0, 1, 2\}^2$ dont la connexité est définie par un opérateur de mutation ajoutant ou retranchant 1 à une des composantes. Le codage (a) utilise une représentation classique des nombres, tandis que le codage (b) utilise une représentation « circulaire » : le successeur de 2 est 0.

Ici, on a $\mathcal{X} = X$, toutes les solutions sont accessibles et font partie de la même composante connexe. Un algorithme n’utilisant qu’un unique individu est donc, en théorie, capable de trouver la solution optimale.

Remarque 1.14 On peut également coder les nombres de telle sorte que le successeur de 2 soit 0. Dans ce cas, la connexité de l’espace de recherche est plus régulière : toutes les solutions ont une connexité de 4 (Figure 1.8 b).

Espace de recherche binaire

Le codage binaire simplifie les opérateurs génétiques, qui sont alors simplement des fonctions opérant sur des bits. Cependant, la représentation des nombres en binaire va entraîner une augmentation de l’espace de recherche. En effet, avec un codage simple, l’espace de recherche aura comme cardinal une puissance de 2. Il faut alors prendre la première puissance de 2 supérieure au cardinal. Dans l’exemple 1.8, $|X| = 9$, il nous faut donc utiliser au moins 4 bits pour coder les solutions, soit de quoi coder 16 possibilités. On a alors $|\mathcal{X}| = 16$.

Comme pour le codage réel, nous ne considérons pas de croisement et utilisons l’opé-

tateur de mutation $m : \mathbb{B}^4 \times [4] \rightarrow \mathbb{B}^4$:

$$m((x_1, x_2, x_3, x_4), n) \mapsto \begin{cases} (\bar{x}_1, x_2, x_3, x_4) & \text{si } n = 1 \\ (x_1, \bar{x}_2, x_3, x_4) & \text{si } n = 2 \\ (x_1, x_2, \bar{x}_3, x_4) & \text{si } n = 3 \\ (x_1, x_2, x_3, \bar{x}_4) & \text{si } n = 4 \end{cases} \quad (1.31)$$

avec $\bar{0} \mapsto 1$ et $\bar{1} \mapsto 0$. Nous obtenons un espace de recherche de plus grande cardinalité que pour le codage réel (figure 1.9)

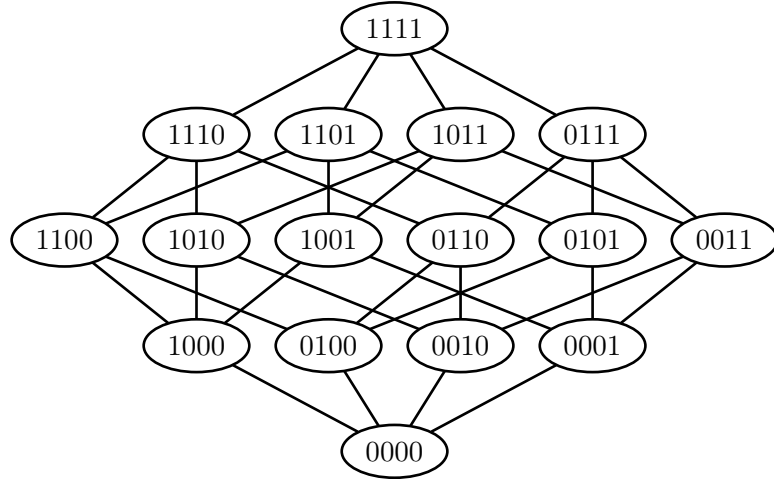


FIGURE 1.9 – Codage binaire de $X = \{0, 1, 2\}^2$. Ici, le cardinal de \mathcal{X} est 16, alors qu'il est de 9 en codage réel.

Selon l'interprétation de la chaîne de bits comme un nombre binaire ou comme un code de Gray, la répartition des solutions dans l'espace de recherche diffère (figure 1.10). En effet, pour le codage binaire simple, les nombres ne sont pas connectés suivant la représentation naturelle des nombres : par exemple $(2, 0)$ est un voisin de $(0, 0)$. En revanche, la connexité des solutions réalisables avec le codage de Gray est identique au codage réel. Le graphe de connexité des solutions réelles est en effet un sous-graphe du graphe de connexité des solutions en codage de Gray. Ce dernier comportant des solutions non admissibles en plus.

1.5 Phénotype : évaluation et sélection des individus

Nous avons vu que le génotype était la représentation de l'espace de décision. Sa structure est liée au codage des variables et aux opérations génétiques définies sur cet espace.

Le phénotype est l'expression du génotype. Il s'agit donc ici d'un ensemble, \mathcal{Y} image de \mathcal{X} par la fonction objectif f , dans lequel sont évaluées les performances des individus. Dans notre modélisation, l'adaptation des individus est évaluée par l'application θ qui en donne un classement dans $(\mathcal{Z}, \preceq_{\mathcal{Z}})$.

Nous présentons ici les mécanismes d'évaluation et de sélection des individus.

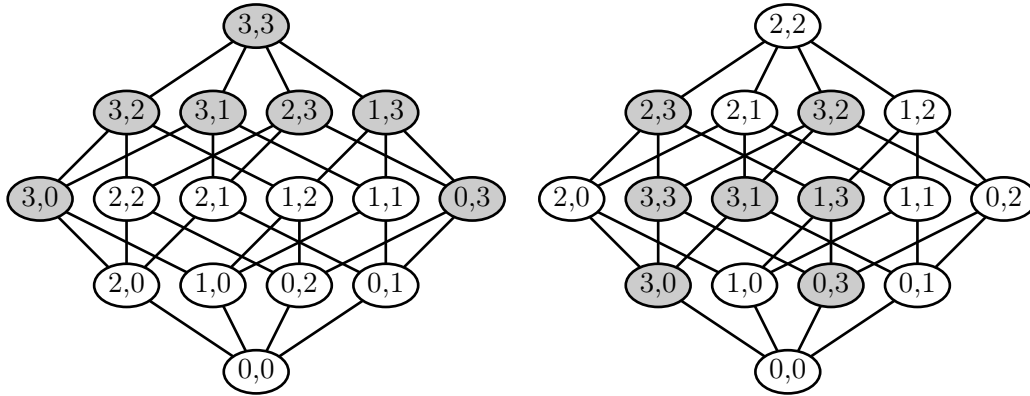


FIGURE 1.10 – Interprétation suivant la représentation binaire (a) et codage de Gray (b) de l'espace de recherche représenté figure 1.9. On remarque qu'en retirant les solutions non admissibles (en gris) le codage de Gray est équivalent au codage réel.

1.5.1 Évaluation des individus

Les individus sont évalués et comparés entre eux suivant leur adaptation (*fitness* en anglais). La mesure d'adaptation peut être différente de la mesure de performance des individus. Dans notre modélisation, la performance est donnée par la fonction objectif f , tandis que l'adaptation est donnée par l'application θ . De plus, si l'on souhaite obtenir plusieurs solutions, il faut à la fois optimiser les performances, mais également prendre en compte la diversité des solutions dans la fonction d'adaptation.

Évaluation des performances

Pour les problèmes où l'objectif est à valeurs dans \mathbb{R} , l'adaptation des individus peut simplement être mesurée sur leur performance. Ainsi, dans les premiers algorithmes évolutionnaires, la probabilité de sélection d'un individu était proportionnelle à sa performance. Cependant, une telle approche donne trop d'importance aux individus qui ont des performances très supérieures aux autres. Afin de limiter ce phénomène, Goldberg (1989) suggère l'utilisation du rang des individus : la probabilité d'un individu d'être sélectionné est ainsi proportionnelle à $r \circ f(x)$, où r est la fonction de rang (définition 1.15).

Dans le cas multiobjectif, une première approche consiste à se ramener à un problème mono-objectif. Le premier algorithme génétique multiobjectif, nommé VEGA (*Vector Evaluated Genetic Algorithm*) de Schaffer (1984), divise aléatoirement la population en autant de groupes qu'il y a d'objectifs et évalue ces groupes sur un unique objectif. Les opérateurs d'évolution ne sont appliqués que dans le même groupe. Puis de nouveaux groupes sont reformés aléatoirement à chaque génération, etc. Le problème multiobjectif est ainsi résolu comme une succession de sous-problèmes mono-objectifs. On peut également se ramener à un problème mono-objectif en utilisant des sommes pondérées, comme dans le WBGA (*Weight-Based Genetic Algorithm*) proposé par Hajela *et al.* (1993). À chaque individu est également associé un vecteur de poids, pour obtenir différentes solutions de compromis.

Une autre approche consiste à classer les individus suivant la dominance de Pareto. Comme pour le cas mono-objectif, le rang des individus peut être utilisé pour les comparer. C'est l'idée mise en pratique dans l'algorithme NSGA (*Non-Dominated Sorting Genetic Algorithm*) par Deb (2001). NSGA minimise donc la fonction ϕ définie par

$$\phi(x) = r \circ f(x) \quad (1.32)$$

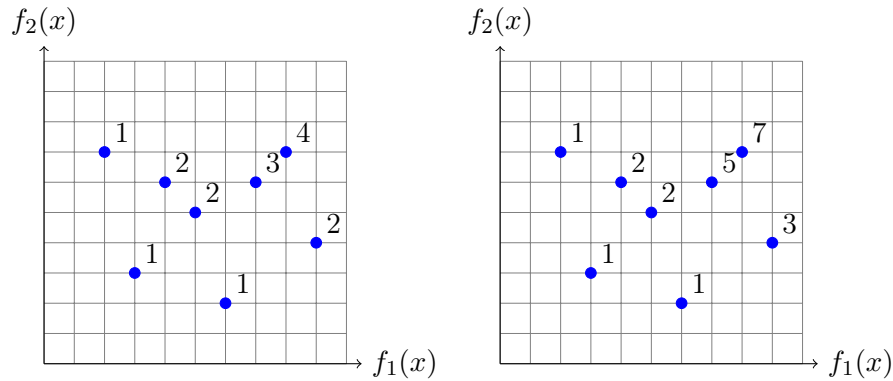


FIGURE 1.11 – Classement suivant le rang (à gauche) et classement de Fonseca et Fleming (à droite) pour un problème de minimisation bi-objectif.

Fonseca et Fleming (1993) proposent, quant à eux, un classement similaire, mais qui ne correspond pas au rang tel que défini en théorie des ordres. Le classement des solutions effectué par Fonseca et Fleming correspond, en fait, au cardinal de l'idéal d'un individu, c'est-à-dire le nombre d'individus qui lui sont inférieurs (lui compris). Soit la minimisation de ϕ telle que

$$\phi(x) = |\downarrow f(x)| \quad (1.33)$$

Le classement de Belegund et Salagame (1995), plus simple, attribue 1 aux éléments minimaux et 2 aux autres⁶. Soit

$$\phi(x) = \begin{cases} 1 & \text{si } f(x) \in \min(Z, \preceq_Z) \\ 2 & \text{sinon} \end{cases} \quad (1.34)$$

Alberto *et al.* (2003) proposent deux méthodes de classement, l'une basée sur le nombre de fois qu'une solution domine les autres avec des pondérations aléatoires et la seconde étant un mélange du rang et du classement de Fonseca et Fleming.

D'autres paramètres que les performances des solutions peuvent être pris en compte dans la fonction d'adaptation, tels que la diversité des solutions.

Distribution des individus

Différents mécanismes pour obtenir une plus grande diversité de solutions ont été proposés. Nous en présentons ici quelques-uns.

La méthode WBGGA doit garantir une diversité des pondérations pour obtenir des solutions bien réparties sur le front de Pareto.

Deb (2001) introduit dans le calcul de classement une fonction de partage qui à l'intérieur d'une même classe, va favoriser les individus les plus dispersés. Ce principe est mis en œuvre dans l'algorithme NSGA-II. Une mesure⁷, appelée *crowding*, est calculée pour chaque point de même rang. Soit une population de taille k où l'ensemble des individus est trié sur chacun des p objectifs par ordre croissant : on note $x_{i,j}$ le j -ième

6. Les auteurs attribuent en fait 0 aux minimums et 1 aux autres, mais nous décalons de 1 pour harmoniser les classements.

7. Deb (2001) parle de *crowding distance* en anglais. Cependant, ne s'agissant pas d'une distance au sens mathématique du terme, nous lui préférons *mesure*.

individu trié sur f_i . On a $a \leq b \Leftrightarrow f_i(x_{i,a}) \leq f_i(x_{i,b})$. La mesure de *crowding* est donnée par la formule suivante :

$$cr(x) = \sum_{i=1}^p \begin{cases} +\infty & \text{si } j = 1 \vee j = k \\ \frac{f_i(x_{i,j+1}) - f_i(x_{i,j-1})}{f_i(x_{i,k}) - f_i(x_{i,1})} & \text{sinon} \end{cases} \quad (1.35)$$

En pratique, il s'agit – à part pour les points extrêmes du front de Pareto –, pour un problème biobjectif du demi-périmètre du rectangle délimité par les points de même rang qui l'entourent (figure 1.12).

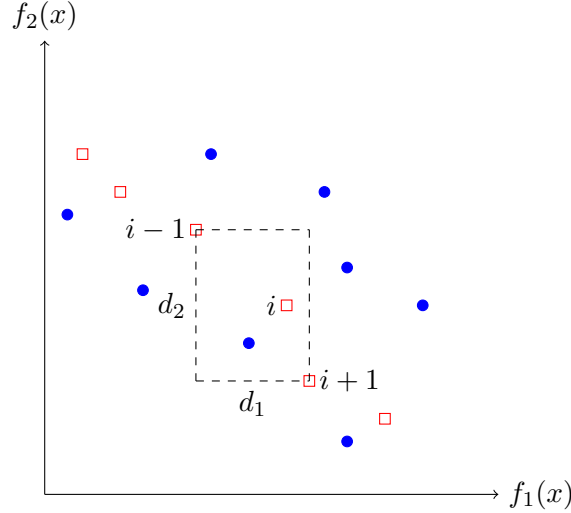


FIGURE 1.12 – Calcul de la distance de crowding. Le crowding autour de l'individu i est défini comme la distance $d_1 + d_2$.

Remarque 1.15 *Le calcul du rang, puis de la mesure de crowding, peut être assimilé à un morphisme θ , en posant Y^i une partie de Y partitionné en rangs :*

$$Y^i := \{y \in Y : r(y) = i\} \quad (1.36)$$

et cr la mesure de crowding, θ est alors l'application de Y dans Z telle que

$$(Z, \leq_Z) = \bigoplus_{i=1}^{r_{\max}} (cr(Y^i), \leq_{\mathbb{R}^{|Y^i|}}) \quad (1.37)$$

où $r_{\max} = \max_{y \in Y} r(y)$. (Z, \leq_Z) est alors un ensemble totalement ordonné. NSGA-II cherche les solutions dont les images sont dans (Z, \leq_Z) (équation 1.37).

L'algorithme SPEA de Zitzler et Thiele (1998) utilise une méthode différente. Une population externe gardant les individus non dominés est maintenue au cours de l'exécution de l'algorithme. Quand la taille de cette population externe dépasse une limite, les individus les plus entourés sont supprimés. Pour cela, les individus sont rassemblés en groupes (*clusters* en anglais). On crée d'abord autant de groupes qu'il y a d'individus, chaque groupe étant un singleton. Puis les groupes les plus proches deux-à-deux selon la distance d_c sont regroupés. La distance entre groupes est donnée par la formule suivante :

$$d_c(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j) \quad (1.38)$$

avec $d(i, j)$ la distance euclidienne entre i et j . Cette opération est répétée tant que le nombre de groupes est supérieur à la taille de population limite. Ensuite, dans chaque groupe, seules les solutions ayant la distance moyenne la plus faible sont conservées.

Selon Bosman et Thierens (2003), les algorithmes évolutionnaires séparant la convergence vers le front de Pareto et le maintien de la diversité réalisent un compromis entre ces deux objectifs. Il faut alors faire un choix entre performance et diversité. Il n'existe cependant pas de formulation explicite de la qualité des solutions prenant en compte ces deux aspects d'après Zitzler et Künzli (2004). De plus, les algorithmes génétiques basés sur la dominance, tels que NSGA-II et SPEA2, ne sont plus efficaces au-delà de trois objectifs (Hughes 2005, Wagner *et al.* 2007).

Indicateurs de qualité

Afin de résoudre des problèmes à beaucoup d'objectifs⁸, Zitzler et Künzli (2004) proposent d'utiliser les indicateurs de qualité des solutions pour évaluer et sélectionner les individus. Les auteurs nomment IBEA (*Indicator-Based Evolutionary Algorithm*) ce type d'algorithme évolutionnaire⁹. La fonction objectif d'IBEA est la suivante :

$$\phi(x) = \sum_{x' \in P \setminus \{x\}} -e^{\frac{-I(\{x'\}, \{x\})}{\kappa}} \quad (1.39)$$

où P est la population, κ une constante (recommandée à 0,05), et I , l'indicateur, est une fonction de $\mathcal{P}(X)^2$ dans \mathbb{R} . Zitzler et Künzli (2004) proposent, par exemple, d'utiliser l'indicateur $I_{\epsilon+}$ (Zitzler *et al.* 2003) :

$$I_{\epsilon+}(A, B) := \min\{x \in B : \exists x' \in A \text{ t.q. } f_i(x') \leq f_i(x) \forall i \in \llbracket p \rrbracket\} \quad (1.40)$$

où $A, B \subseteq Y$. I_{HD} , ou bien l'indicateur I_{HD} :

$$I_{HD}(A, B) := \begin{cases} I_H(B) - I_H(A) & \text{si } \forall x \in B \exists x' \in A : x' \preceq x \\ I_H(A \cup B) - I_H(A) & \text{sinon} \end{cases} \quad (1.41)$$

où I_H est la métrique d'hypervolume, également appelée métrique \mathcal{S} (Zitzler et Thiele 1999, Brockhoff 2009) :

$$I_H(A, R) := \lambda(\{r \in \mathbb{R}^p : \exists a \in A, \exists r \in R \text{ t.q. } f(a) \leq z \leq r\}) \quad (1.42)$$

où $\lambda(\cdot)$ est la mesure de Lebesgue. D'autres indicateurs peuvent être utilisés, l'important étant que l'indicateur utilisé préserve la relation de dominance de Pareto.

Définition 1.28 (Indicateur préservant la dominance) *L'indicateur I préserve la dominance de Pareto si $x \preceq x' \Rightarrow \phi(x) \leq \phi(x')$.*

8. Appelés en anglais *many-objective* par Farina et Amato (2002). En général quand il y a plus de trois ou quatre objectifs. Voir Ishibuchi *et al.* (2008) pour un aperçu de l'état de l'art sur les algorithmes évolutionnaires à beaucoup d'objectifs.

9. Voir (Wagner *et al.* 2007) pour une évaluation de différents IBEA.

1.5.2 Sélection

Des individus sont sélectionnés pour se croiser et former les individus de la génération suivante. La méthode la plus simple est évidemment de prendre au hasard les individus dans la population. Les mécanismes de sélection privilégient les meilleurs individus en leur donnant plus de chances de se reproduire que les autres et de se reproduire un plus grand nombre de fois.

Les algorithmes évolutionnaires comportent deux phases de sélection. La première sélectionne les individus pour constituer la population de parents. La seconde sélectionne les individus parmi les parents et les enfants qui feront partie de la génération suivante.

Les méthodes les plus utilisées sont la sélection par tournois, la sélection proportionnelle et la sélection par classement (Deb 2001).

La sélection par tournois consiste à confronter deux solutions entre elles. La meilleure est retenue dans la population de reproduction. Si le tournoi est réalisé de façon systématique sur l'ensemble de la population, chaque individu peut participer à exactement deux tournois et peut ainsi être présent en deux, un ou zéro exemplaire dans la population de parents.

La sélection proportionnelle consiste à attribuer à chaque individu une probabilité d'être sélectionné en fonction de ses performances. Meilleur est l'individu, plus il a de chances d'être sélectionné. La sélection par roulette est une de ces méthodes. On représente une roulette de casino abstraite divisée en autant de portions qu'il y a d'individus et où la taille des portions dépend des scores des individus. Un nombre est alors tiré aléatoirement (la bille de la roulette).

La méthode de classement consiste à trier les solutions suivant leur score et leur attribuer un rang. Une sélection proportionnelle est ensuite effectuée sur la population, mais en prenant en compte les rangs et non les scores.

Conclusion

Dans ce chapitre, nous avons modélisé les problèmes d'optimisation à l'aide des ensembles ordonnés. Ceci nous permet de définir dans un cadre commun les problèmes d'optimisation mono et multiobjectif, ainsi que d'y intégrer la prise en compte des préférences d'un décideur. Cette modélisation nous permettra par la suite de définir la notion de compromis entre disciplines en décrivant la structure de l'espace de comparaison d'un problème multidisciplinaire.

Nous avons également présenté les principes des algorithmes évolutionnaires et montré comment ils peuvent se définir dans notre formalisme. Nous montrerons ainsi que les algorithmes évolutionnaires ne sont pas adaptés à certaines structures de l'espace de comparaison.

La chapitre suivant présente un type d'optimisation particulier : l'optimisation multidisciplinaire. Les problèmes d'optimisation multidisciplinaire sont, tout d'abord, formellement décrits, puis, les méthodes de résolution de tels problèmes sont présentées.

Formalisation et résolution des problèmes d'optimisation multidisciplinaire

SOMMAIRE

INTRODUCTION	34
2.1 DESCRIPTION DES PROBLÈMES D'OPTIMISATION MULTIDISCIPLINAIRE	34
2.1.1 Analyse disciplinaire	35
2.1.2 Optimisation disciplinaire	37
2.1.3 Analyse multidisciplinaire	38
2.1.4 Problème d'optimisation multidisciplinaire	42
2.2 MÉTHODES DE RÉOLUTION	43
2.2.1 Reformulations de problèmes	44
2.2.2 Théorie des jeux	51
2.2.3 Méthodes basées sur les algorithmes évolutionnaires	53
CONCLUSION	57

Introduction

DANS les problèmes d'optimisation multidisciplinaire, les disciplines représentent généralement une spécialité d'ingénierie (mécanique, mécanique des fluides, thermique. . .). Elles peuvent cependant désigner d'autres types d'entités, telles que des codes de calcul différents ou des équipes de travail. Leur formulation peut être explicitement donnée par des équations – les équations d'état –, représentant des problèmes d'optimisation ou le résultat de simulations par des codes de calculs dédiés, ou représenter des données fournies par une équipe gardant sa méthode confidentielle. C'est pourquoi nous modéliserons les disciplines comme des boîtes noires dont on ne connaît que les entrées et les sorties. Les disciplines sont généralement couplées, c'est-à-dire que les entrées d'une discipline peuvent être le résultat d'un calcul effectué par une autre discipline. Des solutions admissibles, globalement optimales et satisfaisant au mieux chacune des disciplines simultanément doivent alors être trouvées. Ces solutions constituent des solutions de compromis du problème.

La résolution des couplages constitue une des difficultés majeures de l'optimisation multidisciplinaire et est abondamment traitée dans la littérature. Dans ce chapitre, nous donnons des définitions formelles de l'optimisation multidisciplinaire et de ses principales méthodes de résolution. Plusieurs méthodes d'optimisation multidisciplinaire ont été proposées depuis le début des années 1990. Les différences entre ces méthodes sont analysées et en particulier la manière dont les solutions de compromis sont définies. Dans le cas multiobjectif, la définition des solutions optimales et ainsi des solutions de compromis va différer suivant la répartition des objectifs au sein des disciplines (tableau 1 de l'introduction). Nous nous intéressons plus particulièrement aux problèmes d'optimisation multidisciplinaire où chaque discipline a plusieurs objectifs à résoudre, sans objectif global (cas 6 du tableau 1 de l'introduction).

2.1 Description des problèmes d'optimisation multidisciplinaire

L'optimisation multidisciplinaire est définie de manière générale comme l'ensemble des problèmes liés à la conception de systèmes d'ingénierie complexes impliquant plusieurs disciplines (AIA 1991, Sobieszczanski-Sobieski et Haftka 1997, Alexandrov et Lewis 1999). Ces problèmes vont de la modélisation mathématique des problèmes d'optimisation multidisciplinaire, à la gestion des équipes disciplinaires, en passant par le choix algorithmes d'optimisation. Les problèmes d'optimisation multidisciplinaire font souvent référence, plus précisément, à la résolution de problèmes d'optimisation couplés.

Cramer *et al.* (1993) proposent une formalisation des problèmes d'optimisation multidisciplinaire en identifiant les différentes équations et variables mises en jeu dans la résolution du problème (figure 2.1). On retrouvera par la suite des formulations simplifiées des problèmes d'optimisation multidisciplinaire (Alexandrov et Kodiyalam 1998), où les équations sont décrites comme des fonctions. Les formulations précédentes (Sobieszczanski-Sobieski 1988b) présentaient les problèmes d'optimisation multidisciplinaire comme des problèmes de décomposition hiérarchique, l'analyse multidisciplinaire (voir section 2.1.3) étant jugée trop coûteuse.

Dans cette section, nous proposons une formulation des problèmes d'optimisation multidisciplinaire compatible avec celle de Cramer *et al.* (1993). Pour illustrer notre pro-

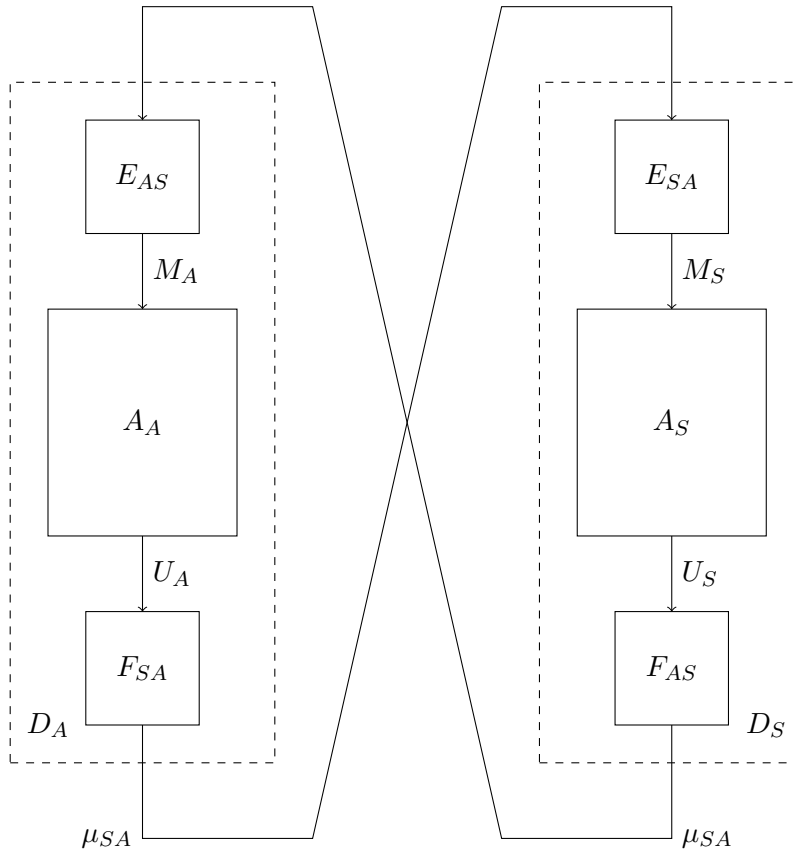


FIGURE 2.1 – Représentation schématique du problème aéroélastique. Les deux disciplines D_A et D_S sont respectivement l'aérodynamique et la structure. Le code d'aérodynamique (A_A) prend en entrée la déflexion de l'aile (M_A) et produit en sortie la pression exercée sur la surface de l'aile (U_A). Le code de calcul de structure (A_S) prend en entrée la charge de l'aile (M_S) et produit en sortie la déflexion U_S . Les données d'entrées d'un des codes ne correspondent pas toujours aux sorties d'un autre : les calculs ne sont pas toujours réalisés avec la même précision et des calculs intermédiaires sont parfois nécessaires. Pour cela, des codes d'interface F_{SA} , F_{AS} , E_{SA} et E_{AS} sont utilisés. D'après Cramer et al. (1993)

pos, nous utiliserons un problème à deux disciplines, dont une vue schématique est présentée figure 2.2.

Nous commençons par définir le problème monodisciplinaire : nous définissons l'analyse disciplinaire (résolution d'une discipline) puis l'optimisation disciplinaire (l'optimisation d'une discipline). Nous considérons enfin le cas où plusieurs disciplines sont impliquées en suivant le même schéma : nous définissons l'analyse multidisciplinaire (résolution des disciplines et couplages) puis l'optimisation multidisciplinaire (le problème d'optimisation impliquant plusieurs disciplines).

2.1.1 Analyse disciplinaire

Le terme *discipline*, dans le cadre de la conception, désigne à l'origine une discipline au sens discipline scientifique ou d'ingénierie (mécanique, mécanique des fluides, électronique...). Plus généralement, une discipline peut faire référence à un sous-problème, un code de calcul ou une équipe, suivant le contexte. En règle générale, une discipline peut

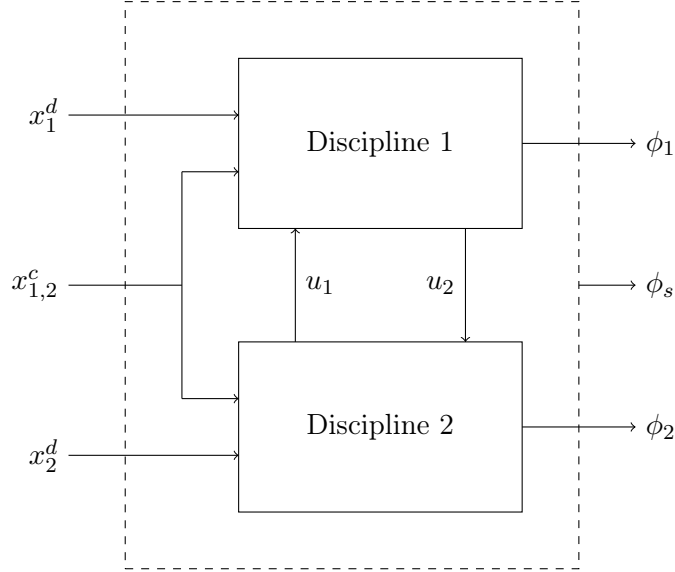


FIGURE 2.2 – Représentation schématique d'un problème d'optimisation multidisciplinaire à deux disciplines. Les variables de décision peuvent être partagées ($x_{1,2}^c$) ou disciplinaires (x_1^d et x_2^d). Des données sont échangées entre les disciplines : la discipline 1 donne les valeurs u_2 à la discipline 2 tandis que cette dernière donne u_1 à la discipline 1. Les objectifs du problème peuvent être propres aux disciplines (ϕ_1 et ϕ_2) ou globaux (ϕ_s).

être considérée comme une boîte noire à laquelle on donne des valeurs d'entrée et qui retourne des valeurs de sortie.

Chaque discipline a un système d'équations d'états à résoudre, qui, pour la discipline i , s'écrit sous la forme :

$$A_i(e_i, s_i) = 0 \tag{2.1}$$

où $e_i \in \mathcal{E}_i$ représente les variables d'entrée et $s_i \in \mathcal{S}_i$ les sorties. En pratique, il s'agit souvent de codes de calculs dont les équations ne sont pas résolues exactement. Un résidu r_i est alors renvoyé. Les disciplines possèdent également leurs propres variables internes décrivant leur physique. Ces variables entrent en compte dans la résolution de l'équation d'état mais n'ont pas d'interaction avec les autres disciplines. Ces variables sont appelées *variables d'état* et nous les noterons t_i . En définitive, une équation d'état est donc de la forme

$$A_i(e_i, s_i, t_i) = r_i \tag{2.2}$$

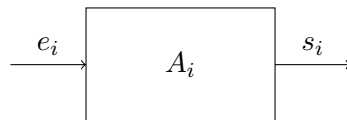


FIGURE 2.3 – Vue schématique d'une discipline représentée comme une boîte noire résolvant une équation d'état A_i , prenant en entrée e_i et renvoyant en sortie s_i .

Les équations sont souvent simplifiées et les modèles multidisciplinaires, dans leur description, ne considèrent pas les variables d'état et le résidu. Dans certains cas¹, les équations

1. Valable que dans le cadre du théorème des fonctions implicites.

d'état sont des fonctions, ce qui correspond alors au modèle de boîte noire (figure 2.3). La discipline est alors vue comme une fonction a_i équivalente à l'équation 2.1 :

$$s_i := a_i(e_i) \quad (2.3)$$

La résolution des équations d'une discipline est appelée l'*analyse disciplinaire*. Elle consiste à trouver un couple d'entrées/sorties (e_i, s_i) de la discipline i qui soit *disciplinairement réalisable*.

Définition 2.1 (Analyse disciplinaire) *Soit $A_i(e_i, s_i, t_i) = r_i$ une équation d'état représentant une discipline i . L'analyse disciplinaire de la discipline i consiste à trouver une valeur de t_i et/ou de s_i telle que (e_i, s_i) soit disciplinairement réalisable (toutes les autres variables étant fixées).*

Définition 2.2 (Réalisation disciplinaire) *Soient $e_i \in \mathcal{E}_i$ les variables d'entrée de la discipline i et $s_i \in \mathcal{S}_i$ ses variables de sortie. Le couple (e_i, s_i) est disciplinairement réalisable s'il existe t_i tel que l'équation d'état $A_i(e_i, s_i, t_i) = r_i$ de la discipline i soit vérifiée.*

Il faut distinguer l'*analyse disciplinaire* d'une *évaluation disciplinaire*. Dans le premier cas, il y a résolution de l'équation (un des termes de l'équation est à déterminer, les autres étant fixés). Dans le second cas, tous les termes de l'équation sont donnés et on ne fait que vérifier la satisfaction de l'équation.

Définition 2.3 (Évaluation disciplinaire) *Soit $A_i(e_i, s_i, t_i) = r_i$ une équation d'état représentant une discipline i . L'analyse disciplinaire de la discipline i consiste à obtenir la valeur du résidu r_i , avec e_i, s_i et t_i fixés.*

L'analyse disciplinaire est plus coûteuse qu'une évaluation. Cependant, bien que dans certains cas une évaluation disciplinaire puisse être effectuée à la place d'une analyse, certains codes de calculs ne le permettent pas (Keane et Nair 2005, chap. 9).

Exemple 2.1 *Dans le problème aéroélastique (figure 2.1), deux analyses disciplinaires sont à réaliser. Si l'on considère que l'aérodynamique est la discipline 1 et que la structure est la discipline 2, alors on a les correspondances suivantes entre les notations proposées et celles de Cramer et al. (1993) : $A_1 = A_A$, $e_1 = M_A$, $s_1 = U_A$, et de même $A_2 = A_S$, $e_2 = M_S$ et $s_2 = U_S$. A_1 correspond alors à un code de calcul de mécanique des fluides par différences finies, et A_2 à un code de structure par éléments finis.*

Afin de réduire les temps de calcul, les analyses disciplinaires sont parfois remplacées par des approximations. Ces approximations sont souvent des approximations linéaires ou des surfaces de réponse. Ces approximations sont d'autant plus utiles quand le problème d'analyse disciplinaire est associé à un problème d'optimisation, car un grand nombre d'analyses est nécessaire.

2.1.2 Optimisation disciplinaire

Si une discipline a un problème d'optimisation qui lui est propre, nous l'appellerons *problème d'optimisation disciplinaire*. Il consiste à trouver les variables d'entrée et de sortie de la discipline qui minimisent une fonction coût ϕ telle que son équation d'état soit vérifiée. Nous appellerons \mathcal{D} le problème décrit par l'équation 2.4 :

$$\begin{cases} \min_{e_i \in E_i} \phi_i(e_i, s_i) \\ A_i(e_i, s_i) = 0 \end{cases} \quad (2.4)$$

s_i est obtenu par résolution de l'équation d'état et $E_i \subseteq \mathcal{E}_i$ est l'ensemble des variables d'entrées admissibles. Il y a donc un aller-retour entre l'optimiseur qui propose des valeurs de e_i à l'analyseur, ce dernier calcule une valeur de s_i qui est retournée à l'optimiseur. Celui-ci évalue la fonction coût et peut proposer un nouveau e_i en conséquence (figure 2.4).

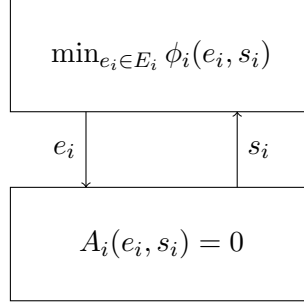


FIGURE 2.4 – Vue schématique de l'optimisation disciplinaire : une analyse disciplinaire est réalisée à chaque itération de l'optimiseur.

Si l'équation d'état A_i peut s'écrire sous la forme d'une fonction a_i équivalente, le problème peut s'écrire sous une forme plus compacte :

$$\min_{e_i \in E_i} \phi_i(e_i, a_i(e_i)) \quad (2.5)$$

La résolution de l'équation d'état peut également être vue comme la satisfaction d'une contrainte. Dans ce cas, s_i est également une variable de décision :

$$\begin{cases} \min_{(e_i, s_i) \in E_i \times S_i} \phi_i(e_i, s_i) \\ A_i(e_i, s_i) = 0 \end{cases} \quad (2.6)$$

Dans l'équation 2.6, $A_i(e_i, s_i) = 0$ n'a pas le même sens que dans l'équation 2.4. Dans le premier cas, il s'agit d'une analyse disciplinaire : les équations sont résolues et les sorties s_i sont calculées. Dans le second cas, il ne s'agit que d'une évaluation disciplinaire² : les valeurs de e_i et s_i sont données et l'équation est évaluée afin de vérifier l'égalité.

La résolution d'un problème d'optimisation disciplinaire consiste à trouver un minimum de la fonction coût qui soit disciplinairement réalisable.

Définition 2.4 (Optimalité disciplinaire) *Une solution (e_i, s_i) est optimale pour la discipline i si :*

- (e_i, s_i) est disciplinairement réalisable,
- (e_i, s_i) est une solution optimale de 2.4.

Après avoir défini les problèmes à une discipline, nous définissons maintenant les problèmes impliquant plusieurs disciplines.

2. Si l'on ne considère pas les variables d'état.

2.1.3 Analyse multidisciplinaire

Les disciplines sont souvent liées de deux manières : par des variables de décision communes et/ou par des couplages. La figure 2.2 donne un aperçu des variables partagées ($x_{1,2}^c$) ou échangées (u_1 et u_2) entre les disciplines. Dans cette section, nous décrivons plus précisément les interactions entre les disciplines.

Les différents espaces et variables en jeu

Dans les sections précédentes, nous n'avons pas évoqué l'espace de décision. Nous avons présenté les problèmes disciplinaires seulement en fonction de leurs variables d'entrée et de sortie. Or, parmi les variables d'entrée d'une discipline, certaines sont des variables de décision du problème et d'autres sont des variables de sortie d'une autre discipline par l'intermédiaire de fonctions de couplage. De même, parmi les variables de sortie d'une discipline, certaines sont des variables de l'espace de décision tandis que d'autres sont des variables des fonctions de couplage à destination des autres disciplines. Nous décomposons donc ainsi les variables d'entrée e_i de la discipline i :

$$e_i = (x_i, u_i) \quad (2.7)$$

où $x_i \in \mathcal{X}_i$ sont les variables de décision de la discipline i , et $u_i \in \mathcal{U}_i$ les variables de couplage données par les autres disciplines à la discipline i . L'espace des variables d'entrée de la discipline i est donné par le produit cartésien

$$\mathcal{E}_i = \mathcal{X}_i \times \mathcal{U}_i \quad (2.8)$$

Plusieurs variables de couplage provenant de plusieurs disciplines peuvent être données en entrée d'une autre discipline. Nous noterons $u_{i,j}$ la variable de couplage en entrée de la discipline i provenant de la discipline j . On a alors

$$\mathcal{U}_i = \prod_{j \in J} \mathcal{U}_{i,j} \quad (2.9)$$

où J est ici l'ensemble des disciplines couplées en entrée avec i . On note alors

$$u_i = (u_{i,j})_{j \in J} \quad (2.10)$$

De même, pour les variables de sortie $s_i \in \mathcal{S}_i$ de la discipline i , que nous décomposons ainsi :

$$s_i = (w_i, v_i) \quad (2.11)$$

où $w_i \in \mathcal{W}_i$ représente les variables de l'espace des objectifs de la discipline i et $v_i \in \mathcal{V}_i$ les variables de sortie de la discipline i données en entrée à d'autres disciplines, par l'intermédiaire des fonctions de couplage.

Remarque 2.1 *Les variables v_i de sortie d'une discipline i impliquées dans une fonction de couplage peuvent également être des variables de l'espace des objectifs w_i . Pour des raisons de simplification d'écriture, nous considérons que les espaces \mathcal{V}_i et \mathcal{W}_i sont disjoints. En pratique, s'ils ne le sont pas, les variables contenues dans leur intersections peuvent être dupliquées dans chacun des espaces.*

L'espace \mathcal{S}_i est également un produit cartésien :

$$\mathcal{S}_i = \mathcal{W}_i \times \mathcal{V}_i \quad (2.12)$$

et comme pour les variables de couplage, nous avons

$$\mathcal{V}_i = \prod_{j \in J} \mathcal{V}_{i,j} \quad (2.13)$$

où J est ici l'ensemble des disciplines couplées en sortie avec i . On note alors

$$v_i = (v_{i,j})_{j \in J} \quad (2.14)$$

Soient deux disciplines i et j partageant des variables de décision. Nous noterons $x_{i,j}^c \in \mathcal{X}_{i,j}^c$ les variables de décision partagées par i et j , et $x_i^d \in \mathcal{X}_i^d$ les variables de décision propres à i . Nous appellerons *variables communes* les variables de décision partagées et *variables disciplinaires* les variables de décision propres à une discipline. Pour une discipline i , l'espace de décision est alors

$$\mathcal{X}_i = \mathcal{X}_i^c \times \mathcal{X}_i^d \quad (2.15)$$

où \mathcal{X}_i^c représente l'espace produit de toutes les variables communes de la discipline i :

$$\mathcal{X}_i^c = \prod_{i \in J} \mathcal{X}_{i,j}^c \quad (2.16)$$

Pour les variables de décision, nous avons donc

$$x_i = (x_i^c, x_i^d) \quad (2.17)$$

avec

$$x_i^c = (x_{i,j}^c)_{j \in J} \quad (2.18)$$

Les variables u_i et v_i , respectivement en entrée et sortie des disciplines, sont liées par les fonctions de couplage.

Fonctions de couplage

Les interactions entre les disciplines interviennent, d'une part, par le partage de variables de décision et, d'autre part, par les fonctions de couplage. Les fonctions de couplage font le lien entre les variables de l'espace \mathcal{U} et celles de l'espace \mathcal{V} : elles transforment les sorties d'une discipline en entrées d'autres disciplines. Soit $c_{i,j} : \mathcal{V}_{i,j} \rightarrow \mathcal{U}_{j,i}$ la fonction de couplage de la discipline i vers la discipline j (figure 2.5). On a

$$u_{j,i} := c_{i,j}(v_{i,j}) \quad (2.19)$$

Nous noterons $\bar{\mathcal{V}}_i$ le produit des \mathcal{V}_j tel que $i \neq j$:

$$\bar{\mathcal{V}}_i = \prod_{j \neq i} \mathcal{V}_j \quad (2.20)$$

et nous noterons \bar{v}_i les éléments de $\bar{\mathcal{V}}_i$. Ceci nous permet de définir la fonction \bar{c}_i de $\bar{\mathcal{V}}_i$ dans \mathcal{X}_i qui nous permet de simplifier le système d'équations $A(x, w) = 0$ modélisant la

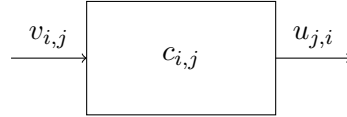


FIGURE 2.5 – Vue schématique de la fonction de couplage $c_{i,j}$ de la discipline i à la discipline j .

résolution des équations d'état couplées des q disciplines du problème multidisciplinaire. Nous notons :

$$A(x, y) := \begin{pmatrix} A_1(x_1, \bar{c}_i(\bar{v}_i), w_i, v_i) \\ \dots \\ A_q(x_q, \bar{c}_q(\bar{v}_q), w_q, v_q) \end{pmatrix} \quad (2.21)$$

L'équation d'état à résoudre est alors :

$$A(x, y) = 0 \quad (2.22)$$

Nous noterons \mathcal{A} le problème consistant à la résolution de l'équation 2.22.

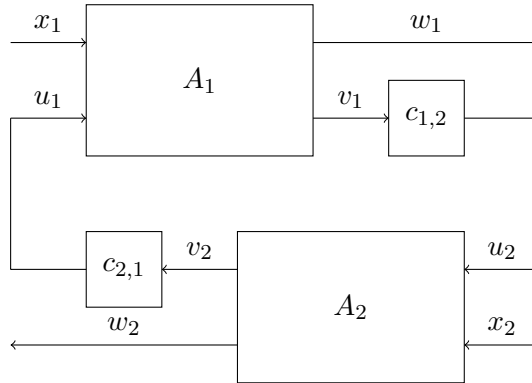


FIGURE 2.6 – Exemple de couplage entre deux disciplines. Ici $v_1 = v_{1,2}$ et $u_2 = u_{2,1}$, idem pour v_2 et u_1 .

La résolution d'un système de disciplines couplées s'appelle *l'analyse multidisciplinaire* et revient à chercher des points fixes du système d'équations 2.22 régissant les disciplines. D'après Cramer *et al.* (1993), un problème d'analyse multidisciplinaire est *multidisciplinairement réalisable* si

- chaque discipline est individuellement réalisable,
- les couplages sont vérifiés.

