



**HAL**  
open science

# Métaheuristiques Coopératives : du déterministe au stochastique

Laetitia Jourdan

► **To cite this version:**

Laetitia Jourdan. Métaheuristiques Coopératives : du déterministe au stochastique. Modélisation et simulation. Université des Sciences et Technologie de Lille - Lille I, 2010. tel-00523274

**HAL Id: tel-00523274**

**<https://theses.hal.science/tel-00523274v1>**

Submitted on 4 Oct 2010

**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.



Année : **2010**

Numéro d'ordre : **40328**

Habilitation à Diriger les Recherches de l'Université Lille I

**Discipline : Sciences Pour l'Ingénieur**

---

## Métaheuristiques Coopératives : du déterministe au stochastique

---

Par : **Laetitia JOURDAN**

Sous la direction de El-Ghazali TALBI, Professeur des Universités

**Membres du jury :**

- Examineur : PIERRE BOULET, Professeur des Universités, Lille I
- Examineur : THOMAS STUTZLE, Chercheur HDR, Université Libre de Bruxelles, Belgique
- Rapporteur : PASCAL BOUVRY, Professeur des Universités, Luxembourg
- Rapporteur : JIN-KAO HAO, Professeur des Universités, Angers
- Rapporteur : CELSO RIBEIRO, Professeur des Universités, Brésil

**Date de soutenance : 15/09/2010**



## Remerciements

Je veux tout d'abord remercier Max DAUCHET directeur de centre de recherche de l'INRIA Lille Nord Europe et Sophie TISON directrice du Laboratoire d'Informatique Fondamentale de Lille pour leur accueil.

Je remercie les membres du Jury pour avoir accepté de participer à ce jury et d'avoir passer du temps à examiner mes travaux.

Je remercie l'équipe DOLPHIN dans laquelle j'ai effectué les travaux présentés dans cette HDR. Tout d'abord El-Ghazali TALBI pour sa confiance dans les tâches et les encadrements qu'il m'a confiés. Cela m'a permis de prendre confiance en moi. Je tiens ensuite à remercier Clarisse et Sébastien, pour leurs encouragements et le temps passé à relire ce manuscrit et celui très intéressant passé à travailler ensemble. Je remercie également deux anciens de l'équipe : Matthieu et Nicolas pour les bons moments passés à discuter recherche et autres depuis nos années "thèse". Je tiens à remercier mes thésards actuels et passés : Arnaud, Jean-Charles, Khedidja, Marie-Eléonore et Mostepha, plus particulièrement Arnaud et Jean-Charles à qui je souhaite une longue et belle carrière d'enseignant-chercheur.

Je remercie mes collègues de l'INRIA et du LIFL pour leur soutien et plus particulièrement Anne-Françoise, Eric, David et Laurence. Je remercie ensuite mes amis, qui permettent à mon cerveau d'appliquer d'autres algorithmes que les métaheuristiques : Ben, Steph et Cec, Sylvain, Ludo, Simon et Anne-Hélène, Hélène et David, Jérém ...

Mes remerciements vont enfin à ma famille : mes parents qui se sont toujours intéressés à mes travaux et qui m'ont toujours poussée à persévérer, mon frère qui a toujours eu un regard objectif sur mon travail, et enfin mon mari et ma fille qui m'ont soutenu et compris quand, le soir ou le week-end, il faut que "*Maman travaille*".

*Il vaut mieux viser la perfection et la manquer que viser l'imperfection et l'atteindre.* [Bertrand Russell]

*No great achievement is possible without persistent work.* [Bertrand Russell]

# Table des matières

<b>I</b>	<b>Introduction générale</b>	<b>1</b>
1	Parcours depuis le doctorat	3
2	Introduction	5
2.1	Contexte . . . . .	5
2.1.1	Optimisation combinatoire . . . . .	5
2.1.2	Méthodes exactes . . . . .	5
2.1.3	Les métaheuristiques . . . . .	6
2.1.4	L'optimisation multiobjectif . . . . .	6
2.2	Organisation du document . . . . .	11
<b>3</b>	<b>ParadisEO</b>	<b>13</b>
3.1	Contexte . . . . .	14
3.1.1	Description . . . . .	14
3.1.2	Caractéristiques principales . . . . .	15
3.2	ParadisEO-MO . . . . .	16
3.2.1	Etat des lieux . . . . .	16
3.2.2	Contributions . . . . .	17
3.3	ParadisEO-MOEO . . . . .	19
3.3.1	Etat de l'art . . . . .	19
3.3.2	Contributions . . . . .	19
3.4	Conclusion . . . . .	24
<b>II</b>	<b>Coopération de méthodes dans le cadre déterministe</b>	<b>27</b>
<b>4</b>	<b>Coopération métaheuristiques-métaheuristiques</b>	<b>29</b>
4.1	Etat de l'art . . . . .	30
4.1.1	Notion de coopération . . . . .	30
4.1.2	Niveaux de Coopération . . . . .	31
4.1.3	Modes de Coopération . . . . .	31
4.1.4	Classes hiérarchiques . . . . .	31
4.2	Méthodes coopératives multiobjectives . . . . .	35
4.3	Nouvelles métaheuristique coopératives . . . . .	38
4.3.1	Une approche de coopération multiobjectif co-évolutionnaire de bas niveau	38