Définition 2.5 (Réalabilité multidisciplinaire) *Une solution (x, w) est multidisciplinairement réalisable si elle est solution de 2.22 (ou ayant un résidu négligeable).*

La composition interne du système $\{1, 2\}$ (fig. 2.7) n'est pas nécessairement visible de l'extérieur. Dans ce cas, elle est considérée, comme une boîte noire. On peut ainsi la considérer comme une unique discipline avec x les variables d'entrée, w les variables de sortie, et le système 2.21 l'équation d'état. Les variables u , internes à cette discipline, doivent alors être initialisées. Elles peuvent soit être données en entrée, ou bien gérées en interne par l'analyseur.

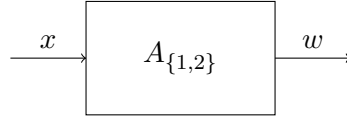


FIGURE 2.7 – Les disciplines couplées 1 et 2 (figure 2.6) vues comme une unique discipline $\{1,2\}$.

Remarque 2.2 Ici, l'ordre des indices dans $A_{\{1,2\}}$ n'a pas d'importance : $A_{\{1,2\}} = A_{\{2,1\}}$.

Cette description est cohérente avec une composition hiérarchique des disciplines où chaque niveau n de la hiérarchie voit les éléments du niveau $n - 1$ comme des disciplines à part entière (figure 2.8).

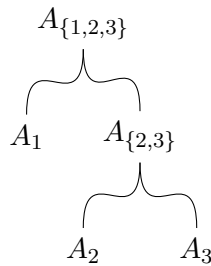


FIGURE 2.8 – Exemple de composition hiérarchique de trois disciplines. Globalement, le problème peut être vu comme une unique discipline $\{1, 2, 3\}$ composée de deux disciplines 1 et $\{2, 3\}$. La discipline $\{2, 3\}$ étant elle-même composée de deux disciplines : 2 et 3.

2.1.4 Problème d'optimisation multidisciplinaire

Les problèmes d'optimisation multidisciplinaire sont souvent formulés comme des problèmes d'optimisation non-linéaire (Alexandrov et Kodiyalam 1998) :

$$\min_{x \in X} \phi(x, a(x)) \tag{2.23}$$

où ϕ est la fonction objectif, X l'ensemble admissible et $a(x)$ les variables d'état définies par le système d'équations d'état. Une telle formulation sous-entend que \mathcal{A} (équation 2.22) puisse s'écrire comme une fonction. La formulation suivante est plus générale :

$$\begin{cases} \min_{x \in X} \phi(x, w) \\ A(x, w) = 0 \end{cases} \tag{2.24}$$

On remarque que le problème d'optimisation multidisciplinaire ainsi formulé est identique au problème d'optimisation disciplinaire \mathcal{D} (équation 2.4). Les variables d'entrée sont ici les variables de décision et les variables de sortie sont un sous-ensemble des variables de sortie des disciplines.

Définition 2.6 (Optimalité multidisciplinaire) Une solution (x, w) est une solution multidisciplinairement optimale si :

- (x, w) est multidisciplinairement réalisable,
- (x, w) est une solution optimale de 2.24.

La fonction coût peut représenter aussi bien des objectifs globaux (réduction des coûts, de la masse, augmentation des performances globales du système...) que des objectifs locaux aux disciplines (problème d'optimisation disciplinaire 2.4).

Les problèmes d'optimisation multidisciplinaire sont généralement classés en deux catégories : hiérarchiques et non hiérarchiques.

Comme l'illustre le tableau 1 du chapitre d'introduction, nous distinguons ainsi plusieurs types de problèmes d'optimisation multidisciplinaire suivant la présence ou non de problèmes d'optimisation locaux ou globaux :

- problème d'optimisation uniquement global,
- problèmes d'optimisation uniquement locaux,
- problèmes d'optimisation globaux et locaux.

Le premier cas ajoute l'analyse multidisciplinaire à un problème d'optimisation. Il s'agit du type de problème d'optimisation le plus étudié dans la littérature (voir par exemple Cramer *et al.* 1993, Balling et Sobieszczanski-Sobieski 1994, Alexandrov et Kodiyalam 1998, Alexandrov et Lewis 1999, Hulme et Bloebaum 2000, Dennis *et al.* 2005, DeMiguel et Murray 2006). Le second peut être vu comme un problème de théorie des jeux (Nash 1951, Chanron et Lewis 2005). En théorie des jeux, cependant, seuls les problèmes mono-objectifs par joueur sont considérés. Dans ce cas, les solutions optimales sont les solutions optimales au sens de Pareto, il ne s'agit cependant pas toujours des solutions d'équilibre.

Quel que soit le type de problème d'optimisation multidisciplinaire, les fonctions coûts locales et globales sont souvent agrégées en une somme pondérée (voir par exemple Balling et Rawlings 2000, Gantois et Morris 2004, Lee et Jeong 2006) ou une fonction multiobjectif (voir par exemple Gunawan *et al.* 2003), soit des sous-ordres de $(\mathbb{R}, \leq_{\mathbb{R}})$ dans le cas de l'agrégation des fonctions en somme pondérée ou de $(\mathbb{R}^p, \leq_{\mathbb{R}^p})$ dans le cas multiobjectif.

2.2 Méthodes de résolution

Les méthodes de résolution des problèmes d'optimisation multidisciplinaire ont fait l'objet de nombreux articles de synthèse de la littérature (Balling et Sobieszczanski-Sobieski 1994, Sobieszczanski-Sobieski et Haftka 1997, Alexandrov et Lewis 1999, Dépincé *et al.* 2007, Yi *et al.* 2008, Tosserams *et al.* 2009). Les méthodes d'optimisation multidisciplinaire y sont généralement classées suivant l'un des critères suivants :

Niveaux d'optimisation (mononiveau ou multiniveau). Une méthode mononiveau est une méthode ne comportant qu'un seul optimiseur, tandis qu'une méthode multiniveau en possède à la fois dans les disciplines et au niveau global.

Intégration de l'analyse et de l'optimisation Balling et Sobieszczanski-Sobieski (1994) introduisent un autre type de classification, en faisant une distinction entre SAND (*Simultaneous Analysis and Design*³) et NAND (*Nested Analysis and Design*⁴). Cette distinction peut être opérée au niveau global ou disciplinaire. SAND signifie que les variables de conception et d'état sont déterminées simultanément par l'optimiseur, tandis que NAND signifie que l'optimiseur ne détermine que les variables de conception, mais requiert les variables d'état à chaque itération.

3. Analyse et optimisation simultanées.

4. Analyse et optimisation emboîtées.

Traitement des contraintes Alexandrov et Lewis (1999) classent les méthodes suivant trois critères, suivant que les contraintes dans l'analyse disciplinaire, l'optimisation ou la cohérence interdisciplinaire soient ouvertes ou fermées.

Ces classifications s'intéressent à la manière dont les phases d'optimisation et d'analyse interagissent. Notre problématique est différente, puisque nous ne nous intéressons non pas à la résolution des couplages, mais à la définition des solutions de compromis. Parmi les méthodes existantes, nous distinguons trois approches différentes :

1. les méthodes de reformulation de problèmes qui, se basant sur des formulations mathématiquement équivalentes, ont pour objectif de donner le maximum d'indépendance aux disciplines, tout en accélérant la convergence des algorithmes ;
2. les méthodes basées sur la théorie des jeux ;
3. les méthodes basées sur les algorithmes évolutionnaires multiobjectifs.

Afin d'illustrer les formulations, nous présenterons le cas d'un problème à deux disciplines. Dans les schémas, les notations suivantes sont utilisées :

- O et O_i pour respectivement l'optimiseur global et l'optimiseur de la discipline i ;
- A et A_i pour respectivement l'analyse multidisciplinaire et l'analyse de la discipline i ;
- E_i pour l'évaluation de la discipline i .

2.2.1 Reformulations de problèmes

Les méthodes présentées dans cette section se basent sur des reformulations mathématiques du même problème. Les solutions recherchées sont équivalentes au problème initial.

Multidisciplinary-Feasible Method (MDF)

Multidisciplinary-Feasible Method (MDF) est la méthode la plus courante pour résoudre un problème d'optimisation multidisciplinaire (Cramer *et al.* 1993). Il s'agit de la méthode la plus naturelle pour résoudre les problèmes d'optimisation multidisciplinaire n'ayant qu'un objectif global. L'analyse multidisciplinaire est réalisée à chaque pas d'optimisation :

$$\begin{cases} \min_{x \in X} \phi(x, w) \\ \text{t.q. } A_1(x_1, c_{2,1}(v_2), v_1, w_1) = 0 \\ \quad A_2(x_2, c_{1,2}(v_1), v_2, w_2) = 0 \end{cases} \quad (2.25)$$

À chaque itération de l'algorithme d'optimisation, une solution multidisciplinairement réalisable est obtenue.

Le processus de résolution est une boucle (figure 2.9) entre l'optimiseur, qui évalue la fonction objectif et propose de nouvelles valeurs de x et l'analyseur multidisciplinaire, qui cherche une solution admissible du système et retourne une solution x' admissible et les valeurs w correspondant à ce x' pour évaluer la fonction coût. À chaque itération, on a donc une solution multidisciplinairement réalisable du problème (x', w) , même si elle n'est pas forcément optimale. Cependant, l'analyse multidisciplinaire est très coûteuse en temps de calcul. De plus, si l'analyse multidisciplinaire est réalisée par une méthode de point fixe, elle peut ne pas converger si l'application considérée n'est pas contractante⁵.

5. Voir le théorème du point fixe de Banach (1922).

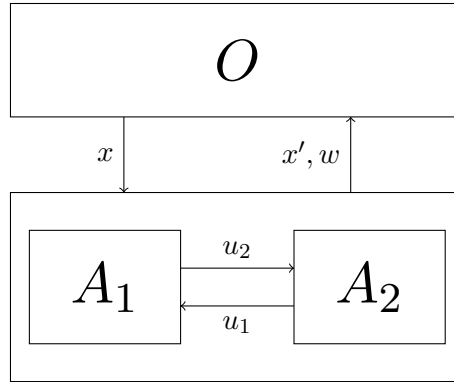


FIGURE 2.9 – La méthode MDF : à chaque itération, l’optimiseur propose une valeur x , une analyse multidisciplinaire complète est réalisée et une nouvelle solution (x', w) est évaluée.

Définition 2.7 (Application contractante) Soient E un espace métrique et f une application de E dans E . f est contractante si $\exists \rho \in]0; 1[$ tel que, $\forall x, y \in E, |f(x) - f(y)| \leq \rho|x - y|$.

Algorithme 1 *Multidisciplinary Feasible Method*

ENTRÉES : solution initiale x

SORTIES : Sortie : solution (x', w) , évaluation ϕ

répéter

$(x', w) \leftarrow \text{MDA}(x)$

$x, \phi \leftarrow \text{optimisation}(x', w)$

jusqu'à critère d'arrêt

retourner $(x', w), \phi$

L’algorithme 1 décrit les étapes de la méthode. La fonction $\text{MDA}()$ est l’analyse multidisciplinaire. On considère ici qu’elle s’occupe de l’initialisation des variables u . La fonction $\text{optimisation}()$ est l’optimiseur global. Il s’agit d’un algorithme itératif. La solution retournée par l’algorithme n’est pas nécessairement la solution optimale. Une analyse complète étant réalisée à chaque pas d’optimisation, une solution satisfaisante peut être obtenue avant la fin du processus d’optimisation. Le critère d’arrêt de l’algorithme peut donc être une condition d’optimalité, un nombre d’itérations fixé, ou un temps donné.

Il n’y a ici qu’un unique optimiseur, donc MDF ne définit pas comment le compromis entre les disciplines est réalisé si chacune d’elles dispose d’un objectif propre. Ce compromis reste donc à la charge de l’optimiseur employé.

Multidisciplinary design optimization based on independent subspaces (MDOIS) proposée par Shin et Park (2005) est similaire à MDF car, à chaque pas d’optimisation, une analyse multidisciplinaire complète est réalisée. Cependant, chaque discipline dispose de son propre objectif qui est traité par un optimiseur indépendant. Le problème d’optimisation global est la somme des objectifs disciplinaires. Selon les auteurs, MDOIS est validé mathématiquement comme satisfaisant les conditions d’optimalité de Karush, Kuhn et Tucker.

All-At-Once (AAO)

Contrairement à MDF, la méthode *All-At-Once* (AAO) ne traite pas séparément l'optimisation et l'analyse. L'analyse multidisciplinaire est intégrée au problème d'optimisation sous forme de contraintes. Il en résulte que cette méthode ne suppose aucune réalisabilité disciplinaire au cours du processus de résolution, et les solutions réalisables et optimales ne sont garanties qu'à convergence de l'algorithme.

Les équations d'état sont converties en contraintes et toutes les variables sont considérées comme variables d'optimisation. Ceci évite les analyses itératives des sous-systèmes, qui sont coûteuses en temps. Si les codes utilisés le permettent, on peut également remplacer l'analyse disciplinaire par des évaluations moins coûteuses en temps de calcul. Le problème devient :

$$\left\{ \begin{array}{l} \min_{(x,u,v,w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{V} \times \mathcal{W}} \phi(x, w) \\ \text{t.q. } r_1 := A_1(x_1, u_1, v_1, w_1) \\ \quad r_2 := A_2(x_2, u_2, v_2, w_2) \\ \quad u_1 = c_{2,1}(v_2) \\ \quad u_2 = c_{1,2}(v_1) \\ \quad r_1 = 0 \\ \quad r_2 = 0 \end{array} \right. \quad (2.26)$$

où r_1 et r_2 sont les résidus des évaluations, qui doivent converger vers zéro. Le symbole $:=$ représente ici une valeur qui est calculée, par opposition au $=$ qui représente l'évaluation d'une expression (ici des contraintes). \mathcal{X} , \mathcal{U} , \mathcal{V} et \mathcal{W} sont les ensembles admissibles et réalisables de respectivement \mathcal{X} , \mathcal{U} , \mathcal{V} et \mathcal{W} .

L'optimiseur propose aux disciplines des variables d'entrée et de sortie et chaque discipline évalue que les solutions proposées par l'optimiseur sont bien disciplinairement réalisables. Les résidus calculés sont ainsi remontés à l'optimiseur (figure 2.10, algorithme 2).

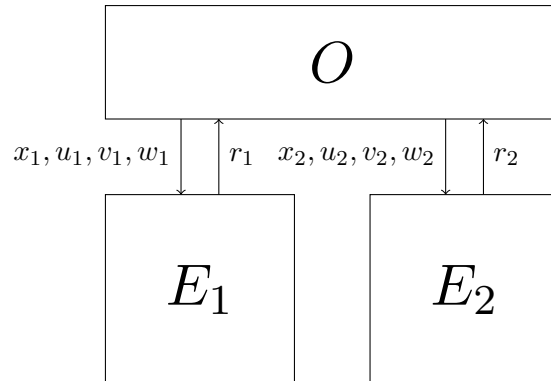


FIGURE 2.10 – La méthode AAO : toutes les variables d'entrée et de sortie des disciplines sont traitées comme des variables d'optimisation, et les équations d'état sont traitées comme des contraintes.

Cette formulation permet d'effectuer des itérations dont la solution n'est pas réalisable. L'augmentation de l'espace des variables permet une plus grande liberté dans l'optimisation et permet ainsi de tendre plus directement vers un optimum en relaxant les

Algorithme 2 *All-At-Once*

ENTRÉES : solution initiale x, u, v, w
SORTIES : Sortie : solution (x, w) , évaluation ϕ
répéter
 pour tout discipline i **faire**
 $r_i \leftarrow \text{évaluation}(i, x_i, u_i, v_i, w_i)$ //évaluations disciplinaires
 pour tout discipline $j \neq i$ **faire**
 $\delta_{i,j} \leftarrow |c_{i,j}(v_i) - u_j|$ //évaluation des couplages
 fin pour
 fin pour
 $(x, u, v, w), \phi \leftarrow \text{optimisation}(x, u, v, w, r, \delta)$
jusqu'à $\forall i, j, r_i = 0 \wedge \delta_{i,j} = 0$ ou optimalité
retourner $(x, w), \phi$

contraintes. Cette augmentation du nombre de variables d'optimisation et de contraintes est cependant plus coûteuse pour l'optimiseur.

Comme pour MDF, il n'y a ici qu'un optimiseur global et la notion de compromis n'est pas définie dans la formulation.

Individual Disciplinary Feasible (IDF)

Individual Disciplinary Feasible (IDF) (Dennis et Lewis 1994) est une sorte de compromis entre MDF et AAO. Elle assure que chaque discipline est indépendamment réalisable à chaque itération, tandis que le système entier ne sera réalisable qu'à la fin. C'est l'optimiseur qui assure la convergence vers une solution multidisciplinairement réalisable et optimale en contrôlant les données multidisciplinaires.

Dans cette formulation, on introduit des variables de substitution pour chaque variable de couplage qui seront utilisées par l'optimiseur et données à chaque discipline. Des contraintes supplémentaires sont également ajoutées au système pour garantir qu'à la convergence, chaque variable de couplage est bien égale à sa variable de substitution correspondante. Le problème est alors :

$$\left\{ \begin{array}{l} \min_{(x,u,v') \in \mathcal{X} \times \mathcal{U} \times \mathcal{V}} \phi(x, w) \\ \text{t.q. } A_1(x_1, u_1, v_1, w_1) = 0 \\ \quad A_1(x_2, u_2, v_2, w_2) = 0 \\ \quad u_1 = c_{2,1}(v'_2) \\ \quad u_2 = c_{1,2}(v'_1) \end{array} \right. \quad (2.27)$$

où v' est une variable de substitution de v . Les équations d'état sont ici résolues (les v_i et w_i sont calculés) et on évalue par des contraintes que les fonctions de couplage sont satisfaites.

À chaque itération, une solution disciplinairement réalisable est calculée par chacune des disciplines. Les variables de couplage de substitution sont données par l'optimiseur pour évaluer, à chaque itération, la satisfaction des fonctions de couplage et ainsi la réalisabilité multidisciplinaire (figure 2.11, algorithme 3).

Comme pour AAO, l'ajout de variables d'optimisation et de contraintes augmente la taille du problème d'optimisation par rapport à MDF. La taille du problème considéré

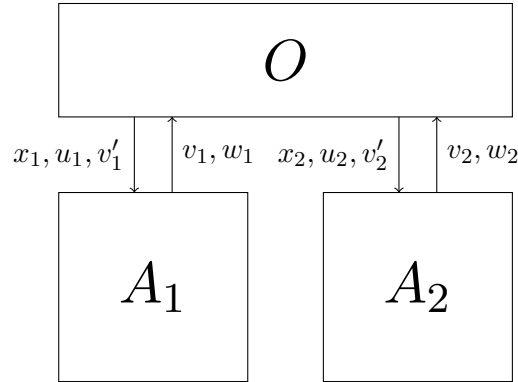


FIGURE 2.11 – La méthode IDF : des variables de substitution des variables de couplage sont ajoutées aux variables d'optimisation et sont passées aux disciplines à chaque itération. Chaque discipline résout ses équations d'état et vérifie la satisfaction des fonctions de couplage.

est cependant inférieure à AAO, du fait du plus faible nombre de variables d'optimisation ajoutées.

Algorithme 3 *Individual Disciplinary Feasible*

ENTRÉES : : solution initiale x, u, v'

SORTIES : Sortie : solution (x, w) , évaluation ϕ

répéter

pour tout discipline i **faire**

$(v_i, w_i) \leftarrow \text{analyse}(i, x_i, u_i)$ //analyse disciplinaire

pour tout discipline $j \neq i$ **faire**

$\delta_{i,j} \leftarrow |c_{i,j}(v_i) - u_j|$ //vérification des couplages

fin pour

fin pour

$(x, u, v'), \phi \leftarrow \text{optimisation}(x, u, v, v', w, \delta)$

jusqu'à $\forall i, j, \delta_{i,j} = 0$ ou optimalité

retourner $(x, w), \phi$

Comme pour MDF et AAO, l'optimisation est globale et le compromis n'y est pas défini. Ces trois méthodes n'utilisent qu'un unique optimiseur et le problème est traité globalement. Elles sont qualifiées de méthodes mononiveaux, par opposition aux méthodes multiniveaux qui utilisent plusieurs optimiseurs.

Collaborative Optimization (CO)

La méthode CO (*Collaborative Optimization*) (Braun 1996) fournit plus de liberté aux disciplines que les méthodes présentées précédemment. Les variables partagées et de couplage sont découplées dans chaque discipline. Contrairement à MDF, AAO et IDF, cette méthode utilise des optimiseurs globaux et locaux aux disciplines. Un optimiseur système (global) est en charge d'optimiser la fonction objectif, tout en garantissant que les variables couplées et partagées soient globalement cohérentes. Les disciplines réalisent, quant à elles, les analyses disciplinaires et ont pour objectif de minimiser l'écart entre les variables globales et les variables disciplinaires.

L'optimiseur système minimise la fonction coût et assure la cohérence des variables :

$$\begin{cases} \min_{(x^c, u') \in \mathcal{X}^c \times \mathcal{U}} \phi(x, w) \\ \text{t.q. } \|x^c - x_1^c\| + \|u'_1 - u_1\| = 0 \\ \|x^c - x_2^c\| + \|u'_2 - u_2\| = 0 \end{cases}$$

où les valeurs de w et de u sont données par les optimiseurs disciplinaires. Ces derniers résolvent leurs équations d'états respectives en minimisant la différence entre les valeurs des variables communes et de couplage données par le système et celles calculées :

$$\begin{cases} \min_{(x_i, u_i) \in \mathcal{X}^c \times \mathcal{U}_i} \|x^c - x_i^c\| + \|u'_j - u_j\| \\ \text{t.q. } A_i(x_i, u_i, v_i, w_i) = 0 \\ u_j := c_{i,j}(v_i) \end{cases}$$

où $i, j \in \{1, 2\}$ avec $i \neq j$ représentent les deux disciplines, chacune composée d'un optimiseur, que nous notons O_i et un analyseur noté A_i (figure 2.12).

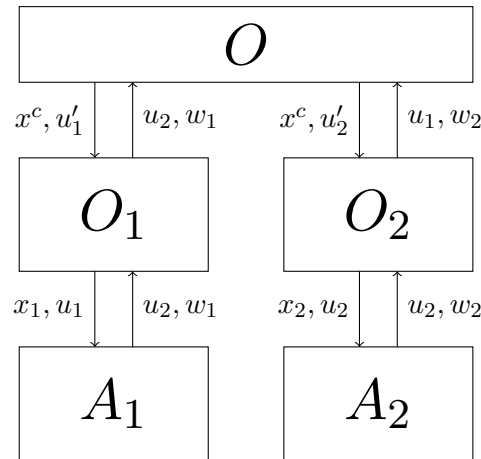


FIGURE 2.12 – La méthode CO : chaque discipline dispose de son propre optimiseur ; dont le rôle est de réduire l'incohérence entre les variables partagées qui sont découplées.

L'algorithme 4 décrit les grandes étapes de la méthode. Les optimisations disciplinaires sont ici réalisées à l'intérieur d'une boucle, mais elles peuvent être exécutées en parallèle.

Il n'existe pas de preuve de convergence de CO et la formulation peut poser des problèmes aux méthodes d'optimisation basées sur le gradient (Alexandrov et Lewis 1999, DeMiguel et Murray 2000). Une amélioration de cette méthode est proposée dans (DeMiguel et Murray 2000) appelée MCO (*Modified CO*) afin de corriger ce problème.

Bien qu'il y ait des optimisations locales, le problème d'optimisation reste global.

Autres formulations

MDF, AAO, IDF et CO sont des formulations mathématiques du même problème initial. D'autres méthodes intègrent des méthodes d'analyse de sensibilité et de surface de réponse pour approximer les analyses disciplinaires et multidisciplinaires et améliorer la convergence de l'algorithme.

Algorithme 4 *Collaborative Optimization*

ENTRÉES : solution initiale x^c, u'
SORTIES : solution (x, w) , évaluation phi
répéter
 pour tout discipline i **faire**
 $(x_i, u_i, w_i) \leftarrow \text{optimisationD}(i, x^c, u_i, u'_i)$
 $(v_i, w_i) \leftarrow \text{analyse}(i, x^c, u'_i)$
 $u_i \leftarrow c_i(v_i)$
 fin pour
 $\forall i, \delta_i = \|x^c - x_i^c\| + \|u'_i - u_i\|$
 $(x^c, u'), \phi \leftarrow \text{optimisationS}(x, u, \delta)$
jusqu'à critère d'arrêt
retourner $(x, w), \phi$

IDP (*Inexact Penalty Decomposition*), EDP (*Exact Penalty Decomposition*) (DeMiguel 2001, DeMiguel et Murray 2006) et DO (*Distributed Optimizations*) (Tribes *et al.* 2005) sont des raffinements de la méthode CO, pour lesquelles l'équivalence de la décomposition par rapport au problème initial est démontrée. Ces méthodes considèrent l'objectif global comme la somme des objectifs disciplinaires.

Tout comme CO, la méthode CSSO (*Concurrent SubSpace Optimization*) proposée par Sobieszcanski-Sobieski (1988a) permet l'utilisation d'optimiseurs disciplinaires. Pour cela, la méthode s'inspire des méthodes de décomposition en sous-domaines (*SubSpace Optimization* - SSO) qui consistent à optimiser un sous-ensemble de variables en gardant les autres fixes. Une telle méthode demande cependant beaucoup d'itérations au niveau système. Elle ne permet pas de réaliser des optimisations concourantes. Pour pallier ce problème, une analyse de sensibilité est réalisée sur les variables de conception afin de connaître leur influence sur les sorties des disciplines. Une matrice des dérivées partielles (appelées équations de sensibilité globale⁶) est calculée pendant l'analyse de sensibilité. Chaque discipline réalise alors une optimisation en fonction de ses variables disciplinaires seulement, en approximant les autres disciplines par une approximation linéaire de leur équations d'état donnée par les équations de sensibilité. Le système sert, quant à lui, à coordonner les sous-systèmes. La convergence de la méthode a été étudiée (Shankar *et al.* 1993), et plusieurs variantes de CSSO ont été proposées (Bloebaum 1991, Renaud et Gabriele 1993), notamment l'ajout de la prise en compte des variables communes (Wujek *et al.* 1996).

CPCO (*Concurrent Process-Circuit Optimization*) (Brockman et Lokanathan 1999) est une méthode basée sur CSSO appliquée à la conception de circuits électroniques où le problème d'optimisation multiobjectif est traité par *goal programming*.

DBD (*Decomposition Based Design method*) (Lee et Jeong 2006) se base sur les mêmes principes que CSSO mais en permettant à chaque discipline de réaliser une optimisation disciplinaire qui lui est propre.

Comme CSSO, la méthode *Bi-Level Integrated System Synthesis* (BLISS) proposée par Sobieszcanski-Sobieski *et al.* (1998) s'appuie sur des outils d'approximation d'analyse disciplinaire et multidisciplinaire pour les optimisations. Les variables disciplinaires sont optimisées dans les disciplines par des optimiseurs dédiés, tandis que les variables communes sont optimisées globalement par un optimiseur système. Quelques améliora-

6. *Global Sensitivity Equations* ou GSE

tions de BLISS ont été proposées, telles que BLISS-2000 (Sobieszczanski-Sobieski *et al.* 2000) et EBLISS 2000 (Xu *et al.* 2009).

La méthode DIVE (*Discipline Interaction Variable Elimination*) (Masmoudi et Parte 2006), s'appuyant également sur des méthodes d'approximation, réduit le nombre de variables en trois étapes :

1. élimination des variables disciplinaires en minimisant les objectifs disciplinaires dans chacune des disciplines,
2. élimination des variables de couplage en minimisant les résidus des équations d'état,
3. élimination des variables communes en minimisant l'objectif global.

Synthèse

Toutes les méthodes présentées précédemment sont des formulations du même problème d'optimisation multidisciplinaire. Certaines de ces méthodes emploient des méthodes d'analyse et d'approximation afin de réduire les temps de calcul et d'améliorer la convergence des algorithmes. Cependant, toutes ces méthodes se concentrent principalement sur l'analyse multidisciplinaire et le problème d'optimisation traité est très souvent un problème d'optimisation global. Lorsque les disciplines ont un problème d'optimisation propre, le problème global traité est simplement une agrégation des problèmes disciplinaires sous forme de somme pondérée, de but à atteindre ou de problème multiobjectif global.

2.2.2 Théorie des jeux

Le cas où chaque discipline a un objectif propre à atteindre peut être modélisé par un jeu (Neumann et Morgenstern 1944). Un jeu est un problème de stratégie où les choix des différents protagonistes impliqués – appelés *joueurs* – ont des répercussions sur les décisions des autres.

Lewis et Mistree (1997), Chen et Lewis (1999) et Chanron *et al.* (2004) proposent de modéliser les problèmes d'optimisation multidisciplinaire sous forme de jeux : dans ce cas, les disciplines sont les joueurs. Trois interactions peuvent alors être modélisées : la coopération, la non-coopération et le modèle de meneur/suiveur.

Majd (2007) décrit l'application de la théorie des jeux à l'optimisation multidisciplinaire d'une voilure d'avion.

Modèle coopératif

Dans le modèle coopératif, chaque joueur a une information complète sur l'autre joueur. Le problème d'optimisation multidisciplinaire est alors traité comme les méthodes mononiveau (MDF, IDF ou AAO) où le problème d'optimisation est un problème d'optimisation multiobjectif. Les solutions optimales du problème sont les solutions efficaces multidisciplinairement réalisables.

Modèle non coopératif

Dans un jeu non coopératif, chaque joueur est en compétition : chacun essaye de maximiser son profit, quel que soit le choix des autres joueurs. Les solutions du modèle non coopératif sont des équilibres de Nash (1951).

1/2	se tait	dénonce
se tait	(-1/2,-1/2)	(-10,0)
dénonce	(0,-10)	(-5,-5)

TABLEAU 2.1 – Représentation des différentes configurations du dilemme du prisonnier (exemple 2.2). Ce jeu n'a qu'un unique équilibre de Nash : (-5, -5)

Définition 2.8 (Équilibre de Nash (1951)) *Soit un jeu à p joueurs. Chaque joueur i minimise une fonction coût ϕ_i de X dans Y seulement par rapport à sa variable propre $x_i \in X_i$. L'ensemble des équilibres de Nash est l'ensemble*

$$X^N := \{x \in X : \forall i \in [p], x_i \in \arg \min_{x_i \in X_i} \phi_i(x)\}$$

Un équilibre de Nash est une solution telle que chaque joueur, connaissant la solution choisie par les autres, ne peut améliorer son profit.

Remarque 2.3 *Un équilibre de Nash n'est pas nécessairement optimal (au sens de Pareto).*

Un exemple d'équilibre de Nash qui n'est pas un optimum au sens de Pareto est donné par le dilemme du prisonnier (exemple 2.2).

Exemple 2.2 *Deux suspects sont arrêtés par la police. Mais les agents n'ont pas assez de preuves pour les inculper, donc ils les interrogent séparément en leur faisant la même offre. « Si tu dénonces ton complice et qu'il ne te dénonce pas, tu seras remis en liberté et l'autre écoperà de 10 ans de prison. Si tu le dénonces et lui aussi, vous écopererez tous les deux de 5 ans de prison. Si personne ne se dénonce, vous aurez tous deux 6 mois de prison. »*

Ce jeu comporte trois solutions Pareto-optimales : les deux se taisent ou l'un des deux se tait ; la quatrième solution étant dominée : chacun dénonce l'autre (tableau 2.1).

Si le premier suspect se tait, le second a intérêt à le dénoncer afin d'éviter la prison. Inversement, si le premier suspect dénonce le second, le second a également intérêt à dénoncer le premier afin de minimiser la peine de prison. Dans tous les cas, le second joueur choisira de dénoncer le premier. Symétriquement, le premier dénoncera le second. Il s'agit d'un équilibre de Nash. Finalement les deux suspects seront condamnés à cinq ans de prison, alors qu'en se taisant tous les deux, ils auraient pu n'être condamnés qu'à six mois.

Pour un problème donné, il peut exister un, plusieurs, voire aucun équilibre de Nash. De plus, trouver les équilibres, s'ils existent, peut être coûteux en temps de calcul, surtout pour des systèmes décentralisés et asynchrones. Gopalakrishnan *et al.* (2006) ont étudié la convergence de tels problèmes et proposent des améliorations des méthodes existantes appliquées à la conception distribuée.

Modèle meneur/suiveur

Le modèle meneur/suiveur n'accorde pas une importance égale à chaque joueur comme dans le modèle non coopératif. Ici les joueurs sont hiérarchisés et les choix des suiveurs sont vus comme des contraintes pour les meneurs qui disposent d'un espace de

recherche plus grand (incluant les variables communes dans le cas de l'optimisation multidisciplinaire, alors que les suiveurs ne s'occupent que de leurs variables disciplinaires). Les solutions d'un tel modèle sont les solutions d'équilibre de Stackelberg.

Définition 2.9 (Équilibre de Stackelberg) *Soit un jeu à deux joueurs. Le meneur du jeu minimise une fonction coût ϕ_1 de X dans Y tandis que le suiveur minimise une fonction ϕ_2 . L'ensemble des équilibres de Stackelberg est l'ensemble*

$$X^S := \{(x_1, x_2) \in X_1 \times X_2 : x \in \underset{(x_1, x_2) \in X_1 \times X_2}{\arg \min} \phi_1(x_1, x_2) \text{ t.q. } x_2 \in \underset{x_2 \in X_2}{\arg \min} \phi_2(x_1, x_2)\}$$

La méthode appelée *Dynamic Leader-Follower Parallel Optimization* (Dy-Leaf) Tava et Suzuki (2003) se base sur ce modèle. Chaque discipline a un objectif disciplinaire à optimiser. Une des disciplines est choisie comme meneuse et optimise son objectif selon les variables partagées et ses variables disciplinaires, les autres disciplines ne modifiant que leurs variables disciplinaires. Le choix du meneur peut changer au cours de l'optimisation. Le meneur est choisi comme celui le plus loin de son objectif.

Synthèse

Peu de méthodes d'optimisation multidisciplinaire s'inspirent de la théorie des jeux. Cette dernière permet pourtant de modéliser les problèmes de décision et d'optimisation distribués où chaque discipline dispose de son propre objectif.

Si l'on considère le problème d'optimisation multidisciplinaire comme un jeu coopératif, on se ramène alors aux méthodes classiques, telles que MDF, IDF ou AAO. Les solutions optimales sont alors les solutions Pareto-optimales.

La notion de compromis est ici abordée comme un équilibre. Une solution d'équilibre n'existe cependant pas dans tous les cas. De plus une solution d'équilibre n'est pas toujours optimale.

Enfin, nous souhaitons traiter les problèmes d'optimisation multidisciplinaire où chaque discipline est un problème d'optimisation multiobjectif. Ce type de problème n'est actuellement pas traité par la théorie des jeux, car chaque joueur n'a qu'un seul objectif.

2.2.3 Méthodes basées sur les algorithmes évolutionnaires

Les méthodes présentées précédemment se basent sur des méthodes mathématiques, soit pour reformuler le problème sous une forme plus adéquate mais équivalente (méthodes de reformulation), soit pour décrire l'évolution du système (théorie des jeux). Une autre approche, plus algorithmique, consiste à se baser sur la théorie de l'évolution pour sélectionner les meilleures solutions potentielles et les faire évoluer jusqu'à convergence. Ces méthodes, basées sur les algorithmes évolutionnaires multiobjectifs (*Evolutionary Multi-Objective* – EMO) ont pour avantage d'être naturellement parallélisables, de traiter simultanément plusieurs solutions et de ne pas nécessiter la connaissance des gradients des fonctions objectifs. Elles permettent, de plus, en utilisant des EMO dans chaque discipline, de traiter les cas où chaque discipline est un problème d'optimisation multiobjectif.

Coevolutionary MDO (CMDO)

Nair et Keane (2002) proposent d'appliquer les algorithmes coévolutionnaires (Potter 1997) à l'optimisation multidisciplinaire.

Un algorithme coévolutionnaire est un algorithme génétique dans lequel plusieurs populations – appelées espèces – évoluent en parallèle. L'adaptation des individus de chaque population est évaluée indépendamment, puis des représentants de chaque espèce sont sélectionnés pour être combinés et réintroduits dans les espèces. L'algorithme est alors semblable à un algorithme génétique classique dans lequel plusieurs populations évoluent en parallèle (algorithme 5). La fonction d'évaluation des espèces est cependant différente, car elle utilise des représentants d'autres espèces (algorithme 6).

Algorithme 5 *Coevolutionary MDO*

ENTRÉES : le nombre d'individus n
SORTIES : Sortie : population solution P
pour tout Discipline $i \in I$ **faire**
 $P_i \leftarrow \text{initialisation}(n)$
 $P_i \leftarrow \text{évaluationF}(P_i)$
fin pour
répéter
 pour tout Discipline $i \in I$ **faire**
 $P_i \leftarrow \text{évolution}(P_i)$
 $E_i \leftarrow \text{évaluation}(i, P)$
 $P_i \leftarrow \text{sélection}(P_i, E_i)$
 fin pour
jusqu'à Critère d'arrêt
retourner P

La fonction `évaluationF()` évalue les individus sur les fonctions coût disciplinaires tandis que `évaluation()` les évalue sur la coopération.

Algorithme 6 `évaluation()`

ENTRÉES : identifiant espèce i , toutes les espèces P
SORTIES : évaluations E_i
 $R \leftarrow \text{choixReprésentants}(P)$
 $C \leftarrow \text{collaboration}(P_i, R)$
 $E_i \leftarrow \text{évaluationF}(C)$
retourner E_i

Les auteurs notent que la problématique des variables communes reste une question ouverte. En effet, leur algorithme considère des problèmes couplés sans variables partagées.

La coévolution se comporte comme une méthode de point fixe. Les solutions recherchées sont des équilibres de Nash. D'autres algorithmes coévolutionnaires faisant explicitement référence à la théorie des jeux ont été développés (Ficici et Pollack 2000, Sim et Kim 2004).

MORDACE

La méthode MORDACE (*Multidisciplinary Optimization and Robust Design Approaches applied to Concurrent Engineering*) proposée par Giassi (2004) consiste à effectuer les optimisations disciplinaires indépendamment, puis à chercher des solutions de compromis sur les variables communes. Pour cela, chaque discipline dispose :

- i) d'un modèle simplifié des autres disciplines, afin d'avoir une approximation des variables de couplage ;
- ii) d'une fonction de sensibilité de la fonction objectif de la discipline en fonction des variables communes. Elle est obtenue par une surface de réponse⁷.

Cette dernière va servir à réaliser une optimisation multiobjectif au sein de chaque discipline, en prenant en compte, d'une part, la fonction objectif de la discipline et, d'autre part, la sensibilité de cette fonction. Ainsi, chaque discipline cherche à la fois les meilleures performances, tout en réduisant l'importance des modifications sur les variables communes. L'optimisation disciplinaire est réalisée par un algorithme génétique multiobjectif.

La méthode part du principe que l'on dispose *a priori* d'une solution initiale. Dans ce cas, après l'étape d'optimisation disciplinaire, chaque discipline renvoie les solutions qui dominent cette solution, plus l'ensemble des solutions optimales au sens de Pareto. Dans le cas contraire, seules les solutions Pareto-optimales sont renvoyées. Un sous-ensemble de ces solutions sont sélectionnées par le décideur pour être comparées deux à deux. Les solutions les plus proches⁸ dans chaque discipline sont ensuite combinées pour donner une solution intermédiaire (une moyenne pondérée) globale.

Il est précisé que la méthode n'est pas appropriée aux problèmes fortement couplés (Voir section 5.4 p.132, Giassi 2004).

EM-MGA et MMSA&O

EM-MGA signifie *Entropy-based Multi-level Multi-objective Genetic Algorithm*. Cette méthode s'appuie sur une décomposition de l'optimisation au niveau disciplinaire. La première version proposée (Gunawan *et al.* 2003) ne prenait pas en compte les couplages et était limitée aux systèmes hiérarchiques (Gunawan *et al.* 2004).

Dans chaque discipline, un algorithme génétique multiobjectif réalise une optimisation en fonction des variables de conception disciplinaires x_i . Chaque discipline dispose d'une population de taille identique. Deux populations de solutions supplémentaires sont maintenues hors des disciplines appelées *grande population* et *grand pool*. Dans chacune de ces deux populations, les individus sont composés de la totalité des variables de conception du problème. La *grande population* est une estimation des solutions du problème global tandis que la *grand pool* est une archive de l'union des solutions des disciplines. La taille de la *grande population* est égale à la taille des populations disciplinaires, alors que la taille de la *grand pool* est k fois cette taille, k étant le nombre de disciplines.

La population de la *grande population* est utilisée pour initialiser les populations de chaque sous-problème. Seuls les chromosomes correspondant aux variables de décision de chaque discipline sont conservés dans les populations disciplinaires. Un algorithme géné-

7. Une surface de réponse est une approximation d'une fonction obtenue à partir d'un échantillon de valeurs de cette fonction. Dans le cas de MORDACE, la surface de réponse est modélisée par un réseau de neurones.

8. Les solutions les plus proches suivant une somme pondérée des distances des variables communes. Les pondérations étant les sensibilités des variables.

tique multiobjectif est appliqué dans chaque discipline, puis chaque population disciplinaire est complétée pour contenir l'ensemble des variables de décision. Les k populations ainsi créées sont ajoutées à la *grand pool*, puis sélectionnées suivant leur performance et une mesure de l'entropie pour préserver la diversité des solutions. Les meilleurs individus remplacent alors la *grande population*.

MMSA&D (*Multidisciplinary Multiobjective System Analysis & Design*) présentée dans (Farhang-Mehr et Tumer 2005, Farhang-Mehr *et al.* 2005) est une méthode d'optimisation pour les problèmes d'optimisation multiobjectif hiérarchiques. Les sous-problèmes disposent de leurs propres objectifs, ainsi qu'une partie des objectifs globaux sur lesquels ils ont une influence. Chaque sous-système résout son problème d'optimisation multiobjectif. Les systèmes au dessus, pour minimiser le biais introduit par les optimisations disciplinaires, optimisent une métrique (H), afin de garantir une plus grande diversité des solutions et une métrique (S), afin d'améliorer la convergence. La méthode est censée converger vers les solutions du problème d'optimisation multiobjectif global.

COSMOS

COSMOS est une méthode d'optimisation multidisciplinaire multiobjectif qui signifie *Collaborative Optimization Strategy for Multi-Objective Systems* (Rabeau 2007).

Le principe de COSMOS, tout comme MORDACE, est d'utiliser des algorithmes génétiques multiobjectifs dans le cadre de la conception collaborative et l'optimisation multidisciplinaire. Chaque discipline travaille sur une population de solutions pour réaliser une optimisation multiobjectif et un superviseur est en charge de trouver un compromis entre les différentes disciplines. Contrairement à MORDACE, COSMOS est cependant conçue pour les problèmes disciplinaires qui sont initialement multiobjectifs. La robustesse des solutions n'est pas recherchée pour le compromis.

Le superviseur propose un ensemble de variables communes, tandis que chaque discipline fournit un ensemble de variables disciplinaires (suivant leur expérience) de même taille, afin de créer une population de solutions initiales. Le superviseur sauvegarde alors la population créée et redescend à chaque discipline l'ensemble des variables communes et ses variables disciplinaires. Chaque discipline peut alors appliquer un algorithme évolutionnaire d'optimisation multiobjectif pour optimiser les solutions données. À la fin de cette procédure d'optimisation, chaque sous-système (discipline) transfère sa nouvelle population disciplinaire au superviseur, qui l'ajoute alors à sa population sauvegardée. Une nouvelle génération de la taille de la population d'origine est obtenue à partir de cette population de taille double (sélection, croisements, mutations sur les variables communes). Cette procédure se poursuit jusqu'à un nombre d'itérations fixé par le superviseur. Les optimisations systèmes ne sont effectuées que sur les variables communes, tandis que les optimisations des sous-systèmes ne sont effectuées que sur les variables disciplinaires.

L'algorithme 7 présente COSMOS sous une forme plus concise. La fonction `initialisation()` crée une population initiale aléatoirement ou suivant l'expérience du décideur. Les fonctions `optimisationDisciplinaire()` et `optimisationGlobale()` sont des algorithmes génétiques multiobjectifs appelés eN-MOGA, qui sont une évolution⁹ de l'algorithme génétique MOSGA (Andersson 2001). `optimisationDisciplinaire()` ne modifie que les variables disciplinaires, alors que `optimisationGlobale()` ne modifie

9. La principale modification de l'algorithme original réside dans l'ajout d'une méthode appelée *GeneToKillTheWorst* consistant, non plus à remplacer systématiquement les enfants par les individus les plus proches dans la population, mais à remplacer périodiquement (à une période fixée par l'utilisateur) les individus les plus mauvais de la population.

que les variables communes. Les fonctions `decompose()` et `recompose()` servent respectivement à obtenir les populations disciplinaires à partir de la population globale et à obtenir la population globale à partir des populations disciplinaires. Les populations disciplinaires ne contiennent pas les variables disciplinaires des autres disciplines.

Algorithme 7 *Collaborative Optimization Strategy for Multi-Objective Systems*

ENTRÉES : le nombre d'individus n

SORTIES : solutions P

$P \leftarrow \text{initialisation}(n)$

répéter

$P' \leftarrow P$ //sauvegarde de la population

pour tout $i \in I$ **faire**

$P_i \leftarrow \text{decompose}(P, i)$ //décomposition en population disciplinaire

$P_i \leftarrow \text{optimisationDisciplinaire}(P_i)$

fin pour

$P \leftarrow \text{recompose}((P_i)_{i \in I})$ //recomposition de la population globale

$P \leftarrow \text{optimisationGlobale}(P \cup P')$

jusqu'à Nombre d'itérations système

retourner Individus non dominés de P

L'analyse du traitement des couplages dans COSMOS est détaillée dans l'annexe C. Les solutions de compromis attendues sont les solutions du front de Pareto du problème multiobjectif incluant l'ensemble des objectifs du problème.