4.3.2	Une approche de coopération multiobjectif de haut niveau en mode relais	45
4.4	Conclusion . . . . .	50
<b>5</b>	<b>Coopération métaheuristiques-méthodes exactes</b>	<b>51</b>
5.1	Extension de la taxonomie . . . . .	52
5.1.1	Les classes hiérarchiques . . . . .	52
5.1.2	Classification à plat . . . . .	56
5.1.3	Implémentation . . . . .	58
5.2	Etat de l'art des coopérations Méta/Exacte . . . . .	59
5.2.1	Grammaire et vocabulaire de la classification . . . . .	59
5.2.2	Classification . . . . .	60
5.3	Conclusion . . . . .	63
<b>6</b>	<b>Coopération métaheuristiques-extraction de connaissances</b>	<b>65</b>
6.1	Brève introduction à l'extraction de connaissances . . . . .	67
6.1.1	Classification supervisée . . . . .	67
6.1.2	Catégorisation ou Clustering . . . . .	68
6.1.3	Recherche de règles d'association . . . . .	68
6.1.4	Sélection d'attributs . . . . .	68
6.2	Synthèse sur les coopérations entre métaheuristiques et extraction de connaissances	68
6.2.1	Hybridation de bas niveau relais . . . . .	69
6.2.2	Hybridation de bas niveau co-évolutionnaire . . . . .	70
6.2.3	Hybridation de haut niveau relais . . . . .	70
6.2.4	Hybridation de haut niveau co-évolutionnaire . . . . .	71
6.3	LEMMO . . . . .	72
6.3.1	LEM . . . . .	72
6.3.2	Adaptation de LEM au cas multiobjectif . . . . .	72
6.3.3	Différents modèles pour LEMMO . . . . .	73
6.3.4	Cadre d'étude de LEMMO : design de systèmes hydrauliques . . . . .	74
6.4	Conclusion et Perspectives . . . . .	78
<b>III</b>	<b>Vers la coopération de méthodes dans le cadre stochastique</b>	<b>79</b>
<b>7</b>	<b>Résolution de problèmes dans un cadre incertain</b>	<b>85</b>
7.1	Optimisation multiobjectif en environnement incertain . . . . .	87
7.1.1	À propos de robustesse . . . . .	87
7.1.2	Incertitude en optimisation multiobjectif . . . . .	87
7.1.3	Approches métaheuristiques . . . . .	90
7.2	Trouver des solutions par approche multiobjectif à un problème monoobjectif incertain . . . . .	92
7.2.1	Cadre d'étude : problème de tournée de véhicule avec domaine incertain (SVRP) . . . . .	92
7.2.2	Une modélisation pour trouver des solutions robustes . . . . .	92
7.2.3	Algorithmes . . . . .	94
7.2.4	Résultats et discussion . . . . .	94

7.2.5	Protocole . . . . .	94
7.3	Résoudre un problème stochastique multiobjectif . . . . .	98
7.3.1	Problématique . . . . .	98
7.3.2	Modélisation de l'incertitude . . . . .	98
7.3.3	Approches métaheuristiques pour l'optimisation multiobjectif en environnement incertain . . . . .	99
7.3.4	Cadre d'étude : le Flowshop de permutation . . . . .	100
7.3.5	Résultats et discussion . . . . .	102
7.4	Conclusion . . . . .	103
<b>8</b>	<b>Méthodes coopératives pour l'optimisation dynamique</b>	<b>105</b>
8.1	Optimisation dynamique : Généralités . . . . .	106
8.1.1	Définition et contexte . . . . .	106
8.1.2	Classification générale des méthodes . . . . .	106
8.1.3	Mesures de qualité . . . . .	106
8.2	Cadre d'étude : le VRP dynamique . . . . .	107
8.2.1	Le problème statique de tournées de véhicules . . . . .	107
8.2.2	Le problème dynamique de tournées de véhicules . . . . .	107
8.2.3	Intérêts industriels . . . . .	110
8.3	Algorithmes hybrides pour le DVRP . . . . .	110
8.3.1	Représentation et initialisation . . . . .	110
8.3.2	Un algorithme à base d'essaim de particules : APSO . . . . .	110
8.3.3	Un algorithme à voisinage variable : VNS . . . . .	111
8.3.4	Recherche locale basée sur le 2-opt pour l'hybridation . . . . .	113
8.4	Résultats . . . . .	114
8.4.1	Jeux de données . . . . .	114
8.4.2	Paramètres . . . . .	114
8.4.3	Résultats numériques . . . . .	115
8.5	Conclusion . . . . .	116
<b>IV</b>	<b>Conclusion Générale et Perspectives</b>	<b>121</b>
<b>V</b>	<b>Annexes</b>	<b>143</b>
	Annexes	145
<b>9</b>	<b>CV Détaillé</b>	<b>145</b>
9.1	Publications . . . . .	155
9.1.1	Invited tutorial . . . . .	164
<b>10</b>	<b>Publications Majeures</b>	<b>165</b>





Première partie

Introduction générale



# Chap. 1

## Parcours depuis le doctorat

Ma thèse, effectuée de 2000 à 2003 au LIFL de l'université de Lille 1, portait sur la modélisation de problèmes d'extraction de connaissances et leur résolution grâce à des métaheuristiques. Les problèmes traités lors de cette thèse étaient relatifs à la génomique et notamment à la recherche des facteurs favorisant des maladies métaboliques comme le diabète. Je me suis ensuite intéressée, lors de mon postdoctorat à l'Université d'Exeter (Angleterre), à l'intégration de connaissances dans des algorithmes multiobjectif afin d'accélérer et d'améliorer le processus de recherche. Le domaine d'application était alors plus orienté vers l'ingénierie et notamment les systèmes hydroliques.

J'ai ensuite été nommée chargée de recherche INRIA dans l'équipe Dolphin au centre INRIA Lille Nord Europe en Septembre 2005. Mon expérience de postdoctorat m'a permis de me focaliser sur l'hybridation de méthodes aussi bien en optimisation mono-objectif que multiobjectif. Depuis mon recrutement, je participe activement à l'extension de la plate-forme ParadisEO qui avait été conçue initialement lors de la thèse dans l'équipe de Sébastien Cahon. Ma participation se concentre plus particulièrement sur les aspects coopérations de méthodes et sur le multiobjectif avec la partie ParadisEO-MOEO. Mon intérêt croissant pour le multiobjectif m'a amené à participer au groupe de travail Roadef PM2O<sup>1</sup> dont je suis actuellement la coordinatrice avec Nicolas Jozefowicz (LAAS Toulouse) et Mathieu Basseur (LERIA Angers) depuis fin 2007. Dans ce cadre, nous organisons régulièrement des réunions et des sessions dans des conférences. Cela m'a notamment amené à co-éditer un livre chez Springer sur l'optimisation multiobjectif dans la collection "Studies in computational intelligence" avec Clarisse Dhaenens et Carlos Coello Coello.

Depuis 2007, je m'intéresse également aux problèmes stochastiques et à l'utilisation de métaheuristiques pour les résoudre. Un axe du projet de recherche DOLPHIN a également suivi cette piste et nos travaux portent maintenant sur l'optimisation dans un contexte incertain et dans un contexte dynamique. Nous travaillons dans le cadre d'une équipe associée INRIA avec l'équipe d'Enrique Alba de l'université de Malaga (Espagne) pour l'application de nouveaux paradigmes de métaheuristiques dans un contexte dynamique.

Les travaux de ce mémoire ont été appliqués à différentes problématiques dans un contexte déterministe ou stochastique. Le tableau 1.1 synthétise les différentes problématiques abordées dans ce manuscrit ainsi que leur contexte : déterministe, incertain ou dynamique. L'application

---

1. <http://www.lifl.fr/PM20/> (Programmation Mathématique Multi-Objective)

Problématique	Contexte déterministe	Contexte Incertain	Contexte Dynamique
Docking moléculaire	X		
Hydrologie	X		
Flowhsop	X	X	
Ringstar	X		
VRP		X	X

TABLE 1.1 – Synthèse des problématiques traitées dans le manuscrit et de leur contexte.

sur le docking moléculaire a été traitée au sein du projet ANR Dock<sup>2</sup>.

L’environnement dynamique de Dolphin m’a permis d’encadrer différents étudiants en stage de DEA /Master recherche d’Informatique dont certains ont poursuivi en thèse sous ma co-direction. Une liste détaillée de mes encadrements figure en annexe dans la partie CV de ce document. Les travaux présentés dans ce mémoire s’appuient en partie sur les travaux menés avec ces étudiants notamment ceux de DEA et de thèse de Jean-Charles Boisson, Arnaud Liefoghe et Mostepha Redouane Khouadjia.

Ci-dessous se trouve la liste de mes co-auteurs depuis mon doctorat.

- Enrique Alba (Université de Malaga, Espagne)
- Matthieu Basseur (Université d’Angers, France)
- Jean-Charles Boisson (Université de Lille puis Université de Reims, France)
- Edmund K. Burke (Université de Nottingham, Angleterre)
- David W. Corne (Université d’Exeter puis Edhimbou, Angleterre)
- Clarisse Dhaenens (Université de Lille 1, France)
- Sophie Gallina (Institut Pasteur de Lille, France)
- José García-Nieto (Université de Malaga, Espagne)
- Dragos Horvath (CNRS, Strasbourg, France)
- Jérémie Humeau (INRIA Lille Nord Europe, France)
- Nicolas Jozefowicz (LAAS, INSA Toulouse, France)
- Thomas Legrand (INRIA Lille Nord Europe, France)
- Arnaud Liefoghe (Université de Lille 1, France)
- Salma Mesmoudi (Université Paris 6, France)
- Christian Rolando (Dpt Chimie, Université de Lille 1, France)
- Dragan A. Savic (Université d’Exeter, Angleterre)
- Oliver Schütze (Université de Mexico, Mexique)
- El-Ghazali Talbi (Université de Lille 1, France)
- Godfrey A. Walters (Université d’Exeter, Angleterre)
- Jean Luc Wojkiewicz (Dpt Matériaux et Polymères, Ecole des mines de Douai, France)

Pour information, nous avons opté pour deux styles de référence différentes : [nombre] indiquant les références de la bibliographie générale et [LetteNombre] indiquant les références de ma bibliographie personnelle.

---

2. <http://dockinggrid.gforge.inria.fr/>

## Chap. 2

# Introduction

Ce document synthétise une partie de mes travaux de recherche effectués au sein du projet INRIA DOLPHIN et de l'équipe OPAC/DOLPHIN du laboratoire LIFL CNRS/Université de Lille 1 dirigés par El-Ghazali TALBI. Avant de rentrer dans le cœur du travail effectué depuis la fin de mon doctorat, je vais donner les principales notions nécessaires à la compréhension du manuscrit puis le plan du document sera décrit.

### 2.1 Contexte

#### 2.1.1 Optimisation combinatoire

Un problème d'optimisation combinatoire est généralement caractérisé par un ensemble fini de solutions admissibles  $\Omega$  et une fonction objectif  $f : \Omega \rightarrow \mathbb{R}$  associant une valeur à chaque solution admissible. La résolution du problème consiste à déterminer la (ou les) solution(s) de  $\Omega$  minimisant ou maximisant  $f$ .

Dans la résolution de problèmes d'optimisation, deux types de méthodes sont souvent mises en opposition : les *méthodes exactes* et les *méthodes approchées*.

#### 2.1.2 Méthodes exactes

Les méthodes exactes ont l'avantage d'obtenir des solutions dont l'optimalité est garantie. Ces méthodes sont génériques et demandent souvent une particularisation vis-à-vis d'un problème spécifique. Parmi les méthodes exactes, on peut citer :

- la programmation dynamique consistant à placer le problème dans une famille de problèmes de même nature mais de difficulté différente puis à trouver une relation de récurrence liant les solutions optimales de ces problèmes ;
- le Branch & Bound consistant à faire une énumération implicite en séparant le problème en sous-problèmes et en évaluant ceux-ci à l'aide d'une relaxation (continue ou lagrangienne principalement) jusqu'à ne plus avoir que des problèmes faciles à résoudre ou dont on sait avec certitude qu'ils ne peuvent pas contenir de solution optimale ;
- les méthodes polyédrales consistant à ajouter progressivement des contraintes supplémentaires afin de ramener le domaine des solutions admissibles à un domaine convexe (sans en enlever la ou les solutions optimales bien évidemment).

Il existe aussi des outils logiciels génériques (AMPL, CPLEX, LINDO, MPL, OMP, XPRESS...) permettant de résoudre l'ensemble des problèmes pouvant s'écrire sous la forme algébrique d'un problème en variables binaires, entières ou mixtes.