Cette décomposition de problèmes multidisciplinaires multiobjectifs en une optimisation locale sur les variables disciplinaires, et une optimisation globale sur les variables partagées, est similaire à l'approche proposée par Papalambros (2002) : le système global est décomposé en sous-systèmes ayant chacun un objectif à atteindre. La décomposition comporte des variables communes, mais pas de variables de couplage. Comme pour COSMOS, chaque sous-système optimise son problème disciplinaire en ne modifiant que ses variables locales. Les variables sont, quant à elles, modifiées par un optimiseur global qui minimise la somme des objectifs disciplinaires.

Une seconde version de COSMOS, nommée COSMOS-L (local)¹⁰, donne plus d'autonomie aux disciplines en leur permettant de réaliser leurs optimisations sur l'ensemble des variables (communes et disciplinaires). L'optimiseur système est alors en charge de trouver des valeurs de compromis pour les variables communes.

Synthèse

Les méthodes d'optimisation multidisciplinaire basées sur les algorithmes évolutionnaires sont par nature des méthodes multiobjectifs permettant de proposer au concepteur un ensemble de solutions. Les solutions recherchées sont soit les solutions d'équilibre de Nash (CMDO), soit les solutions Pareto-optimales (EM-MGA, MMSA&O, COSMOS) ou bien encore des solutions Pareto-optimales robustes (MORDACE).

Nous pouvons également citer les méthodes proposées par Mouelhi (2010). Il ne s'agit pas de méthodes basées sur des algorithmes évolutionnaires, mais sur une métaheuristique (recherche tabou) et une méthode hybride (*branch&bound*/recherche locale). Dans ces méthodes, comme pour les méthodes basées sur les algorithmes évolutionnaires, le

10. La version originale étant alors nommée COSMOS-G (global).

problème d'optimisation multidisciplinaire est traité comme un problème d'optimisation multiobjectif.

Conclusion

Dans ce chapitre, nous avons tout d'abord posé la formalisation des problèmes d'optimisation multidisciplinaire. Nous avons réalisé une synthèse des différentes méthodes de résolution proposées.

Fadel et Wiecek (2006) notent que les méthodes MDF, AAO, IDF et CO ne sont que des reformulations du même problème, tandis que des méthodes telles CSSO et BLISS se déroulent en plusieurs étapes et utilisent des outils et formulations mathématiques complexes. Nous pouvons y ajouter BLISS-2000, DIVE et MDOIS.

De la même manière, nous pouvons considérer que COSMOS est une implémentation particulière de *Non-convex Coordination Strategy* (Papalambros 2002) dédiée aux problèmes disciplinaires multiobjectifs et utilisant des algorithmes génétiques. De même que CMDO est une implémentation par des algorithmes évolutionnaires des méthodes basées sur la théorie des jeux.

La majorité des méthodes sont des reformulations d'un problème d'optimisation multidisciplinaire ne considérant qu'un objectif global. Les méthodes se concentrent sur la résolution des couplages (l'analyse multidisciplinaire) et n'abordent pas la question des compromis entre les objectifs disciplinaires. Les méthodes considérant que les disciplines ont des objectifs propres traitent ces objectifs globalement en les agrégeant au niveau système. Les solutions sont alors les minimums d'une fonction d'agrégation (souvent une somme pondérée) ou les solutions efficaces d'un problème d'optimisation multiobjectif dans lequel tous les objectifs sont considérés au même niveau.

La théorie des jeux permet d'appréhender les problèmes d'optimisation multidisciplinaire sans objectif global mais dans lesquels chaque discipline dispose de son propre objectif. Les solutions optimales sont alors les solutions Pareto-optimales. Cependant, suivant la nature des relations entre les disciplines, les méthodes vont converger vers des points d'équilibre (de Nash ou de Stackelberg) qui ne sont pas toujours optimaux.

Enfin, les méthodes basées sur les algorithmes évolutionnaires prennent plus naturellement en compte les multiples objectifs que peuvent avoir les disciplines. Les solutions recherchées au niveau global sont là aussi les solutions efficaces du problème d'optimisation multiobjectif.

Dans le chapitre suivant, nous montrons tout d'abord que les solutions proposées ne sont pas toutes des solutions de compromis pertinentes pour des problèmes d'optimisation multidisciplinaire dont les disciplines sont des problèmes d'optimisation multiobjectif. Nous proposons alors plusieurs définitions du compromis que nous analysons.

CHAPITRE 3

Le compromis en optimisation multidisciplinaire multiobjectif

SOMMAIRE

INTRODUCTION	60
3.1 PROBLÉMATIQUE	60
3.1.1 Définition du problème	60
3.1.2 Traitement du compromis dans les méthodes d'OMD	62
3.1.3 Limites des méthodes actuelles	62
3.2 \mathcal{C}_1 : LE PRODUIT D'ORDRE	64
3.2.1 Principe	64
3.2.2 Formalisation	64
3.3 \mathcal{C}_2 : UNE EXTENSION DU PRODUIT	66
3.3.1 Principe	66
3.3.2 Formalisation	67
3.3.3 Propriétés	72
3.4 \mathcal{C}_3 : LE COMPROMIS COMME LE RANG DES RANGS	74
3.4.1 Principe	74
3.4.2 Formalisation	75
3.4.3 Propriétés	76
3.5 INCONVÉNIENTS DES COMPROMIS \mathcal{C}_2 ET \mathcal{C}_3	76
3.6 \mathcal{C}_4 : UN PRODUIT FAIBLE ENTRE LES DISCIPLINES	79
3.6.1 Principe	79
3.6.2 Formalisme	79
3.6.3 Propriétés	81
CONCLUSION	81

Introduction

LES méthodes de résolution de problèmes d'optimisation multidisciplinaire s'intéressent principalement à la résolution des couplages (*Cf.* les méthodes de reformulation, section 2.2.1, et les méthodes basées sur la théorie des jeux, section 2.2.2). Bien que les disciplines soient souvent hiérarchisées (du fait de la décomposition du système en sous-systèmes eux-mêmes décomposables en sous-systèmes) et qu'elles possèdent plusieurs objectifs propres, les solutions proposées par les méthodes d'optimisation multidisciplinaire sont équivalentes à la résolution d'un problème d'optimisation mono-objectif ou multiobjectif, sans considération des préférences disciplinaires.

Dans ce chapitre, nous proposons des définitions de solutions de compromis entre disciplines. Dans le cas où les problèmes d'optimisation disciplinaires sont mono-objectifs, le compromis est défini comme l'ensemble des solutions efficaces du problème global, qui est un problème d'optimisation multiobjectif. Mais dans le cas où chaque problème disciplinaire est multiobjectif, il faut définir les solutions de compromis entre plusieurs problèmes d'optimisation multiobjectif.

Nous définissons les solutions de compromis comme des solutions optimales de fonctions de coûts modélisant les préférences des disciplines sur les solutions. L'optimalité d'une solution peut être définie par la relation d'ordre appliquée à l'espace de comparaison du problème (*Cf.* section 1.2.3). Dans ce chapitre, nous décrivons quatre structures ordonnées différentes donnant chacune une modélisation possible de l'espace de comparaison dans le cadre du problème multidisciplinaire où chaque discipline a un problème d'optimisation multiobjectif à résoudre. À ces quatre structures ordonnées correspondent quatre définitions de compromis. À partir des propriétés de ces structures, nous déduisons les capacités de convergence d'un algorithme d'optimisation évolutionnaire implémentant ces compromis.

La relation d'ordre étant définie dans l'espace ordonné, le compromis sera défini dans ce même espace. Les quatre compromis définis sont notés \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 . $X^{\mathcal{C}_i} \in X$ et $Z^{\mathcal{C}_i} \in Z$ sont respectivement les solutions de compromis pour \mathcal{C}_i dans l'espace de décision et de comparaison.

3.1 Problématique

Nous distinguons les problèmes d'optimisation multiobjectif hiérarchisés des problèmes d'optimisation multiobjectif classiques. Nous montrons que les méthodes existantes d'optimisation multidisciplinaire ne sont pas adaptées à la résolution de problèmes d'optimisation multiobjectif hiérarchisés.

3.1.1 Définition du problème

Soit un problème d'optimisation multidisciplinaire impliquant q disciplines. Chaque discipline i doit optimiser un problème à p_i objectifs, et l'ensemble des solutions admissibles du problème est noté X . Le nombre total d'objectifs du problème est noté p avec $p := \sum_1^q p_i$. Soit \mathcal{M} ce problème d'optimisation :

$$\min_{x \in X} \phi(x) := (\phi_1(x), \dots, \phi_q(x)) \quad (3.1)$$

avec pour tout $i \in \llbracket q \rrbracket$, $\phi_i(x) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{p_i}$ la fonction coût de la discipline i .

Chaque discipline i a un problème d'optimisation multiobjectif \mathcal{M}_i à résoudre :

$$\min_{x \in X_i} \phi_i(x) := (\phi_{i,1}(x), \dots, \phi_{i,p_i}(x)) \quad (3.2)$$

Dans une première analyse, nous ne considérerons que les problèmes sans variable locale et définis sur le même ensemble admissible : $X_i = X \forall i$.

Remarque 3.1 *Les variables locales peuvent néanmoins être prises en compte en considérant un ensemble admissible X_i identique pour chaque discipline incluant les variables locales. Les variables inutiles à une discipline ne sont simplement pas prises en compte dans sa fonction coût.*

Les fonctions coûts sont naturellement hiérarchiquement organisées¹ (figure 3.1).

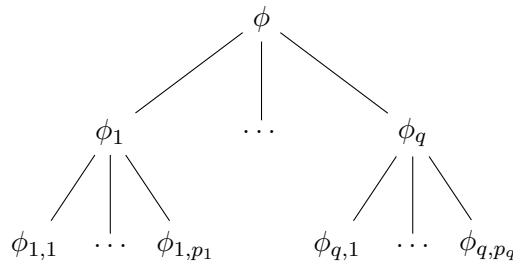


FIGURE 3.1 – Les objectifs d'un problème d'optimisation multidisciplinaire multiobjectif sont hiérarchiquement organisés : l'objectif global (la racine : ϕ) est décomposé en objectifs disciplinaires (premier niveau : ϕ_i), qui sont eux-mêmes décomposés en plusieurs objectifs (les feuilles : $\phi_{i,j}$).

Nous appellerons \mathcal{M}' le problème d'optimisation multiobjectif composé de l'union de tous les objectifs disciplinaires :

$$\min_{x \in X} \phi(x) := \min_{x \in X} (\phi_{1,1}(x), \dots, \phi_{1,p_1}(x), \dots, \phi_{q,1}(x), \dots, \phi_{q,p_q}(x)) \quad (3.3)$$

Contrairement au problème \mathcal{M} , il n'y a aucune hiérarchie entre les objectifs dans le problème \mathcal{M}' (figure 3.2).

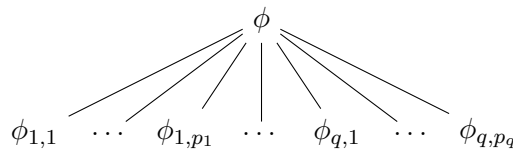


FIGURE 3.2 – Dans le problème d'optimisation multiobjectif \mathcal{M}' , tous les objectifs sont considérés au même niveau.

1. Il est important de noter que cette décomposition hiérarchique des objectifs n'est pas la décomposition des problèmes d'optimisation multidisciplinaire suivant les couplages. En effet, un problème d'optimisation multidisciplinaire peut ne pas avoir de décomposition hiérarchique (par exemple toutes les disciplines sont directement couplées), mais tout de même avoir une hiérarchie des objectifs.

3.1.2 Traitement du compromis dans les méthodes d'optimisation multidisciplinaire

Nous rappelons ici brièvement les méthodes d'optimisation multiobjectif et la manière dont le compromis est obtenu².

Les méthodes d'optimisation multidisciplinaire mononiveaux – telles que MDF, IDF et AAO (*Cf.* section 2.2.1) – traitent le problème comme un unique problème d'optimisation global, le plus souvent ayant un unique objectif à valeurs dans \mathbb{R} . La différence entre ces trois méthodes est leur manière de résoudre les couplages interdisciplinaires. Les disciplines n'ont pas d'objectif propre autre que la résolution locale de leurs équations d'états.

Dans les méthodes d'optimisation multiniveaux, comme CO, CSSO ou BLISS (*Cf.* section 2.2.1), des optimiseurs locaux sont ajoutés aux disciplines. Leur rôle n'est pas le traitement d'un objectif propre à chaque discipline, mais la résolution locale d'un objectif global distribué entre les disciplines, ou la résolution d'un problème permettant d'assurer la convergence de la méthode.

Dans les méthodes précédemment citées, il n'y a qu'un objectif global, alors que chaque discipline a souvent un ou plusieurs objectifs propres. Une approche possible de modélisation d'un problème où plusieurs acteurs ont des objectifs à atteindre est la théorie des jeux (*Cf.* section 2.2.2). Lewis et Mistree (1997) ont modélisé des problèmes d'optimisation multidisciplinaire en théorie des jeux où les disciplines sont les joueurs. Plusieurs modèles de coopération ont été adaptés à l'optimisation multidisciplinaire (Chen et Lewis 1999, Chanron *et al.* 2004). Une des limitations de la théorie des jeux est que chaque joueur n'a qu'un seul objectif à optimiser.

Récemment, des méthodes basées sur des algorithmes évolutionnaires multiobjectifs telles que EM-MOGA, MORDACE ou COSMOS ont été proposées (*Cf.* section 2.2.3) pour traiter des problèmes disciplinaires multiobjectifs. Ces méthodes possèdent trois inconvénients : elles considèrent le problème \mathcal{M}' (équation 3.3) plutôt que \mathcal{M} (équation 3.1), l'union des optimisations disciplinaires ne permet pas d'atteindre la totalité des solutions efficaces du problème mais seulement un sous-ensemble de celui-ci, et certaines solutions trouvées sont des solutions faiblement dominées.

Bien qu'il ne s'agisse pas d'une méthode de résolution de problèmes d'optimisation multidisciplinaire au sens où nous l'avons défini, nous pouvons citer la méthode d'Engau et Wiecek (2007). Les auteurs proposent une méthode interactive de recherche de compromis dans les problèmes d'optimisation multiobjectif ayant un très grand nombre d'objectifs. Les problèmes sont décomposés en sous-problèmes biobjectifs afin de faciliter la visualisation et la sélection des solutions de compromis par le décideur. Le but est de pouvoir potentiellement atteindre toutes les solutions efficaces et de guider le décideur vers une unique solution de son choix. Une solution de compromis entre les différents sous-problèmes biobjectifs est obtenue en proposant au décideur un ensemble de solutions obtenues par relaxation (ϵ -dominance) de l'efficacité locale d'un sous-problème. La solution obtenue est donc une solution proche des solutions efficaces de chaque sous-problème.

3.1.3 Limites des méthodes actuelles

Les méthodes d'optimisation multidisciplinaire traitant le cas où chaque discipline a un problème d'optimisation multiobjectif à résoudre cherchent les solutions efficaces

2. Voir section 2.2 pour le détail des méthodes.

du problème d'optimisation multiobjectif regroupant l'ensemble des objectifs disciplinaires. Elles résolvent donc le problème \mathcal{M}' (équation 3.3) plutôt que le problème \mathcal{M} (équation 3.1).

Cette modélisation du problème d'optimisation multidisciplinaire en problème d'optimisation multiobjectif est pertinente si le découpage en disciplines n'est qu'un découpage en sous-problèmes plus simple, afin de rendre plus aisée la résolution du problème global (comme la méthode d'Engau et Wiecek). Cependant, cette méthode n'est pas pertinente si les disciplines veulent garder une autonomie de décision.

Pour illustrer notre propos, prenons un exemple didactique que nous suivrons tout au long du chapitre.

Exemple 3.1 Soit un problème d'optimisation multidisciplinaire multiobjectif à deux disciplines. Chaque discipline possède deux objectifs à minimiser : $f_{1,1}$ et $f_{1,2}$ pour la première discipline, $f_{2,1}$ and $f_{2,2}$ pour la seconde. Soit $f_1(x) = (f_{1,1}(x), f_{1,2}(x))$ et $f_2(x) = (f_{2,1}(x), f_{2,2}(x))$ respectivement les fonctions objectifs de la première et seconde discipline, et $X = \{x_a, x_b, x_c\}$ l'ensemble admissible. Les performances de ces solutions sont reportées dans le tableau 3.1 et illustrées figure 3.3.

	Discipline 1		Discipline 2	
x	$f_{1,1}(x)$	$f_{1,2}(x)$	$f_{2,1}(x)$	$f_{2,2}(x)$
x_a	1	3	2	1
x_b	3	2	1	3
x_c	2	1	3	2

TABLEAU 3.1 – Performances des solutions admissibles par disciplines de l'exemple 3.1.

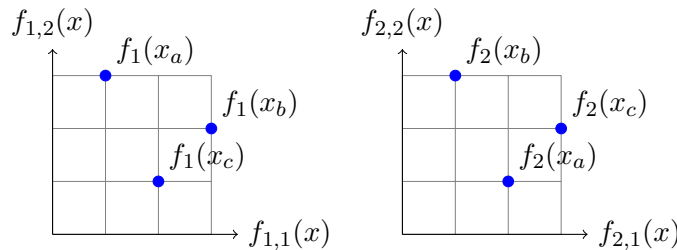


FIGURE 3.3 – Les espaces des objectifs des deux disciplines de l'exemple 3.1 : les solutions réalisables du problème sont représentées dans l'espace des objectifs de la première discipline à gauche, et de la seconde discipline à droite.

Au niveau disciplinaire, nous avons deux solutions efficaces par discipline (figure 3.3) : x_a et x_c dans la première discipline et x_a et x_b dans la seconde. Puisque x_a est l'unique solution efficace simultanément dans les deux disciplines, c'est le compromis naturel du problème multidisciplinaire. Cependant, si on considère le problème multiobjectif (formulation \mathcal{M}'), toutes les solutions sont efficaces, car leur image dans l'espace des objectifs ne se dominent pas mutuellement. En effet, résoudre le problème \mathcal{M}' peut être interprété comme trouver les solutions de compromis entre les objectifs, sans tenir compte des préférences de chaque discipline. Par exemple, la solution x_a sera respectivement préférée à x_b dans la première discipline et x_c dans la seconde, mais cette information n'apparaît pas dans la formulation multiobjectif.

Arrow (1950) a proposé quatre critères pour les fonctions de choix social que nous adaptons au problème de compromis en optimisation multidisciplinaire multiobjectif :

1. L'universalité : les préférences de chacune des disciplines doivent être prises en compte dans le compromis.
2. La non-dictature : Il n'existe aucune discipline dont les préférences coïncident avec le compromis indépendamment des préférences des autres disciplines.
3. L'unanimité : si toutes les disciplines ont les mêmes préférences, alors le compromis doit refléter exactement ces préférences.
4. L'indifférence aux options non pertinentes : la préférence relative entre deux solutions dans le compromis ne doit pas dépendre de la présence ou non d'une troisième solution.

Parmi ces critères, le troisième est ambigu dans le cas où chaque discipline possède plusieurs solutions préférées (les solutions efficaces de leur problème d'optimisation multiobjectif). Nous proposons de considérer qu'un choix entre les disciplines est unanime s'il est préféré simultanément dans toutes les disciplines. Formellement :

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow X^C = \bigcap_{i=1}^q X_i^E \quad (3.4)$$

avec X^C l'ensemble des solutions de compromis du problème. Ainsi, dans l'exemple 3.1, x_a est unanimement préféré, tandis que x_b et x_c ne le sont pas. Dans ce cas, le compromis doit être x_a .

3.2 \mathcal{C}_1 : Le produit d'ordre

Nous proposons ici une première définition formelle du compromis en optimisation multidisciplinaire, que nous noterons \mathcal{C}_1 . Il s'inspire de la structure de l'ensemble de comparaison du compromis entre les objectifs pour définir le compromis entre les disciplines.

3.2.1 Principe

Dans un problème d'optimisation multiobjectif, le front de Pareto est interprété comme l'ensemble des compromis entre les objectifs. C'est également l'ensemble des minimums de l'ensemble réalisable³ ordonné suivant l'ordre produit des ensembles réalisables des objectifs (figure 3.4). Une première approche du compromis entre les disciplines est de définir les minimums du problème multidisciplinaire comme le produit des ordres des ensembles de comparaison des disciplines. Les disciplines étant elles-même ordonnées suivant l'ordre produit des ensembles de comparaison de leurs objectifs (figure 3.5).

3.2.2 Formalisation

Chaque discipline i est un problème d'optimisation multiobjectif ayant p_i objectifs. Les espaces de comparaison de chaque discipline sont donc ordonnés ainsi :

$$(Z_i, \leq_{Z_i}) = (\mathbb{R}^{p_i}, \leq_{\mathbb{R}^{p_i}})[Y_i] \quad (3.5)$$

Le produit des ordres de chaque discipline est alors :

3. L'image de l'ensemble admissible par la fonction objectif.

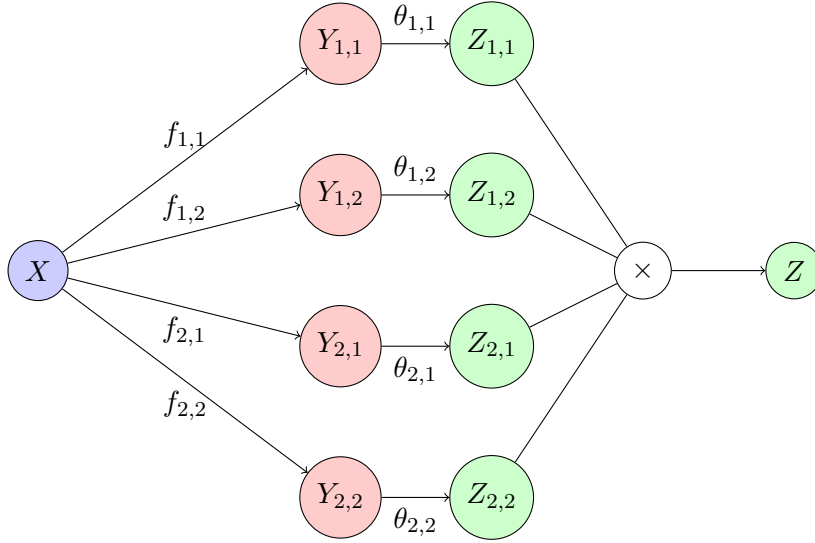


FIGURE 3.4 – Illustration de la construction des ensembles de comparaison de l'exemple 3.1 traité comme un problème d'optimisation multiobjectif (eq. 3.3) : l'ensemble de comparaison du problème est obtenu par produit des ensembles de comparaison de chacune des fonctions coûts prises indépendamment. Ici, les fonctions de préférence $\theta_{i,j}$ sont des fonctions identité : aucune préférence n'est ajoutée.

$$(Z, \leq_Z) = \left(\prod_{i=1}^q ((\mathbb{R}^{p_i}, \leq_{\mathbb{R}^{p_i}})[Y_i]) \right)[Y] \quad (3.6)$$

$$= \left(\prod_{i=1}^q (\mathbb{R}^{p_i}, \leq_{\mathbb{R}^{p_i}}) \right)[Y] \quad (3.7)$$

Or

$$\prod_{i=1}^q (\mathbb{R}^{p_i}, \leq_{\mathbb{R}^{p_i}}) = (\mathbb{R}^p, \leq_{\mathbb{R}^p}) \quad (3.8)$$

avec $p = \sum_{i=1}^q p_i$. Nous avons donc finalement :

$$(Z, \leq_Z^{\mathcal{C}_1}) = (\mathbb{R}^p, \leq_{\mathbb{R}^p})[Y] \quad (3.9)$$

Le résultat du produit des q disciplines est identique au produit des p objectifs du problème. Autrement dit, les ensembles de comparaison de \mathcal{M} et de \mathcal{M}' sont identiques. L'ensemble des solutions de compromis est donc l'ensemble des solutions efficaces de \mathcal{M}' . Ce type de compromis n'est cependant pas satisfaisant pour un problème d'optimisation multidisciplinaire, car il ne tient pas compte des préférences des disciplines (Cf. section 3.1.3).

Le compromis \mathcal{C}_1 est formulé comme suit dans l'espace de comparaison :

$$Z^{\mathcal{C}_1} := \min((\mathbb{R}^p, \leq_{\mathbb{R}^p})[Y]) \quad (3.10)$$

et on a donc

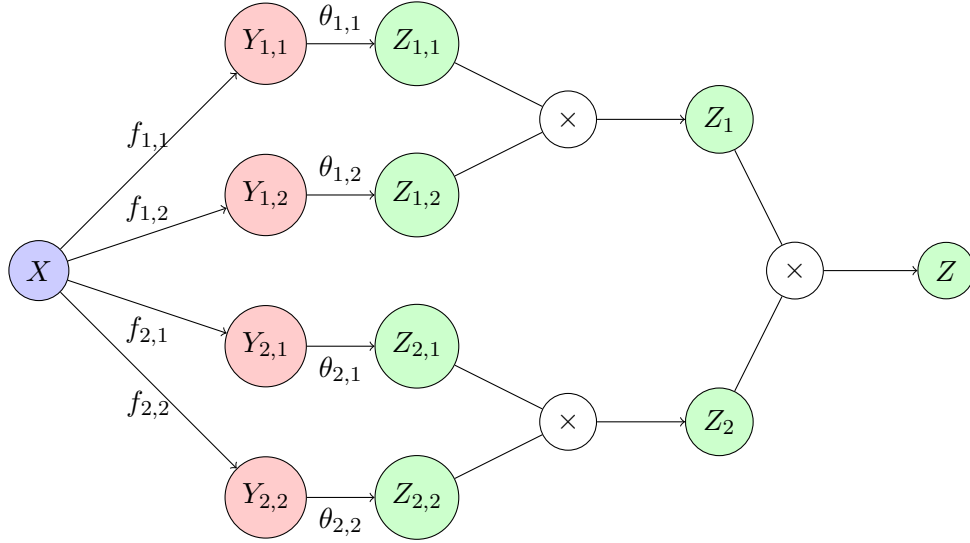


FIGURE 3.5 – Le compromis entre objectifs étant le produit des ensembles de comparaison de chaque objectif (figure 3.4), le compromis \mathcal{C}_1 entre disciplines et le produit des ensembles de comparaison disciplinaires (Z_1 et Z_2), eux-mêmes obtenus par produit des ensembles de comparaison de leurs objectifs (resp. $Z_{1,1}$, $Z_{1,2}$ et $Z_{2,1}$ et $Z_{2,2}$). Les fonctions de préférence $\theta_{i,j}$ sont des fonctions identité.

$$X^{\mathcal{C}_1} = X^E \quad (3.11)$$

Exemple 3.2 Dans l'exemple 3.1, chacun des quatre objectifs $f_{i,j}$ a le même ensemble réalisable $Y_{i,j} = \{1, 2, 3\}$ et le même ensemble de comparaison $Z_{i,j} = (\mathbb{N}, \leq_{\mathbb{N}})[\llbracket 3 \rrbracket]$ avec $i, j \in \llbracket 2 \rrbracket$. L'ensemble réalisable est $Y = \{(1, 3, 2, 1), (3, 2, 1, 3), (2, 1, 3, 2)\}$ et l'ensemble de comparaison du problème multiobjectif est donc $(Z, \leq_Z) = (\mathbb{N}^4, \leq_{\mathbb{N}^4})[Y]$.

Si nous considérons maintenant chacune des deux disciplines comme un problème d'optimisation biobjectif, les espaces réalisables des deux disciplines sont respectivement $Y_1 = \{(1, 3), (3, 2), (2, 1)\}$ et $Y_2 = \{(2, 1), (1, 3), (3, 2)\}$. Les ensembles ordonnés correspondants sont $(Z_1, \leq_{Z_1}) = (\mathbb{N}^2, \leq_{\mathbb{N}^2})[Y_1]$ et $(Z_2, \leq_{Z_2}) = (\mathbb{N}^2, \leq_{\mathbb{N}^2})[Y_2]$. Le produit de ces deux ensembles est aussi $(Z, \leq_Z) = (\mathbb{N}^4, \leq_{\mathbb{N}^4})[Y]$. Les solutions de compromis \mathcal{C}_1 sont donc $\{x_a, x_b, x_c\}$.

3.3 \mathcal{C}_2 : une extension du produit

Le second compromis, noté \mathcal{C}_2 , est une extension de \mathcal{C}_1 : des relations modélisant les préférences disciplinaires sont intégrées au niveau des disciplines.

3.3.1 Principe

Le compromis précédent n'est pas satisfaisant, car des informations pertinentes sur l'efficacité des solutions au sein d'une discipline donnée ne sont pas considérées lors du produit de leurs ordres. On peut considérer que les solutions efficaces d'une discipline sont *meilleures* que les solutions non efficaces. Cependant, l'image d'une solution efficace dans l'espace des objectifs n'est pas toujours comparable à l'image d'une solution qui n'est pas efficace. Par exemple, dans la figure 3.6, a est non dominé et h est dominé, alors

que a et h sont incomparables. Dès lors, leurs antécédents dans l'espace des solutions sont potentiellement tous deux dans l'ensemble efficace de \mathcal{M}' si a et h ne sont pas comparables non plus dans d'autres disciplines.

Exemple 3.3 Dans l'exemple 3.1, on peut observer que dans la première discipline, x_a est efficace alors que x_b ne l'est pas. Bien que $f_1(x_a)$ et $f_1(x_b)$ soient incomparables, $f_1(x_a)$ sera jugé meilleur que $f_1(x_b)$. Mais comme ils sont tous deux incomparables, ils appartiennent tous les deux aux solutions efficaces du problème à quatre objectifs (formulation 3.3).

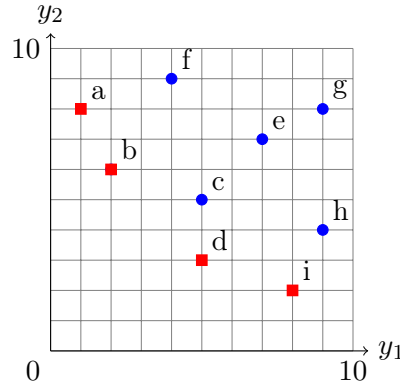


FIGURE 3.6 – Dans une discipline, tous les points non dominés (carrés) peuvent être considérés meilleurs que les points dominés (ronds).

Remarque 3.2 Dans chaque discipline (figure 3.6) :

1. toutes les solutions efficaces se valent a priori,
2. une solution efficace est préférée à une solution qui n'est pas efficace.

La remarque 3.2 exprime des préférences entre les solutions au sein des disciplines. Le compromis \mathcal{C}_2 consiste à ajouter ces préférences à l'espace de comparaison de chaque discipline avant d'en effectuer le produit.

3.3.2 Formalisation

Afin d'ajouter les préférences disciplinaires (remarque 3.2) au compromis \mathcal{C}_1 , nous étendons la relation d'ordre de l'espace de comparaison de chaque discipline.

Soient $Y_i^D := Y_i \setminus Y_i^N$ l'ensemble des points dominés⁴ de la discipline i . On peut étendre, l'ordre de chaque discipline $(Z_i, \leq_{Z_i}) \sqsubseteq (\mathbb{R}^{p_i}, \leq_{\mathbb{R}^{p_i}})$ pour obtenir le préordre (Z_i, \preceq_{Z_i}) , avec $Z_i = Y_i$, $\theta_i = id$ et les propriétés suivantes :

1. $\forall y_1, y_2 \in Y_i^D, y_1 \leq_{Z_i} y_2 \Rightarrow y_1 \preceq_{Z_i} y_2$
2. $\forall y_1, y_2 \in Y_i^N, y_1 \preceq_{Z_i} y_2$
3. $\forall (y_1, y_2) \in Y_i^N \times Y_i^D, y_1 \preceq_{Z_i} y_2$

Le premier point signifie simplement que \preceq_{Z_i} est une extension de \leq_{Z_i} (il conserve toutes ses relations). Le second point est une traduction formelle de « toutes les solutions

4. L'ensemble des points non dominés de l'ensemble réalisable étant noté Y^N (équation 1.15).

efficaces d'une discipline se valent *a priori* ». Le troisième point traduit « une solution efficace est préférée à une solution qui n'est pas efficace ».

Remarque 3.3 Une conséquence du second point est que tous les points du front Pareto d'une discipline appartiennent à la même classe d'équivalence, donc $\forall y_1, y_2 \in Y_i^N, \theta_i(y_1) \equiv_{Z_i} \theta_i(y_2)$

Remarque 3.4 Le troisième point peut être remplacé par la fermeture transitive.

Remarque 3.5 Pour chaque discipline i , (Z_i, \preceq_{Z_i}) n'est pas un ordre mais un pré-ordre, car tous les éléments de Y_i^N sont fortement connectés (définition A.13), mais tous différents.

Une fois l'extension de l'ordre de l'ensemble de comparaison de chaque discipline réalisée, on en fait le produit pour obtenir l'ensemble de comparaison du problème d'optimisation multidisciplinaire. Le compromis \mathcal{C}_2 est donc, par rapport à \mathcal{C}_1 , un ajout de préférences au sein de chaque discipline.

Exemple 3.4 Dans l'exemple 3.1, x_a et x_c sont efficaces dans la première discipline et x_a et x_b sont efficaces dans la seconde. On a donc

$$\begin{aligned} Y_1^N &= \{f_1(x_a), f_1(x_c)\} \quad \wedge \quad Y_1^D = \{f_1(x_b)\}, \\ Y_2^N &= \{f_2(x_a), f_2(x_b)\} \quad \wedge \quad Y_2^D = \{f_2(x_c)\}. \end{aligned}$$

En appliquant les trois règles, nous avons :

1. $f_1(x_b) \preceq_{Z_1} f_1(x_c); f_2(x_c) \preceq_{Z_2} f_2(x_b);$
2. $f_1(x_a) \preceq_{Z_1} f_1(x_b), f_1(x_c) \preceq_{Z_1} f_1(x_b), f_1(x_a) \preceq_{Z_1} f_1(x_c), f_1(x_c) \preceq_{Z_1} f_1(x_a);$
 $f_2(x_a) \preceq_{Z_2} f_2(x_b), f_2(x_b) \preceq_{Z_2} f_2(x_c), f_2(x_a) \preceq_{Z_2} f_2(x_c), f_2(x_b) \preceq_{Z_2} f_2(x_a);$
3. $f_1(x_a) \preceq_{Z_1} f_1(x_b), f_1(x_c) \preceq_{Z_1} f_1(x_b); f_2(x_a) \preceq_{Z_2} f_2(x_c), f_2(x_b) \preceq_{Z_2} f_2(x_c).$

Ce qui nous donne (en considérant la fermeture réflexive⁵) :

$$\begin{aligned} \preceq_{Z_1} &= \{(f_1(x_a), f_1(x_c)), (f_1(x_c), f_1(x_a)), (f_1(x_a), f_1(x_b)), (f_1(x_c), f_1(x_b))\}^1 \\ \preceq_{Z_2} &= \{(f_2(x_a), f_2(x_b)), (f_2(x_b), f_2(x_a)), (f_2(x_a), f_2(x_c)), (f_2(x_b), f_2(x_c))\}^1 \end{aligned}$$

Le produit de (Z_1, \preceq_{Z_1}) par (Z_2, \preceq_{Z_2}) nous donne :

$$\preceq_Z = \{(f(x_a), f(x_b)), (f(x_a), f(x_c))\}^1$$

Le minimum de (Z, \preceq_Z) pour \mathcal{C}_2 est donc $f(x_a)$. L'unique solution de compromis \mathcal{C}_2 est donc x_a .

Remarque 3.6 L'extension présentée peut aussi être interprétée comme une somme ordinale (définition 1.8) entre les points non dominés et dominés dans chaque discipline, puis le produit des préordres ainsi obtenus.

Dans chaque discipline i , l'ensemble réalisable forme un ordre (Y_i, \leq_i) en désignant par \leq_i la relation d'ordre produit classique. Tous les éléments de Y_i^N sont connectés entre eux, ce qui donne le préordre $(Y_i^N, (Y_i^N)^2)$; tous les éléments de Y_i^D gardent leur

5. Se reporter à la définition A.5 : la fermeture réflexive est noté par un 1 en exposant.

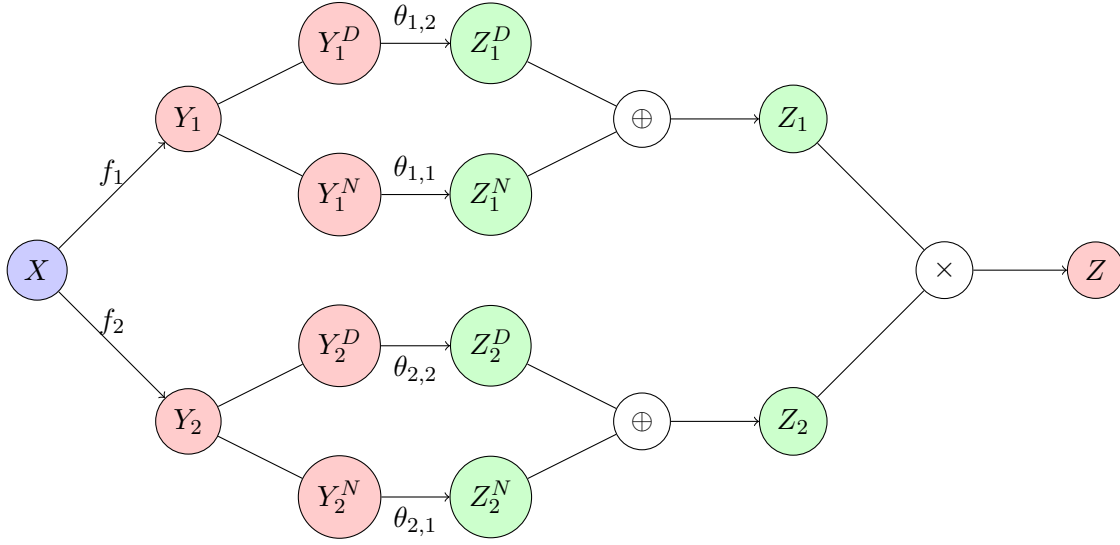


FIGURE 3.7 – Compromis \mathcal{C}_2 : dans les deux disciplines, les points non dominés (Y_i^N) sont séparés des points dominés (Y_i^D). L'ensemble de comparaison Z_i de chaque discipline est alors obtenu par la somme ordinaire de Y_i^N avec Y_i^D . L'ensemble de comparaison du compromis est le produit des Z_i .

ordre initial : $(Y_i, \leq_{Y_i})[Y_i^D]$. L'ensemble de comparaison (Z_i, \preceq_{Z_i}) de chaque discipline est obtenu en sommant $(Y_i^N, (Y_i^N)^2)$ et $(Y_i, \leq_{Y_i})[Y_i^D]$:

$$(Z_i, \preceq_{Z_i}) = (Y_i^N, (Y_i^N)^2) \oplus (Y_i, \leq_{Y_i})[Y_i^D], \forall i \in \llbracket q \rrbracket \quad (3.12)$$

Le compromis \mathcal{C}_2 entre q disciplines est alors le minimum du produit entre ces q préordres :

$$Z^{\mathcal{C}_2} := \min \left(\prod_{i=1}^q ((Y_i^N, (Y_i^N)^2) \oplus (Y_i, \leq_{Y_i})[Y_i^D]) \right) [Y] \quad (3.13)$$

La figure 3.7 illustre cette interprétation avec deux disciplines ayant chacune deux objectifs, avec

$$\theta_{i,1} : Y_i^N \mapsto (Y_i^N, (Y_i^N)^2) \quad (3.14)$$

$$\theta_{i,2} : Y_i^D \mapsto (Y_i, \leq_{Y_i})[Y_i^D] \quad (3.15)$$

Les fonctions de préférence $\theta_{i,1}$ connectent tous les éléments non dominés, et les fonctions $\theta_{i,2}$ préservent l'ordre des éléments dominés.

Exemple 3.5 Dans l'exemple 3.1, chaque discipline peut être ordonnée suivant l'ordre produit de ses objectifs, ce qui nous donne les deux ordres suivants :

$$(Y_1, \leq_{Y_1}) = (\{f_1(x_a), f_1(x_b), f_1(x_c)\}, \{(f_1(x_c), f_1(x_b))\}^1)$$

$$(Y_2, \leq_{Y_2}) = (\{f_2(x_a), f_2(x_b), f_2(x_c)\}, \{(f_2(x_a), f_2(x_c))\}^1)$$

Pour chaque discipline, les ordres (Y_1, \leq_{Y_1}) et (Y_2, \leq_{Y_2}) peuvent être décomposés en leurs points non dominés :

$$\begin{aligned} (Y_1, \leq_{Y_1})[Y_1^N] &= (\{f_1(x_a), f_1(x_c)\}, \{(f_1(x_a), f_1(x_a)), (f_1(x_c), f_1(x_c))\}) \\ (Y_2, \leq_{Y_2})[Y_2^N] &= (\{f_2(x_a), f_2(x_b)\}, \{(f_1(x_a), f_1(x_a)), (f_1(x_b), f_1(x_b))\}) \end{aligned}$$

et leurs points dominés :

$$\begin{aligned} (Y_1, \leq_{Y_1})[Y_1^D] &= (\{f_1(x_b)\}, \{(f_1(x_b), f_1(x_b))\}) \\ (Y_2, \leq_{Y_2})[Y_2^D] &= (\{f_2(x_c)\}, \{(f_2(x_c), f_2(x_c))\}) \end{aligned}$$

Nous commençons par ajouter que tous les points non dominés sont équivalents dans chaque discipline (application respectivement de $\theta_{1,1}$ et de $\theta_{2,1}$) :

$$\begin{aligned} (Y_1, (Y_1)^2)[Y_1^N] &= (\{f_1(x_a), f_1(x_c)\}, \{(f_1(x_a), f_1(x_c)), (f_1(x_c), f_1(x_a))\})^1 \\ (Y_2, (Y_2)^2)[Y_2^N] &= (\{f_2(x_a), f_2(x_b)\}, \{(f_2(x_a), f_2(x_b)), (f_2(x_b), f_2(x_a))\})^1 \end{aligned}$$

Nous ajoutons ensuite, pour chaque discipline, l'ordre obtenu précédemment à l'ensemble des points non dominés. Soit pour la première discipline :

$$\begin{aligned} (Z_1, \preceq_{Z_1}) &= (Y_1, (Y_1)^2)[Y_1^N] \oplus (Y_1, (Y_1)^2)[Y_1^D] \\ &= (Y_1, \{(f_1(x_a), f_1(x_c)), (f_1(x_c), f_1(x_a)), (f_1(x_a), f_1(x_b)), (f_1(x_c), f_1(x_b))\})^1 \end{aligned}$$

et pour la seconde :

$$\begin{aligned} (Z_2, \preceq_{Z_2}) &= (Y_2, (Y_2)^2)[Y_2^N] \oplus (Y_2, (Y_2)^2)[Y_2^D] \\ &= (Y_2, \{(f_2(x_a), f_2(x_b)), (f_2(x_b), f_2(x_a)), (f_2(x_a), f_2(x_c)), (f_2(x_b), f_2(x_c))\})^1 \end{aligned}$$

Enfin, le produit est calculé entre les deux préordres disciplinaires :

$$\begin{aligned} (Z, \preceq) &= ((Z_1, \preceq_{Z_1}) \times (Z_2, \preceq_{Z_2}))[Y] \\ &= (Y, \{(f(x_a), f(x_b)), (f(x_a), f(x_c))\})^1[Y] \end{aligned}$$

L'ensemble minimum du préordre obtenu est alors $\{f(x_a)\}$. La solution x_a est donc l'unique solution de compromis \mathcal{C}_2 .

Une autre interprétation du compromis \mathcal{C}_2 donne des résultats équivalents, mais avec une fonction de préférence θ_i explicite à valeurs dans \mathbb{N}^{p_i} . La relation d'ordre utilisée pour l'espace de comparaison de chaque discipline est alors l'ordre produit dans \mathbb{N}^{p_i} .