Les méthodes exactes bien que fournissant une solution optimale au problème traité ont généralement le défaut d'être très coûteuses en temps d'exécution pour des problèmes difficiles. Au contraire, les méthodes approchées visent à générer des solutions de haute qualité en un temps de calcul raisonnable, mais il n'existe aucune garantie de trouver la solution optimale. Ces méthodes approchées sont elles-mêmes divisées en deux sous-catégories : les *algorithmes d'approximation* et les *heuristiques*. Contrairement aux heuristiques, les algorithmes d'approximation garantissent la qualité de la solution trouvée par rapport à l'optimal. Enfin, il existe encore deux sous-classes de méthodes heuristiques : les heuristiques spécifiques à un problème donné, et les *métaheuristiques* qui regroupent des méthodes plus génériques.

### 2.1.3 Les métaheuristiques

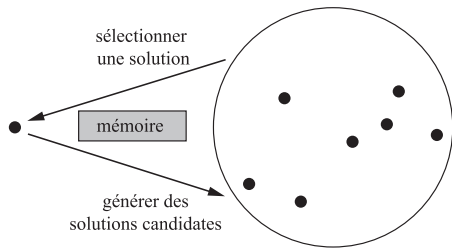
Les métaheuristiques sont des algorithmes pouvant être appliqués à la résolution d'un grand nombre de problèmes d'optimisation. Elles peuvent être vues comme des méthodologies de niveau supérieur servant à guider la conception d'heuristiques implicitement dédiées à la résolution d'un problème spécifique. Elles sont donc composées d'éléments génériques ou invariants, ainsi que d'éléments spécifiques au problème considéré, tels que la représentation ou l'évaluation d'une solution.

Durant les vingt dernières années, les métaheuristiques ont reçu un intérêt grandissant et ont montré leur efficacité dans de vastes domaines d'application en résolvant de nombreux problèmes d'optimisation [165]. Deux types de métaheuristiques peuvent être distingués : les *métaheuristiques à base de solution unique* (S-META) et les *métaheuristiques à base de population* (P-META). Les S-META (telles que les algorithmes de recherche locale [138, 139], de recherche tabou [67], de recuit simulé [31], etc.) manipulent et transforment une seule solution durant le processus de recherche, alors que dans les P-META (algorithmes évolutionnaires, algorithmes à essaim de particules, etc.), un ensemble de solutions, appelé *population*, évoluent en parallèle. En termes de conception, deux critères contradictoires sont à prendre en compte lors du développement d'une métaheuristique : l'exploration de l'espace de recherche (*diversification*), et l'exploitation des meilleures solutions trouvées (*intensification*). Les S-META sont plutôt axées sur l'exploitation de l'espace de recherche. Les régions prometteuses sont explorées localement dans l'espoir de trouver de meilleures solutions. Les P-META sont généralement plutôt exploratoires et permettent une meilleure diversification de l'espace de recherche. Les principes généraux d'une S-META et d'une P-META sont respectivement illustrés sur les figures 2.1 et 2.2 [165].

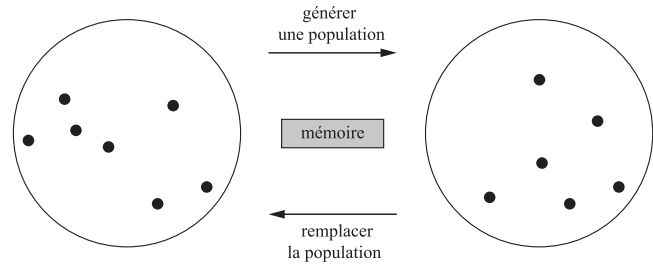
### 2.1.4 L'optimisation multiobjectif

L'optimisation multiobjectif est un champ de recherche de plus en plus étudié car de nombreux problèmes rencontrés sont intrinsèquement multiobjectif car ils nécessitent d'optimiser plusieurs critères simultanément.

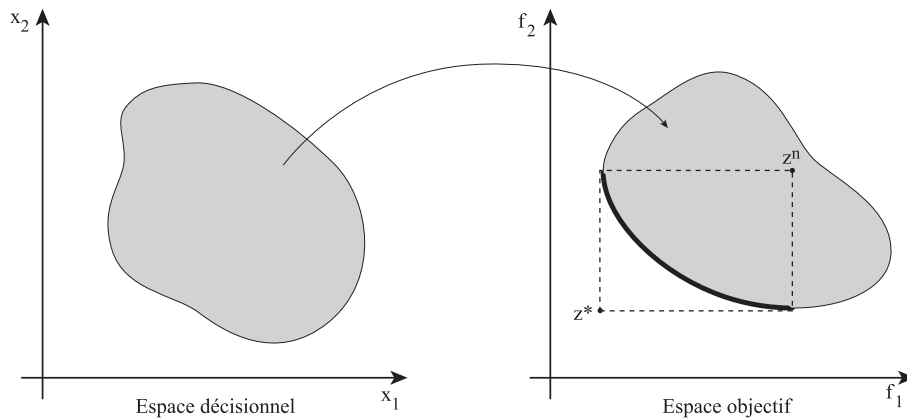
Les racines de l'optimisation multiobjectif se trouvent dans les travaux des économistes Edgeworth [50] et Pareto [142]. Ces travaux ont été utilisés plus tard en sciences de l'ingénieur.



**Fig. 2.1** – Illustration des principes généraux d’une métaheuristique à base de solution unique (S-META).



**Fig. 2.2** – Illustration des principes généraux d’une métaheuristique à base de population (P-META).



**Fig. 2.3** – Espace décisionnel et espace objectif d’un problème d’optimisation multiobjectif (exemple avec deux variables de décision  $X_1$  et  $X_2$  et deux fonctions objectif  $f_1$  et  $f_2$ ).