Il faut établir l'ordre dans \mathcal{Z} par la fonction θ_i que l'on peut définir ainsi :

$$\begin{aligned} \theta_i : \mathcal{Y}_i &\rightarrow \mathcal{Z}_i \\ y &\mapsto \begin{cases} 0_{p_i} & \text{si } y \in Y_i^N \\ (r_i(y_1), \dots, r_i(y_{p_i})) & \text{sinon} \end{cases} \end{aligned} \quad (3.16)$$

où 0_{p_i} est le vecteur nul de \mathbb{N}^{p_i} et $r_i(y)$ le rang de y dans la discipline i . Contrairement aux interprétations précédentes du même compromis, nous n'avons plus $|Y_i| = |Z_i|$,

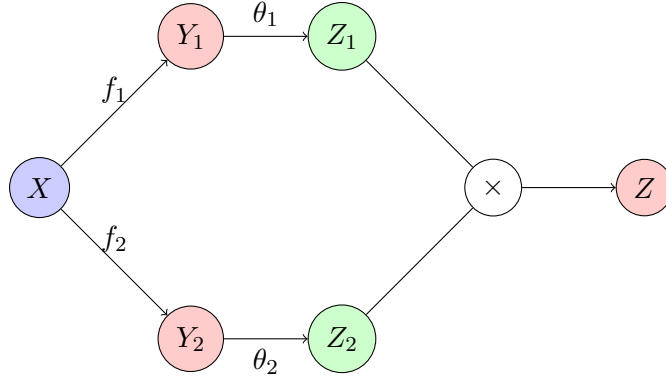


FIGURE 3.8 – Autre interprétation du compromis \mathcal{C}_2 : plutôt que d'effectuer la somme ordinale entre les points non dominés et les points dominés au sein de chaque discipline (figure 3.7), nous utilisons une fonction θ_i qui attribue $0_{\mathbb{N}}$ aux éléments non dominés et une valeur de $\mathbb{N}^p \geq 1_{\mathbb{N}}$ aux autres.

mais $|Y_i| \geq |Z_i|$. Les images des points non dominés de Y_i dans chaque discipline sont confondues en un unique point de Z_i . Les images des éléments de même rang dans chaque discipline sont également confondues. Pour chaque discipline i , les (Z_i, \leq_{Z_i}) sont ici des ordres :

$$(Z_i, \leq_{Z_i}) := (\mathbb{N}^{p_i}, \leq_{\mathbb{N}^{p_i}})[\theta_i(Z_i)], \forall i \in \llbracket q \rrbracket \quad (3.17)$$

On a alors

$$Z^{\mathcal{C}_2} := \min\left(\prod_{i=1}^q (\mathbb{N}^{p_i}, \leq_{\mathbb{N}^{p_i}})[\theta(Z_i)]\right) \quad (3.18)$$

Exemple 3.6 Dans l'exemple 3.1, x_a et x_c sont efficaces dans la première discipline et x_a et x_b sont efficaces dans la seconde. On a donc

$$\begin{aligned} Y_1^N &= \{f_1(x_a), f_1(x_c)\} \quad \wedge \quad Y_1^D = \{f_1(x_b)\}, \\ Y_2^N &= \{f_2(x_a), f_2(x_b)\} \quad \wedge \quad Y_2^D = \{f_2(x_c)\}. \end{aligned}$$

Dans la première discipline, les rangs sur chaque objectif de x_b sont respectivement 3 et 2 pour $f_{1,1}$ et $f_{1,2}$. De même, dans la seconde discipline, les rangs de x_c sont respectivement 3 et 2 pour $f_{2,1}$ et $f_{2,2}$. Les autres solutions sont efficaces, donc on leur attribue $0_{\mathbb{N}^2}$. Ce qui nous donne :

$$\begin{aligned} \theta_1 \circ f_1(x_a) &= 0_{\mathbb{N}^2} \\ \theta_1 \circ f_1(x_b) &= (3, 2) \\ \theta_1 \circ f_1(x_c) &= 0_{\mathbb{N}^2} \\ \theta_2 \circ f_1(x_a) &= 0_{\mathbb{N}^2} \\ \theta_2 \circ f_1(x_b) &= 0_{\mathbb{N}^2} \\ \theta_2 \circ f_1(x_c) &= (3, 2) \end{aligned}$$

Nous obtenons donc

$$(Z_1, \leq_{Z_1}) = (Z_2, \leq_{Z_2}) = (\{0_{\mathbb{N}^2}, (3, 2)\}, \{(0_{\mathbb{N}^2}, (3, 2))\})^1$$

et donc

$$\begin{aligned} (Z, \leq_Z) &= ((Z_1, \leq_{Z_1}) \times (Z_2, \leq_{Z_2}))[Z] \\ &= (\{0_{\mathbb{N}^4}, (0, 0, 3, 2), (3, 2, 0, 0)\}, \{(0_{\mathbb{N}^4}, (0, 0, 3, 2)), (0_{\mathbb{N}^4}, (3, 2, 0, 0))\})^1 \end{aligned}$$

Le plus petit élément de Z est évidemment $0_{\mathbb{N}^4}$. L'unique antécédent de $0_{\mathbb{N}^4}$ par ϕ est x_a . On trouve donc x_a comme unique solution de compromis de \mathcal{C}_2 .

3.3.3 Propriétés

Les solutions de compromis \mathcal{C}_2 diffèrent des solutions de compromis \mathcal{C}_1 , mais toutes les solutions de \mathcal{C}_2 sont également des solutions de \mathcal{C}_1 .

Proposition 3.1 *Toutes les solutions de compromis \mathcal{C}_2 sont des solutions efficaces du problème d'optimisation multiobjectif $\mathcal{M}' : X^{\mathcal{C}_2} \subseteq X^E$*

Démonstration. Nous voulons montrer qu'une solution x de compromis appartenant à $X^{\mathcal{C}_2}$ appartient également à $X^{\mathcal{C}_1} = X^E$. $x \in X^{\mathcal{C}_2}$ signifie que son image par la fonction coût appartient à $Z^{\mathcal{C}_2}$:

$$x \in X^{\mathcal{C}_2} \Leftrightarrow \phi(x) \in Z^{\mathcal{C}_2}$$

Or $Z^{\mathcal{C}_2}$ est l'ensemble des minimums de (Z, \preceq_Z) . On a donc

$$x \in X^{\mathcal{C}_2} \Leftrightarrow \phi(x) \in \min(Z, \preceq_Z)$$

Ce qui, d'après la définition des minimums d'un préordre (définition 1.14), nous donne

$$x \in X^{\mathcal{C}_2} \Leftrightarrow \forall x' \in X, \phi(x') \preceq_Z \phi(x) \Rightarrow \phi(x') \equiv \phi(x)$$

Or la fonction coût ϕ est identique pour \mathcal{C}_1 et \mathcal{C}_2 : $\phi = f$. Soit

$$x \in X^{\mathcal{C}_2} \Leftrightarrow \forall x' \in X, f(x') \preceq_Z f(x) \Rightarrow f(x') \equiv f(x)$$

Dans le cas du compromis \mathcal{C}_2 , deux éléments de l'ensemble de comparaison sont équivalents s'ils sont égaux ou s'ils appartiennent au front de Pareto de toutes les disciplines :

$$f(x') \equiv f(x) \Leftrightarrow (x' = x) \vee (\forall i \in \llbracket q \rrbracket : f_i(x'), f_i(x) \in Y_i^N)$$

Dans le premier cas ($x' = x$), on retrouve la définition d'une solution efficace, donc $x \in X^{\mathcal{C}_1}$. Dans le second cas ($\forall i \in \llbracket q \rrbracket : f_i(x'), f_i(x) \in Y_i^N$), si une solution est localement efficace dans chaque discipline, alors elle est globalement efficace. Finalement, on a bien

$$x \in X^{\mathcal{C}_2} \Rightarrow x \in X^{\mathcal{C}_1}$$

□

Le compromis \mathcal{C}_2 possède également d'intéressantes propriétés que n'a pas le compromis \mathcal{C}_1 . La proposition 3.2 montre que, dans le cas où il existe des solutions efficaces communes à toutes les disciplines, alors elles sont les uniques solutions de compromis (comme x_a dans l'exemple 3.1), ce qui satisfait notre interprétation du troisième critère d'Arrow (section 3.1.3, équation 3.4).

Proposition 3.2 *S'il existe des solutions efficaces communes à toutes les disciplines simultanément, alors l'ensemble des solutions de compromis est exactement l'intersection des solutions efficaces des disciplines :*

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow X^{C_2} = \bigcap_{i=1}^q X_i^E$$

Démonstration. Pour la démonstration, nous utilisons la dernière interprétation de C_2 . Dans chaque discipline, les solutions efficaces sont évaluées à $0_{\mathbb{N}^{p_i}}$:

$$\forall i \in \llbracket q \rrbracket, \forall x \in X_i^E, \phi_i(x) = 0_{\mathbb{N}^{p_i}} \quad (3.19)$$

Donc s'il existe des solutions efficaces à toutes les disciplines simultanément, leur évaluation par la fonction coût sera $0_{\mathbb{N}^q}$:

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow \forall x \in \bigcap_{i=1}^q X_i^E, \phi(x) = 0_{\mathbb{N}^q} \quad (3.20)$$

Donc si l'intersection des solutions efficaces des disciplines est non vide, il existe au moins une solution évaluée à $0_{\mathbb{N}^q}$ par la fonction coût :

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow \exists x \in X : \phi(x) = 0_{\mathbb{N}^q} \quad (3.21)$$

Or s'il existe une solution évaluée à $0_{\mathbb{N}^q}$, alors $0_{\mathbb{N}^q}$ est le plus petit élément de l'ensemble de comparaison Z :

$$\exists x \in X : \phi(x) = 0_{\mathbb{N}^q} \Rightarrow \{0_{\mathbb{N}^q}\} = \min(\mathbb{N}^q, \leq_{\mathbb{N}^q})[Z] \quad (3.22)$$

Or si $0_{\mathbb{N}^q}$ est le plus petit élément de l'ensemble de comparaison Z , par définition :

$$\{0_{\mathbb{N}^q}\} = Z^{C_2} \quad (3.23)$$

$$\Leftrightarrow \{x \in X : \phi(x) = 0\} = X^{C_2} \quad (3.24)$$

D'après les équations 3.21, 3.22 et 3.24, nous avons par transitivité :

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow \{x \in X : \phi(x) = 0\} = X^{C_2} \quad (3.25)$$

et d'après les équations 3.20 et 3.25 :

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow X^{C_2} = \bigcap_{i=1}^q X_i^E \quad (3.26)$$

□

La proposition 3.3 montre que, dans le cas à deux disciplines, les solutions de compromis se trouvent entre les projections des solutions efficaces de chaque discipline dans les différents espaces des objectifs (voir figure 3.9). Il s'agit d'une représentation intuitive des solutions de compromis, qui peut être exploitée pour éliminer des solutions non pertinentes dans une méthode de résolution.

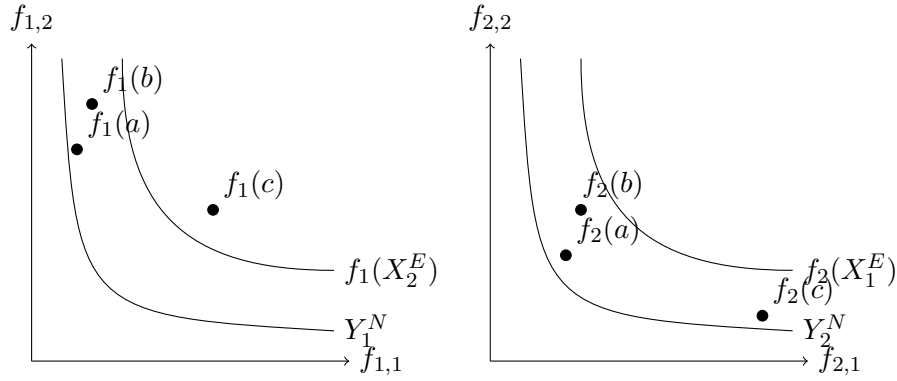


FIGURE 3.9 – L'élément c ne peut pas être une solution de compromis pour \mathcal{C}_2 car son image par f_1 est « au dessus » de l'image par f_1 de l'ensemble efficace de la seconde discipline. Bien que l'image de b soit « au dessous » de l'image de l'ensemble efficace de chaque discipline dans sa propre discipline, ce n'est pas une solution de compromis, parce qu'elle est dominée par a .

Proposition 3.3 Soient deux problèmes d'optimisation multiobjectif i et j définis sur le même ensemble admissible X , et $\forall x \in X$, si $\exists x' \in X_j^E$ telle que $f_i(x') \leq f_i(x)$, alors $x \notin X^{\mathcal{C}_2}$.

Démonstration. Pour deux disciplines i et j , si l'image, dans l'ensemble de comparaison, d'une solution x' efficace dans j domine une solution x dans i

$$\exists x' \in X_j^E : f_i(x') \leq f_i(x) \Rightarrow \phi_i(x') \leq \phi_i(x) \wedge \phi_j(x') \leq \phi_j(x) \quad (3.27)$$

alors x' est inférieur à x dans les deux disciplines, dont au moins une strictement. Par conséquent, en effectuant le produit des ensembles de comparaison, on obtient

$$\phi_i(x') \leq \phi_i(x) \wedge \phi_j(x') \leq \phi_j(x) \Rightarrow \phi(x') \leq \phi(x) \quad (3.28)$$

x ne peut donc pas être solution du compromis \mathcal{C}_2 . \square

La proposition 3.3 est illustrée figure 3.9 où c n'est pas une solution de compromis, car elle est dominée par un point de $f_1(X_2^E)$. Malheureusement, l'opposée n'est pas vraie : une solution dont toutes les images sont « au dessous » de $f_i(X_j^E)$ n'est pas toujours une solution de compromis. Sur cette même figure, b est dominé par a dans toutes les disciplines, donc elle ne peut être une solution de compromis, bien qu'elle soit placée entre $f_1(X_1^E)$ et $f_2(X_2^E)$.

3.4 \mathcal{C}_3 : Le compromis comme le rang des rangs

Le compromis \mathcal{C}_3 s'inspire du compromis \mathcal{C}_2 , mais affine les préférences disciplinaires en tenant compte des rangs des solutions.

3.4.1 Principe

Le compromis \mathcal{C}_2 décompose l'ensemble réalisable de chaque discipline en deux sous-ensembles : les solutions efficaces et les solutions non efficaces. On peut réaliser des partitions plus fines des ensembles réalisables de chaque discipline en utilisant les rangs (figure 3.10).

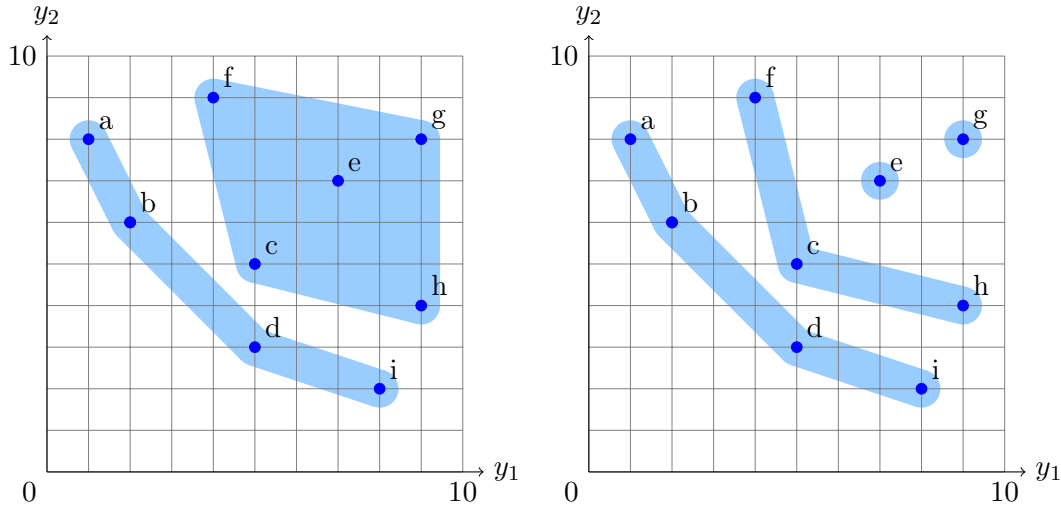


FIGURE 3.10 – Décomposition de l'espace des objectifs d'une discipline suivant C_2 à gauche et suivant C_3 à droite.

3.4.2 Formalisation

Si on note Y_i^j les éléments de rang j dans l'espace des objectifs d'une discipline i , nous obtenons :

$$Y_i^j := \{y \in Y_i : r_i(y) = j\}, \quad (3.29)$$

et r_i^{\max} le rang le plus élevé de la discipline i :

$$r_i^{\max} := \max_{y \in Y_i} r_i(y) \quad (3.30)$$

L'ensemble de comparaison de C_3 est construit de manière similaire à celui de C_2 , mais sur chacun des rangs : les solutions de chaque rang sont considérées équivalentes, puis une somme ordinale par ordre croissant du rang est réalisée. L'ensemble des solutions de compromis C_3 dans l'ensemble de comparaison est alors

$$Z^{C_3} := \min \left(\prod_{i=1}^q \bigoplus_{j=1}^{r_i^{\max}} (Y_i^j, (Y_i^j)^2) \right) [Y] \quad (3.31)$$

Exemple 3.7 Dans l'exemple 3.1, chaque discipline se décompose en deux rangs :

$$\begin{aligned} (Y_1, (\leq_{Y_1})) [Y_1^1] &= (\{f_1(x_a), f_1(x_c)\}, \emptyset) \\ (Y_2, (\leq_{Y_2})) [Y_2^1] &= (\{f_2(x_a), f_2(x_b)\}, \emptyset) \end{aligned}$$

pour les premiers rangs et

$$\begin{aligned} (Y_1, (\leq_{Y_1})) [Y_1^2] &= (\{f_1(x_b)\}, \{(f_1(x_b), f_1(x_b))\}) \\ (Y_2, (\leq_{Y_2})) [Y_2^2] &= (\{f_2(x_c)\}, \{(f_2(x_c), f_2(x_c))\}) \end{aligned}$$

pour les seconds. Le résultat est donc ici identique à celui trouvé pour le compromis C_2 (exemple 3.5), car les disciplines ne contiennent que deux rangs chacune.

Remarque 3.7 *Le compromis \mathcal{C}_3 peut également être interprété comme la transformation de chaque problème multiobjectif (\mathcal{M}_i) en un problème mono-objectif qui est la minimisation du rang r_i dans chaque discipline i :*

$$\min_{x \in X} (r_1(x), \dots, r_q(x)) \quad (3.32)$$

Le problème \mathcal{M} est transformé en un problème d'optimisation multiobjectif avec q objectifs qui sont les rangs de chaque discipline : pour chaque discipline, on a la fonction de préférence $\phi_i = r_i$.

Exemple 3.8 *On peut considérer le problème de l'exemple 3.1 comme un problème d'optimisation biobjectif consistant à minimiser les rangs dans chaque discipline. Le tableau 3.2 donne les rangs de chaque solution dans les deux disciplines.*

x	$r_1(x)$	$r_2(x)$
x_a	1	1
x_b	2	1
x_c	1	2

TABLEAU 3.2 – Rang des points dans chaque discipline : $r_1(x)$ (resp. $r_2(x)$) est le rang de x dans la discipline 1 (resp. discipline 2).

On en déduit que x_a est le minimum, car $(1, 1)$ domine $(2, 1)$ et $(1, 2)$.

3.4.3 Propriétés

Les propriétés montrées pour \mathcal{C}_2 sont également valables pour \mathcal{C}_3 . Puisque la distinction entre les points non dominés Y^N et les points dominés Y^D du compromis \mathcal{C}_2 est conservée dans le compromis \mathcal{C}_3 , les démonstrations de ces propositions sont similaires à celles de \mathcal{C}_2 .

Proposition 3.4 *Toutes les solutions de compromis \mathcal{C}_3 sont des solutions efficaces de $\mathcal{M}' : X^{\mathcal{C}_3} \subseteq X^E$*

Proposition 3.5 *S'il existe des solutions efficaces communes à toutes les disciplines simultanément, alors l'ensemble des solutions de compromis est exactement l'intersection des solutions efficaces des disciplines :*

$$\bigcap_{i=1}^q X_i^E \neq \emptyset \Rightarrow X^{\mathcal{C}_3} = \bigcap_{i=1}^q X_i^E$$

Proposition 3.6 *Soient deux problèmes d'optimisation multiobjectif i et j définis sur le même ensemble admissible X , et $\forall x \in X$, si $\exists x' \in X_j^E$ telle que $f_i(x') \preceq f_i(x)$, alors $x \notin X^{\mathcal{C}_3}$.*

3.5 Inconvénients des compromis \mathcal{C}_2 et \mathcal{C}_3

Les compromis \mathcal{C}_2 et \mathcal{C}_3 partagent cependant des inconvénients pour leur implémentation dans un algorithme évolutionnaire (remarque 3.8 et proposition 3.7).

x	$f_1(x)$	$f_2(x)$
x_a	3	5
x_b	4	1
x_c	5	5
x_d	1	3
x_e	2	4

TABLEAU 3.3 – Performances d’un sous-ensemble de solutions admissibles de X de l’exemple 3.9.

Remarque 3.8 Un ordre basé sur les rangs tels que \mathcal{C}_2 ou \mathcal{C}_3 dépend de la répartition des points dans l’ensemble réalisable.

La remarque 3.8 entre en contradiction avec le critère d’indifférence aux options non pertinentes (quatrième critère d’Arrow, section 3.1.3) : la relation d’ordre entre deux éléments dans une discipline dépend de la présence d’un troisième élément (voir l’exemple 3.9).

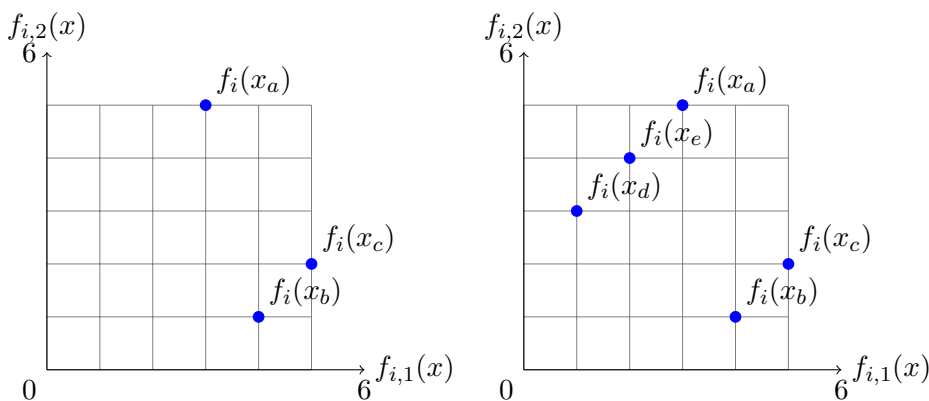


FIGURE 3.11 – Le rang de $f_i(x_a)$ ($r_i(x_a) = 1$) est inférieur au rang de $f_i(x_c)$ ($r_i(x_c) = 2$) dans le graphique de gauche, alors que cette relation est inversée dans le graphique de droite ($r_i(x_a) = 3$ et $r_i(x_c) = 2$) avec l’ajout de deux éléments.

Exemple 3.9 Soit un problème d’optimisation disciplinaire défini sur l’ensemble admissible X incluant $\{x_a, x_b, x_c, x_d, x_e\}$. Les performances de ces solutions sont indiquées dans le tableau 3.3.

Si on considère simplement les solutions x_a, x_b et x_c , le point $f_i(x_a)$ est de rang 1 et $f_i(x_c)$ est de rang 2. En revanche, si on ajoute x_d et x_e , alors $f_i(x_a)$ est de rang 3 et $f_i(x_c)$ est de rang 2. En ajoutant deux solutions à l’ensemble admissible, une inversion de la relation entre $f_i(x_a)$ et $f_i(x_c)$ s’est produite (figure 3.11).

Une propriété liée à la remarque 3.8 est que le minimum n’est pas idempotent pour \mathcal{C}_2 et \mathcal{C}_3 (proposition 3.7).

Proposition 3.7 Pour tout préordre O obtenu par les compromis \mathcal{C}_2 et \mathcal{C}_3 , la propriété suivante n’est pas toujours vérifiée :

$$\min(O[\min O]) = \min O$$

x	$f_{1,1}(x)$	$f_{1,2}(x)$	$f_{2,1}(x)$	$f_{2,2}(x)$
x_a	1	4	3	3
x_b	2	2	5	5
x_c	4	1	5	2
x_d	3	3	1	4
x_e	5	5	4	1
x_f	5	2	2	2

TABEAU 3.4 – Performances d'un sous-ensemble de solutions admissibles de X de l'exemple 3.10.

La remarque 3.8 et la proposition 3.7 indiquent que des algorithmes d'optimisation itératifs implémentant les compromis \mathcal{C}_2 et \mathcal{C}_3 ne convergeront pas toujours vers les bonnes solutions, car le minimum est susceptible de changer à chaque itération (exemple 3.10).

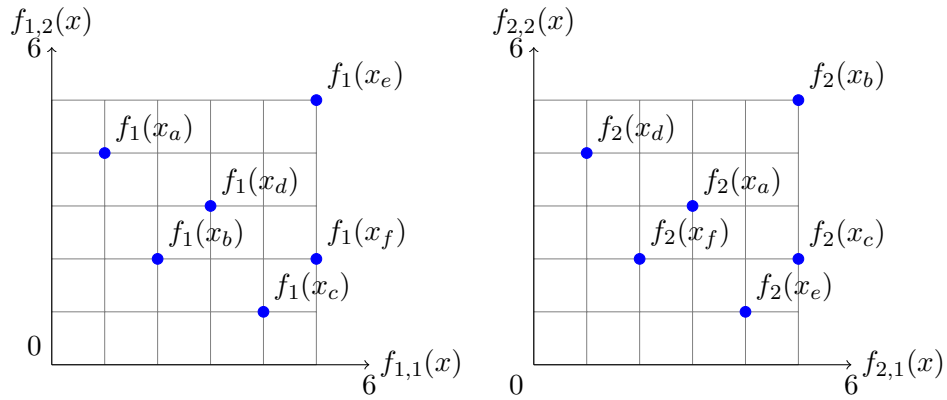


FIGURE 3.12 – Les solutions de compromis \mathcal{C}_2 et \mathcal{C}_3 pour ce problème sont $\{x_a, x_c, x_d, x_f\}$, mais le minimum pour le sous-préordre restreint à $\{x_a, x_c, x_d, x_f\}$ est $\{x_d\}$. Le minimum de $Z^{\mathcal{C}_2}$ et $Z^{\mathcal{C}_3}$ n'est alors pas idempotent.

Exemple 3.10 Soit un problème d'optimisation à deux disciplines défini sur l'ensemble admissible X incluant $\{x_a, x_b, x_c, x_d, x_e, x_f\}$. Chaque discipline est un problème d'optimisation biobjectif. Les performances de ces solutions sont présentées dans le tableau 3.4 et illustrées figure 3.12. Dans \mathcal{C}_2 et \mathcal{C}_3 , les solutions de compromis sont :

$$\min\{\phi(x_a), \phi(x_b), \phi(x_c), \phi(x_d), \phi(x_e), \phi(x_f)\} = \{\phi(x_a), \phi(x_c), \phi(x_d), \phi(x_f)\}$$

Si on applique une nouvelle fois le minimum à l'ensemble obtenu, on obtient

$$\min\{\phi(x_a), \phi(x_c), \phi(x_d), \phi(x_f)\} = \{\phi(x_d)\}$$

Le minimum n'est donc pas idempotent pour \mathcal{C}_2 et \mathcal{C}_3 .

Ces inconvénients des compromis \mathcal{C}_2 et \mathcal{C}_3 sont issus de l'ajout de préférences dans les disciplines qui rendent leurs ordres sensibles à la répartition des solutions dans l'espace des objectifs. Pour pallier ces problèmes, un autre compromis (\mathcal{C}_4), n'ajoutant pas de préférence aux disciplines, est proposé.

3.6 \mathcal{C}_4 : Un produit faible entre les disciplines

Le quatrième compromis, noté \mathcal{C}_4 , utilise une approche différente de celle des compromis \mathcal{C}_2 et \mathcal{C}_3 . Plutôt que d'ajouter des préférences dans les disciplines, on modifie la façon dont ces préférences sont agrégées.

3.6.1 Principe

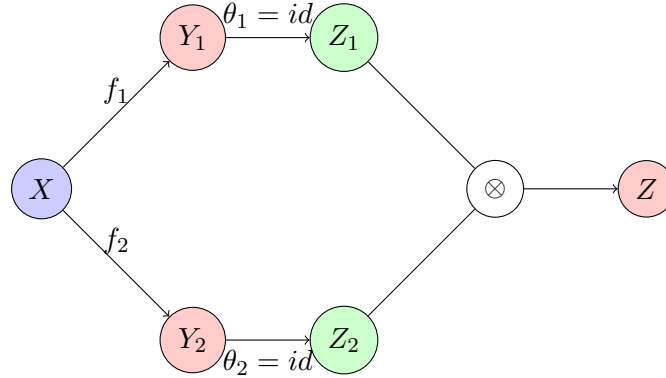


FIGURE 3.13 – Dans le compromis \mathcal{C}_4 , aucune préférence n'est ajoutée aux disciplines : les ordres disciplinaires (Z_1, \leq_{Z_1}) et (Z_2, \leq_{Z_2}) sont les ordres produits classiques. L'ensemble de comparaison Z est obtenu par le produit faible (définition 3.1) de (Z_1, \leq_{Z_1}) et de (Z_2, \leq_{Z_2}) .

La dominance de Pareto prend implicitement comme hypothèse que chaque composante est totalement ordonnée. Cependant, quand les problèmes disciplinaires sont des problèmes d'optimisation multiobjectif, les solutions sont partiellement ordonnées, du fait des ordres produits. Le compromis \mathcal{C}_3 consiste à totalement ordonner les solutions disciplinaires par leur rang. Mais, dans ce cas, la relation d'ordre est sensible à la répartition des points dans l'espace des objectifs. Le compromis \mathcal{C}_4 consiste à changer la relation de dominance pour qu'elle prenne en compte les ensembles partiellement ordonnés de chaque discipline. Plutôt que de considérer que « a domine b si et seulement si a est inférieur à b pour tous les objectifs et strictement inférieur pour au moins un objectif » (relation de Pareto), nous considérons que « a est inférieur à b si a domine b dans au moins une discipline et b ne domine a dans aucune discipline ».

3.6.2 Formalisme

Nous définissons un autre produit d'ordre que nous appelons produit faible :

Définition 3.1 (Produit faible) Soient $O_1 = (E_1, \leq_1), \dots, O_q = (E_q, \leq_q)$ q ensembles ordonnés. On définit le produit faible $O = \bigotimes_{i=1}^q O_i$ tel que $E = \prod_{i=1}^q E_i$ et

$$a \preceq b \iff \begin{cases} \exists i \in [q] & a_i \leq_i b_i \\ \nexists j \in [q] & b_j \prec_j a_j \end{cases}$$

Remarque 3.9 L'application du produit faible ne donne pas un préordre, car la relation résultante n'est pas transitive. En appliquant la fermeture transitive (définition A.6), on obtient alors un préordre – et non pas un ordre – car des relations symétriques peuvent apparaître (exemple 3.11).

x	$f_{1,1}(x)$	$f_{1,2}(x)$	$f_{2,1}(x)$	$f_{2,2}(x)$
x_a	2	4	1	3
x_b	3	1	2	4
x_c	1	3	4	2
x_d	4	2	3	1

TABLEAU 3.5 – Vecteurs de performances de l'exemple 3.11.

Exemple 3.11 Soit l'ensemble admissible $X = \{x_a, x_b, x_c, x_d\}$ et deux disciplines ayant chacune un problème d'optimisation biobjectif à résoudre. Les performances des solutions sont reportées dans le tableau 3.5 et illustrées figure 3.14. En appliquant le produit faible, nous obtenons la relation $\{(x_a, x_a), (x_a, x_b), (x_b, x_b), (x_b, x_d), (x_c, x_a), (x_c, x_c), (x_d, x_c), (x_d, x_d)\}$ qui est réflexive et antisymétrique (elle conserve les propriétés de ses composantes), mais pas transitive. Par exemple $x_a \preceq x_b$ et $x_b \preceq x_d$, mais on n'a pas $x_a \preceq x_d$. Sa fermeture transitive n'est plus antisymétrique : par exemple $x_a \preceq x_b$ et $x_b \preceq^+ x_a$ ($x_b \preceq x_d \wedge x_d \preceq x_c \wedge x_c \preceq x_a$).

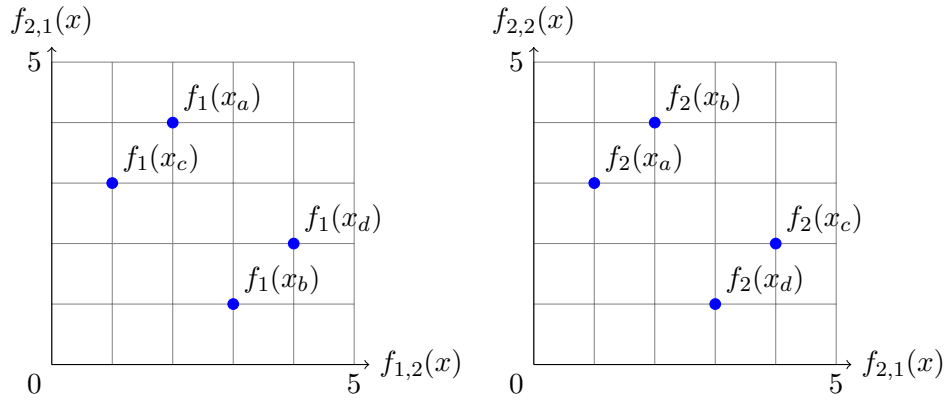


FIGURE 3.14 – La fermeture transitive du produit faible génère des préordres.

Le compromis \mathcal{C}_4 est proche du compromis \mathcal{C}_1 dans sa formulation. Seule l'opération de produit des ensembles de comparaison disciplinaire change. Le compromis \mathcal{C}_4 est formulé ainsi :

$$Z^{\mathcal{C}_4} := \min\left(\left(\bigotimes_{i=1}^q (Y_i, \leq_{\mathbb{R}^{p_i}})\right)[Y]\right)^+ \quad (3.33)$$

en désignant par $+$ la fermeture transitive (définition A.6).

Exemple 3.12 Dans l'exemple didactique 3.1, soit $f(x) = (f_1(x), f_2(x))$. Le point $f(x_a)$ domine $f(x_c)$ et le point $f(x_c)$ domine $f(x_b)$ dans une discipline chacun, et il n'y a pas d'autre relation de dominance. Nous avons donc $f(x_a) \preceq f(x_c) \preceq f(x_b)$. Donc, x_a est l'unique solution de compromis \mathcal{C}_4 de ce problème.

3.6.3 Propriétés

Comme pour les compromis précédents, toutes les solutions de \mathcal{C}_4 sont dans l'ensemble de Pareto.

Proposition 3.8 *Toutes les solutions de compromis \mathcal{C}_4 sont des solutions efficaces de 3.3 : $X^{\mathcal{C}_4} \subseteq X^E$*

En revanche, les autres propriétés des compromis \mathcal{C}_2 et \mathcal{C}_3 ne sont plus respectées.

Conclusion

Le compromis en optimisation multidisciplinaire est souvent traité comme un compromis entre les objectifs des disciplines. Bien que cette conception du compromis soit pertinente dans le cas d'un problème massivement multiobjectif décomposé en sous-problèmes plus simples à résoudre, elle ne tient pas compte des préférences locales des disciplines.

Nous avons proposé quatre définitions du compromis dans le cadre de l'optimisation multidisciplinaire. Chaque compromis est formellement défini à l'aide de la théorie des ensembles ordonnés. Le premier compromis, noté \mathcal{C}_1 , se révèle identique au compromis entre les objectifs et ne correspond donc pas à nos critères. Les compromis \mathcal{C}_2 et \mathcal{C}_3 reposent sur des extensions de la relation d'ordre de chaque discipline. Bien qu'ayant des propriétés intéressantes théoriquement, ils ne sont pas adaptés à une utilisation dans un algorithme évolutionnaire. En effet, les extensions d'ordre induisent une sensibilité de la relation d'ordre à la répartition des points dans l'espace des objectifs. Ainsi, à chaque génération de l'algorithme évolutionnaire, la répartition des individus change dans la population et la définition du minimum avec eux. Afin de pallier ce problème, le quatrième compromis – \mathcal{C}_4 –, plutôt que de modifier la relation d'ordre au sein de chaque discipline avant d'en réaliser le produit, modifie la définition du produit pour prendre en compte les incomparabilités dans les disciplines. Ce dernier compromis est donc plus adapté aux algorithmes évolutionnaires multiobjectifs.

Dans les différents compromis, nous avons utilisé des applications θ préservant la relation de dominance de Pareto. D'autres préférences, modélisées par θ , pourraient être appliquées, selon les choix des concepteurs dans les disciplines.

Dans le chapitre suivant, les compromis sont implémentés au sein d'un algorithme évolutionnaire multiobjectif du type NSGA-II. Le comportement de cet algorithme en fonction du compromis utilisé est étudié sur plusieurs problèmes.

EMO-MDO : un algorithme évolutionnaire multiobjectif pour l'optimisation multidisciplinaire

SOMMAIRE

INTRODUCTION	84
4.1 ALGORITHME	84
4.1.1 Algorithme général	84
4.1.2 Calcul du rang	85
4.2 EXPÉRIMENTATIONS	90
4.2.1 Protocole expérimental	90
4.2.2 Problème 1	92
4.2.3 Problème 2	94
4.2.4 Problème 3 : problème à quatre barres	102
4.2.5 Problème 4 : le réducteur de vitesse	108
4.2.6 Synthèse des résultats	114
CONCLUSION	117

Introduction

NOUS présentons une implémentation des compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 introduits au chapitre précédent à l'aide d'un algorithme évolutionnaire multiobjectif. Cet algorithme est appelé EMO-MDO pour *Evolutionary Multiobjective Optimization & Multi-disciplinary Design Optimization*. Nous présentons, dans un premier temps, l'algorithme et en particulier ce qui le différencie de NSGA-II (Deb *et al.* 2002) dont il s'inspire. Dans un second temps, le comportement de l'algorithme en fonction du compromis utilisé est étudié sur quatre problèmes de la littérature. Ces analyses ont pour objectif de vérifier les propriétés théoriques énoncées au chapitre 3, et d'observer que les solutions obtenues sont pertinentes dans le cadre des problèmes considérés. Cependant, il n'existe pas, à notre connaissance, de mesure d'évaluation objective de la qualité des solutions pour ce type de problème. Les résultats obtenus sont donc évalués de manière qualitative et non quantitative.

4.1 Algorithme

Nous présentons, dans un premier temps, la structure générale de l'algorithme, puis nous décrivons, dans un second temps, la méthode de calcul des rangs utilisée, qui est la particularité de cet algorithme.

4.1.1 Algorithme général

L'algorithme proposé suit le schéma classique des algorithmes génétiques. Il est inspiré de l'algorithme NSGA-II (Deb *et al.* 2002) dont il reprend les opérateurs de sélection, croisement et mutation. Sa particularité réside dans le calcul de la relation d'ordre existant entre les individus de la population en fonction de la définition du compromis (\mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 ou \mathcal{C}_4). Cette relation d'ordre est utilisée pour calculer le rang des individus à chaque génération.

Algorithme 8 EMO-MDO

ENTRÉES : compromis utilisé c

SORTIES : population solution

```
population ← initialisation()
rangs ← calcule_rangs(c,population)
pour  $i = 1$  to max_iteration faire
    parents ← selection(population,rangs)
    enfants ← mutation(croisements(parents))
    population ← population  $\cup$  enfants
    rangs ← calcule_rangs(c,population)
    population ← generation(population,rangs)
fin pour
retourner population
```

Une population initiale de N individus est créée à partir de solutions aléatoirement choisies dans l'ensemble admissible du problème. La relation d'ordre entre les individus est ensuite calculée par la fonction `calcule_rangs` à partir de la *population* et du compromis $c \in \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$. De cette relation d'ordre sont déduits les *rangs* des individus de la

population. Sur la base des *rangs* des individus et de leur distance phénotypique, une population de *parents* est sélectionnée dans la *population*. Des opérations de **croisement** et de **mutation** sont effectuées sur la population de *parents* afin de créer une population d'*enfants*. Les *enfants* sont ajoutés à la *population* courante pour créer une *population* de taille $2 \times N$. Les *rangs* sont ensuite calculés pour chaque individu de la *population* double à partir de laquelle seront sélectionnés les individus faisant partie de la prochaine *population* par la fonction **generation** (algorithme 8).

Cet algorithme est identique à NSGA-II (Deb *et al.* 2002) quand il est utilisé avec le compromis \mathcal{C}_0 :

- il utilise un codage réel des individus ;
- les croisements sont effectués via l'opérateur *SBX*¹ (*Simulated Binary Crossover*) (Deb et Agrawal 1995, Deb et Beyer 1999) avec prise en charge des contraintes de bornes (Deb et Agrawal 1999) ;
- la mutation est effectuée via l'opérateur de mutation polynomiale² (Deb et Agrawal 1995) modifiée pour prendre en compte les bornes (Deb et Tiwari 2008) ;
- la sélection est une sélection binaire par tournoi avec prise en compte de la distance phénotypique³ (Deb 2001) ;
- la génération suivante est sélectionnée via une procédure de tri sur le respect des contraintes et sur les rangs, en prenant en compte la distance phénotypique⁴ (Deb 2001).
- les contraintes sont prises en compte dans le calcul du rang.

La seule différence avec NSGA-II réside dans la procédure de calcul des rangs, qui est plus générique et peut utiliser des structures préordonnées quelconques. En effet, NSGA-II étant conçu pour le multiobjectif, les rangs calculés sont les rangs d'un espace produit d'ordres totaux et donc d'un ordre (définition 1.15). L'utilisation d'un préordre permet plus de liberté dans l'expression des préférences, et permet en particulier la modélisation des compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 .

4.1.2 Calcul du rang

Les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 , dans le cas général, ne produisent pas d'ordre mais des préordres. L'ensemble des minimums dans un préordre (définition 1.14) peut être interprété comme l'union des classes d'équivalence des minimums de son ordre quotient. Le calcul des rangs d'un préordre nécessite le calcul de son ordre quotient, puis le calcul des rangs de l'ordre quotient obtenu.

Algorithme

Afin de calculer l'ordre quotient, nous cherchons tout d'abord les composantes fortement connexes du préordre. Celles-ci peuvent être calculées efficacement en utilisant l'algorithme de Tarjan (1972) (voir aussi Kosaraju 1978, Aho *et al.* 1983, Sharir 1981), et en ordonnant les classes d'équivalence selon le préordre. Les rangs des éléments du préordre sont alors les rangs des éléments de la classe d'équivalence dans l'ordre quotient. L'exemple 4.1 illustre le calcul du rang (algorithme 9).

1. Cf. section 1.4.2.
2. Cf. section 1.4.3.
3. *crowded binary constrained tournament selection operator*.
4. *non-constrain-dominated sorting procedure with crowding*.

Algorithme 9 Procédure de rang

ENTRÉES : compromis c utilisé ; population
SORTIES : Sortie : tableau de rangs

```

population ← évaluation(population)
ordre ← compromis(c,population) //calcul de la relation d'ordre
ordre_quotient ← quotient(population,ordre)
r ← 1 //rang courant
classes_equivalence ← minimum(ordre_quotient)
ordre_quotient ← ordre_quotient \ classes_equivalence
∀e ∈ ∪ classes_equivalence, rangs[e] ← r
tant que ordre_quotient ≠ ∅ faire
    r ← r + 1
    classes_equivalence ← minimum(ordre_quotient)
    ordre_quotient ← ordre_quotient \ classes_equivalence
    ∀e ∈ ∪ classes_equivalence, rangs[e] ← r
fin tant que
retourner rangs
    
```

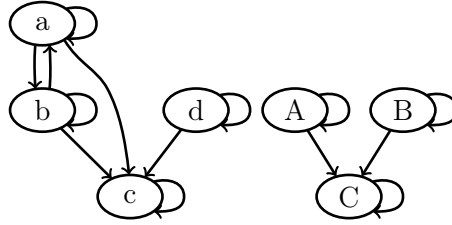


FIGURE 4.1 – Illustration du préordre P et de son ordre quotient Q de l'exemple 4.1.

Exemple 4.1 Soit un préordre $P = (\{a, b, c, d\}, \{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, c), (d, c), (d, d)\})$. L'ordre quotient de P est l'ordre $Q = (\{A, B, C\}, \{(A, A), (A, C), (B, B), (B, C), (C, C)\})$ avec $A = \{a, b\}$, $B = \{d\}$ et $C = \{c\}$ (figure 4.1). Les rangs dans l'ordre quotient sont $r(A) = r(B) = 1$ et $r(C) = 2$. Donc $r(a) = r(b) = r(d) = 1$ et $r(c) = 2$.

Calcul des compromis

La relation d'ordre de l'ensemble de comparaison calculée dans la fonction `compromis` de l'algorithme 9 dépend du compromis souhaité.

L'algorithme 10 détaille l'implémentation du compromis \mathcal{C}_1 . Pour chaque *objectif* de chaque *discipline*, on calcule l'*ordre* total correspondant à l'aide de la fonction `ordonne`. Le produit de chacun de ces *ordres* est effectué pour obtenir l'*ordre* correspondant à l'ensemble de comparaison.

Pour le compromis \mathcal{C}_2 , le produit des *ordres* totaux des ensembles réalisables de chaque objectif f est d'abord calculé pour chaque *discipline* afin d'obtenir l'ordre produit disciplinaire `ordre_disc`. Puis les points *non dominés* d'`ordre_disc` sont séparés des points *dominés* pour être sommés par une somme ordinale. Les ordres disciplinaires `ordre_disc` obtenus pour chaque discipline sont multipliés entre eux (algorithme 11).

L'algorithme utilisé pour le compromis \mathcal{C}_3 (algorithme 12) est similaire à celui du

Algorithme 10 Ordre du compromis \mathcal{C}_1

ENTRÉES : population
SORTIES : Sortie : ordre
ordre $\leftarrow \emptyset$
pour $i = 1$ à q **faire**
 objectifs \leftarrow discipline[i]
 pour $j = 1$ à p_i **faire**
 $f \leftarrow$ objectifs[j]
 ordre \leftarrow ordre \times ordonne(population, f)
 fin pour
fin pour
retourner ordre

Algorithme 11 Ordre du compromis \mathcal{C}_2

ENTRÉES : population
SORTIES : Sortie : ordre
ordre $\leftarrow \emptyset$
pour $i = 1$ à q **faire**
 ordre_disc $\leftarrow \emptyset$
 objectifs \leftarrow discipline[i]
 pour $j = 1$ à p_i **faire**
 $f \leftarrow$ objectifs[j]
 ordre_disc \leftarrow ordre_disc \times ordonne(population, f)
 fin pour
 (non dominés,dominés) \leftarrow partition(ordre_disc)
 ordre_disc \leftarrow non dominés \oplus dominés
 ordre \leftarrow ordre \times ordre_disc
fin pour
retourner ordre

\mathcal{C}_2 à part que les ordres partiels disciplinaires $ordre_disc$ sont décomposés en autant de partitions que de rangs et sommés par une somme ordinale par rangs croissants.