**Définition d’un problème d’optimisation combinatoire multiobjectif**

Un problème d’optimisation combinatoire multiobjectif (*multiobjective combinatorial optimization problem, MOP*) peut être défini comme suit :

$$(MOP) = \begin{cases} \text{“min”} & f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{tel que :} & x \in X, \end{cases} \tag{2.1}$$

où  $n$  est le nombre d’objectifs ( $n \geq 2$ ),  $X$  est l’ensemble des solutions admissibles dans l’*espace décisionnel*, et  $x = (x_1, x_2, \dots, x_k) \in X$  est un vecteur représentant les variables de décision. Dans le cas combinatoire,  $X$  est un ensemble discret. À chaque solution  $x \in X$  est associé un vecteur objectif  $z \in Z$  sur la base d’un vecteur de fonctions  $f : X \rightarrow Z$  tel que  $z = (z_1, z_2, \dots, z_n) = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ , où  $Z = f(X)$  représente l’ensemble des points réalisables de l’*espace objectif*. La figure 2.3 présente la liaison entre l’*espace décisionnel* et l’*espace objectif*. Sans perte de généralité, nous supposons par la suite que  $Z \subseteq \mathbb{R}^n$  et que toutes les fonctions objectif sont à minimiser.



### Dominance et notion d'optimalité

Dans le domaine de l'optimisation multiobjectif, le décideur évalue généralement une solution par rapport à chacun des critères, et se positionne donc naturellement dans l'espace objectif. Néanmoins, contrairement au cas mono-objectif où il existe un ordre total sur  $\mathbb{R}$  parmi les solutions réalisables, il n'existe généralement pas de solution qui serait à la fois optimale pour chaque objectif, étant donnée la nature conflictuelle de ces derniers. Ainsi, une *relation de dominance*, d'ordre partiel, est généralement définie.

**Définition 2.1 (Dominance Pareto)** Un vecteur objectif  $z \in Z$  domine un vecteur objectif  $z' \in Z$  ssi  $\forall i \in \{1, 2, \dots, n\}, z_i \leq z'_i$  et  $\exists j \in \{1, 2, \dots, n\}$  tel que  $z_j < z'_j$ . Cette relation sera notée  $z \succ z'$ . Par extension, une solution  $x \in X$  domine une solution  $x' \in X$ , noté  $x \succ x'$ , ssi  $f(x) \succ f(x')$ .

**Définition 2.2 (Vecteur non-dominé)** Un vecteur objectif  $z \in Z$  est non-dominé ssi  $\forall z' \in Z, z' \not\succeq z$ .

**Définition 2.3 (Solution Pareto optimale)** Une solution  $x \in X$  est Pareto optimale (ou non-dominée) ssi  $\forall x' \in X, x' \not\succeq x$ .

**Définition 2.4 (Dominance faible)** Un vecteur objectif  $z \in Z$  domine *faiblement* un vecteur objectif  $z' \in Z$  ssi  $\forall i \in \{1, 2, \dots, n\}, z_i \leq z'_i$ . Cette relation sera notée  $z \succeq z'$ .

**Définition 2.5 (Point idéal)** Le *point idéal*  $z^* = (z_1^*, z_2^*, \dots, z_n^*)$  est le vecteur qui minimise chaque fonction objectif individuellement,  $z_i^* = \min_{x \in X} f_i(x)$  pour  $i \in \{1, 2, \dots, n\}$ .

### Analyse de performance

En optimisation multiobjectif, l'existence d'un ensemble de solutions de meilleur compromis et l'absence d'ordre total entre ces solutions rendent la mesure de qualité d'une approximation de l'ensemble Pareto optimal difficile. En outre, à ceci vient s'ajouter la nécessité de mesurer la qualité d'une approximation à la fois en terme de convergence et de diversité. Ainsi, de nombreuses métriques ont été proposées pour mesurer la qualité d'un ensemble non-dominé, ou pour comparer deux approximations [190, 187] et chaque métrique a ses objectifs.

- **Indicateurs basés sur la convergence** qui indiquent la proximité des solutions trouvées par rapport au front pareto optimal. Dans cette catégorie se trouvent notamment la contribution [128, 177], la distance générationnelle [177], l'indicateur  $\epsilon$  [190].
- **Indicateurs basés sur la diversité** qui indiquent l'uniformité de la distribution des solutions trouvées en terme de dispersion et d'étendue. On peut notamment citer dans cette classe les indicateurs tels que le "spacing", le "spread" [186], l'entropie [128, 177].
- **Indicateurs hybrides** qui mesurent à la fois la convergence et la diversité. Parmi ces indicateurs les plus connus sont : l'hypervolume [107], les R-metrics [100].

Nous ne détaillerons que les indicateurs qui serviront dans le reste du document. Dans les descriptions d'indicateur,  $A$  et  $B$  représentent des ensembles d'approximation,  $Z_N^*$  le front optimal s'il est connu ou le front issu de l'union de tous les ensembles de solutions obtenus précédemment si le front optimal est inconnu.

**Mesure de contribution** La mesure de *contribution* [128] est un indicateur binaire permettant la comparaison de deux ensembles  $A, B \in \Omega$ . Soit  $S^*$  l'ensemble des solutions non-dominées de  $A \cup B$ . La contribution de  $A$  sur  $B$ , donnée par  $I_C(A, B)$ , évalue la proportion de solutions représentées par  $A$  dans  $S^*$ .

L'avantage de cette mesure est qu'elle possède une complexité temporelle faible.

**$\epsilon$ -indicateur** L'indicateur epsilon additif binaire ( $I_{\epsilon+}$ ) donne le plus petit terme additif par lequel une approximation  $A \in \Omega$  doit être translatée dans l'espace objectif pour dominer faiblement une approximation  $B \in \Omega$ .

$$I_{\epsilon+}(A, B) = \min_{\epsilon \in \mathbb{R}} \{ \forall x \in B, \exists x' \in A : x' \succeq_{\epsilon+} x \}.$$

Par extension, l'*indicateur epsilon additif unaire* ( $I_{\epsilon+}^1$ ) peut être défini comme suit.

$$I_{\epsilon+}^1(A) = I_{\epsilon+}(A, R),$$

où  $R$  est un ensemble de référence. Cet indicateur est à minimiser.