Algorithme 12 Ordre du compromis \mathcal{C}_3

ENTRÉES : population
SORTIES : Sortie : ordre
ordre $\leftarrow \emptyset$
pour $i = 1$ à q **faire**
 ordre_disc $\leftarrow \emptyset$
 objectifs $\leftarrow discipline[i]$
 pour $j = 1$ à p_i **faire**
 f $\leftarrow objectifs[j]$
 ordre_disc $\leftarrow ordre_disc \times ordonne(population, f)$
 fin pour
 o $\leftarrow \emptyset$
 tant que ordre_disc $\neq \emptyset$ **faire**
 rang $\leftarrow \min(ordre_disc)$
 ordre_disc $\leftarrow ordre_disc \setminus rang$
 o $\leftarrow o \oplus rang$
 fin tant que
 ordre $\leftarrow ordre \times o$
fin pour
retourner ordre

L'algorithme 13 décrit le compromis \mathcal{C}_4 . Les ordres partiels disciplinaires $ordre_disc$ sont calculés dans chaque *discipline* par le produit des ensembles réalisables de leurs *objectifs*. Puis, le produit faible (définition 3.1) des ordres disciplinaires est réalisé, afin d'obtenir l'*ordre* décrivant l'ensemble de comparaison du compromis.

Algorithme 13 Ordre du compromis \mathcal{C}_4

ENTRÉES : population
SORTIES : ordre
ordre $\leftarrow \emptyset$
pour $i = 1$ à q **faire**
 ordre_disc $\leftarrow \emptyset$
 objectifs $\leftarrow discipline[i]$
 pour $j = 1$ à p_i **faire**
 f $\leftarrow objectifs[j]$
 ordre_disc $\leftarrow ordre_disc \times ordonne(population, f)$
 fin pour
 ordre $\leftarrow ordre \otimes ordre_disc$
fin pour
retourner ordre

Prise en compte des contraintes

Dans NSGA-II et dans EMO-MDO, la prise en compte des contraintes se fait au moment des deux phases de sélection : pendant la sélection des parents et pendant la sélection de la nouvelle génération.

L'évaluation de la violation des contraintes se fait par une application c à valeurs dans \mathbb{R} telle que

$$c(x) := \sum_{i=1}^k \delta(g_i(x)) \cdot g_i(x) \quad (4.1)$$

en désignant par $g_i(x) \leq 0$ la $i^{\text{ème}}$ contrainte du problème et δ l'application à valeurs binaires :

$$x \mapsto \begin{cases} 0 & \text{si } x \leq 0 \\ 1 & \text{sinon} \end{cases} \quad (4.2)$$

La relation prise en compte pour l'établissement de la relation d'ordre (et donc du calcul du rang) est la relation de contrainte-dominance⁵ :

Définition 4.1 (Relation de contrainte-dominance (Deb 2001)) *Une solution a domine une solution b si une des conditions suivantes est vérifiée :*

- i) a est admissible et b ne l'est pas ;*
- ii) a et b ne sont pas admissibles, mais $c(a) \leq c(b)$;*
- iii) a et b sont admissibles et $a \preceq b$ (dominance de Pareto).*

Dans EMO-MDO, les contraintes sont donc directement prises en compte dans la relation d'ordre et vont ainsi être propagées dans le calcul du rang. Dans la relation d'ordre, pour chacune des composantes d'une solution, $c(x_1) \leq c(x_2) \Rightarrow \phi(x_1) \leq \phi(x_2) \forall x_1, x_2 \in X$. Ainsi, une solution ne respectant pas les contraintes se retrouvera à un rang supérieur par rapport à une solution respectant les contraintes.

Prise en compte de la distance phénotypique

Tout comme dans NSGA-II, et contrairement aux contraintes, la distance phénotypique (ou *crowding* – équation 1.35) n'entre pas en compte dans le calcul du rang. Cette mesure est utilisée dans les opérateurs de sélection pour choisir le plus isolé de deux individus lorsqu'ils ont le même rang.

La mesure de *crowding* concerne la diversité des solutions sur le front de Pareto global (points non dominés de \mathcal{M}') alors que la diversité souhaitée dans le cadre de l'optimisation multidisciplinaire devrait se situer dans les espaces des objectifs de chaque discipline.

4.2 Expérimentations

L'algorithme est testé sur quatre problèmes différents de la littérature. Les deux premiers sont des problèmes artificiels et les deux suivants correspondent à des problèmes de conception mécanique.

5. En anglais *constrain dominance*.

Il n'existe pas, à notre connaissance, de mesure de la qualité d'un compromis en optimisation multidisciplinaire, ce qui nous permettrait d'avoir des évaluations objectives des performances de l'algorithme (qualité des solutions, vitesse de convergence, diversité des solutions...). Cependant, le but de ces expérimentations n'est pas d'évaluer l'algorithme évolutionnaire en tant que tel, mais d'étudier l'influence des compromis et leur applicabilité dans un algorithme évolutionnaire. Les objectifs sont les suivants :

1. vérifier les propriétés théoriques des compromis exposées au chapitre 3 :
 - les solutions de compromis sont des solutions efficaces (propositions 3.1, 3.4 et 3.8) ;
 - s'il existe des solutions efficaces communes à toutes les disciplines simultanément, alors l'ensemble des solutions de compromis est exactement l'intersection des solutions efficaces des disciplines (propositions 3.2 et 3.5) ;
 - les solutions de compromis se trouvent entre les projections des solutions efficaces des disciplines (propositions 3.3 et 3.6).
 - le compromis \mathcal{C}_4 est plus robuste que les compromis \mathcal{C}_2 et \mathcal{C}_3 (remarque 3.8 et proposition 3.7).
2. analyser le comportement d'un algorithme évolutionnaire implémentant ces compromis :
 - comparaison avec des solutions de la littérature ;
 - vérifier si les solutions sont similaires à des solutions obtenues par filtrage⁶ ;
 - évaluer la pertinence des solutions proposées ;
 - vérifier la pertinence de la prise en compte de la distance phénotypique, telle qu'elle existe dans NSGA-II, pour ce type de problème.

Pour cela, l'algorithme a été testé dans différentes conditions.

4.2.1 Protocole expérimental

Les problèmes sont tout d'abord évalués sur chacun des compromis avec une méthode de filtrage où des solutions sont tirées uniformément ou aléatoirement (procédure de Monte-Carlo) dans l'ensemble admissible. Les solutions trouvées par cette méthode, en plus des solutions de la littérature, servent de référence aux comparaisons des solutions obtenues avec EMO-MDO.

Puis, pour chacun de ces problèmes, différentes configurations de l'algorithme sont testées. Elles sont résumées dans le tableau 4.1. Chacune de ces configurations est réalisée avec les paramètres suivants de l'algorithme génétique⁷ :

- probabilité de croisement 0,8 ;
- probabilité de mutation 0,2 ;
- $\eta_c = 2,0$ (pour le croisement SBX) ;
- $\eta_m = 4,0$ (pour la mutation).

Chacune des configurations est testée trois fois afin de vérifier la stabilité des résultats obtenus.

Les populations initiales sont réalisées à partir de solutions prises aléatoirement suivant une loi de probabilité uniforme dans l'ensemble admissible. Ainsi, les algorithmes sont initialisés avec des valeurs admissibles. Seuls les résultats les plus représentatifs seront présentés.

6. Solutions obtenues en sélectionnant l'ensemble des minimums parmi des solutions tirées uniformément ou aléatoirement dans l'ensemble admissible.

7. Notre but n'étant pas d'évaluer les performances de l'algorithme, mais simplement son comportement, nous n'avons pas cherché à régler les paramètres.

nomenclature	compromis	crowding	population	itérations
c1_n_10x50	\mathcal{C}_1	non	10	50
c1_o_10x50	\mathcal{C}_1	oui	10	50
c1_n_20x20	\mathcal{C}_1	non	20	20
c1_o_20x20	\mathcal{C}_1	oui	20	20
c1_n_50x50	\mathcal{C}_1	non	50	50
c1_o_50x50	\mathcal{C}_1	oui	50	50
c1_n_50x500	\mathcal{C}_1	non	50	500
c1_o_50x500	\mathcal{C}_1	oui	50	500
c2_n_10x50	\mathcal{C}_2	non	10	50
c2_o_10x50	\mathcal{C}_2	oui	10	50
c2_n_20x20	\mathcal{C}_2	non	20	20
c2_o_20x20	\mathcal{C}_2	oui	20	20
c2_n_50x50	\mathcal{C}_2	non	50	50
c2_o_50x50	\mathcal{C}_2	oui	50	50
c2_n_50x500	\mathcal{C}_2	non	50	500
c2_o_50x500	\mathcal{C}_2	oui	50	500
c3_n_10x50	\mathcal{C}_3	non	10	50
c3_o_10x50	\mathcal{C}_3	oui	10	50
c3_n_20x20	\mathcal{C}_3	non	20	20
c3_o_20x20	\mathcal{C}_3	oui	20	20
c3_n_50x50	\mathcal{C}_3	non	50	50
c3_o_50x50	\mathcal{C}_3	oui	50	50
c3_n_50x500	\mathcal{C}_3	non	50	500
c3_o_50x500	\mathcal{C}_3	oui	50	500
c4_n_10x50	\mathcal{C}_4	non	10	50
c4_o_10x50	\mathcal{C}_4	oui	10	50
c4_n_20x20	\mathcal{C}_4	non	20	20
c4_o_20x20	\mathcal{C}_4	oui	20	20
c4_n_50x50	\mathcal{C}_4	non	50	50
c4_o_50x50	\mathcal{C}_4	oui	50	50
c4_n_50x500	\mathcal{C}_4	non	50	500
c4_o_50x500	\mathcal{C}_4	oui	50	500

TABLEAU 4.1 – Configurations de tests pour un problème donné. Chaque problème est testé dans 32 configurations différentes.

4.2.2 Problème 1

Le premier problème est une adaptation d'un exemple issu de (Rabeau 2007) duquel les variables locales ont été enlevées.

Description du problème

Le problème d'origine comporte des variables locales qui n'ont aucune incidence sur les valeurs des variables communes, nous les avons donc supprimées. Les valeurs numériques du problème ont également été remplacées par des paramètres $\alpha, \beta, \lambda, \omega \in \mathbb{R}$ afin de rendre le problème plus générique.

Le problème est composé de deux disciplines D_1 et D_2 ayant chacune deux objectifs. La première discipline D_1 :

$$D_1 \begin{cases} \min_{x \in X} f_{1,1}(x) = (x - \alpha)^2 \\ \min_{x \in X} f_{1,2}(x) = (x - \beta)^2 \end{cases} \quad (4.3)$$

et la seconde discipline D_2 :

$$D_2 \begin{cases} \min_{x \in X} f_{2,1}(x) = (x - \lambda)^2 \\ \min_{x \in X} f_{2,2}(x) = (x - \omega)^2 \end{cases} \quad (4.4)$$

avec $\alpha < \beta$, $\lambda < \omega$ et $\alpha < \omega$.

Remarque 4.1 Chaque problème disciplinaire est similaire au problème de test de Schaffer (1984) SCH1 où $\alpha = \lambda = 0$ et $\beta = \omega = 2$.

Analyse du problème

Chaque fonction objectif est une fonction quadratique, convexe, n'acceptant qu'un unique minimum :

$$\arg \min_{x \in X} f_{1,1}(x) = \alpha \quad (4.5)$$

$$\arg \min_{x \in X} f_{1,2}(x) = \beta \quad (4.6)$$

$$\arg \min_{x \in X} f_{2,1}(x) = \lambda \quad (4.7)$$

$$\arg \min_{x \in X} f_{2,2}(x) = \omega \quad (4.8)$$

Ainsi, les solutions efficaces de D_1 et D_2 sont respectivement (figure 4.2)

$$X_1^E = [\alpha; \beta] \quad (4.9)$$

$$X_2^E = [\lambda; \omega] \quad (4.10)$$

Deux cas se présentent suivant les paramètres du problème :

$\beta < \lambda$: il n'y a aucune solution efficace commune entre les deux disciplines. Autrement dit, $X_1^E \cap X_2^E = \emptyset$.

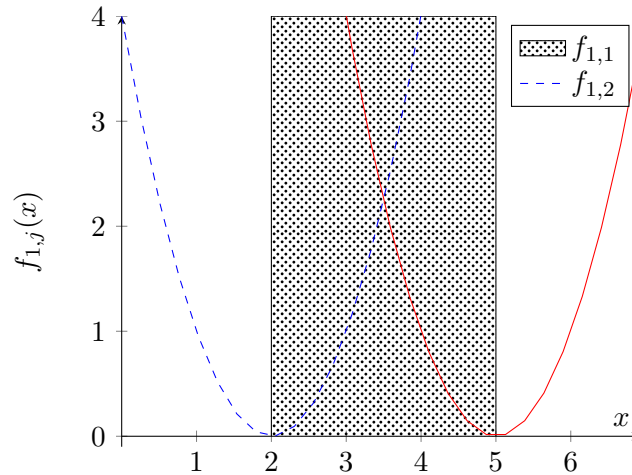


FIGURE 4.2 – Solutions efficaces de la discipline D_1 avec $\alpha = 2$ et $\beta = 5$. Les solutions efficaces de la discipline se trouvent dans l'intervalle $[2; 5]$.

compromis	$\beta < \lambda$	$\lambda \leq \beta$
\mathcal{C}_1	$[\alpha, \omega]$	$[\alpha, \omega]$
\mathcal{C}_2	$[\beta, \lambda]$	$[\lambda, \beta]$
\mathcal{C}_3	$[\beta, \lambda]$	$[\lambda, \beta]$
\mathcal{C}_4	$[\beta, \lambda]$	$[\lambda, \beta]$

TABLEAU 4.2 – Solutions théoriques du problème 1 en fonction du compromis choisi et des paramètres.

$\lambda \leq \beta$: il y a un intervalle de solutions efficaces communes entre les deux disciplines. Autrement dit, $X_1^E \cap X_2^E = [\lambda; \beta]$.

Cet exemple a été conçu pour tester la propriété suivante : s'il y a un sous-ensemble commun de solutions dans chaque discipline, les solutions de compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 doivent se trouver dans ce sous-ensemble (propositions 3.1, 3.4 et 3.8). Les solutions de compromis \mathcal{C}_1 , quant à elles, se trouvent théoriquement dans l'union des intervalles des solutions efficaces des disciplines. Le tableau 4.2 résume les intervalles de solution attendus suivant le compromis et les paramètres du problème.

Les résultats attendus sont identiques avec $\beta < \lambda$ et $\lambda \leq \beta$. Dans le cas où $\lambda \leq \beta$, les résultats pour les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 correspondent à l'intersection des solutions efficaces des disciplines ($X_1^E \cap X_2^E$). Dans le cas où $\beta < \lambda$, les résultats correspondent à l'intervalle $[\sup(X_1^E); \inf(X_2^E)]$ se trouvant entre les intervalles de solutions efficaces des disciplines. En effet, les solutions se trouvent théoriquement dans l'intervalle $[\inf(X_1^E); \sup(X_2^E)]$, c'est-à-dire toutes les solutions se trouvant entre α et ω . Mais les solutions β et λ dominent respectivement toutes les solutions inférieures à β dans la première discipline, et toutes les solutions supérieures à λ dans la seconde discipline, ce qui exclut les intervalles $[\alpha; \beta[$ et $]\lambda; \omega]$ de l'ensemble des solutions de compromis.

Les résultats obtenus par filtrage (simulation de Monte-Carlo) sont conformes aux résultats théoriques : il existe des solutions de compromis \mathcal{C}_1 qui ne sont pas efficaces dans une des deux disciplines (figure 4.3). Les solutions de compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sont en revanche les solutions efficaces des deux disciplines simultanément (figure 4.4).

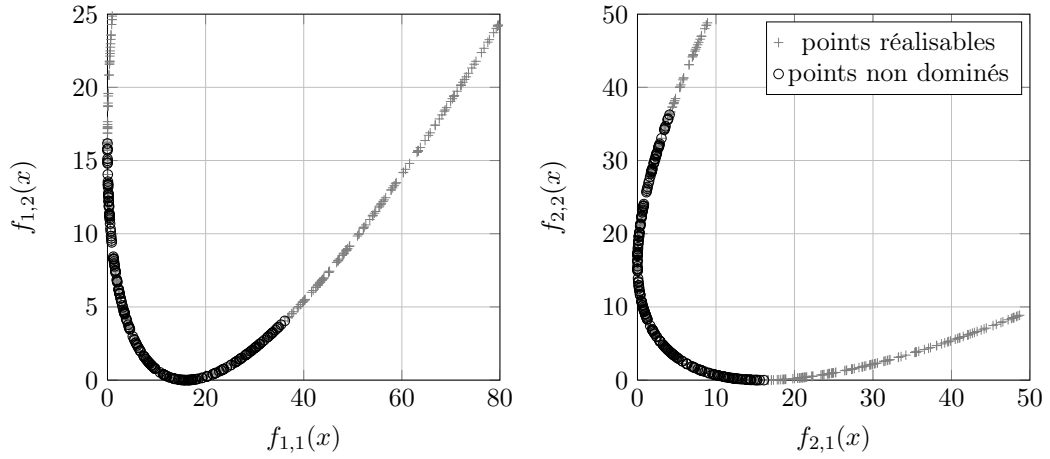


FIGURE 4.3 – Solution du problème 1 en appliquant le compromis C_1 sur 400 solutions prises aléatoirement dans l'ensemble admissible.

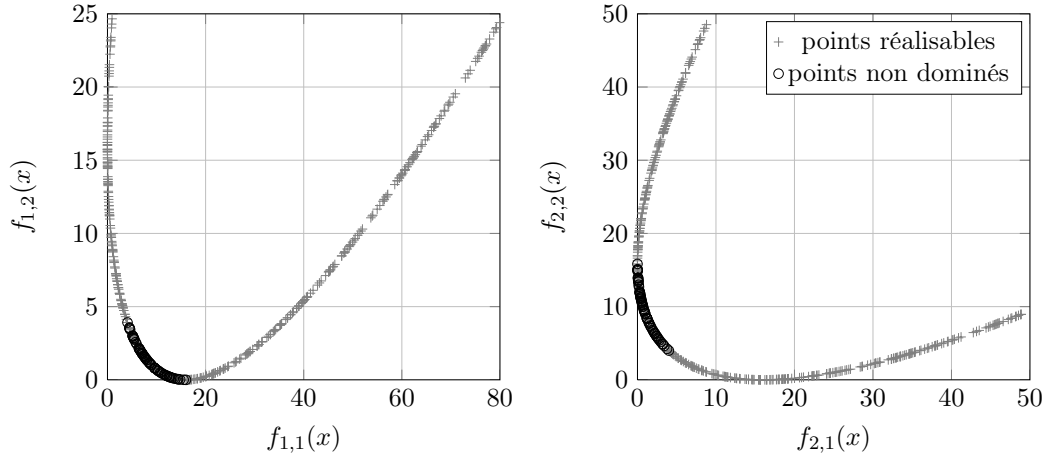


FIGURE 4.4 – Solution du problème 1 en appliquant le compromis C_2 sur 400 solutions prises aléatoirement dans l'ensemble admissible. Les solutions sont identiques pour C_3 et C_4 .

Résultats

Le problème 1 a été testé avec les paramètres $\alpha = 1, \beta = 5, \lambda = 3, \omega = 7$ ($X_1^E \cap X_2^E \neq \emptyset$) puis $\alpha = 1, \beta = 3, \lambda = 5, \omega = 7$ ($X_1^E \cap X_2^E = \emptyset$), et $x \in [0; 10]$ dans les différentes configurations de test.

Les solutions trouvées dans toutes les configurations d'EMO-MDO sont conformes aux solutions théoriques (tableau 4.2) : Pour le compromis C_1 , les solutions trouvées correspondent à l'union des solutions efficaces des deux disciplines (figure 4.5). Il existe, dans chaque discipline, des solutions de compromis dominées. Tandis que pour les compromis C_2, C_3 et C_4 , les solutions correspondent aux solutions appartenant à l'intervalle entre β et λ (figure 4.6) : il n'existe pas de solution de compromis dominée dans les disciplines.

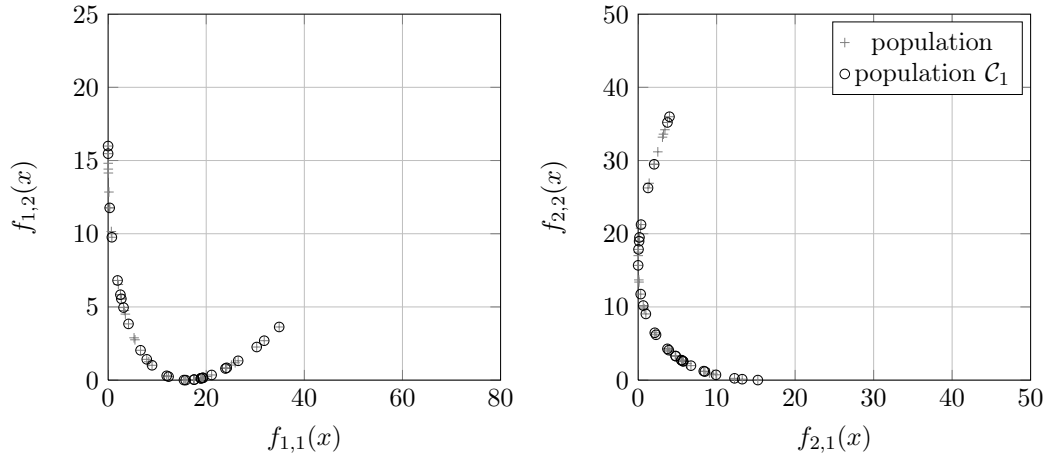


FIGURE 4.5 – Solutions obtenues sur le problème 1 avec la configuration `c1_n_50x50` d'EMO-MDO. Les solutions sont semblables à celles obtenues par une simulation de Monte-Carlo (figure 4.3).

4.2.3 Problème 2

Le second problème est présenté dans (Engau et Wiecek 2007). Il s'agit, tout comme le problème précédent, d'un problème artificiel.

Description du problème

Ce problème est présenté comme un problème de conception à quatre objectifs pour lequel une solution efficace est recherchée. Pour aider à la sélection de cette solution, le problème à quatre objectifs est décomposé en deux sous-problèmes à deux objectifs chacun, afin de visualiser plus simplement les solutions. Ce problème ne correspond pas à un problème réel et est formulé sous la forme de deux disciplines D_1 et D_2 :

$$D_1 \begin{cases} \min_{x \in X} f_{1,1}(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \min_{x \in X} f_{1,2}(x) = x_1^2 + (x_2 - 3)^2 \end{cases} \quad (4.11)$$

$$D_2 \begin{cases} \min_{x \in X} f_{2,1}(x) = (x_1 - 1)^2 + (x_2 + 1)^2 \\ \min_{x \in X} f_{2,2}(x) = (x_1 + 1)^2 + (x_2 - 1)^2 \end{cases} \quad (4.12)$$

Sous les contraintes :

$$\begin{cases} (x_1^2 - x_2) \leq 0 \\ (x_1 + x_2 - 2) \leq 0 \\ (-x_1) \leq 0 \end{cases} \quad (4.13)$$

Les domaines des variables ne sont pas donnés par les auteurs. En observant les contraintes du problème, nous en déduisons les intervalles suivants :

$$x_1 \in [0; 2] \quad (4.14)$$

$$x_2 \in [0; 2] \quad (4.15)$$

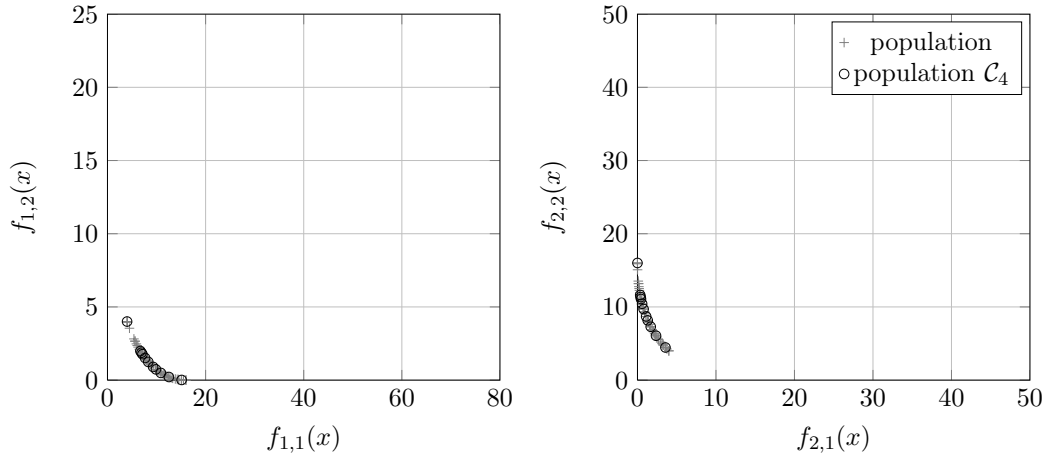


FIGURE 4.6 – Solutions obtenues sur le problème 1 avec la configuration `c4_o_50x50` d'EMO-MDO. Le résultat est similaire aux différentes configurations de C_2 , C_2 et C_4 . Les solutions sont semblables à celles obtenues par une simulation de Monte-Carlo (figure 4.4).

Ces intervalles, compatibles avec les contraintes du problème, seront utilisés dans les algorithmes de résolution.

Analyse du problème

Dans ce problème, les ensembles de solutions efficaces des deux disciplines prises indépendamment sont disjoints ($X_1^E \cap X_2^E = \emptyset$). Si l'on considère ce problème comme un unique problème à quatre objectifs, l'ensemble des solutions admissibles et l'ensemble des solutions efficaces sont égaux ($X = X^E$).

Engau et Wiecek (2007) donnent les résultats présentés figure 4.7. Les auteurs notent que les points non dominés dans une discipline sont à l'opposé dans l'autre discipline (figure 4.8).

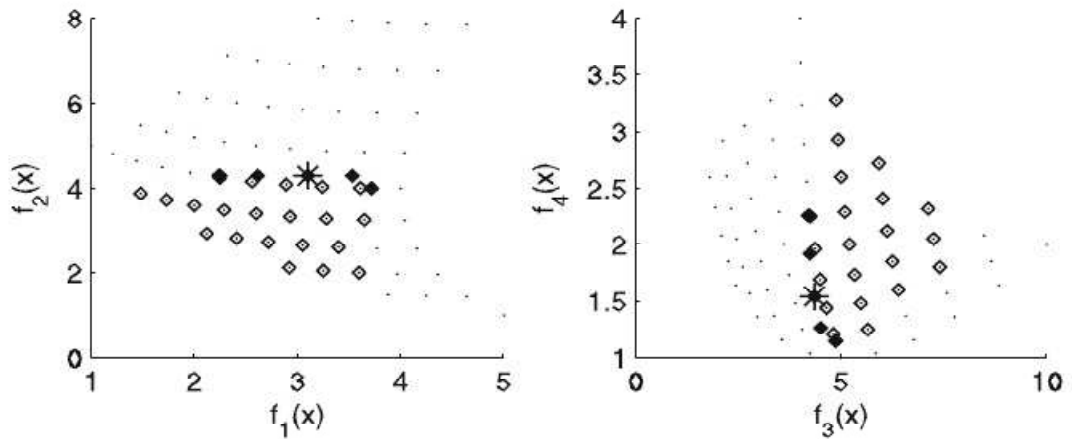


FIGURE 4.7 – Résultats d'Engau et Wiecek (2007) pour le problème 2. Les points correspondent aux solutions réalisables dans l'espace des objectifs. Les losanges sont les solutions Pareto optimales de la première discipline relâchée à un ϵ près. Les losanges noirs correspondent aux losanges gris non dominés dans la discipline 2. Parmi ces points, une solution est choisie par le décideur : l'étoile.

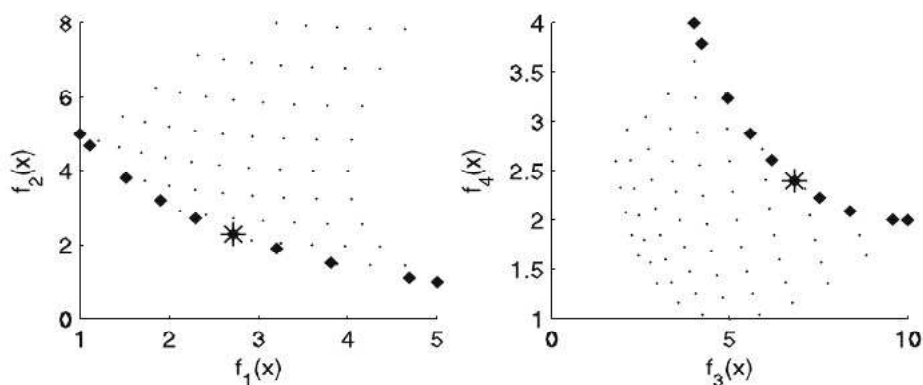


FIGURE 4.8 – Illustration d'Engau et Wiecek (2007) du problème 2. Les points correspondent aux solutions réalisables dans l'espace des objectifs. Les losanges sont les solutions efficaces pour la première discipline projetées dans les deux disciplines. Parmi ces points, une solution est choisie par le décideur : l'étoile.

On peut également observer que les rangs des solutions dans chaque discipline (compromis C_3) sont très dépendants de la répartition des points dans l'espace réalisable du problème. Si les points sont équitablement répartis, les rangs les plus faibles se trouvent dans la partie basse de l'espace des objectifs de la discipline D_1 et dans la partie droite de l'espace des objectifs de la discipline D_2 . Dans chacune de ces disciplines, la valeur des rangs augmente à mesure que l'on s'éloigne de cette zone (figure 4.9).

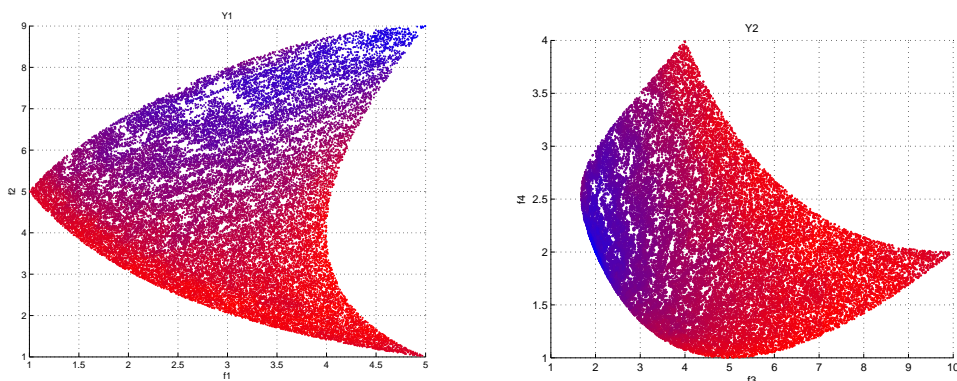


FIGURE 4.9 – Problème 2 : valeurs des rangs du compromis C_3 calculées à partir de solutions tirées aléatoirement dans l'espace de décision du problème 2. Les solutions de rangs les plus faibles sont en rouge tandis que les solutions de rangs les plus élevés sont en bleu.

En choisissant de façon non équitable des solutions de l'espace réalisable (beaucoup plus de solutions sont tirées dans la zone de compromis C_3 obtenue avec un tirage uniforme), les rangs s'inversent (figure 4.10).

Les solutions obtenues par filtrage (tirage uniforme dans l'ensemble admissible – figure 4.11) sont différentes pour les quatre compromis. L'ensemble des solutions de compromis C_3 est plus restreint que pour le compromis C_2 .

D'après ces observations, nous devrions observer avec l'algorithme EMO-MDO les faits suivants :

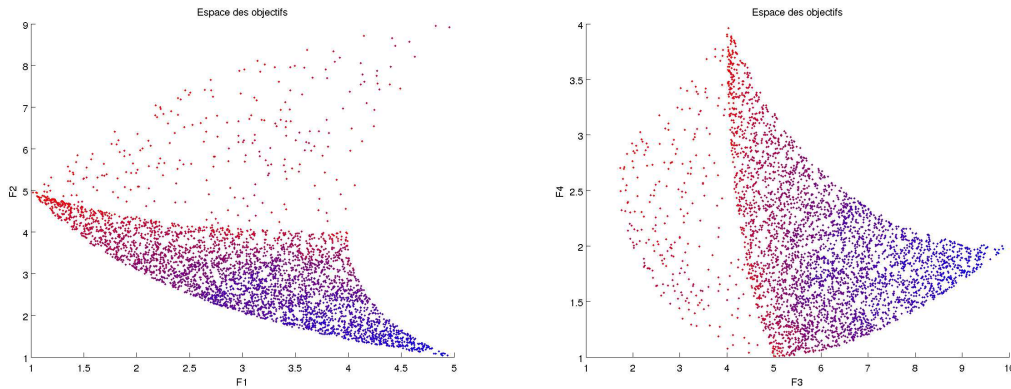


FIGURE 4.10 – *Problème 2 : valeurs des rangs du compromis \mathcal{C}_3 calculées à partir de solutions tirées aléatoirement dans l'espace de décision du problème 2. Les solutions de rang les plus faibles sont en rouge tandis que les solutions de rangs les plus élevés sont en bleu. On observe que les valeurs des rangs sont inversées par rapport à la figure 4.9.*

- les solutions de compromis \mathcal{C}_1 doivent être aléatoirement réparties dans l'ensemble réalisable ;
- les solutions de compromis trouvées doivent être différentes entre \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 avec les inclusions suivantes : $Z^{\mathcal{C}_3} \subseteq Z^{\mathcal{C}_2} \subseteq Z^{\mathcal{C}_1}$ et $Z^{\mathcal{C}_4} \subseteq Z^{\mathcal{C}_1}$;
- les compromis \mathcal{C}_2 et \mathcal{C}_3 peuvent faire converger la population vers des solutions différentes de celles trouvées par filtrage ;
- les solutions de compromis \mathcal{C}_4 doivent être, dans les différentes configurations de l'algorithme, similaires aux solutions obtenues par filtrage.

Résultats

Les figures 4.12 et 4.13 représentent les résultats trouvés par EMO-MDO pour les différents compromis, respectivement sans et avec la prise en compte de la diversité phénotypique. Les résultats présentés sont obtenus avec une population de 50 individus et 50 itérations de l'algorithme. Les résultats obtenus avec les autres configurations sont similaires.

Pour le compromis \mathcal{C}_1 , toutes les solutions admissibles du problème sont également des solutions optimales. Les solutions de compromis \mathcal{C}_1 trouvées par EMO-MDO sont donc des solutions admissibles distribuées aléatoirement. On peut remarquer que la prise en compte de la distance phénotypique n'améliore pas la diversité des solutions dans les espaces des objectifs des disciplines.

Avec le compromis \mathcal{C}_2 , les solutions trouvées par EMO-MDO ne recouvrent pas tout l'espace et sont beaucoup plus regroupées aux centres des espaces réalisables de chaque discipline. L'utilisation de la distance phénotypique permet néanmoins une plus grande dispersion des solutions.

Les solutions de compromis \mathcal{C}_3 trouvées par les différentes configurations d'EMO-MDO sont éloignées des solutions trouvées par la méthode de filtrage (avec tirage uniformément réparti et aléatoire). Ces différences s'expliquent par la plus grande sensibilité de la méthode de compromis \mathcal{C}_3 à la répartition des individus. Dans un premier temps, la population se déplace vers la zone de compromis (la moitié inférieure pour D_1 et la moitié droite pour D_2 de leur ensemble réalisable respectif – figure 4.9). Or, quand la po-

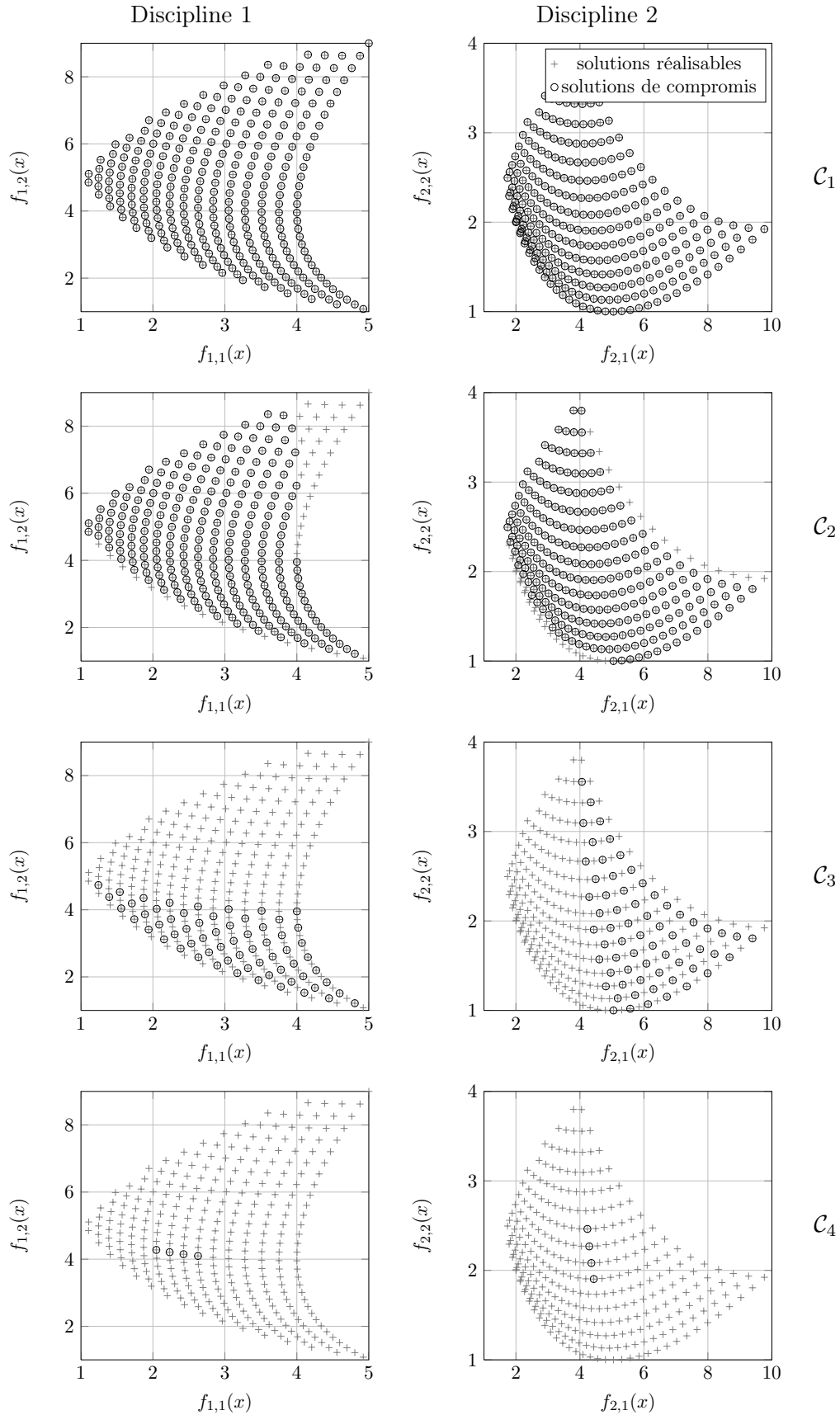


FIGURE 4.11 – Solutions de compromis avec \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 pour le problème 2 sur un tirage uniforme de 400 points de solutions admissibles.

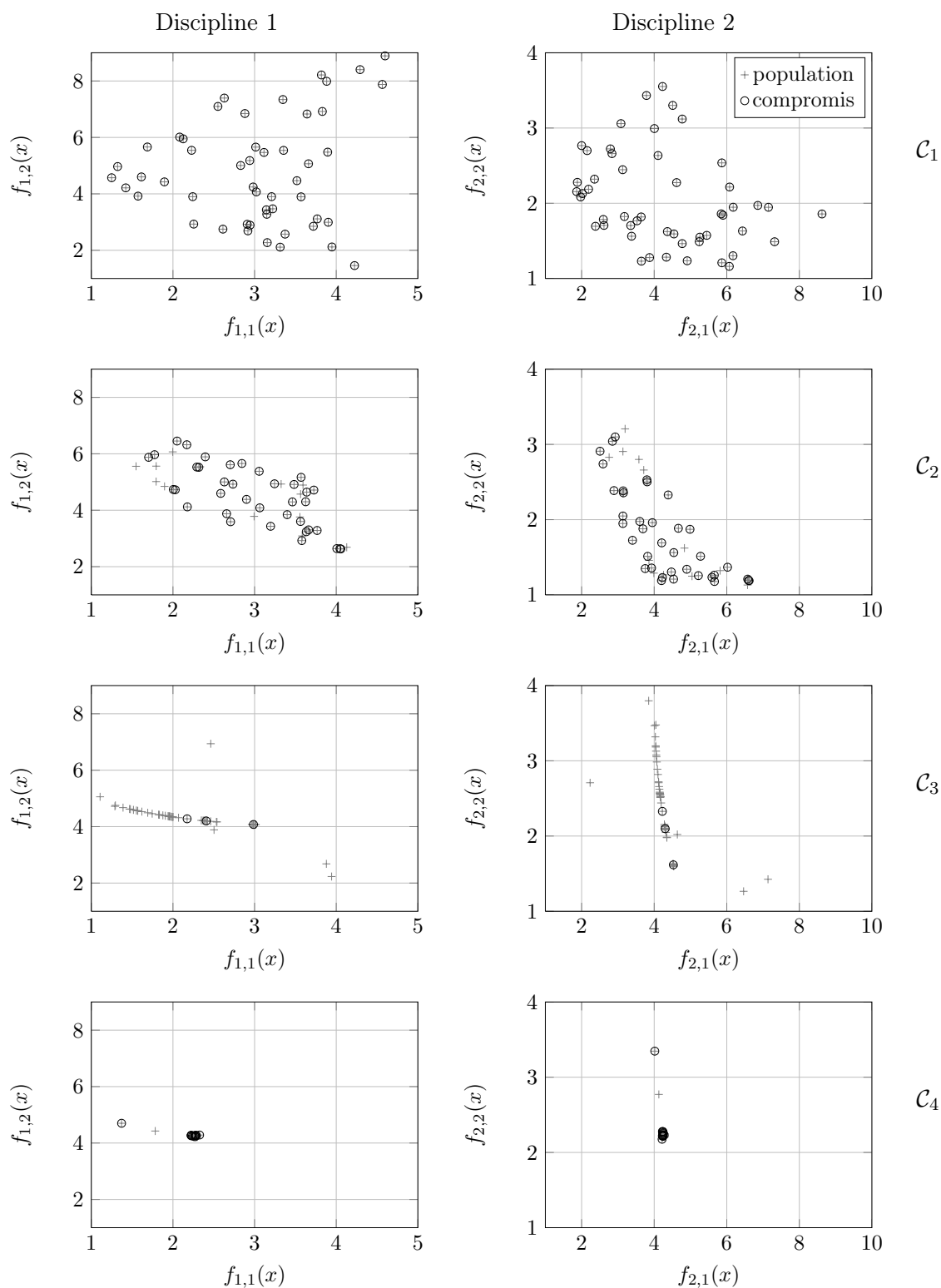


FIGURE 4.12 – Exemple 2 avec les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sans prise en compte de la distance phénotypique. Les résultats sont obtenus avec une population de 50 individus et 50 itérations. Les solutions entourées d'un cercle (o) sont les solutions de compromis sur la dernière population seulement.