**Hypervolume** L'indicateur hypervolume ( $I_H$ ), proposé par [107], mesure le volume de la portion multidimensionnelle de l'espace objectif faiblement dominée par une approximation  $A \in \Omega$ . Cet indicateur unaire nécessite la spécification d'un point de référence  $z^{ref}$ . Ce dernier doit au moins être faiblement dominé par toutes les solutions de l'approximation considérée.

Une variante de cet indicateur mesure la différence, en terme d'hypervolume, entre une approximation  $A \in \Omega$  et un ensemble de référence  $R$ . Ceci peut être vu comme le volume faiblement dominé par  $R$  et non par  $A$ .

Cet indicateur *hypervolume-différence* sera noté ( $I_H^-$ ) et contrairement à l'hypervolume est à minimiser.

$$I_H^-(A) = I_H(R) - I_H(A). \quad (2.2)$$

**R-Metrics** Il existe trois indicateurs  $R^R$  [74] basés sur un ensemble de fonctions d'utilité  $u$ . La métrique  $R_1$  [74] mesure la probabilité qu'une approximation  $A$  soit meilleure que  $B$  sur l'ensemble des fonctions d'utilité.  $R_1^R$  est  $R_1$  quand un ensemble de référence est utilisé et permet alors d'avoir un ordre total sur les ensembles d'approximation.

$$R_1(A, B, U, p) = \int_{u \in U} C(A, B, u) p(u) du$$

où  $U$  est un ensemble de fonctions d'utilité,  $A$  et  $B$  les deux approximations.

$$C(A, B, u) = \begin{cases} 1 & \text{si } u(A) > u(B) \\ 1/2 & \text{si } u(A) = u(B) \\ 0 & \text{si } u(A) < u(B) \end{cases}$$

Nous pouvons remarquer que  $R_1(A, B, u, p) = 1 - R_1(B, A, u, p)$ .

Si  $R_1(A, B, u, p) > 1/2$  alors  $A$  est meilleure que  $B$ .

$R_2$  et  $R_3$  sont définis comme suit :

$$R_2(A, B) = \frac{\sum_{\lambda \in \Lambda} u(\lambda, A) - u(\lambda, B)}{|\Lambda|}$$

$$R_3(A, B) = \frac{\sum_{\lambda \in \Lambda} [u(\lambda, B) - u(\lambda, A)]/u(\lambda, B)}{|\Lambda|}$$

Nous avons proposé un ensemble de définitions permettant de mieux appréhender les concepts présents dans ce manuscrit et notamment ceux de l'optimisation multiobjectif. Nous présentons maintenant le contenu du document.

## 2.2 Organisation du document

La première partie d'introduction générale se termine par une présentation de la plateforme ParadisEO à laquelle j'ai contribué et qui servira d'environnement de développement pour les travaux présentés dans ce mémoire.

Le reste du document est divisé en deux grandes parties.

La **deuxième partie** présente nos travaux sur la coopération de méthodes dans le cadre *déterministe*.

Le **chapitre 4** présente différentes coopérations entre métaheuristiques aussi bien en mono-objectif qu'en multiobjectif. Après un positionnement sur les métaheuristiques coopératives, nous présenterons différentes coopérations entre métaheuristiques notamment des méthodes qui sont adaptatives *ACS* et *PCS*. Ces méthodes seront testées sur différentes applications en optimisation multiobjectif.

Dans le **chapitre 5** nous montrons que la classification des métaheuristiques hybrides présentée par Talbi [164] s'adapte aux méthodes hybrides entre des métaheuristiques et des méthodes exactes et nous proposons une extension de cette classification afin de mieux cerner le type de résolution (exacte ou approchée) proposé par la méthode coopérative. Nous présentons ensuite une revue des différentes coopérations de ce type dans la littérature.

Le **chapitre 6** présente les possibilités offertes par l'hybridation de métaheuristiques avec des méthodes d'extraction de connaissances. Après avoir présenté une classification des différentes hybridations possibles selon la taxonomie globale présente dans le document, une nouvelle méthode hybride, *LEMMO*, est décrite et testée sur une application en hydrologie. *LEMMO* est une des premières méthodes hybride entre une métaheuristique multiobjectif et une méthode d'extraction de connaissances.

La **troisième partie** est consacrée à nos travaux sur les métaheuristiques en environnement *stochastique*.

Le **chapitre 7** présente nos contributions en optimisation sous incertitude. Nous présentons différentes approches multiobjectives pour résoudre des problèmes soumis à des incertitudes dans leurs données. La première approche proposée étudie l'importance du critère de robustesse lors de la modélisation d'un problème monoobjectif incertain en un problème multiobjectif. La seconde approche propose d'étudier différentes stratégies de résolution pour un problème multiobjectif incertain.

Dans le **chapitre 8**, nous présentons un nouvel algorithme, *APSO*, basé sur les essaims de particules pour un problème de transport dynamique. Nous montrons ensuite les apports d'une hybridation avec une recherche locale classique pour améliorer les résultats obtenus.

Le **partie IV** conclut ce travail et présente différentes perspectives de recherche.

Le reste du manuscrit est composé de trois annexes :

- un CV détaillé,
- une liste des publications réalisées,
- un ensemble de publications représentatif des travaux présentés dans ce mémoire.