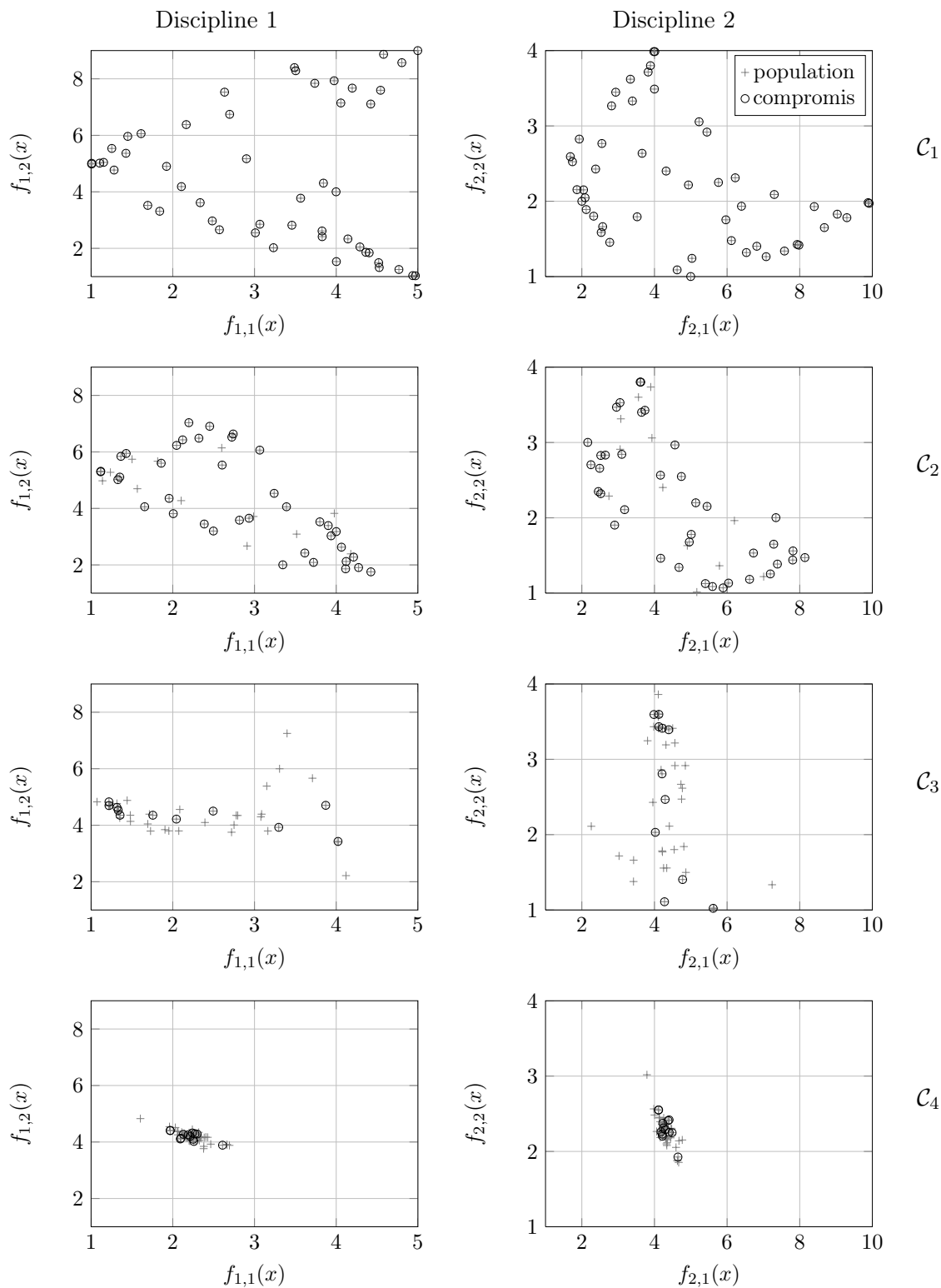


FIGURE 4.13 – Exemple 2 avec les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 avec prise en compte de la distance phénotypique. Les résultats sont obtenus avec une population de 50 individus et 50 itérations. On peut remarquer que les solutions sont plus diversifiées que sans la prise en compte de la distance phénotypique (figure 4.12). Les solutions entourées d'un cercle (o) sont les solutions de compromis sur la dernière population seulement.

pulation est majoritairement dans cette zone, le compromis se trouve alors dans la partie haute de cette zone pour la discipline D_1 et dans la partie gauche pour la discipline D_2 (figure 4.10). Les solutions générées par l'algorithme convergent donc naturellement vers cet axe.

Les solutions de compromis \mathcal{C}_4 sont proches des solutions obtenues par filtrage. On peut également noter qu'elles sont proches de la solution choisie par Engau et Wiecek (2007)⁸ (figure 4.7).

Globalement, sur les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 , la prise en compte de la distance phénotypique dans EMO-MDO améliore la diversité des solutions générées. Les solutions ne sont cependant pas uniformément distribuées dans l'espace des objectifs des disciplines. Ceci est dû à la fonction de *crowding*, qui ne considère pas la diversité au sein de chaque discipline, mais sur le front de Pareto global du problème. Parmi les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 , ce sont les solutions de compromis \mathcal{C}_4 qui sont les plus proches des solutions obtenues par filtrage. Les solutions de compromis \mathcal{C}_2 sont également plus proches des solutions obtenues par filtrage que ne le sont les solutions de compromis \mathcal{C}_3 .

4.2.4 Problème 3 : problème à quatre barres

Le troisième exemple, également présenté dans (Engau et Wiecek 2007), est un problème de structure à quatre barres.

Description du problème

Il s'agit d'un problème à quatre objectifs au total organisés en trois disciplines ayant chacune deux objectifs. Les trois disciplines correspondent à trois scénarios de charge de la structure (figure 4.14). Pour chacun des scénarios i , l'objectif est de minimiser le volume V et le déplacement d_i . Des solutions de compromis doivent donc être trouvées pour que la structure supporte au mieux les contraintes dans ces trois configurations. Les objectifs sont les suivants :

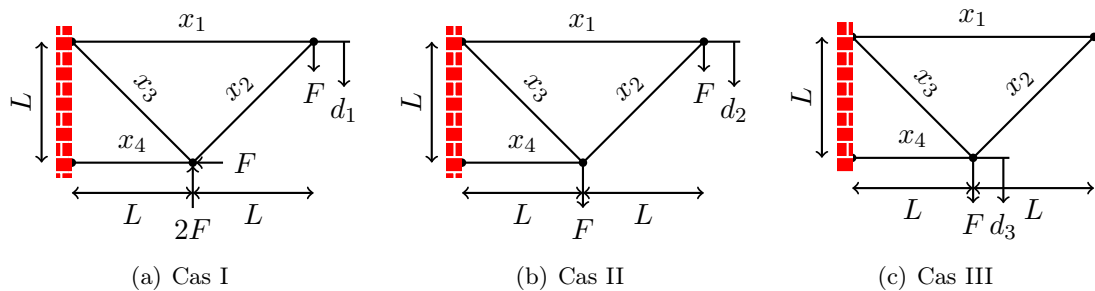


FIGURE 4.14 – Illustration du problème à quatre barres. Les trois cas correspondent à trois scénarios de charge du même problème.

8. Cf. le dernier paragraphe de la section 3.1.3 pour une présentation de leur méthode.

$$\begin{cases} V(x) = L(2x_1 + \sqrt{2}x_2 + \sqrt{2}x_3 + x_4) \\ d_1(x) = \frac{FL}{E} \left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4} \right) \\ d_2(x) = \frac{FL}{E} \left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} + \frac{4\sqrt{2}}{x_3} + \frac{6}{x_4} \right) \\ d_3(x) = \frac{FL}{E} \left(\frac{6\sqrt{2}}{x_3} + \frac{3}{x_4} \right) \end{cases} \quad (4.16)$$

sous les contraintes :

$$\begin{cases} x_1 \geq \frac{F}{\sigma} \\ x_2 \geq \sqrt{2} \frac{F}{\sigma} \\ x_3 \leq 3 \frac{F}{\sigma} \\ x_4 \leq 3 \frac{F}{\sigma} \end{cases} \quad (4.17)$$

avec : $F = 10\text{kN}$, $L = 200\text{cm}$, $E = 2.10^5\text{kN/cm}^2$ et $\sigma = 10\text{kN/cm}^2$. Le problème est décomposé en trois sous-problèmes de la façon suivante :

$$f_1 = (f_{1,1}, f_{1,2}) = (d_1, V) \quad (4.18)$$

$$f_2 = (f_{2,1}, f_{2,2}) = (d_2, V) \quad (4.19)$$

$$f_3 = (f_{3,1}, f_{3,2}) = (d_3, V) \quad (4.20)$$

Les auteurs n'ont pas spécifié les intervalles des variables. Nous avons pris les intervalles [1; 3] pour les quatre variables du problème.

Analyse du problème

Engau et Wiecek (2007) proposent, comme exemple, une solution de compromis pour ce problème (figure 4.15). La solution de compromis proposée est efficace pour deux des disciplines, mais n'est pas efficace pour la première. On peut cependant remarquer que, parmi les solutions admissibles, il en existe une qui est efficace pour chacune des disciplines simultanément (figure 4.16).

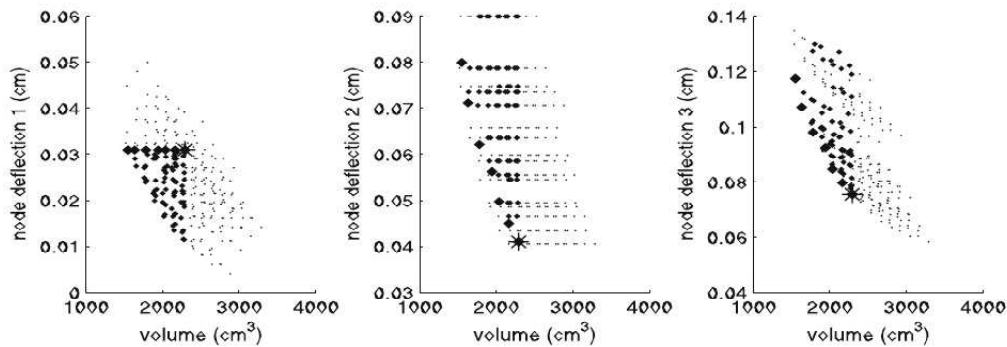


FIGURE 4.15 – La solution du problème 3 donnée en exemple par Engau et Wiecek (2007) (l'étoile) est efficace pour les disciplines 2 et 3, mais pas pour la 1.

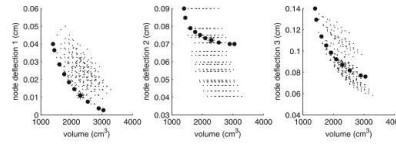


FIGURE 4.16 – Pour le problème 3, Engau et Wiecek (2007) observent que les solutions efficaces dans la discipline 1 ne sont pas nécessairement efficaces dans les deux autres disciplines. Les points plus foncés représentent les projections des solutions efficaces de la discipline 1. Dans ce problème, les solutions sont distribuées dans le même ordre dans les différents espaces des objectifs. Le point le plus à gauche de chaque espace des objectifs correspond à une unique solution. Cette solution est non dominée dans chacune des disciplines.

Les solutions trouvées avec la méthode de filtrage (Monte-Carlo avec 400 points) donnent un grand nombre de solutions de compromis \mathcal{C}_1 . Ces solutions recouvrent tout l'espace réalisable, la sélection d'une solution pertinente est difficile. Les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 donnent un ensemble plus restreint de solutions. Ces solutions sont similaires pour ces trois compromis. Les solutions de compromis varient légèrement suivant la répartition des points, mais se trouvent globalement dans la même zone (figure 4.17). Cette zone correspond, suivant la répartition des points, à une solution efficace dans chacune des disciplines simultanément.

Résultats

Il est difficile de vérifier la validité des solutions de compromis \mathcal{C}_1 , car elles recouvrent quasiment l'ensemble des solutions admissibles. On peut, en revanche, constater que les solutions trouvées par EMO-MDO sont proches de celles trouvées par la simulation de Monte-Carlo, et que la prise en compte de la distance phénotypique améliore la diversité des solutions (figure 4.18).

Les solutions de compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sont similaires dans les différentes configurations d'EMO-MDO. Elles sont cohérentes et dominent les solutions trouvées par filtrage (figure 4.19). Il s'agit de solutions simultanément efficaces pour chacune des disciplines.

4.2.5 Problème 4 : le réducteur de vitesse

Ce problème a pour la première fois été proposé par Golinski (1970), comme un problème mono-objectif. Il a ensuite été formulé en problème biobjectif et triobjectif par Azarm *et al.* (1989).

Description du problème

L'objectif est de minimiser la taille d'un réducteur de vitesse, tout en minimisant les contraintes élastiques dans les deux arbres (figure 4.20). Ce problème peut se décomposer en deux disciplines ayant chacune deux objectifs. Pour chaque sous-système, nous avons deux objectifs :

- $f_{i,1}$: minimiser le volume contenant le roulement de l'arbre i ;
- $f_{i,2}$: minimiser les contraintes élastiques de l'arbre i .

L'objectif global étant de minimiser le volume total de la boîte, soit $f_{1,1}(x) + f_{2,1}(x)$. Les variables de conception du problème sont au nombre de sept, se décomposant en trois parties :

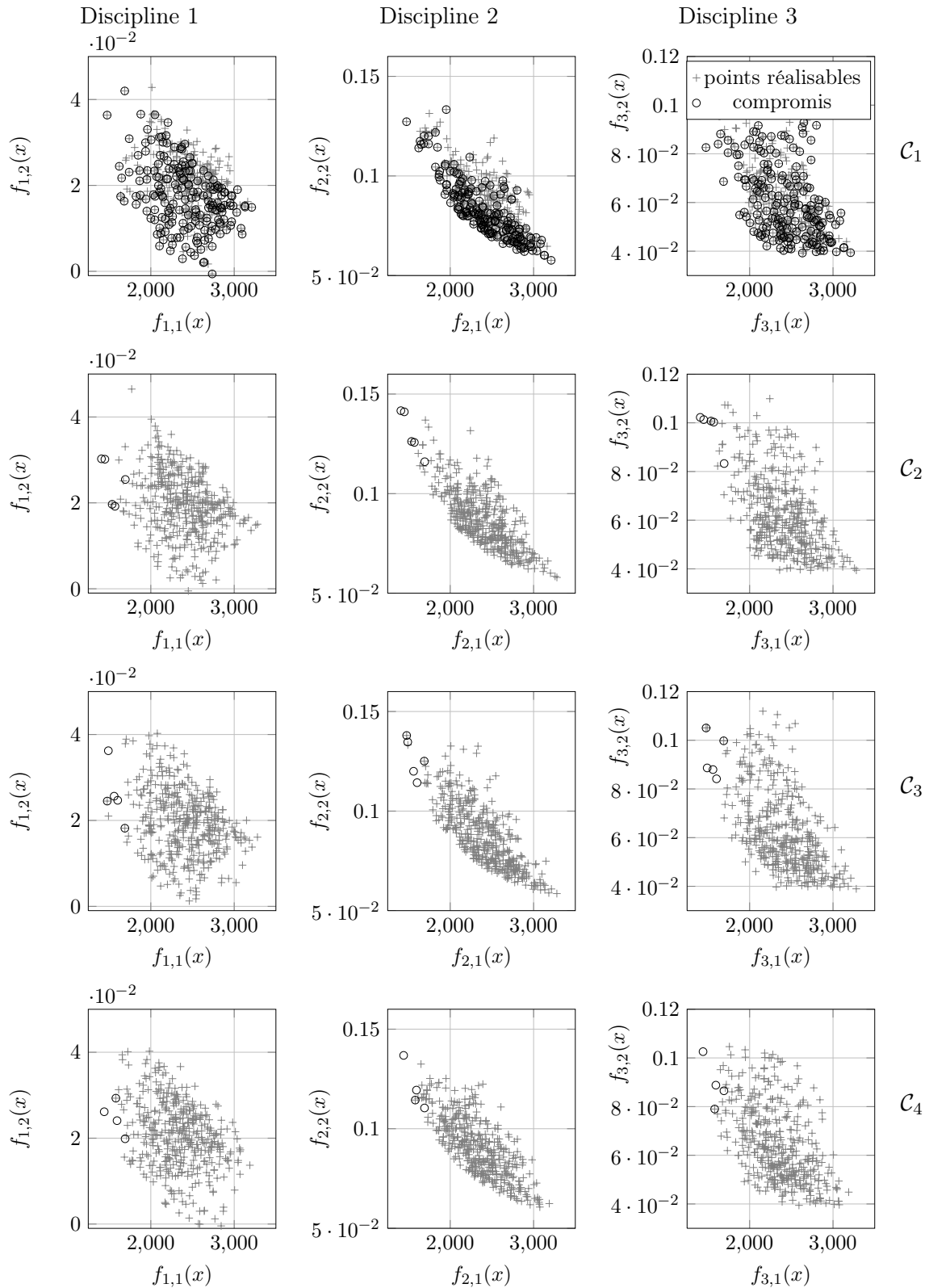


FIGURE 4.17 – Résultats obtenus sur l'exemple 3 par simulation de Monte-Carlo sur les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 avec 400 points. Chaque test est ici réalisé avec des ensembles admissibles différents. Les solutions trouvées pour les trois séries de tests successifs sont superposées pour les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 .

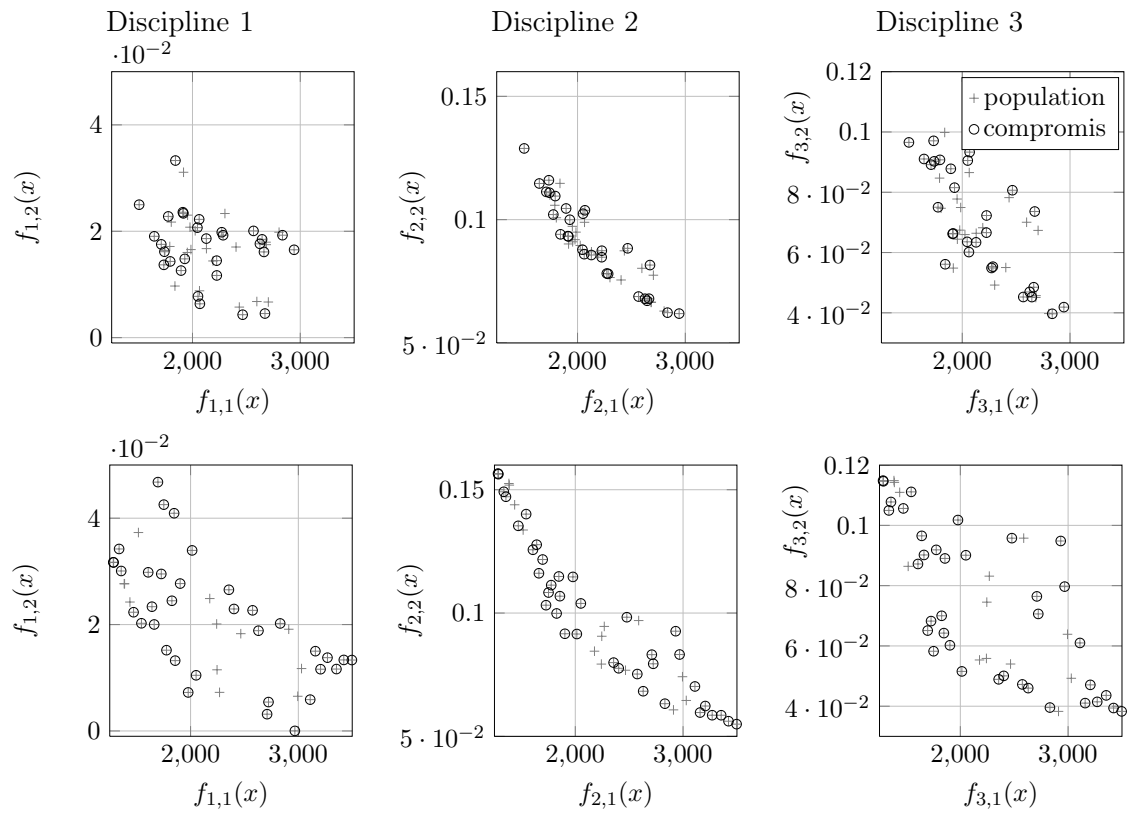


FIGURE 4.18 – Problème 3 : des solutions trouvées avec la configuration $c1_n_50x500$ (en haut) et avec la configuration $c1_o_50x500$ (en bas). Les résultats sont similaires, bien que la dispersion des points soit meilleure avec la prise en compte de la distance phénotypique.

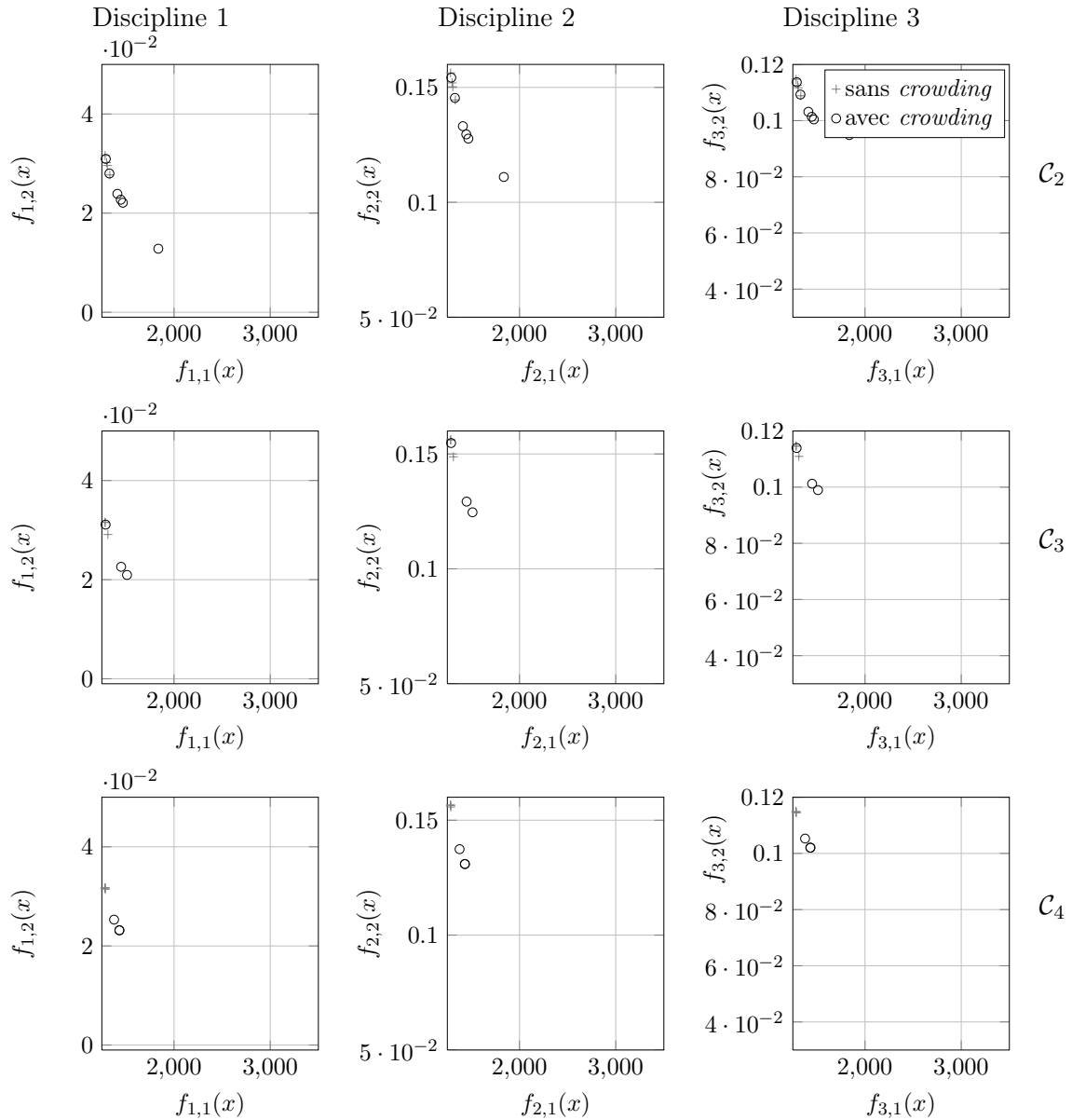
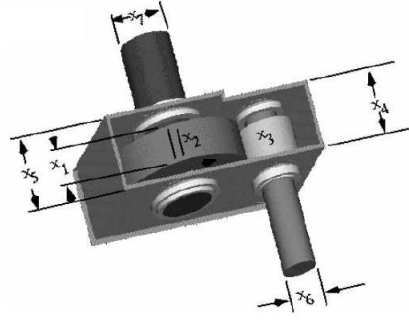


FIGURE 4.19 – Solutions de compromis du problème 3 avec respectivement \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 . Pour chacun de ces compromis, nous comparons les solutions trouvées dans la version avec et sans prise en compte de la distance phénotypique. Ces solutions dominent celles obtenues par filtrage dans toutes les disciplines (figure 4.17).


 FIGURE 4.20 – Représentation tridimensionnelle du réducteur de vitesse d'après www.nasa.gov.

1. Les variables communes $x^c = (x_1^c, x_2^c, x_3^c)$ qui concernent les roues dentées :
 - x_1^c : l'épaisseur de la denture ;
 - x_2^c : le module des dents des roues dentées ;
 - x_3^c : le nombre de dents du pignon.
2. Les variables propres au sous-système 1, $x_1^d = (x_{1,1}^d, x_{1,2}^d)$:
 - $x_{1,1}^d$: distance entre les roulements du sous-système 1 ;
 - $x_{1,2}^d$: diamètre de l'arbre 1.
3. Les variables propres au sous-système 2, $x_2^d = (x_{2,1}^d, x_{2,2}^d)$:
 - $x_{2,1}^d$: distance entre les roulements du sous-système 2 ;
 - $x_{2,2}^d$: diamètre de l'arbre 2.

Les domaines des variables sont les suivants :

$$\left\{ \begin{array}{l} x_1^c \in [2,6; 3,6] \\ x_2^c \in [0,7; 0,8] \\ x_3^c \in [17; 28] \\ x_{1,1}^d \in [7,3; 8,3] \\ x_{1,2}^d \in [2,9; 3,9] \\ x_{2,1}^d \in [7,3; 8,3] \\ x_{2,2}^d \in [5; 5,5] \end{array} \right. \quad (4.21)$$

Remarque 4.2 La variable x_3^c est, normalement, entière. Cependant, pour simplifier le problème, nous la considérons continue. Cela n'a pas d'incidence sur ce que nous voulons montrer.

Les fonctions objectif du problème sont les suivantes :

$$D_1 \left\{ \begin{array}{l} f_{1,1}(x) = 0,7854x_1^c(x_2^c)^2 \left(\frac{10(x_3^c)^2}{3} + 14,933x_3^c - 43,0934 \right) \\ \quad - 1,508x_1^c(x_{1,2}^d)^2 + 7,477(x_{1,2}^d)^3 + 0,7854x_{1,1}^d(x_{1,2}^d)^2 \\ f_{1,2}(x) = \frac{\sqrt{\left(\frac{745(x_{1,1}^d)^2}{x_2^c x_3^c} \right)^2 + 1,69 \times 10^7}}{0,1(x_{1,2}^d)^3} \end{array} \right. \quad (4.22)$$

$$D_2 \begin{cases} f_{2,1}(x) = -1,508x_1^c(x_{2,2}^d)^2 + 7,477(x_{2,2}^d)^3 + 0,7854x_{2,1}^d(x_{2,2}^d)^2 \\ f_{2,2}(x) = \frac{\sqrt{\left(\frac{745x_{2,1}^d}{x_2^c x_3^c}\right)^2 + 1,575 \times 10^8}}{0,1(x_{2,2}^d)^3} \end{cases} \quad (4.23)$$

Le problème contient onze contraintes : trois contraintes propres à chaque sous-système (ne dépendant que des variables locales) et cinq contraintes sur les variables communes. Les contraintes communes sont :

$$g_{s1}(x) : \frac{1}{x_1^c(x_2^c)^2 x_3^c} - \frac{1}{27} \leq 0 \quad (4.24)$$

$$g_{s2}(x) : \frac{1}{x_1^c(x_2^c)^2(x_3^c)^2} - \frac{1}{397,5} \leq 0 \quad (4.25)$$

$$g_{s3}(x) : x_2^c x_3^c - 40 \leq 0 \quad (4.26)$$

$$g_{s4}(x) : \frac{x_1^c}{x_2^c} - 12 \leq 0 \quad (4.27)$$

$$g_{s5}(x) : 5 - \frac{x_1^c}{x_2^c} \quad (4.28)$$

Les contraintes de la discipline D_1 sont :

$$g_{1,1}(x) : \frac{(x_{1,1}^d)^3}{x_2^c x_3^c (x_{1,2}^d)^4} - \frac{1}{1,93} \quad (4.29)$$

$$g_{1,2}(x) : 1,9 - x_{1,1}^d + 1,5x_{1,2}^d \leq 0 \quad (4.30)$$

$$g_{1,3}(x) : f_{1,2} - 1300 \leq 0 \quad (4.31)$$

Les contraintes de la discipline D_2 sont :

$$g_{2,1}(x) : \frac{(x_{2,1}^d)^3}{x_2^c x_3^c (x_{2,2}^d)^4} - \frac{1}{1,93} \quad (4.32)$$

$$g_{2,2}(x) : 1,9 - x_{2,1}^d + 1,1x_{2,2}^d \leq 0 \quad (4.33)$$

$$g_{2,3}(x) : f_{1,2} - 1100 \leq 0 \quad (4.34)$$

Analyse du problème

Gunawan, Farhang-Mehr, et Azarm (2004) comparent les résultats obtenus sur ce problème avec EM-MOGA et MOGA. Ils agrègent les fonctions $f_{1,1}$ et $f_{2,1}$ en une unique fonction F_1 représentant le volume total du réducteur de vitesse (figure 4.21). Les solutions présentées contiennent peu de solutions dominées dans la première discipline, alors qu'elles semblent aléatoirement distribuées dans l'espace réalisable de la seconde.

Nous constatons que, suivant les contraintes, certaines valeurs des domaines des variables ne sont pas admissibles. Ces intervalles peuvent ainsi être réduits, comme le montre l'exemple 4.2.

Exemple 4.2 Prenons l'exemple de $x_1^c \in [2,6; 3,6]$. D'après la contrainte g_{s5} , nous avons :

$$\begin{aligned} 5 - \frac{x_1^c}{x_2^c} &\leq 0 \\ \Leftrightarrow x_1^c &\geq 5x_2^c \end{aligned}$$

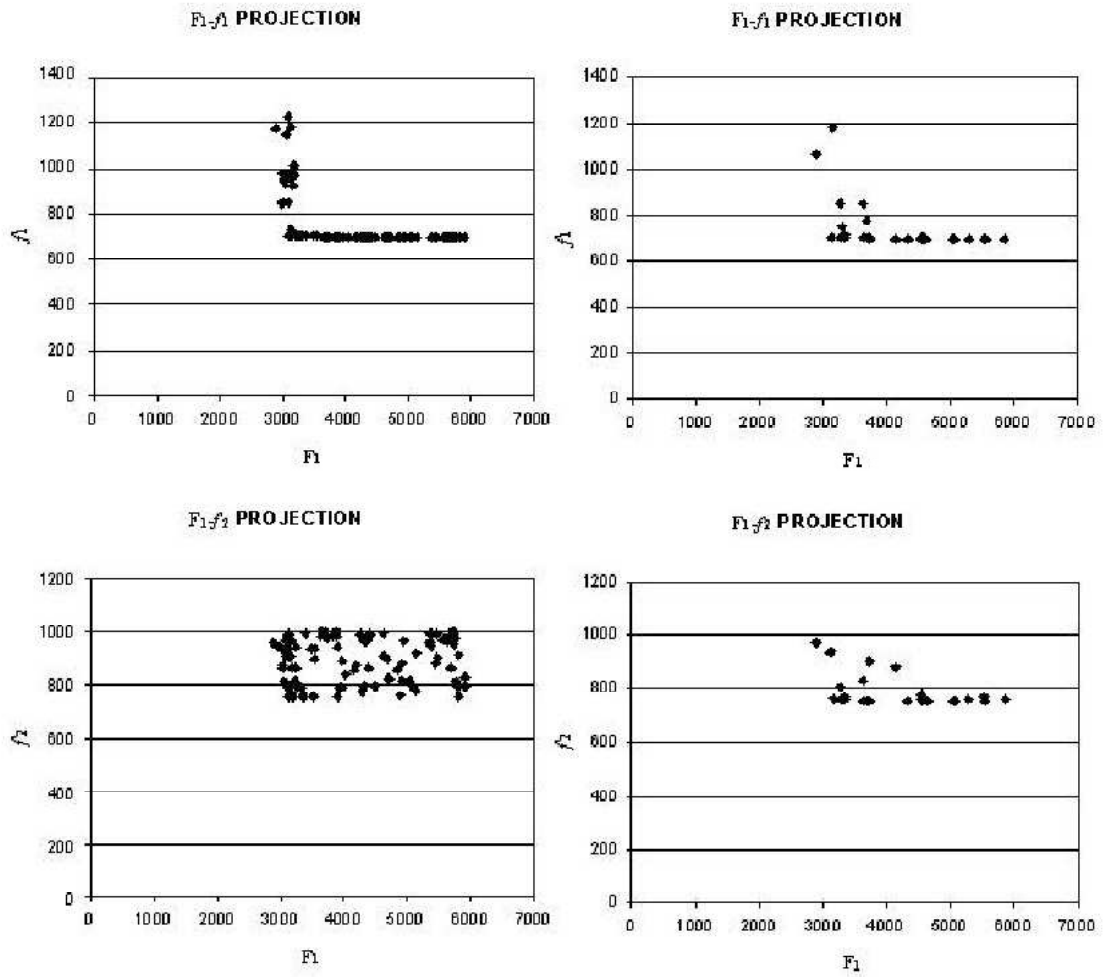


FIGURE 4.21 – Solutions obtenues par Gunawan et al. (2004) comparant les solutions générées par EM-MOGA (à gauche) et MOGA (à droite).

Var	initial	réduit	gain
x_1^c	[2,6; 3,6]	[3,5; 3,6]	90%
x_2^c	[0,7; 0,8]	[0,7; 0,72]	80%
x_3^c	[17; 28]	[17; 28]	0%
$x_{1,1}^d$	[7,3; 8,3]	[7,3; 8,3]	0%
$x_{1,2}^d$	[2,9; 3,9]	[3,164543146138814; 3,9]	26,45%
$x_{2,1}^d$	[7,3; 8,3]	[7,3; 8,3]	0%
$x_{2,2}^d$	[5; 5,5]	[5; 5,5]	0%

TABLEAU 4.3 – Domaines réduits du problème 4 en utilisant RealPaver (Granvilliers et Benhamou 2006).

Or :

$$x_2^c \in [0,7; 0,8]$$

Donc :

$$\begin{aligned} x_1^c &\geq 5 \times [0,7; 0,8] \\ \Leftrightarrow x_1^c &\geq 5 \times \inf([0,7; 0,8]) \\ \Leftrightarrow x_1^c &\geq 5 \times 0,7 \\ \Leftrightarrow x_1^c &\geq 3,5 \end{aligned}$$

On a alors :

$$\begin{aligned} x_1^c &\in [3,5; +\infty] \cap [2,6; 3,6] \\ \Leftrightarrow x_1^c &\in [3,5; 3,6] \end{aligned}$$

En appliquant la réduction des domaines de toutes les variables de conception avec toutes les contraintes grâce à *RealPaver* (Granvilliers et Benhamou 2006), nous obtenons les nouveaux domaines présentés tableau 4.3. Ce sont ces domaines, plus précis, que nous utiliserons pour générer des solutions admissibles.

Les solutions générées par filtrage (simulations de Monte-Carlo sur 400 solutions admissibles, figure 4.24) indiquent l'existence d'une corrélation entre les objectifs $f_{2,1}$ et $f_{2,2}$: plus le volume du deuxième axe est élevé, moins les contraintes élastiques seront importantes et inversement. En revanche, il semble qu'il n'y ait aucune corrélation entre les objectifs $f_{1,1}$ et $f_{2,2}$.

Les solutions de compromis \mathcal{C}_1 (figure 4.23) devraient, en principe, ressembler aux solutions présentées figure 4.21. Gunawan, Farhang-Mehr, et Azarm (2004) trouvent cependant une répartition des solutions moins dispersée dans la première discipline (il y a moins de solutions dominées pour cette discipline). Ceci peut s'expliquer par le fait que les méthodes basées sur les algorithmes évolutionnaires multiobjectifs telles qu'EM-MGA ou COSMOS (dont les optimisations disciplinaires sont réalisées indépendamment par des algorithmes évolutionnaires multiobjectifs distincts) ne permettent pas de converger vers l'ensemble des solutions efficaces du problème global. En effet, les optimisations disciplinaires biaisent la recherche de solutions globales.

Résultats

Les solutions obtenues pour le compromis \mathcal{C}_1 sont aléatoirement distribuées dans l'espace de décision. Les solutions obtenues avec les autres compromis sont, quant à elles, similaires et donnent de meilleurs résultats pour la première discipline (des solutions non-

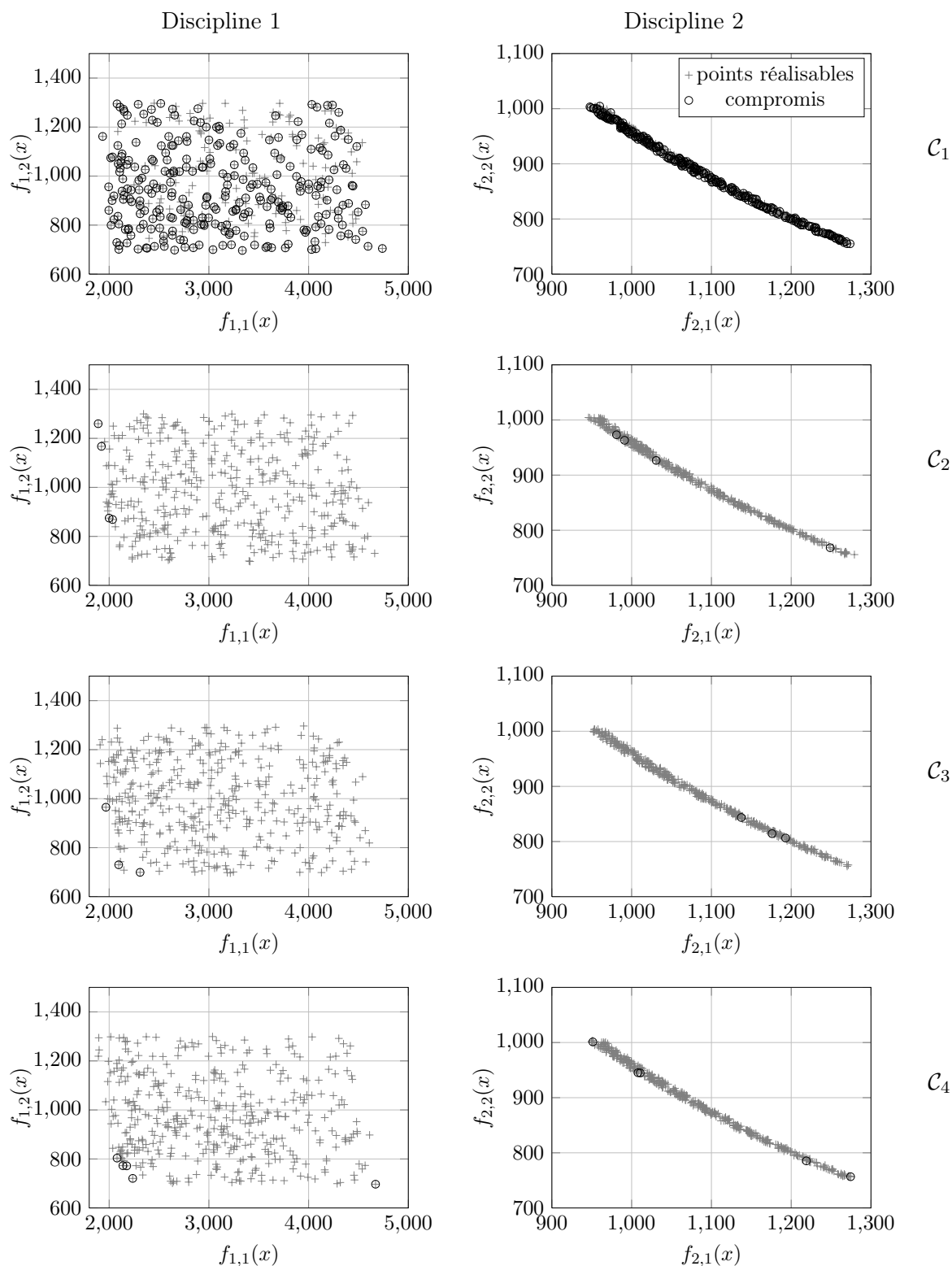


FIGURE 4.22 – Solutions du problème 4 pour les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 obtenues par simulation de Monte-Carlo avec 400 points. Les projections sont les espaces des objectifs des disciplines.

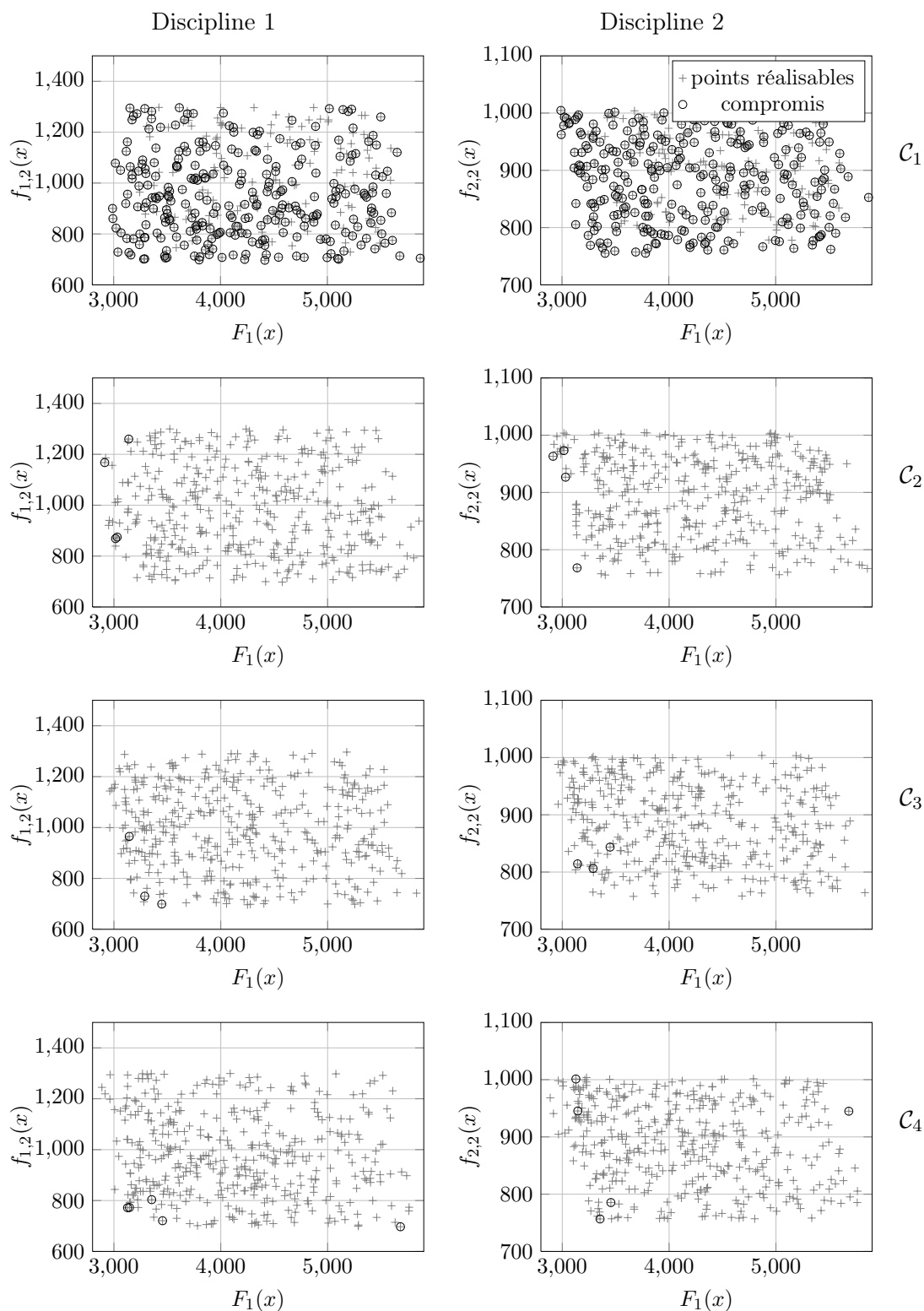


FIGURE 4.23 – Solutions du problème 4 pour les compromis C_1 , C_2 , C_3 et C_4 obtenues par simulation de Monte-Carlo avec 400 points. Les projections sont les mêmes que celles utilisées par Gunawan et al. (2004) (figure 4.21).

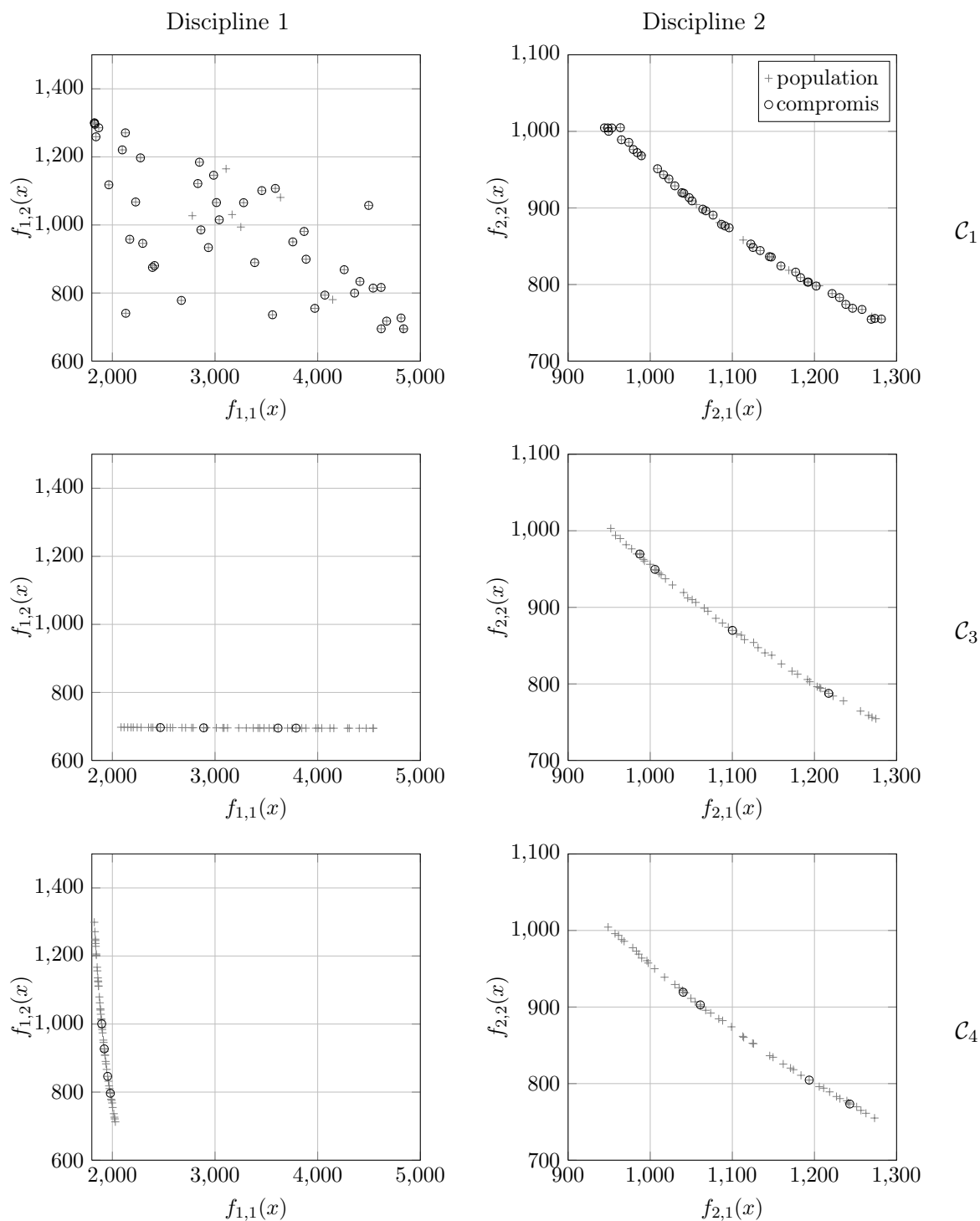


FIGURE 4.24 – Solutions générées pour le problème 4 avec les configurations $c1_o_50x500$, $c3_o_50x500$, $c4_o_50x500$. Les résultats obtenus avec les compromis C_2 , C_3 et C_4 sont similaires, mais les solutions obtenues par différentes simulations peuvent différer, comme l'illustrent les résultats obtenus par C_3 et C_4 sur cette simulation (nous avons obtenu des résultats inverses sur d'autres simulations, Cf. figure 4.25).