## Chap. 3

# ParadisEO : Une plateforme pour la coopération de méthodes d'optimisation

### Sommaire

---

<b>3.1</b>	<b>Contexte</b>	<b>14</b>
3.1.1	Description	14
3.1.2	Caractéristiques principales	15
<b>3.2</b>	<b>ParadisEO-MO</b>	<b>16</b>
3.2.1	Etat des lieux	16
3.2.2	Contributions	17
<b>3.3</b>	<b>ParadisEO-MOEO</b>	<b>19</b>
3.3.1	Etat de l'art	19
3.3.2	Contributions	19
<b>3.4</b>	<b>Conclusion</b>	<b>24</b>

---

Ce chapitre a donné lieu aux publications suivantes :

- A. Liefooghe, L. Jourdan, T. Legrand, J. Humeau, E.-G. Talbi. *ParadisEO-MOEO : A Software Framework for Evolutionary Multi-objective Optimization*. *Advances in Multi-objective Nature Inspired Computing*, Studies in Computational Intelligence vol. 272, pp. 87-117, Springer, 2010
- A. Liefooghe, L. Jourdan, and E.-G. Talbi. *A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework*. *Computational Intelligence in Multi-Criteria Decision-Making, 2009. MCDM '09. IEEE Symposium on*, 88–95. 2009.
- L. Jourdan. E.-G. Talbi, J.-C. Boisson, S. Mesmoudi. *Paradiseo-MO : une plateforme pour le développement de métaheuristiques à base de solution unique*. *Dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF 2009)*, 208–209. 2009.
- A. Liefooghe, M. Basseur, L. Jourdan, and E.-G. Talbi. *Paradiseo-MOEO : a framework for multiobjective optimization*. *EMO 2007*, 4408 LNCS, 457–471. 2007.

Dans ce chapitre, nous présentons ParadisEO que nous utiliserons comme environnement de développement des différentes méthodes présentées dans ce document. Nous nous focalisons sur nos principales contributions à cette plateforme logicielle dans les modules ParadisEO-MO (Moving Object) dans le cadre de la thèse de Jean-Charles Boisson et ParadisEO-MOEO (Multi-Objective Evolving Objects) dans le cadre de la thèse de Arnaud Liefoghe.

## 3.1 Contexte

### 3.1.1 Description

ParadisEO<sup>1</sup> est une plateforme logicielle « boîte blanche » orientée-objet dédiée à la conception flexible des métaheuristiques pour la résolution de problèmes d'optimisation de nature continue, discrète ou combinatoire. Basé sur la librairie EO (*Evolving Objects*)<sup>2</sup> [92], ParadisEO s'appuie sur la notion de Template C++, et est portable sur les systèmes Unix (Linux, MacOS) et Windows. Ce logiciel est régi par la licence CeCILL soumise au droit français et respectant les règles de diffusion des logiciels libres<sup>3</sup>. Il tend à être utilisé à la fois par des non-spécialistes et des experts de l'optimisation. Comme illustré dans la figure 3.1, il est composé de quatre modules inter-connectés qui constituent une plateforme globale. Chaque module est fondé sur une séparation conceptuelle claire entre les méthodes de résolution et les problèmes qu'elles sont destinées à résoudre. Cette séparation confère une réutilisation maximum de code et de conception pour l'utilisateur.

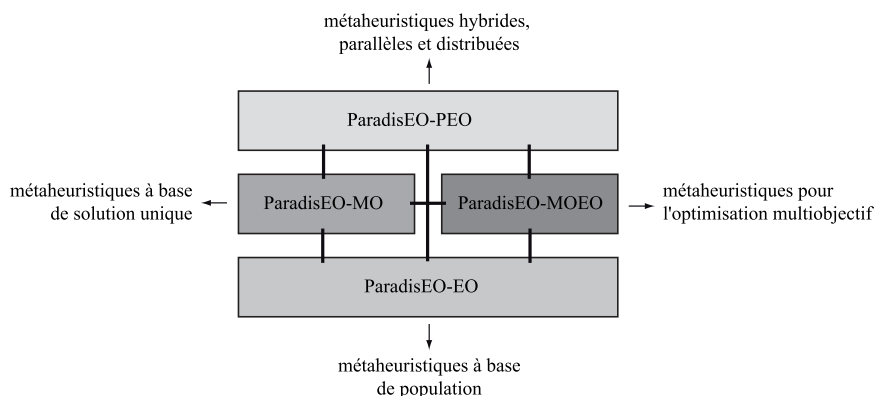


Fig. 3.1 – Les modules de la plateforme ParadisEO.

Les modules constituant la plateforme ParadisEO sont les suivants.

1. **ParadisEO-EO** (*Evolving Objects*) pour l'implémentation de métaheuristiques à base de population, comprenant entre autres les algorithmes évolutionnaires [92] et les algorithmes à essaim de particules.
2. **ParadisEO-MO** (*Moving Objects*) [24] pour l'implémentation de métaheuristiques à base de solution unique, c'est-à-dire les méthodes de recherche locale, de recuit simulé, de recherche tabou, etc.

1. ParadisEO est disponible à l'URL : <http://paradiseo.gforge.inria.fr>.

2. EO est disponible à l'URL : <http://eodev.sourceforge.net>.

3. Des informations sur la licence CeCILL sont disponibles à l'URL : <http://www.cecill.info>.

3. **ParadisEO-MOEO** (*Multi-Objective Evolving Objects*) [114, 117] pour l'implémentation de métaheuristiques dédiées à l'optimisation multiobjectif.
4. **ParadisEO-PEO** (*Parallel Evolving Objects*) [30, 165] pour l'implémentation de métaheuristiques parallèles et distribuées, que ce soit au niveau algorithmique (*multi-start*, modèle en îles), itératif (parallélisation de l'évaluation des solutions de la population) ou de la solution (parallélisation de la fonction d'évaluation).

### 3.1.2 Caractéristiques principales

Une plateforme logicielle est en général destinée à être exploitée par un grand nombre d'utilisateurs. Son exploitation ne peut être efficace que si un ensemble de critères est satisfait. Par conséquent, les principaux objectifs de la plateforme ParadisEO sont les suivants [30, 165].