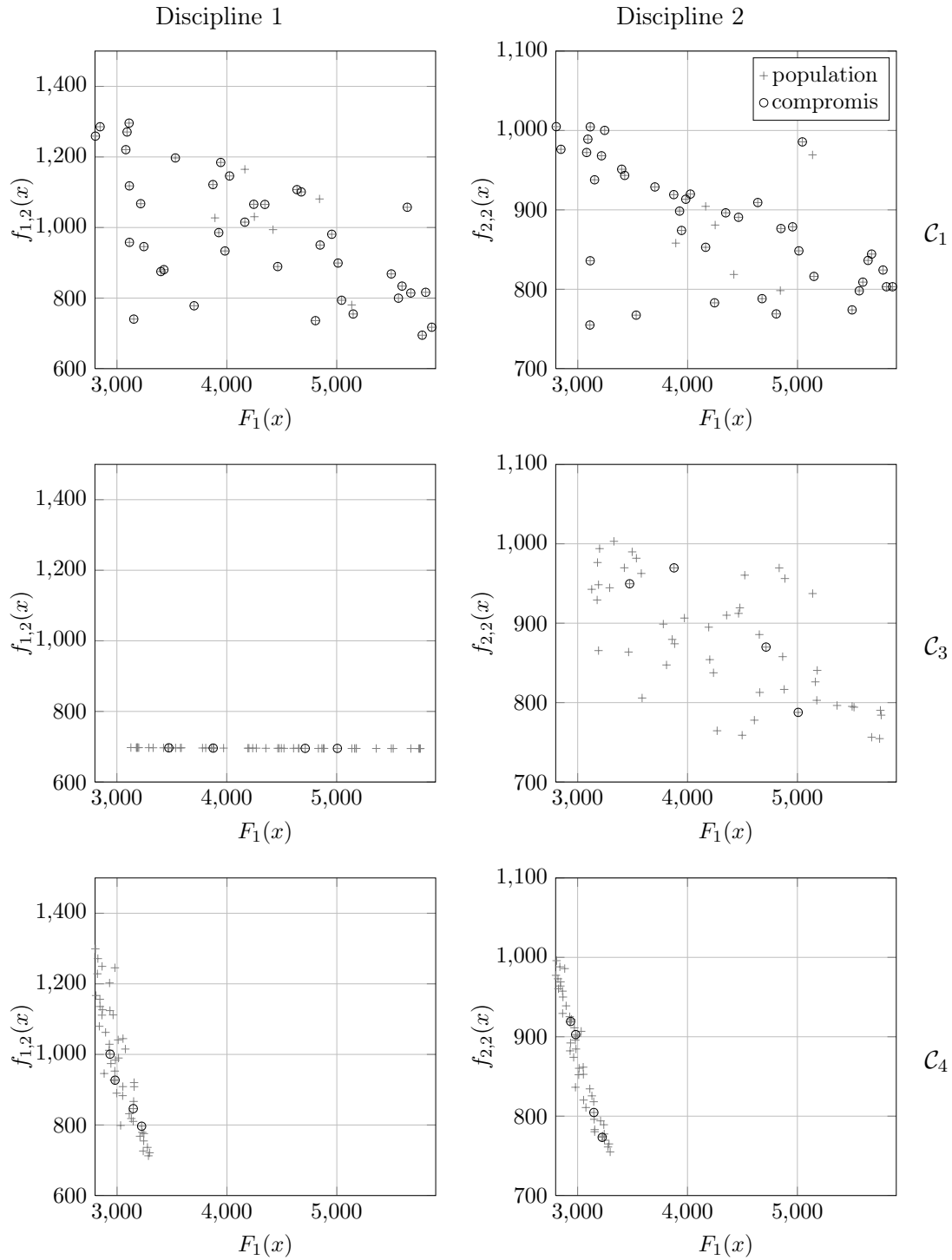


FIGURE 4.25 – Solutions générées pour le problème 4 avec les configurations $c1_o_50x500$, $c3_o_50x500$, $c4_o_50x500$. Il s'agit des mêmes résultats que ceux présentés figure 4.24, mais projetés selon les mêmes axes que Gunawan et al. (2004) (figure 4.21).

dominées dans la première discipline) (figure 4.24). Tous ces résultats sont conformes aux résultats obtenus par filtrage (Monte-Carlo).

Le front de Pareto de la première discipline est formé de deux parties : une verticale et une horizontale. L'algorithme converge globalement vers l'une ou l'autre de ces parties avec les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 . Sur la figure 4.24, la version \mathcal{C}_3 a convergé vers la partie horizontale tandis que \mathcal{C}_4 a convergé vers la partie verticale. Sur d'autres essais, l'inverse se produit (figure 4.25). Ces deux scénarios de convergence donnent des résultats différents sur les projections $F_1/f_{1,2}$ et $F_2/f_{2,2}$ (figure 4.25). Ces différences ne sont cependant pas dépendantes du compromis utilisé parmi \mathcal{C}_2 , \mathcal{C}_3 ou \mathcal{C}_4 .

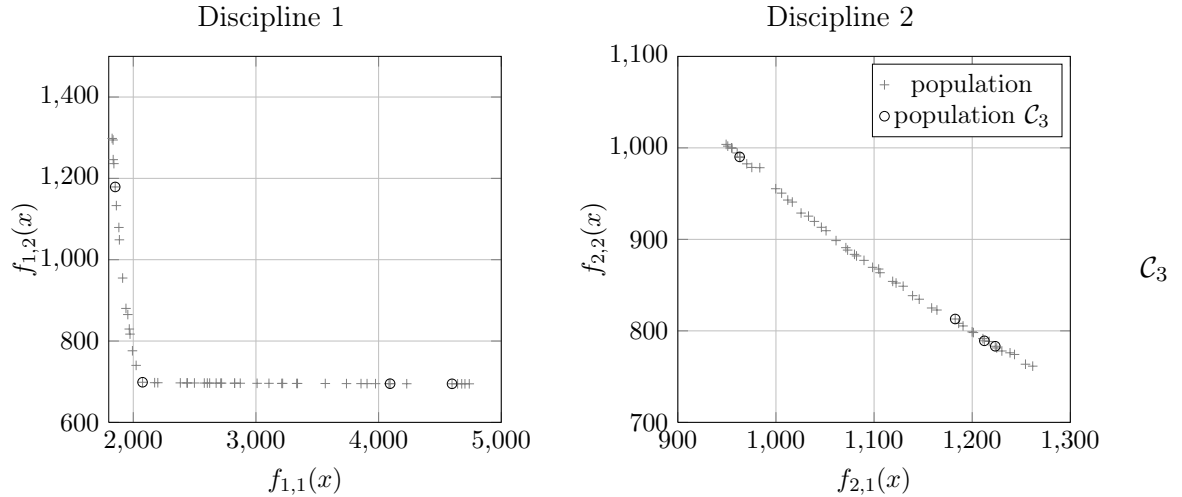


FIGURE 4.26 – Solutions générées pour le problème 4 avec la configuration $c3_o_50x50$. Les résultats sont similaires pour les compromis \mathcal{C}_2 et \mathcal{C}_3 , avec ou sans prise en compte de la diversité phénotypique.

On peut remarquer qu'avec un nombre d'itérations moindre (50 itérations, plutôt que 500), les individus sont mieux répartis sur le front de Pareto de la première discipline (figure 4.26).

L'obtention de ces deux scénarios, ainsi que la meilleure répartition des individus sur le front de Pareto de la première discipline avec moins d'itérations, suggèrent d'une part la présence de deux niches de solutions dans le problème (annexe E), et d'autre part le peu d'efficacité de la diversité phénotypique pour ce type de problème. Cet effet peut être atténué en utilisant une mesure de distance phénotypique adaptée et/ou une méthode de nichage.

4.2.6 Synthèse des résultats

Les différents tests menés nous ont permis de vérifier les prédictions théoriques des compromis, ainsi que d'étudier le comportement d'EMO-MDO sur différents types de problèmes avec les différents compromis. Nous avons fait les constatations suivantes :

- Lorsqu'il existe un ensemble de solutions efficaces communes à toutes les disciplines simultanément, alors elles sont des solutions de compromis \mathcal{C}_2 et \mathcal{C}_3 .
- Cette dernière assertion n'est théoriquement pas valable dans le cas général pour le compromis \mathcal{C}_4 . Elle est néanmoins valable expérimentalement pour les problèmes 1 et 3.

- Si de telles solutions n’existent pas, les performances des solutions semblent équitablement réparties entre les disciplines. En ce sens, les performances des solutions trouvées sont équilibrées dans les disciplines.
- Les solutions générées par les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sont plus concentrées dans des sous-espaces de l’espace des objectifs que les solutions de compromis \mathcal{C}_1 .
- Les solutions de compromis \mathcal{C}_4 sont moins sensibles à la distribution de la population que les solutions de compromis \mathcal{C}_2 et \mathcal{C}_3 : les solutions sont cohérentes avec les méthodes de filtrage et dans les différentes configurations de l’algorithme évolutionnaire.
- La prise en compte de la distance phénotypique permet effectivement une meilleure distribution des solutions. Cette distribution n’est cependant pas adaptée aux problèmes d’optimisation multidisciplinaire.

Conclusion

L’algorithme EMO-MDO, présenté dans ce chapitre, est une adaptation de l’algorithme génétique multiobjectif NSGA-II aux problèmes d’optimisation multidisciplinaire. Son calcul du rang des solutions basé sur des préordres quelconques permet d’y adapter différents types de compromis pouvant modéliser des préférences des disciplines, et en particulier les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 .

Les analyses expérimentales menées sur EMO-MDO confirment les propriétés théoriques évoquées au chapitre précédent. Il en ressort que les compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 donnent des solutions plus pertinentes que le compromis \mathcal{C}_1 . Le compromis \mathcal{C}_4 est cependant le plus adapté à l’utilisation dans un algorithme évolutionnaire multiobjectif.

La prise en compte de la distance phénotypique apporte de la diversité aux solutions. Cependant, cette diversité peut être améliorée en tenant compte des spécificités du problème, c’est-à-dire en tenant compte des distances phénotypiques propres à chaque discipline. Il peut par exemple s’agir d’une agrégation de distances phénotypiques calculées indépendamment dans chaque discipline.

Les performances d’EMO-MDO (vitesse de convergence, qualité des résultats, diversité des solutions. . .) n’ont pas été évaluées avec des mesures précises. En effet, de telles mesures manquent à ce type de problème.

La prise en compte des variables locales dans EMO-MDO pourrait accélérer la convergence de l’algorithme en générant des enfants plus pertinents dans les cas où les problèmes comportent des variables locales (voir annexe D).

CONCLUSION

Nous nous sommes intéressés à la recherche de solutions de compromis dans les problèmes d'optimisation multidisciplinaire à l'aide d'algorithmes évolutionnaires. L'analyse des méthodes d'optimisation multidisciplinaire de la littérature nous a montré que les méthodes existantes n'étaient pas adaptées au cas où chaque discipline résout un problème d'optimisation multiobjectif qui lui est propre. Ces méthodes traitent soit le problème d'optimisation multidisciplinaire comme un problème d'optimisation dont l'objectif est global au système (méthodes basées sur des reformulations mathématiques), soit comme la recherche de solutions d'équilibre entre des disciplines n'ayant qu'un objectif chacune (méthodes basées sur la théorie des jeux), soit comme des problèmes d'optimisation multiobjectif ne prenant pas en compte les préférences disciplinaires (méthodes basées sur les algorithmes évolutionnaires multiobjectifs). En effet, les solutions générées correspondent à des solutions de compromis entre les objectifs, mais pas nécessairement à des solutions de compromis entre les disciplines.

La définition de compromis, et donc des préférences entre les solutions dans un problème d'optimisation, se modélise par les relations d'ordre existantes dans l'ensemble des solutions. Nous avons alors défini formellement quatre structures ordonnées représentant autant de structures de préférences dont les minimums représentent respectivement quatre compromis possibles. Ces quatre compromis sont notés \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 . Le compromis \mathcal{C}_1 correspond aux solutions efficaces du problème d'optimisation multiobjectif constitué de l'ensemble des objectifs du problème. Ce compromis n'est, selon nous, pas adapté au cas multidisciplinaire, car il ne prend pas en compte les préférences des disciplines : des solutions efficaces simultanément dans toutes les disciplines seront considérées aussi bonnes que certaines solutions localement dominées dans une discipline. Le problème est que le produit d'ordre, utilisé pour agréger les préférences disciplinaires, ne prend pas en compte les préférences disciplinaires qui ne sont pas explicitement modélisées en terme de relation d'ordre (une solution efficace est meilleure qu'une solution qui n'est pas efficace). Pour pallier ce problème, le compromis \mathcal{C}_2 étend, dans chaque discipline, la relation définie dans le compromis \mathcal{C}_1 . Les préférences disciplinaires sont ainsi ajoutées : les solutions efficaces d'une discipline se valent toutes *a priori*, et une solution efficace dans une discipline est meilleure qu'une solution qui n'est pas efficace. Le compromis \mathcal{C}_3 affine ces préférences en ne distinguant pas seulement les solutions efficaces des solutions non efficaces, mais en partitionnant les solutions suivant leur rang. Dans une discipline, une solution de rang inférieur à une autre est considérée meilleure. Bien que les compromis \mathcal{C}_2 et \mathcal{C}_3 aient d'intéressantes propriétés théoriques, ils ont aussi des inconvénients pour leur implémentation dans un algorithme évolutionnaire. L'ajout de préférences disciplinaires rend la relation d'ordre, et donc le compromis, dépendants de la répartition des solutions dans l'espace des objectifs. Le compromis \mathcal{C}_4 aborde le

problème différemment : il n'étend pas les relations d'ordre des disciplines. La définition du compromis n'est ainsi pas dépendante de la répartition des solutions dans l'espace des objectifs. C'est au niveau du produit des ordres disciplinaires que les préférences entre les solutions disciplinaires sont prises en compte. La relation de dominance de Pareto est ainsi modifiée : une solution a domine une solution b si a domine b dans au moins une discipline et si b ne domine a dans aucune discipline.

Les compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 ont été implémentés dans un algorithme évolutionnaire multiobjectif appelé EMO-MDO⁹. Cet algorithme s'inspire de l'algorithme NSGA-II dont il reprend les principes et les différents opérateurs génétiques. Sa particularité réside dans la procédure de calcul des rangs des solutions plus générique que celle de NSGA-II puisqu'elle est applicable à tout préordre. EMO-MDO a pour objectif de valider expérimentalement les propriétés théoriques des différents compromis et leur applicabilité au sein d'un algorithme évolutionnaire. L'algorithme a été évalué sur cinq problèmes de la littérature dans différentes configurations. Les résultats obtenus sont conformes aux prédictions théoriques :

- les solutions de compromis \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sont des solutions efficaces du problème ;
- s'il existe des solutions simultanément efficaces dans toutes les disciplines, alors ce sont exactement les solutions de compromis \mathcal{C}_2 et \mathcal{C}_3 ;
- l'utilisation de \mathcal{C}_2 et \mathcal{C}_3 dans EMO-MDO ne donne pas les mêmes résultats que par filtrage. En revanche, les résultats avec \mathcal{C}_4 sont similaires par filtrage et par EMO-MDO ;

Plus généralement, l'intégration de préférences disciplinaires permet de générer des solutions plus pertinentes pour le concepteur. De plus, l'approche adoptée pour la modélisation des compromis peut être adaptée à l'intégration d'autres types de préférences spécifiques à chaque discipline.

Nous observons également que la prise en compte de la diversité phénotypique dans la sélection des individus améliore la dispersion des solutions dans l'espace des objectifs. La diversité des solutions peut cependant être améliorée en prenant en compte les spécificités du problème d'optimisation multidisciplinaire. En effet, la diversité considérée est la diversité des solutions sur le front de Pareto global du problème. Les projections de ces solutions dans les espaces des objectifs des disciplines ne rendent pas compte de cette diversité. Or, c'est dans les disciplines que les décisions sont prises et c'est dans les espaces des objectifs des disciplines que les solutions doivent être diversifiées.

La diversité phénotypique dans un problème d'optimisation multidisciplinaire reste à définir. Il faut pour cela agréger les diversités phénotypiques de chaque discipline. Les distances phénotypiques ne sont également pas prises en compte dans le calcul de la relation d'ordre comme l'est, par exemple, la violation des contraintes. Sa prise en compte dans la relation d'ordre permettrait une plus grande cohérence dans la formulation du problème, puisque la distance phénotypique entre en jeu dans la comparaison des solutions. Une première perspective d'amélioration de l'algorithme EMO-MDO consisterait à intégrer une mesure de la diversité phénotypique adaptée à l'optimisation multidisciplinaire directement dans la relation d'ordre.

Une mesure de la diversité phénotypique adaptée aux problèmes d'optimisation multidisciplinaire permettrait également de fournir une mesure de la la qualité des solutions de compromis obtenues. Il manque en effet des mesures de la qualité des solutions, telles qu'elles existent pour les algorithmes évolutionnaires multiobjectifs, mesurant à la fois la

9. Evolutionary Multiple Objective & Multidisciplinary Design Optimisation

proximité des solutions trouvées avec les meilleures solutions théoriques, et la diversité des solutions (voir section 8.2 de Deb 2001). L'évaluation de la qualité des compromis est cependant difficile, car il s'agit en partie d'appréciations pratiques dépendantes du problème traité. La mise en place d'une méthodologie impliquant le concepteur serait alors nécessaire. L'évaluation des compromis passe également par la mise en place de problèmes de tests standardisés servant de références aux comparaisons. Les solutions de référence serviraient aussi à la comparaison des algorithmes d'optimisation. Une autre perspective de notre travail consiste en la mise en place de procédures d'évaluation objectives des solutions obtenues.

Une seconde perspective d'amélioration d'EMO-MDO porte sur l'efficacité des algorithmes utilisés. Ils peuvent être améliorés en prenant en compte les différentes propriétés des compromis. Les propositions 3.2 et 3.5 peuvent être exploitées pour limiter la recherche de solutions de compromis dans les solutions efficaces des disciplines quand une solution est trouvée. Les propositions 3.3 et 3.6 permettent d'éliminer rapidement des solutions non pertinentes, en connaissant les solutions efficaces des disciplines. La prise en compte des variables locales dans EMO-MDO pourrait également accélérer la convergence de l'algorithme, en générant des enfants plus pertinents dans les cas où les problèmes comportent des variables locales (annexe D).

La définition des compromis fournit une méthode d'évaluation des solutions permettant de les comparer et offre ainsi une mesure d'adaptation utilisable dans un algorithme évolutionnaire. Elle tient compte de la structure des problèmes multidisciplinaires dans la définition des solutions optimales, mais ne fournit pas de méthode de résolution d'optimisation multidisciplinaire offrant, à chaque discipline, la liberté de réaliser ses optimisations indépendamment et de résoudre les couplages. EMO-MDO réalise en effet une optimisation globale et n'inclut pas de mécanisme de résolution des couplages. Il doit pour cela être utilisé conjointement avec une méthode de type MDF, IDF ou AAO pour laquelle il peut servir d'optimiseur. Notre objectif est, à terme, de disposer d'une méthode unifiée incluant la résolution des couplages et offrant plus d'autonomie aux disciplines pour les optimisations locales.

Relations : définitions et notations

Nous considérons un ensemble E fini dénombrable.

Définition A.1 (Relation binaire) Une relation binaire \mathcal{R} sur un ensemble E est un sous-ensemble de $E \times E$. On dit que a est en relation avec b , et on écrit $a\mathcal{R}b$ si et seulement si $(a, b) \in \mathcal{R}$, et $a \not\mathcal{R}b$ si et seulement si $(a, b) \notin \mathcal{R}$.

Définition A.2 (Prédécesseur et successeur direct) Soit \mathcal{R} une relation binaire sur un ensemble E . Si $(a, b) \in \mathcal{R}$, on dit que a est un prédécesseur direct de b et b est un successeur direct de a dans \mathcal{R} .

Prenons un exemple que nous allons suivre tout au long de cette annexe :

Exemple A.1 Soit $\mathcal{Q} = \{(a, b), (b, b), (b, g), (d, e), (e, f), (e, g), (f, d), (f, f), (g, b)\}$ une relation sur l'ensemble $S = \{a, b, c, d, e, f, g\}$.

Dans (S, \mathcal{Q}) , d est un prédécesseur direct de e alors que b est à la fois prédécesseur direct et successeur direct de g .

Définition A.3 Une relation binaire \mathcal{R} sur un ensemble de base E est dite

- réflexive si $\forall a \in E, a\mathcal{R}a$
- symétrique si $\forall a, b \in E, a\mathcal{R}b \Rightarrow b\mathcal{R}a$
- antisymétrique si $\forall a, b \in E, a\mathcal{R}b \wedge b\mathcal{R}a \Rightarrow (a = b)$
- transitive si $\forall a, b, c \in E, a\mathcal{R}b \wedge b\mathcal{R}c \Rightarrow a\mathcal{R}c$
- totale si $\forall a, b \in E, a\mathcal{R}b \vee b\mathcal{R}a$

On peut noter que (S, \mathcal{Q}) n'est ni réflexive, ni symétrique, ni antisymétrique, ni transitive ni totale. Toutefois, une relation réflexive ou transitive peut être obtenue par fermeture.

Commençons par définir les puissances d'une relation correspondant non pas au produit cartésien, mais à l'union.

Définition A.4 Soit \mathcal{R} une relation binaire sur un ensemble de base E . Nous définissons récursivement la relation \mathcal{R}^i telle que

- $\mathcal{R}^0 := \{(x, x) \in E \times E\}$

$$- \mathcal{R}^i := \mathcal{R}^{i-1} \cup \{(a, c) \in E \times E : (a, b) \in \mathcal{R}^{i-1} \wedge (b, c) \in \mathcal{R}^{i-1}\}$$

Définition A.5 (Fermeture réflexive) *Soit \mathcal{R} une relation binaire sur un ensemble de base E . La fermeture réflexive de \mathcal{R} est \mathcal{R}^1 soit $\mathcal{R} \cup \mathcal{R}^0$.*

Définition A.6 (Fermeture transitive) *Soit \mathcal{R} une relation binaire sur un ensemble de base E . La fermeture transitive \mathcal{R}^+ de \mathcal{R} est définie par :*

$$\mathcal{R}^+ := \bigcup_{i \in \mathbb{N}^*}$$

Nous avons $\mathcal{Q}^1 = \{(a, a), (a, b), (b, b), (b, g), (c, c), (d, d), (d, e), (e, e), (e, f), (e, g), (f, f), (f, d), (f, f), (g, g), (g, b)\}$ et $\mathcal{Q}^+ = \{(a, b), (a, g), (b, b), (b, g), (d, b), (d, d), (d, e), (d, f), (d, g), (e, b), (e, d), (e, f), (e, g), (e, d), (f, b), (f, d), (f, e), (f, f), (f, g), (g, b)\}$.

Définition A.7 (Fermeture réflexo-transitive) *Soit \mathcal{R} une relation binaire sur un ensemble de base E . La fermeture réflexo-transitive \mathcal{R}^* de \mathcal{R} est définie par :*

$$\mathcal{R}^* := \bigcup_{i \in \mathbb{N}}$$

Définition A.8 (Prédécesseur) *Soient a et b deux éléments de (E, \mathcal{R}) . Nous dirons que a est un prédécesseur (resp. un successeur) de b si et seulement si $a\mathcal{R}^+b$ (resp. $b\mathcal{R}^+a$).*

(E, \mathcal{R}) peut être représenté comme un graphe orienté où E est l'ensemble des sommets et \mathcal{R} l'ensemble des arêtes.

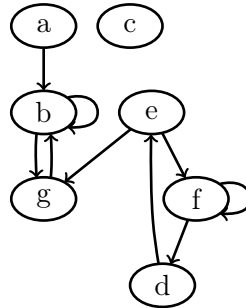


FIGURE A.1 – La représentation sous forme de graphe de (S, \mathcal{Q}) donné en exemple.

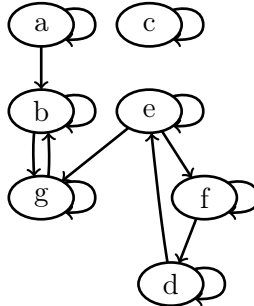


FIGURE A.2 – La représentation sous forme de graphe de la fermeture réflexive de (S, \mathcal{Q}) : (S, \mathcal{Q}^1) .

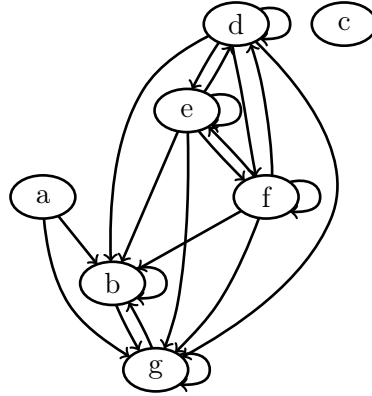


FIGURE A.3 – La représentation sous forme de graphe de la fermeture transitive de $(S, \mathcal{Q}) : (S, \mathcal{Q}^+)$.

Définition A.9 (Chemin) On dit que $(a_1, \dots, a_n) \in E^n$ est un chemin de a_1 à a_n dans (E, \mathcal{R}) si et seulement si $\forall 1 \leq i < n, a_i \mathcal{R} a_{i+1}$.

Définition A.10 (Chaîne) On dit que $(a_1, \dots, a_n) \in E^n$ est un chemin de a_1 à a_n dans (E, \mathcal{R}) si et seulement si $\forall 1 \leq i < n, a_i \mathcal{R} a_{i+1} \vee a_{i+1} \mathcal{R} a_i$.

Définition A.11 (Cycle) On dit que $(a_1, \dots, a_n) \in E^n$ est un cycle dans (E, \mathcal{R}) si et seulement si il y a un chemin de a_1 à a_n et $a_1 = a_n$.

Définition A.12 (Ensemble connexe) Un ensemble E est dit connexe dans \mathcal{R} si et seulement si $\forall a, b \in E$, il y a une chaîne de a à b .

Définition A.13 (Ensemble fortement connexe) Un ensemble E est dit fortement connexe dans \mathcal{R} si et seulement si $\forall a, b \in E$, il y a un chemin de a à b et de b à a .

Définition A.14 (Relation d'équivalence) Une relation binaire \mathcal{R} sur un ensemble de base E est une relation d'équivalence si et seulement si elle est réflexive, symétrique et transitive.

Définition A.15 (Classe d'équivalence) Soit une relation d'équivalence \mathcal{R} . On note $[a]_{\mathcal{R}}$ l'ensemble des éléments de E qui sont équivalents par \mathcal{R} :

$$[a]_{\mathcal{R}} := \{b \in E : a \mathcal{R} b\}$$

L'ensemble $[a]_{\mathcal{R}}$ est appelé la classe d'équivalence de a pour \mathcal{R} .

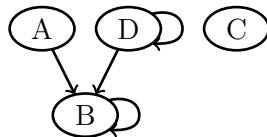


FIGURE A.4 – Exemple du graphe de S/\mathcal{Q}^+ .

Définition A.16 (Ensemble quotient) Soit un ensemble E et \mathcal{R} une relation d'équivalence sur E . L'ensemble quotient E/\mathcal{R} est défini ainsi :

$$E/\mathcal{R} := \{[a]_{\mathcal{R}} : a \in E\}$$

De même, on peut définir la relation quotient.

Définition A.17 (Relation quotient) *Soit une relation \mathcal{R} définie sur E et \mathcal{Q} une relation d'équivalence sur E . La relation quotient \mathcal{R}/\mathcal{Q} est définie ainsi :*

$$\mathcal{R}/\mathcal{Q} := \{([a]_{\mathcal{Q}}, [b]_{\mathcal{Q}}) : (a, b) \in \mathcal{R}\}$$

Définition A.18 (Idéal) *Soit a un élément de (E, \mathcal{R}) . L'idéal de a est l'ensemble des prédécesseurs de a dans (E, \mathcal{R}) :*

$$\downarrow a := \{b \in E : b\mathcal{R}^t a\}$$

Définition A.19 (Filtre) *Soit a un élément de (E, \mathcal{R}) . Le filtre de a est l'ensemble des successeurs de a dans (E, \mathcal{R}) :*

$$\uparrow a := \{b \in E : a\mathcal{R}^t b\}$$

Définition A.20 (Relation restreinte) *Soit \mathcal{R} une relation sur l'ensemble de base E , et F un sous-ensemble de E . Nous noterons la sous-relation de \mathcal{R} restreinte aux éléments de F comme suit :*

$$(E, \mathcal{R})[F] := (F, \mathcal{R} \cap F^2)$$

Définition A.21 (Relation duale) *Soit \mathcal{R} une relation binaire sur un ensemble de base E . La relation duale de \mathcal{R} notée \mathcal{R}^d est définie comme*

$$\mathcal{R}^d := \{(a, b) \in E^2 : b\mathcal{R}a\}$$

Diagrammes de Hasse

LES diagrammes de Hasse offrent un moyen commode et intuitif de représenter graphiquement les ordres : les sommets sont les éléments de l'ordre, et les arcs représentent la relation de couverture entre ces éléments.

Définition B.1 (Couverture) Soient un ordre $O = (E, \leq)$ et trois éléments a, b, c de E . On dit que a est couvert par b et on note $a \prec b$ si $(a < b \wedge a \leq c < b) \Rightarrow a = c$.

Définition B.2 (Graphe de couverture) Le graphe de couverture d'un ordre $O = (E, \leq)$ est un graphe orienté $C = (E, V)$ tel que $\forall a, b \in E, (a, b) \in V$ ssi $a \prec b$.

Définition B.3 (Diagramme de Hasse) Le diagramme de Hasse d'un ordre $O = (E, \leq)$ est une représentation dans le plan cartésien de son graphe de couverture où tous les sommets (éléments de E) sont représentés par des points tels que si $a, b \in E, a < b$, l'ordonnée de a est inférieure à l'ordonnée de b .

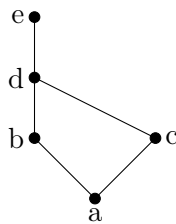


FIGURE B.1 – Diagramme de Hasse de l'exemple B.1.

Exemple B.1 Soit l'ordre $O = (\{a, b, c, d, e\}, \{(a, b), (a, c), (a, d), (a, e), (b, d), (b, e), (c, d), (c, e), (d, e)\})$. Le diagramme de Hasse de O est représenté figure B.1. Sur cette figure, on voit que b couvre a . Bien que a soit inférieur à d dans O , il n'y a pas d'arc entre a et d , car d ne couvre pas a . En effet, on a $a < b < d$.

Approximation des couplages dans COSMOS-G

Introduction

NOUS détaillons ici la façon dont les fonctions de couplage sont traitées dans COSMOS (section 2.2.3). Pendant les optimisations disciplinaires, COSMOS utilise des approximations des variables de couplage calculées à l'itération système précédente dans les autres disciplines. D'après Rabeau (2007), la valeur de l'approximation doit converger vers la valeur réelle de la variable de couplage en même temps que la population converge vers les solutions optimales du problème.

Cette analyse des couplages dans COSMOS a fait l'objet d'une publication en conférence internationale (Guédas et Dépincé 2008a) sélectionnée pour une publication en journal (Guédas et Dépincé 2008b).

C.1 Traitement des variables couplées

Considérons un modèle simplifié de deux disciplines D_1 et D_2 (il peut être généralisé à q disciplines). Chaque discipline D_i a une équation d'état $A_i(x_c, x_i, u_i, v_i) = 0$. Nous considérerons qu'il existe une fonction implicite $a_i : \mathcal{X}_c \times \mathcal{X}_i \times \mathcal{U}_i \times \rightarrow \mathcal{V}_i$ où $x_c \in \mathcal{X}_c$ est la variable de conception partagée entre les disciplines (variable commune), $x_i \in \mathcal{X}_i$ est la variable disciplinaire (variable locale), $u_i \in \mathcal{U}_i$ est un paramètre donné par l'autre discipline, et v_i est la variable d'état calculée par la discipline i . Nous avons le problème suivant :

$$D_1 \begin{cases} v_1 = a_1(x_c, x_1, u_1) \\ u_2 = c_1(v_1) \end{cases} \quad D_2 \begin{cases} v_2 = a_2(x_c, x_2, u_2) \\ u_1 = c_2(v_2) \end{cases} \quad (\text{C.1})$$

Trouver les solutions admissibles de ce problème est appelé l'analyse multidisciplinaire (MDA). Nous notons A l'ensemble des solutions admissibles du problème (les solutions multidisciplinairement réalisables) :

$$\begin{aligned}
 X = \{ & (x_c, x_1, x_2, v_1, v_2) \in \mathcal{X}_c \times \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{V}_1 \times \mathcal{V}_2 : \\
 & v_1 = a_1(x_c, x_1, c_2(v_2)) \wedge v_2 = a_2(x_c, x_2, c_1(v_1)) \}
 \end{aligned} \tag{C.2}$$

L'optimisation multidisciplinaire consiste à ajouter un problème d'optimisation à la satisfaction du problème couplé. Nous nous intéressons au cas où chaque discipline a son propre problème d'optimisation à résoudre, qui est la minimisation d'une fonction coût $f_i : \mathcal{X}_c \times \mathcal{X}_i \times \mathcal{V}_i \rightarrow \mathbb{R}^{p_i}$, avec p_i le nombre d'objectifs. Une formulation du problème à deux disciplines est alors :

$$\begin{aligned}
 D_1 \left\{ \begin{array}{l} \min_{x_c, x_1 \in \mathcal{X}_c \times \mathcal{X}_1} f_1(x_c, x_1, v_1) \\ v_1 = a_1(x_c, x_1, u_1) \\ u_2 = c_1(y_1) \end{array} \right. & \quad D_2 \left\{ \begin{array}{l} \min_{x_c, x_2 \in \mathcal{X}_c \times \mathcal{X}_2} f_2(x_c, x_2, v_2) \\ v_2 = e_2(x_c, x_2, u_2) \\ u_1 = c_2(y_2) \end{array} \right.
 \end{aligned} \tag{C.3}$$

L'ensemble S des solutions du problème est alors :

$$\begin{aligned}
 X^* = \{ & (x_c^*, x_1^*, x_2^*) | \exists (v_1^*, v_2^*) \in V_1 \times V_2 \\
 \text{t.q. } & (x_c, x_1, x_2, v_1, v_2) \in X \wedge \nexists (x_c, x_1, x_2, v_1, v_2) \in X \\
 & \text{t.q. } f(x_c, x_1, x_2, v_1, v_2) \preceq f(x_c^*, x_1^*, x_2^*, v_1^*, v_2^*) \}
 \end{aligned} \tag{C.4}$$

où \preceq est la relation de dominance de Pareto.

Malheureusement, un tel ensemble de solutions n'est pas atteignable sous l'hypothèse de la partition des variables dans chaque discipline. En effet, une discipline i n'a pas accès aux variables locales d'une discipline j et ne connaît pas ses fonctions de couplage.

Les problèmes d'optimisation multidisciplinaire sont souvent écrits sous une forme simplifiée, où les variables d'état sont directement données à l'autre discipline. Elles sont ainsi également des variables de couplage et les fonctions a_i sont les fonctions de couplage :

$$\begin{aligned}
 D_1 \left\{ \begin{array}{l} \min_{x_c, x_1} f_1(x_c, x_1, v_1) \\ v_1 = a_1(x_c, x_1, v_2) \end{array} \right. & \quad D_2 \left\{ \begin{array}{l} \min_{x_c, x_2} f_2(x_c, x_2, v_2) \\ v_2 = a_2(x_c, x_2, v_1) \end{array} \right.
 \end{aligned} \tag{C.5}$$

C'est cette dernière modélisation que nous utiliserons.

C.1.1 Traitement des variables de couplage dans COSMOS

Toutes les valeurs des variables de couplage v_i calculées dans la discipline i à la fin de l'optimisation disciplinaire sont stockées dans un tableau. Ce tableau contient tous les couples $(v_{i,k}, x_{c,k})$, où $x_{c,k}$ est le vecteur des variables couplées du $k^{\text{ème}}$ individu de la population, et $v_{i,k}$ est la variable de couplage calculée par la fonction de couplage a_i dans la discipline i avec la variable $x_{c,k}$ et les variables locales $x_{i,k}^*$ obtenues à la fin de l'optimisation disciplinaire. Quand une autre discipline j a besoin de la valeur de v_i , pour tous ces $x_{c,k}$ elle cherchera le $v_{i,k}$ correspondant – quand il existe – dans le tableau. Si le tableau ne contient pas le $x_{c,k}$ voulu, le plus proche est pris. Autrement dit, pour chaque discipline i , il y a un tableau T_i de couples (x_c, v_i) tels que $T_i \subseteq X_c \times V_i$. Le tableau T_i décrit donc une relation binaire entre les éléments de X_c et de V_i établie par la discipline i .

Si on note \tilde{v}_i l'approximation de v_i dans T_i , il semble que lorsque la population converge vers les solutions optimales, la distance entre \tilde{v}_i et v_i tend vers zéro.

Notre objectif est ici d'étudier le comportement de COSMOS sur les problèmes couplés, et plus précisément l'évolution de l'erreur d'approximation des variables couplées induite par la relation T_i au lieu de la fonction a_i .

C.2 Problèmes de test

COSMOS n'a pas été évalué sur des problèmes réels faisant intervenir des fonctions de couplage.

C.2.1 Problème #1

Cet exemple est présenté dans la thèse de Rabeau (2007). Il est composé de deux disciplines ayant chacune deux objectifs à minimiser. Les domaines des variables d'optimisation sont : $x_c \in [0; 10]$, $x_1, x_2, x_3, x_4 \in [0; 20]$.

$$D_1 \begin{cases} \min f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 \\ \min f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 9)^2 \\ v_1 = \frac{x_1}{f_2(x_c, x_2) + 0.1} \end{cases} \quad (C.6)$$

$$D_2 \begin{cases} \min f_3(x_c, x_3) = (x_3 - 4)^2 + (x_c - 5)^2 \\ \min f_4(x_c, x_4) = (v_1 - 5)^2 + (x_4 - 5)^2 + (x_c - 9)^2 \end{cases} \quad (C.7)$$

Problème #2

Les domaines des variables d'optimisation sont : $x_c, x_1, x_2, x_3, x_4, x_5, x_6 \in [-20; 20]$.

$$D_1 \begin{cases} \min f_1(x_c, x_1, x_2) = (x_1 - 1)^2 + x_2^2 \\ \min f_2(x_c, x_1, x_2) = x_c \times (x_1 - 3)^2 \times (x_2 - 7)^2 \\ \min f_3(x_c, x_1, x_2) = x_1 + x_2 + 1 \\ v_{12} = v_{13} = 10 \times x_1 + x_c \end{cases} \quad (C.8)$$

$$D_2 \begin{cases} \min f_4(x_c, x_3, x_4) = x_3^2 + (x_4 - 3)^2 \\ \min f_5(x_c, x_3, x_4) = v_{12} + (x_3 + x_4)^2 + (x_c - 4)^2 \\ \min f_6(x_c, x_3, x_4) = v_{32} \times x_3 \times x_4 \times x_c \end{cases} \quad (C.9)$$

$$D_3 \begin{cases} \min f_7(x_c, x_5, x_6) = 7 \times v_{13}^2 / x_c^2 + 0,1 \\ \min f_8(x_c, x_5, x_6) = (x_5 - 1)^2 + (x_6 - 9)^2 \\ v_{32} = 10 \times x_c + x_5 \end{cases} \quad (C.10)$$

Problème #3

Les domaines des variables d'optimisation sont : $x_c, x_1, x_2, x_3, x_4 \in [0; 20]$.

$$D_1 \begin{cases} \min f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 + (v_3 - 5)^2 \\ \min f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 7)^2 + (v_4 - 7)^2 \\ v_1 = \frac{x_2}{\sqrt{x_c + x_1 + 0,1}} \\ v_2 = \frac{x_1}{\sqrt{x_c + x_2 + 0,1}} \end{cases} \quad (C.11)$$

$$D_2 \left\{ \begin{array}{l} \min f_3(x_c, x_3) = (x_3 - 3)^2 + (x_c - 5)^2 + (v_1 - 3)^2 \\ \min f_4(x_c, x_4) = (x_4 - 5)^2 + (x_c - 9)^2 + (v_2 - 9)^2 \\ v_3 = \frac{x_3}{\sqrt{x_c + x_4 + 0,1}} \\ v_4 = \frac{x_4}{\sqrt{x_c + x_3 + 0,1}} \end{array} \right. \quad (C.12)$$

Problème #4

Les domaines des variables d'optimisation sont : $x_c \in [-50; 50], x_1, x_2, x_3, x_4 \in [-10; 10]$.

$$D_1 \left\{ \begin{array}{l} \min f_1(x_c, x_1) = x_1 \times \frac{(x_1 - x_c + 0,1)^2}{(x_1 - v_2 + 0,1)^2} \\ \min f_2(x_c, x_1) = x_2 \times \frac{(x_2 - x_c + 0,1)^2}{(x_2 - v_2 + 0,1)^2} \\ v_1 = x_c \end{array} \right. \quad (C.13)$$

$$D_2 \left\{ \begin{array}{l} \min f_1(x_c, x_3) = -x_3 \times \frac{(x_3 - x_c + 0,1)^2}{(x_3 - v_1 + 0,1)^2} \\ \min f_2(x_c, x_4) = -x_4 \times \frac{(x_4 - x_c + 0,1)^2}{(x_4 - v_1 + 0,1)^2} \\ v_2 = x_c \end{array} \right. \quad (C.14)$$

Problème #5

Les domaines des variables d'optimisation sont : $x_c, x_1, x_2, x_3, x_4 \in [-50; 50]$.

$$D_1 \left\{ \begin{array}{l} \min f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 + (v_3 - 5)^2 \\ \min f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 7)^2 + (v_4 - 7)^2 \\ v_1 = (v_3 + 2)^2 - 22 \\ v_2 = (v_4 + 2)^2 - 22 \end{array} \right. \quad (C.15)$$

$$D_2 \left\{ \begin{array}{l} \min f_3(x_c, x_3) = (x_3 - 3)^2 + (x_c - 5)^2 + (v_1 - 3)^2 \\ \min f_4(x_c, x_4) = (x_4 - 5)^2 + (x_c - 9)^2 + (v_2 - 9)^2 \\ v_3 = (y_1 + 3)^2 - 33 \\ v_4 = (y_2 + 3)^2 - 33 \end{array} \right. \quad (C.16)$$

C.3 Résultats

COSMOS a été lancé sur chaque problème de test avec deux jeux de paramètres : le premier avec une population de 20 individus, 20 itérations système et 20 itérations sous-système (*léger*) ; le second avec 100 individus, 50 itérations système et 50 itérations sous-système (*lourd*).

Deux critères sont vérifiés pour la validation des résultats : premièrement, la convergence de l'approximation de la variable couplée vers sa valeur réelle, et enfin la qualité des solutions obtenues.

opt.	test	var.	min	moy	max
Light	#1	v_1	0.0007	0.0157	0.1038
	#2	v_{1x}	0.0192	9.2963	38.2864
		v_{32}	0.0041	1.4345	21.3760
	#3	v_1	0.0047	0.1540	0.4441
		v_2	0.0047	0.2923	0.9366
		v_3	0.0051	0.1391	0.9548
		v_4	0.0972	0.4443	0.8602
	#4	v_1	0.0495	2.4839	15.0109
		v_2	0.0495	2.4839	15.0109
	Heavy	#1	v_1	0.0001	0.0030
#2		v_{1x}	0.1293	15.5426	41.0447
		v_{32}	0.0002	0.1891	2.4605
#3		v_1	0.0041	0.2539	1.3165
		v_2	0.0035	0.2215	1.0609
		v_3	0.0022	0.2020	0.8487
		v_4	0.0008	0.3336	1.3102
#4		v_1	0.0054	0.4863	2.3182
		v_2	0.0054	0.4863	2.3182

TABLEAU C.1 – Valeurs minimum, moyenne et maximum de l’erreur d’approximation (%); $x \in \{1, 2\}$.

C.3.1 Qualité de l’approximation

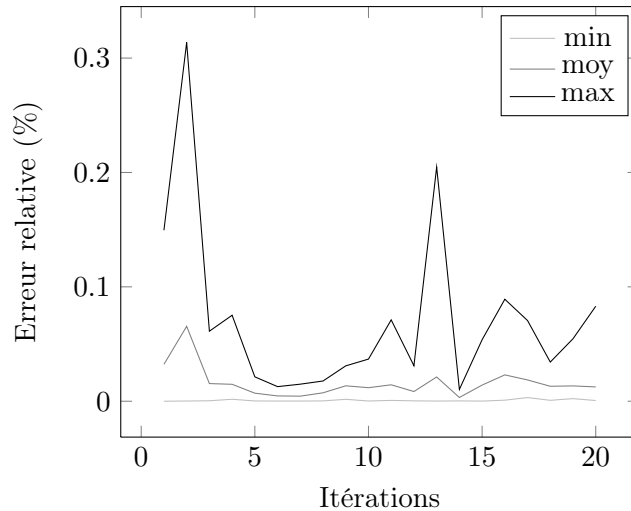
Pendant l’optimisation disciplinaire, chaque discipline connaît seulement la variable de couplage calculée dans l’autre sous-système à l’itération précédente, et l’utilise pour trouver une approximation de sa valeur réelle.

Nous évaluons seulement l’évolution de l’erreur de l’approximation de la fonction de couplage dans le temps (c’est-à-dire à chaque itération) et non l’évolution de la variable de couplage vers sa valeur optimale à la fin du processus d’optimisation. Nous vérifions aussi si chaque v_i , qui est une entrée du tableau T_i , est celle qui donnerait la meilleure approximation sur tous les v_i qui ont été calculés pendant l’optimisation disciplinaire.

Le tableau C.1 représente les résultats calculés pour les problèmes #1, #2, #3 et #4 avec les paramètres *léger* et *lourd*. Les résultats sont donnés en % de l’erreur relative à la taille du domaine de v_i .

Nous observons que l’erreur entre l’approximation et la valeur réelle est assez petite, exceptée pour v_{12} et v_{13} dans l’exemple #2, comme le montrent les figures C.2 et C.3. Le fait qu’une fonction de couplage soit utilisée dans deux disciplines différentes en même temps semble perturber son approximation. Nous observons également que la valeur moyenne de l’erreur est proche de la valeur minimum, sauf pour cette variable de couplage, ce qui semble indiquer qu’il y a autant de bonnes valeurs que de mauvaises. Deux phénomènes peuvent expliquer ces erreurs :

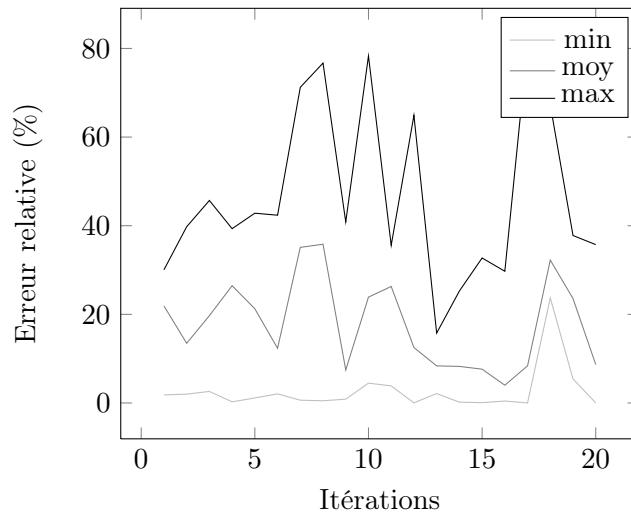
- Les valeurs de x_c^* ne couvrent pas tout son domaine : les x_c optimaux pour les disciplines 1, 2 et 3 sont respectivement dans $\{-20\}$, $[-20, 4]$ et $\{20\}$ donc $x_c^* \in [-20, 4] \cup \{20\}$. Pour les valeurs de $x_c \in]4, 20[$, l’approximation ne sera pas bonne.
- La valeur optimale de x_1 n’est pas unique, mais dans $[-20; 3]$. Quand cette valeur est unique, les autres disciplines n’ont qu’un seul choix pour trouver le x_c corres-

FIGURE C.1 – Erreur relative de v sur #1 (léger)

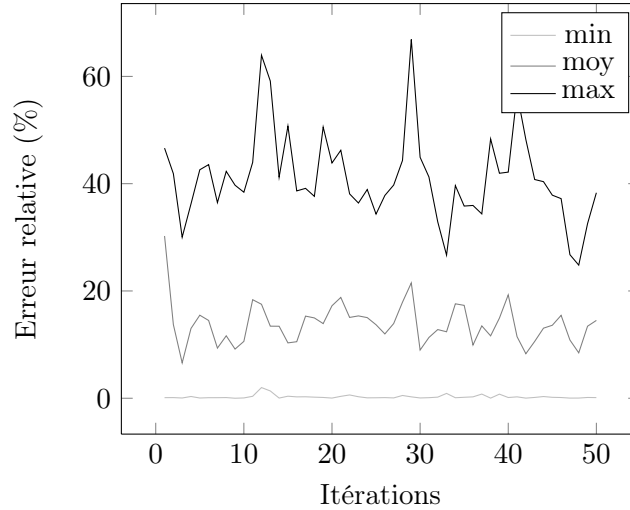
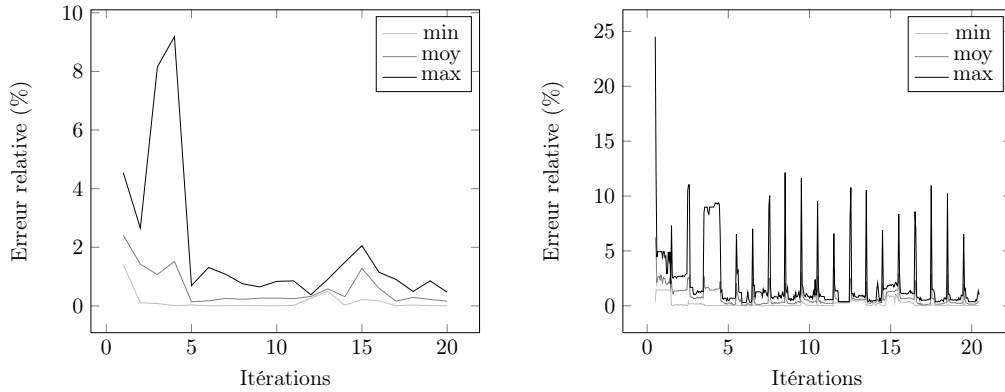
pondant. Ici, la discipline accroît sa probabilité de choisir un x_c correspondant au x_1 qui ne donne pas la valeur optimale de v_1x pour sa propre discipline.

D'autres tests ont montré que l'utilisation de la variable de couplage dans plus d'une discipline n'influence pas l'approximation.

Il n'y a pas de décroissance stricte de l'erreur au cours du temps, mais nous notons une décroissance globale autour des cinq premières itérations, qui se stabilise ensuite (voir figures C.1 et C.4).

FIGURE C.2 – Erreur relative de $v_{12} = v_{13}$ sur #2 (léger)

L'erreur d'approximation, dans le cas général, ne converge pas, mais reste assez petite au cours du temps. De plus, les tests effectués avec le jeu de paramètres *lourd* sont moins chaotiques (Cf. la différence entre C.2 et C.3). Nous expliquons cela par dispersion des valeurs de x_c dans la table de couplage. Effectivement, plus la taille de la population est grande, plus les chances de trouver un x_c dans la table qui est proche de la valeur

FIGURE C.3 – Erreur relative de $v_{12} = v_{13}$ sur #2 (lourd)FIGURE C.4 – Erreur relative de $v_{12} = v_{13}$ sur #2 (lourd) au niveau système (gauche) et des sous-systèmes (droite).

dont nous avons besoin est grande. Cette erreur devrait également tendre vers zéro si la population tend vers un unique x_c .

En résumé, dans la plupart des cas, nous distinguons deux phases dans la convergence des variables de couplage illustrées par la figure C.5. La première phase est une décroissance globale de l'erreur due à la convergence des individus (variables disciplinaires). La seconde phase est un état où l'erreur est stabilisée et ne montre pas de changements notables. L'évolution est chaotique et son amplitude décroît quand nous ajoutons plus d'individus à la population.

Cette description semble correspondre aux exemples où les fonctions objectifs sont convexes. Sous cette hypothèse, pour chaque x_c , il y a un seul et unique x_i^* qui minimise f_i . Donc il existe une fonction $s_i : \mathcal{X}_c \rightarrow \mathcal{X}_i$ qui permet de remplacer la fonction a_i par une fonction simplifiée t_i avec $t_i(x_c, v_i) = a_i(x_c, s(x_c), v_i)$. Les éléments de T_i sont des éléments du graphe de la fonction t_i , donc si les fonctions objectifs sont convexes, T_i est un bon candidat pour l'approximation de a_i .

De plus, - comme l'indique la figure C.4 - l'approximation au niveau système n'est pas toujours la meilleure calculée au niveau sous-système. Cela est vrai sous deux conditions :

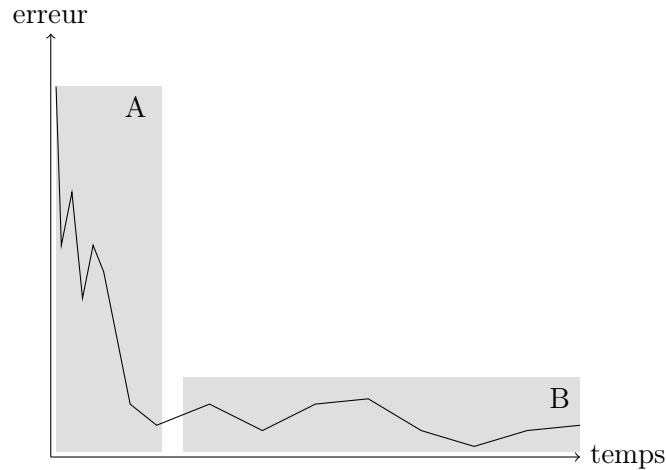


FIGURE C.5 – Schéma de l'évolution du taux d'erreur au cours du temps. Dans une première phase (A), la courbe décroît globalement, puis elle se stabilise dans une seconde phase (B).

- Les optimums des x_i sont uniques et le système n'a pas encore convergé vers ces solutions.
- Les optimums des x_i ne sont pas uniques et, comme expliqué au-dessus, les x_i pris par la première discipline ne correspondent pas aux x_i qui donnent le meilleur couple (x_c, x_i) qui permet d'obtenir la solution optimale de la variable de couplage pour l'autre discipline.

L'exemple #5 n'entre pas dans la même catégorie de problème. Les fonctions de couplage ont été choisies comme un système d'équations qui ne converge pas avec un algorithme de point fixe. Sur ce problème, COSMOS ne converge pas non plus. L'erreur d'approximation augmente au cours du temps. En fait, sur ce type de problème, COSMOS se comporte comme un algorithme de point fixe.

Qualité des solutions

Les solutions trouvées pour les problèmes #1 et #3 sont optimales. Elles correspondent aux solutions Pareto-optimales dans chaque discipline en même temps. Nous sommes donc sûrs qu'elles sont sur le front de Pareto global. En effet, pour le premier problème de test, COSMOS a globalement convergé vers la solution $x^* = (a, 1, 3, 3, 5)$ avec $a \in [3; 7]$, ce qui correspond aux solutions optimales du problème. Les résultats du troisième problème de test sont aussi satisfaisantes (figure C.6).

Dans l'exemple #4, les variables de couplage convergent, mais COSMOS n'a pas trouvé la bonne solution. En effet, l'optimum est unique $(x_c^*, x_1^*, x_2^*) = (-10, -10, 10, 10)$, mais COSMOS a convergé vers $(x_c, x_1, x_2) = (0, 0, 0, 0)$. Nous expliquons ce comportement par le fait que la population est utilisée en même temps pour correspondre aux objectifs et pour correspondre à l'approximation des fonctions de couplage. Ici, cette interférence est explicite, puisque les variables communes sont aussi les variables de couplage.

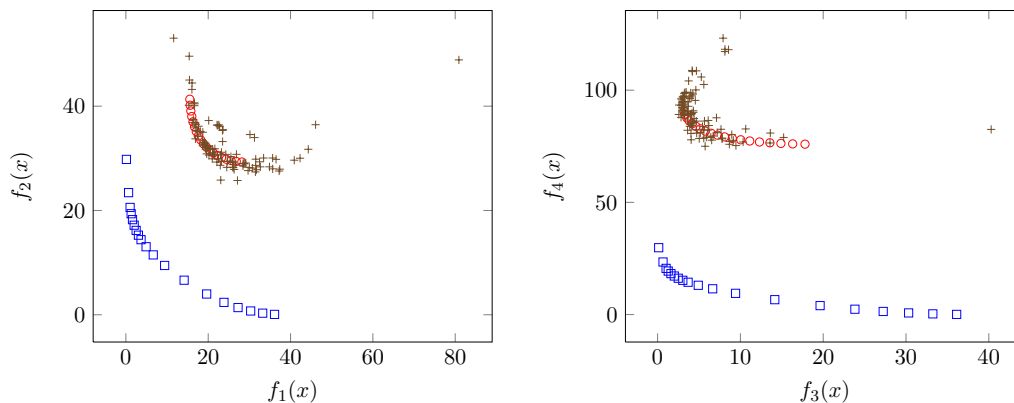


FIGURE C.6 – Solutions sur le problème #3 : Les carrés (\square) représentent les fronts de Pareto des deux disciplines prises indépendamment. Les ronds (\circ) correspondent au front de Pareto pour des valeurs fixées à l'optimum de l'autre discipline (ex. $x_3 = 3$ et $x_4 = 5$ dans la première discipline, et $x_1 = 1$ et $x_2 = 3$ pour la seconde). Les solutions trouvées par COSMOS (+) sont proches de ce second front de Pareto.

Conclusion

Nous avons montré des problèmes rencontrés par COSMOS dans le traitement des variables couplées. Pour une première analyse, COSMOS a seulement été testé sur des problèmes simples. Il serait intéressant de pouvoir analyser son comportement sur d'autres problèmes, et en particulier des problèmes industriels.

Il apparaît néanmoins que, sous l'hypothèse de l'unicité des optimums des variables locales pour une valeur de variable commune donnée, le tableau utilisé par COSMOS pour approximer les fonctions de couplage est pertinent. Cette hypothèse est en particulier vérifiée si les fonctions objectifs sont convexes. Cependant, dans le cas général où cette hypothèse n'est pas vérifiée, nos tests indiquent que la méthode n'est pas capable de déterminer une bonne approximation des couplages. Nous avons de plus observé que les valeurs gardées dans le tableau pour l'approximation ne sont pas les meilleures valeurs calculées au cours des optimisations disciplinaires.

Prise en compte des variables locales dans EMO-MDO

LES compromis (chapitre 3) et l'algorithme (chapitre 4) proposés ne considèrent pas les variables locales. Le compromis étant défini sur l'espace de comparaison et non sur l'espace de décision, sa définition est indépendante de la présence de ces variables. En revanche, bien qu'EMO-MDO fonctionne avec des problèmes contenant des variables locales et globales (voir l'exemple 4), son efficacité peut être améliorée en tenant compte des spécificités de ces variables, comme le montre l'exemple D.1.

Exemple D.1 *Pour illustrer le problème, considérons un problème d'optimisation avec deux disciplines ayant chacune deux objectifs. Les variables communes du problème sont définies sur l'ensemble $X_{\{1,2\}}^c$ tandis que les variables locales aux disciplines sont respectivement définies sur X_1^d et X_2^d pour la première et seconde discipline. Soient $X_{\{1,2\}}^c = \{a, b, c, d\}$, $X_1^d = \{x, y, z\}$ et $X_2^d = \{\alpha, \beta\}$. Le tableau D.1 référence un ensemble de solutions admissibles et leurs valeurs dans l'espace des variables et des objectifs. Les solutions dans l'espace des objectifs sont représentées figure D.1.*

On peut remarquer, par exemple, que les points A et C partagent la valeur a pour la variable x^c . Dans la première discipline $A_1 \leq C_1$ et dans la seconde $C_2 \leq A_2$. Comme $A = f(a, y, \alpha)$ et $C = f(a, z, \beta)$, on peut créer une nouvelle solution (a, y, β) dont l'image est équivalente à A_1 dans la première discipline et C_2 dans la seconde. Cette nouvelle solution domine donc A et C.

L'exemple D.1 peut être généralisé dans une méthode de croisement utilisée conjointement au croisement habituel pour créer de meilleures solutions.

$f(x)$	x^c	x_1^d	x_2^d	$f_{1,1}(x^c, x_1^d)$	$f_{1,2}(x^c, x_1^d)$	$f_{2,1}(x^c, x_2^d)$	$f_{2,2}(x^c, x_2^d)$
A	a	y	α	1	6	8	7
B	b	x	α	3	8	1	3
C	a	z	β	7	7	4	1
D	c	y	α	8	4	3	8
E	b	x	β	5	3	6	6
F	d	y	β	4	5	5	4
G	d	z	α	2	2	7	2
H	a	x	β	7	1	2	4

TABLEAU D.1 – Solutions admissibles de l'exemple D.1. Une variable est commune x^c et les deux disciplines disposent de leurs variables locales (x_1^d et x_2^d). Elles sont représentées graphiquement figure D.1.

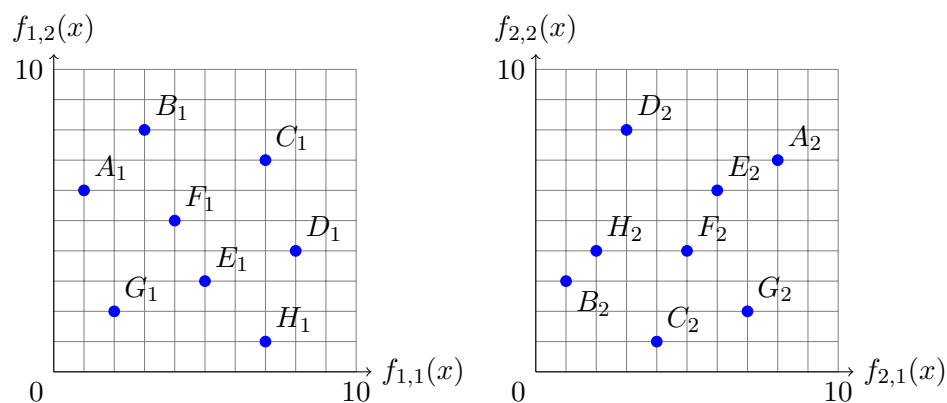


FIGURE D.1 – Illustration des solutions présentées dans l'exemple D.1.

Niches de solutions pour le problème du réducteur de vitesse

NOUS détaillons les valeurs des solutions trouvées par l'algorithme EMO-MDO sur le problème du réducteur de vitesse (section 4.2.5) avec les configurations $c2_n_50 \times 500$, $c2_o_50 \times 500$, $c3_n_50 \times 500$, $c3_o_50 \times 500$ et $c4_n_50 \times 500$ $c4_o_50 \times 500$. Chacune de ces configurations a été lancée trois fois (le numéro est indiqué entre parenthèses dans les tableaux E.1, E.2, E.3 et E.4).

Les solutions de compromis \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 sont en principe similaires dans ce problème (figure 4.26). Cependant, sur certains tests effectués, les populations ne convergent pas vers la totalité de la zone de compromis dans la première discipline (figure 4.24). Les différences observées se trouvant essentiellement dans la première discipline, nous n'avons considéré que cette dernière. Nous ne nous intéressons donc pas ici aux valeurs des variables locales de la seconde discipline. Dans la première discipline, les solutions se répartissent entre deux scénarios : dans un premier cas, la population converge vers la partie horizontale du front de Pareto ; dans un second cas, la population converge vers la partie verticale du front de Pareto. Ces deux scénarios correspondent respectivement à deux niches de solutions :

Niche 1 : les valeurs de la variable $x_{1,2}^d$ sont toutes égales à la valeur de leur borne inférieure. Les valeurs de la variable x_2^c sont proches des valeurs de leur borne inférieure (tableaux E.1 et E.2) ;

Niche 2 : les valeurs des variables x_1^c , x_2^c et x_3^c sont toutes égales aux valeurs de leur borne inférieure (tableaux E.3 et E.4).

On peut remarquer que l'algorithme a plus tendance à converger vers la niche 1 (13 fois sur 18) que vers la niche 2 (5 fois sur 18).

	x_1^c	x_2^c	x_3^c	$x_{1,1}^d$	$x_{1,2}^d$
c2_n_50x500 (1)					
moy	3,5546	0,7003	23,5479	7,7542	3,9000
min	3,5344	0,7000	17,7191	7,7500	3,9000
max	3,5756	0,7009	27,9756	7,8180	3,9000
c3_n_50x500 (1)					
moy	3,5150	0,7015	25,7717	7,7506	3,9000
min	3,5012	0,7000	18,1206	7,7500	3,9000
max	3,5380	0,7073	28,0000	7,7515	3,9000
c4_n_50x500 (1)					
moy	3,5484	0,7017	22,321	7,7517	3,9000
min	3,5287	0,7001	17,0306	7,7500	3,9000
max	3,5718	0,7057	27,8801	7,7544	3,9000
c3_n_50x500 (2)					
moy	3,5140	0,7009	25,4418	7,7527	3,9000
min	3,5027	0,7000	19,2756	7,7506	3,9000
max	3,5267	0,7049	28,0000	7,7570	3,9000
c3_n_50x500 (3)					
moy	3,5231	0,7006	25,3662	7,7531	3,9000
min	3,5078	0,7000	17,4975	7,7501	3,9000
max	3,5321	0,7043	27,9905	7,8064	3,9000
c4_n_50x500 (3)					
moy	3,5011	0,7000	27,2717	7,7511	3,9000
min	3,5003	0,7000	25,3973	7,7502	3,9000
max	3,5035	0,7001	27,9893	7,7520	3,9000

TABLEAU E.1 – Scénario 1 (horizontal), sans crowding.

	x_1^c	x_2^c	x_3^c	$x_{1,1}^d$	$x_{1,2}^d$
c2_n_50x500 (2)					
moy	3,5000	0,7	17	7,3	3,2263
min	3,5000	0,7000	17,0000	7,3000	3,1742
max	3,5000	0,7000	17,0000	7,3000	3,3779
c4_n_50x500 (2)					
moy	3,5003	0,7000	17,000	7,3009	3,3529
min	3,5001	0,7000	17,000	7,3000	3,1703
max	3,5008	0,7000	17,000	7,3110	3,6062
c2_n_50x500 (3)					
moy	3,5002	0,7	17,0001	7,3000	3,4622
min	3,5000	0,7000	17,0000	7,3000	3,2248
max	3,5008	0,7000	17,0003	7,3001	3,5895

TABLEAU E.2 – Scénario 2 (vertical), sans crowding.

	x_1^c	x_2^c	x_3^c	$x_{1,1}^d$	$x_{1,2}^d$
c2_o_50x500 (1)					
moy	3,5694	0,7057	22,3588	7,7556	3,9000
min	3,5384	0,7002	17,1897	7,7501	3,9000
max	3,5882	0,7149	27,9974	7,8152	3,9000
c3_o_50x500 (1)					
moy	3,5261	0,7011	22,5622	7,75488	3,9000
min	3,5025	0,7000	17,1701	7,75002	3,9000
max	3,5536	0,7033	27,9557	7,76453	3,9000
c2_o_50x500 (2)					
moy	3,5054	0,7002	21,975	7,7530	3,9000
min	3,5014	0,7000	17,0233	7,7501	3,9000
max	3,5121	0,7013	27,7706	7,7660	3,9000
c3_o_50x500 (2)					
moy	3,5137	0,7006	23,1684	7,7530	3,9000
min	3,5014	0,7000	17,0323	7,7502	3,9000
max	3,5606	0,7057	27,9931	7,8028	3,9000
c2_o_50x500 (3)					
moy	3,5105	0,7005	22,34	7,7539	3,9000
min	3,5012	0,7000	17,0642	7,7501	3,9000
max	3,5251	0,7048	27,9955	7,8282	3,9000
c3_o_50x500 (3)					
moy	3,5252	0,7028	22,8702	7,7549	3,9000
min	3,5036	0,7000	17,3931	7,7506	3,9000
max	3,5970	0,7174	27,9967	7,8142	3,9000
c4_o_50x500 (3)					
moy	3,5259	0,7022	22,4648	7,7558	3,9000
min	3,5038	0,7000	17,0530	7,7510	3,9000
max	3,5869	0,7162	27,9529	7,7831	3,9000

TABLEAU E.3 – Scénario 1 (horizontal), avec crowding.

	x_1^c	x_2^c	x_3^c	$x_{1,1}^d$	$x_{1,2}^d$
c4_o_50x500 (1)					
moy	3,5001	0,7	17,0001	7,4116	3,5023
min	3,5000	0,7000	17,0000	7,3003	3,1694
max	3,5005	0,7000	17,0007	7,872	3,8729
c4_o_50x500 (2)					
moy	3,5006	0,7000	17,0001	7,4346	3,5282
min	3,5000	0,7000	17,0000	7,3000	3,1690
max	3,5019	0,7000	17,0007	7,7983	3,8975

TABLEAU E.4 – Scénario 2 (vertical), avec crowding.

RÉFÉRENCES BIBLIOGRAPHIQUES

- AIAA technical committee on multidisciplinary design optimization (MDO). white paper on current state of the art, 1991.
- Agte, J., de Weck, O., Sobieszczanski-Sobieski, J., Arendsen, P., Morris, A. et Spieck, M. : MDO : assessment and direction for advancement - an opinion of one international group. *Structural and Multidisciplinary Optimization*, 38(online first), avr. 2009.
- Aho, A. V., Hopcroft, J. E. et Ullman, J. D. : *Data Structures and Algorithms*. Addison-Wesley, 1983.
- Alberto, I., Azcarate, C., Mallor, F. et Mateo, P. : Multiobjective evolutionary algorithms. pareto ranking. vol. 27 de *Monografias del Seminario Matematico Garcia de Galdeano*, p. 25–35, 2003.
- Alexandrov, N. M. et Kodiyalam, S. : Initial results of an MDO method evaluation study. In *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, sept. 1998.
- Alexandrov, N. M. et Lewis, R. M. : Comparative properties of collaborative optimization and other approaches to MDO. Rap. tech., Institute for Computer Applications in Science and Engineering (ICASE), 1999.
- Andersson, J. : *Multiobjective optimization in engineering design*. Thèse de doctorat, Linköpings Universitet, 2001.
- Arrow, K. J. : A difficulty in the concept of social welfare. *The Journal of Political Economy*, 58(4):328–346, août 1950.
- Azarm, S., Tits, A. et Fan, M. : Tradeoff driven optimization-based design of mechanical systems. In *Proc. 4-th AIAA/ASAF/NASA/OAI Symp. on Multidisciplinary Analysis and Optimization*, p. 551–558, 1989.
- Balling, R. J. et Rawlings, M. R. : Collaborative optimization with disciplinary conceptual design. *Structural and multidisciplinary optimization*, 20(3):232–241, 2000.
- Balling, R. J. et Sobieszczanski-Sobieski, J. : Optimization of coupled systems : a critical overview of approaches. Rap. tech. 195019, NASA Langley Research Center, Hampton, VA 23681-0001, déc. 1994.
- Banach, S. : Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 1922.

- Belegund, A. et Salagame, R. : Optimization of laminated ceramic composites for minimum residual stress and cost. *Microcomputers in civil engineering*, 10(4):303–306, 1995.
- Bergounioux, M. : *Optimisation et controle des systèmes linéaires*. Cours. Dunod, Paris, France, 2001.
- Bloebaum, C. L. : *Formal and heuristic system decomposition methods in multidisciplinary synthesis*. Thèse de doctorat, Florida University, Gainesville, 1991.
- Boothroyd, G. : *Assembly automation and product design*. Marcel Dekker, 1992.
- Bosman, P. et Thierens, D. : The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.
- Braun, R. D. : *Collaborative optimization : an architecture for large-scale distributed design*. Thèse de doctorat, Stanford University, mai 1996.
- Brockhoff, D. : *Many-Objective Optimization and Hypervolume Based Search*. Thèse de doctorat, ETH Zurich, Zurich, Switzerland, 2009.
- Brockman, J. B. et Lokanathan, A. N. : A methodology for concurrent process-circuit optimization. *IEEE Transactions on Computer-Aided Design*, 18(7):889–902, 1999.
- Casparid, N., Leclerc, B. et Monjardet, B. : *Ensembles ordonnés finis : concepts, résultats et usages*, vol. 60 de *Mathématiques & Applications*. Springer, 2007.
- Chanron, V. et Lewis, K. : A study of convergence in decentralized design processes. *Research in Engineering Design*, 16(3):133–145, déc. 2005.
- Chanron, V., Singh, T. et Lewis, K. : An investigation of equilibrium stability in decentralized design using nonlinear control theory. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, sept. 2004.
- Chen, W. et Lewis, K. : A robust design approach for achieving flexibility in multidisciplinary design. *AIAA Journal*, 7(8):982–989, 1999.
- Cramer, E. J., Jr, J. E. D., Frank, P. D., Lewis, R. M. et Shubin, G. R. : Problem formulation for multidisciplinary optimization. Rap. tech. CRPC-TR93334, Center for Research on Parallel Computation, Rice University, 6100 South Main Street, Houston TX 77005, août 1993.
- Darwin, C. R. : *On the origin of species by means of natural selection, or The preservation of favoured races in the struggle for life*. D. Appleton and company, 1869.
- Deb, K. : *Multi-objective optimization using evolutionary algorithms*. Wiley-Intersciences series in systems and optimization. Wiley, juin 2001.
- Deb, K. et Agrawal, R. B. : Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.
- Deb, K. et Agrawal, R. B. : A Niche-Penalty approach for constraint handling in genetic algorithms. p. 235 — 243, 1999.

- Deb, K. et Beyer, H. : Self-Adaptive genetic algorithms with simulated binary crossover. *Complex Systems*, 9:431–454, 1999.
- Deb, K., Pratap, A., Agarwal, S. et Meyarivan, T. : A fast and elitist multiobjective genetic algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2):182–197, avr. 2002.
- Deb, K. et Tiwari, S. : Omni-optimizer : A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062–1087, mars 2008.
- DeMiguel, A. : *Two Decomposition Algorithms for Nonconvex Optimization Problems with Global Variables*. Thèse de doctorat, Stanford University, juin 2001.
- DeMiguel, A. et Murray, W. : An analysis of collaborative optimization methods. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, sept. 2000.
- DeMiguel, V. et Murray, W. : A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering*, 7(2):99–133, juin 2006.
- Dennis, J. E. J., Arroyo, S. F., Cramer, E. J. et Frank, P. D. : Problem formulations for systems of systems. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, p. 64–71, oct. 2005.
- Dennis, J. E. J. et Lewis, R. M. : Problem formulations and other optimization issues in multidisciplinary optimization. Rap. tech. CRPC-TR94469, Center for Research on Parallel Computation, Rice University, Houston, USA, 1994.
- Dépincé, P., Guédas, B. et Picard, J. : Multidisciplinary and multiobjective optimization : Comparison of several methods. Seoul, Korea, mai 2007.
- Ehrgott, M. : *Multicriteria Optimization*. Springer Berlin / Heidelberg, 2 éd., 2005.
- Engau, A. et Wiecek, M. : 2D decision-making for multicriteria design optimization. *Structural and Multidisciplinary Optimization*, 34(4):301–315, oct. 2007.
- Fadel, G. et Wiecek, M. M. : Integration through optimization : An interdisciplinary Graduate-Level course in engineering design and mathematical optimization. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, sept. 2006.
- Farhang-Mehr, A. et Tumer, I. Y. : Two-Level optimization of systems health monitoring in systems with Top-Down hierarchical architecture. In Chang, F., éd. : *The 5th International Workshop on Structural Health Monitoring*, CA, USA, sept. 2005.
- Farhang-Mehr, A., Tumer, I. Y. et Barszcz, E. : Optimal design of integrated systems health management (ISHM) for improving the safety of NASA’s exploration missions : A multidisciplinary design approach. In *6th World Congress on Structural and Multidisciplinary Optimization*, juin 2005.
- Farina, M. et Amato, P. : On the optimal solution definition for many-criteria optimization problems. In *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622)*, p. 233–238, New Orleans, LA, USA, 2002.

- Ficici, S. et Pollack, J. : A Game-Theoretic approach to the simple coevolutionary algorithm. *In Parallel Problem Solving from Nature PPSN VI*, p. 467–476. 2000.
- Fonseca, C. M. et Fleming, P. J. : Genetic algorithms for multiobjective optimization : Formulation discussion and generalization. *In Proceedings of the 5th International Conference on Genetic Algorithms*, p. 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.
- Gantois, K. et Morris, A. : The multi-disciplinary design of a large-scale civil aircraft wing taking account of manufacturing costs. *Structural and Multidisciplinary Optimization*, 28(1):31–46, 2004.
- Giassi, A. : *Optimisation et conception collaborative dans le cadre de l'ingénierie simultanée*. Thèse de doctorat, École Centrale de Nantes, sept. 2004.
- Goldberg, D. E. : *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.
- Golinski, J. : Optimal synthesis problems solved by means of nonlinear programming and random methods. *Journal of Mechanisms*, 5:287–309, 1970.
- Gopalakrishnan, J., Singh, T. et Lewis, K. : Enhanced convergence in distributed design processes. *In Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, sept. 2006.
- Granvilliers, L. et Benhamou, F. : Realpaver : An interval solver using constraint satisfaction techniques. *ACM Trans. on Mathematical Software*, 32:138–156, 2006.
- Gray, F. : Pulse code communication, u.s. patent 2,632,058, mars 1953.
- Gunawan, S., Farhang-Mehr, A. et Azarm, S. : Multi-level multi-objective genetic algorithm using entropy to preserve diversity. *In Evolutionary Multi-Criterion Optimization*, p. 10. 2003.
- Gunawan, S., Farhang-Mehr, A. et Azarm, S. : On maximizing solution diversity in a multiobjective multidisciplinary genetic algorithm for design optimization. *Mechanics based design of structures and machines*, 32(4):491–514, 2004.
- Guédas, B. et Dépincé, P. : Coupling functions treatment in a bi-level optimization process. *In Seventh International Symposium on Tools and Methods for Concurrent Engineering (TMCE 2008)*, Izmir, Turkey, 2008a.
- Guédas, B. et Dépincé, P. : Coupling functions treatment in a Bi-Level optimization process. *Strojniški vestnik*, p. 413–425, 2008b.
- Hajela, P., Lee, E. et Lin, C. Y. : Genetic algorithms in structural topology optimization. *In Proceedings of the NATO Advanced Research Workshop on Topology Design of Structures*, 1993.
- Hodges, D. et Pierce, G. A. : *Introduction to Structural Dynamics and Aeroelasticity*. Cambridge University Press, 2002.
- Holland, J. H. : Outline for a logical theory of adaptive systems. *Journal of the ACM*, 9(3):297–314, juil. 1962.

- Holland, J. H. : *Adaptation in Natural and Artificial Systems*. MIT Press, 2^e éd., 1992.
- Hughes, E. J. : On the optimal solution definition for many-criteria optimization problems. In *J. : Evolutionary Many-Objective Optimisation : Many Once or Evolutionary Computation 'Congress (CEC'05)*, p. 222–227, Edinburgh, UK, 2005.
- Hulme, K. F. et Bloebaum, C. L. : A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE. *Structural and Multidisciplinary Optimization*, 19(1):17–35, mars 2000.
- Ishibuchi, H., Tsukamoto, N. et Nojima, Y. : Evolutionary many-objective optimization : A short review. In *Proc. of 2008 IEEE Congress on Evolutionary Computation*, p. 2424–2431, Hong Kong, China, juin 2008.
- Jacquenot, G. : *Méthode générique pour l'optimisation d'agencement géométrique et fonctionnel*. Thèse de doctorat, École Centrale de Nantes, Nantes, France, jan. 2010.
- Keane, A. J. et Nair, P. B. : *Computational Approaches for Aerospace Design*. John Wiley & Sons, Ltd, Chichester, UK, 2005.
- Klein, M., Sayama, H., Faratin, P. et Bar-Yam, Y. : The dynamics of collaborative design : Insights from complex systems and negotiation research. In *Complex Engineered Systems*, p. 158–174. 2006.
- Kosaraju, R. S. : Non publié, 1978.
- Lee, J. et Jeong, H. : A decomposition based design method coordinated by disciplinary subspace optimization. *JSME International Journal Series C*, 49(3):935–941, 2006.
- Lewis, K. et Mistree, F. : Modeling interactions in multidisciplinary design - a game theoretic approach. *AIAA Journal of Aircraft*, 35(8):1387–1392, 1997.
- Majd, B. A. E. : *Algorithmes hiérarchiques et stratégies de jeux pour l'optimisation multidisciplinaire. Application à l'optimisation de la voilure d'un avion d'affaires*. Thèse de doctorat, Université de Nice-Sophia Antipolis, 2007.
- Masmoudi, M. et Parte, Y. S. : Disciplinary interaction variable elimination (DIVE) approach for MDO. In *European Conference on Computational Fluid Dynamics 2006*, 2006.
- Mouelhi, O. : *Contribution à l'optimisation multiobjectif en conception multidisciplinaire*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2010.
- Nair, P. B. et Keane, A. J. : Coevolutionary architecture for distributed optimisation of complex coupled systems. *AIAA Journal*, 40(7):1434–1443, 2002.
- Nash, J. F. : Non-Cooperative games. *The Annals of Mathematics*, 54(2):286–295, sept. 1951.
- Neumann, J. et Morgenstern, O. : *Theory Of Games And Economic Behavior*. Princeton University Press, 1944.
- Papalambros, P. Y. : The optimization paradigm in engineering design : promises and challenges. *computer-Aided Design*, 34(12):939–951, 2002.

- Potter, M. A. : *The Design and Analysis of a Computational Model of Cooperative Coevolution*. Thèse de doctorat, George Mason University, Fairfax, Virginia, USA, 1997.
- Rabeau, S. : *Optimisation Multi-objectif en conception collaborative*. Thèse de doctorat, École Centrale de Nantes, 2007.
- Rechenberg, I. : Cybernetic solution path of an experimental problem. Rap. tech., 1965.
- Renaud, J. E. et Gabriele, G. A. : Improved coordination in nonhierarchic system optimization. *AIAA journal*, 31(12):2367–2373, 1993.
- Schaffer, D. J. : *Some Experiments in Machine Learning using Vector Evaluated Genetic Algorithms*. Thèse de doctorat, Vanderbilt University, 1984.
- Schröder, B. S. W. : *Ordered Sets : An introduction*. Birkhäuser, déc. 2002.
- Shankar, J., Haftka, R. T. et Watson, L. T. : Computational study of a nonhierarchical decomposition algorithm. *Computational Optimization and Applications*, 2(3):273–293, nov. 1993.
- Sharir, M. : A strong-connectivity algorithm and its applications in data-flow analysis. *Computers and Mathematics with Applications*, 7(1):67–72, 1981.
- Shin, M. et Park, G. : Multidisciplinary design optimization based on independent subspaces. *International Journal for Numerical Methods in Engineering*, 64(5):599 – 617, oct. 2005.
- Sim, K. et Kim, J. : Solution of multiobjective optimization problems : coevolutionary algorithm based on evolutionary game theory. *Artificial Life and Robotics*, 8(2):174–185, juin 2004.
- Sobieszczanski-Sobieski, J. : Optimization by decomposition : A step from hierarchic to Non-Hierarchic systems. Rap. tech., 1988a.
- Sobieszczanski-Sobieski, J. : Sensitivity analysis and multidisciplinary optimization for aircraft design : Recent advances and results. Rap. tech. NAS 1.15 :100630 ; NASA-TM-100630, NASA, Langley Research Center, 1988b.
- Sobieszczanski-Sobieski, J., Agte, J. S. et Jr, R. R. S. : Bi-Level integrated system synthesis. Rap. tech. TM-1998-208715, NASA, août 1998.
- Sobieszczanski-Sobieski, J., Emiley, M. S., Agte, J. S. et Sandusky, R. R. J. : Advancement of Bi-Level integrated system synthesis (BLISS). *In AIAA 38th Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2000.
- Sobieszczanski-Sobieski, J. et Haftka, R. T. : Multidisciplinary aerospace design optimization : survey of recent developments. *Structural and Multidisciplinary Optimization*, 14(1):1–23, août 1997.
- Stalk, G. et Hout, T. M. : *Competing against time*. Simon & Schuster, fév. 2003.
- Stalk, G. J. : Time—The next source of competitive advantage. *Harvard Business Review*, p. 41, juil. 1988.

- Tarjan, R. : Depth-First search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, juin 1972.
- Tava, M. et Suzuki, S. : Integrated multidisciplinary and multicriteria optimization of a space transportation system and its trajectory. *In 54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, oct. 2003.
- Tosserams, S., Etman, L. F. P. et Rooda, J. E. : A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization*, 38(online first), 2009.
- Tribes, C., Dubé, J. et Trépanier, J. : Decomposition of multidisciplinary optimization problems : formulations and application to a simplified wing design. *Engineering Optimization*, 37(8):775 – 796, déc. 2005.
- Wagner, T., Beume, N. et Naujoks, B. : Pareto-, aggregation-, and Indicator-Based methods in Many-Objective optimization. *In Evolutionary Multi-Criterion Optimization*, p. 742–756. 2007.
- Winner, R. I., Pennell, J. P., Bertrand, H. E. et Slusarczuk, M. M. : The role of concurrent engineering in weapons system acquisition. *Rap. tech.*, déc. 1988.
- Wujek, B. A., Renaud, J. E., Batill, S. M. et Brockman, J. B. : Concurrent subspace optimization using design variable sharing in a distributed computing environment. *Concurrent Engineering*, 4(4):361–377, 1996.
- Xu, L., Yan, L. et Chen, X. Q. : Enhanced bi-level integrated system synthesis 2000 (EBLISS 2000). Palm Springs, California, USA, mai 2009.
- Yi, S. I., Shin, J. K. et Park, G. J. : Comparison of MDO methods with mathematical examples. *Structural and Multidisciplinary Optimization*, 35(5):391–402, mai 2008.
- Zitzler, E. et Künzli, S. : Indicator-Based selection in multiobjective search. *In Parallel Problem Solving from Nature - PPSN VIII*, p. 832–842. 2004.
- Zitzler, E. et Thiele, L. : An evolutionary algorithm for multiobjective optimization : The strength pareto approach. *Rap. tech.* 43, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1998.
- Zitzler, E. et Thiele, L. : Multiobjective evolutionary algorithms : A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. et da Fonseca, V. G. : Performance assessment of multiobjective optimizers : An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7:117—132, 2003.

Titre Proposition de compromis pour le calcul de solutions préférées à l'aide d'un algorithme évolutionnaire multiobjectif en optimisation multidisciplinaire

Résumé L'optimisation multidisciplinaire fait référence à la conception et l'optimisation de problèmes d'ingénierie nécessitant l'intervention simultanée d'au moins deux disciplines, chacune pouvant avoir plus d'un objectif à optimiser. Les méthodes usuelles n'abordent pas le cas où chaque discipline a un problème d'optimisation multiobjectif à résoudre. Des méthodes ont été récemment proposées, transformant le problème d'optimisation multidisciplinaire en un problème d'optimisation multiobjectif. Ces méthodes reposent sur des algorithmes évolutionnaires multiobjectifs. Cependant, l'ensemble des solutions obtenues ne reflète pas les préférences disciplinaires : des solutions peuvent être globalement efficaces alors qu'elles sont localement dominées. En nous basant sur les propriétés des relations d'ordre, nous proposons quatre définitions de compromis qui tiennent compte du regroupement des objectifs en disciplines. Les propriétés théoriques de ces compromis sont étudiées, et notamment leur capacité à converger vers l'ensemble de solutions attendues, lorsqu'ils sont utilisés avec des algorithmes évolutionnaires. Ces compromis sont intégrés dans un algorithme évolutionnaire multiobjectif. Des analyses expérimentales de cet algorithme sur les quatre compromis proposés sont effectuées. Elles confirment les prédictions théoriques et montrent la pertinence des solutions obtenues.

Mots-clés optimisation multidisciplinaire; optimisation multiobjectif; algorithmes évolutionnaires; préférences; compromis; ensembles ordonnés.

Title Proposition of compromises for the computation of preferred solutions with a multiobjective evolutionary algorithm in multidisciplinary design optimisation

Abstract Multidisciplinary Design Optimization refers to the design and optimization of complex engineering systems, requiring simultaneous work of at least two disciplines. Each of them could have more than one objective to achieve. The traditional methods do not consider the case where each discipline has a multiobjective optimization problem to solve. Recently, methods which transform the multidisciplinary design optimization problem into a multiobjective optimization problem have been proposed. These methods are based on evolutionary multiobjective optimization algorithms. Unfortunately, the set of solutions found does not reflect the preferences of the disciplines : solutions can be globally efficient whereas they are locally dominated. Using the properties of order relations, we propose four definitions of the compromise between disciplines which take into account the grouping of the objectives. The theoretical properties of these compromises are studied, and especially their capacity to converge to the wanted solution set when they are used within an evolutionary algorithm. The compromises are integrated within an evolutionary multiobjective optimization algorithm. Experimental analysis of this algorithm with the four compromises are conducted. They confirm the theoretical predictions and show the relevance of the obtained solutions.

Keywords multidisciplinary design optimization; multiobjective optimization; evolutionary algorithms; preferences; compromise; ordered sets.