- **Réutilisabilité maximum de la conception et du code.** La plateforme doit fournir une architecture de conception complète pour la métaheuristique considérée. L'utilisateur ne doit écrire que le minimum de code spécifique à son problème et le processus de développement doit être fait de façon incrémentale, ceci afin de considérablement simplifier la mise en œuvre et d'alléger le temps de développement, et donc le coût. Cet objectif nécessite une séparation conceptuelle maximale entre les méthodes de résolution et le problème à résoudre.
- **Flexibilité et adaptabilité.** Il doit être possible de facilement ajouter de nouvelles fonctionnalités ou de modifier les fonctionnalités existantes, sans pour autant qu'il y ait de répercussions sur les autres composants. L'utilisateur doit avoir accès au code source et doit utiliser les concepts d'héritage ou de spécialisation de la programmation orientée-objet pour créer de nouveaux composants à partir de composants de base ou de classes abstraites. Par ailleurs, de nouveaux problèmes d'optimisation surgissent et certains problèmes d'optimisation existants se développent, les composants doivent donc être adaptés et spécialisés de façon appropriée.
- **Utilité.** La plateforme doit couvrir un large éventail de métaheuristiques, de composants, des modélisations de problèmes classiques, des modèles parallèles et distribués, des mécanismes d'hybridation, etc. Bien sûr, une utilisation avancée ne doit pas ajouter de difficultés pour les utilisateurs désireux d'utiliser les implémentations basiques.
- **Accès transparent à la performance et à la robustesse.** Étant donné que les méthodes d'optimisation sont souvent coûteuses en temps, en particulier pour les problèmes réels difficiles, la question de performance des métaheuristiques proposées est cruciale. Aussi, le calcul parallèle et distribué est une façon intéressante d'atteindre un niveau élevé de performance à l'exécution. Par ailleurs, l'exécution des algorithmes doit être robuste afin de garantir la fiabilité et la qualité des résultats. Lorsqu'ils sont conçus de manière astucieuse, les mécanismes d'hybridation permettent souvent d'obtenir de solides et de meilleures solutions.
- **Portabilité.** Afin de satisfaire le plus grand nombre d'utilisateurs, la plateforme doit supporter un grand nombre d'architectures matérielles (séquentielle, parallèle, distribuée) et leurs systèmes d'exploitation associés (Windows, Linux, MacOS).
- **Facilité d'utilisation et efficacité.** La plateforme doit être facile d'utilisation et ne doit contenir aucun coût supplémentaire en termes de complexité en temps ou en espace, ceci afin de conserver une efficacité similaire à une implémentation spécifique. Au contraire, celle-ci est destinée à être moins sujette aux erreurs que les métaheuristiques développées de façon spécifique.



ParadisEO honore tous les critères mentionnés ci-dessus et vise à être utilisée à la fois par des non-spécialistes et des experts de l’optimisation.

La tableau 3.1 propose un éventail, non exhaustif, des plateformes existantes pour développer des métaheuristiques.

TABLE 3.1 – Liste des plateformes existantes pour développer des métaheuristiques. RL signifie recherches locales, AE algorithmes évolutionnaires, RD recherche par descente, AG algorithmes génétiques, SS, scatter search, SE stratégies évolutionnaires et PS métaheuristiques à population de solutions.

Nom de la plateforme	Type de métaheuristique	Optimisation		Parallélisme
		mono-objectif	multi-objectif	
EasyLocal++	RL	X	-	-
Localizer++	RL	X	-	-
INCOP	RL	X	-	-
PISA	AE	-	X	-
MAFRA	RD, AE	X	-	-
iOpt	RL, AG	X	-	-
OptQuest	SS	X	-	-
GAlib	AG	X	-	X
GenocopIII	AE	X	-	-
DREAM	AE	X	-	X
MALLBA	RD, AE	X	-	X
Hotframe	RL, AE	X	-	X
TEMPLAR	RL, AG	X	-	X
JDEAL	AG, SE	X	-	X
ECJ	AE	X	-	X
Distributed BEAGLE	AE	X	-	X
COMET	RL			
ParadisEO	RL, PS	X	X	X

## 3.2 ParadisEO-MO

Dans un premier temps, avec J-C. Boisson nous avons repris le module ParadisEO-MO afin de l’étendre puis avec J. Humeau, A. Liefoghe et S. Verel nous avons repensé le modèle afin d’unifier les concepts pour gagner en flexibilité. La conception initiale de ParadisEO-MO comportait certains défauts et une refonte de ce module a donc été récemment effectuée. Nous présentons ici la conception originelle en montrant ses limites puis la nouvelle conception.

### 3.2.1 Etat des lieux

La conception initiale de ParadisEO-MO était basée sur une notion de “mouvement” et comprenait quatre fonctionnalités :

1. le *mouvement lui-même*, c’est-à-dire la façon de modifier une solution afin d’obtenir une solution voisine.

2. l'*initialisation* d'un premier mouvement valide. Aucune vérification sur l'existence d'un tel mouvement n'était faite, ce qui pouvait entraîner des erreurs d'exécution dans des cas extrêmes.
3. le *test* et le *calcul* du mouvement suivant le mouvement courant. Ces deux notions étant regroupées dans une seule méthode, cela rendait plus délicat certains traitements.
4. l'*évaluation incrémentale* d'un mouvement. Celle-ci ne pouvant être définie pour tout problème, une alternative moins élégante et moins efficace devait être implémentée afin d'utiliser une évaluation complète de la solution générée par le mouvement. De plus, seule la valeur de fitness pouvait être sauvegardée aisément.

Toutes les recherches locales utilisaient cette même notion de mouvement, mais aucun traitement générique à toutes ces recherches n'avait été identifié... De ce fait, chaque recherche locale était implémentée indépendamment.

De plus, il n'existait aucun moyen de "tracer" le déroulement d'une recherche locale. Et du fait de l'indépendance de l'implémentation des recherches locales, il était difficile d'intégrer des outils génériques le permettant.

### 3.2.2 Contributions

La nouvelle conception de ParadisEO-MO reprend les mêmes fonctionnalités que l'ancienne et en apporte de nouvelles. La notion de "mouvement" est remplacée par les notions de "voisinage" et de "voisin". Le "voisin" englobe la fonctionnalité de *mouvement lui-même* et permet en plus de stocker facilement des informations supplémentaires. Le "voisinage" regroupe les fonctionnalités d'*initialisation* du premier mouvement, du *calcul* et de *test* d'existence d'un mouvement suivant. Ces deux fonctionnalités sont maintenant bien distinctes pour les raisons mentionnées dans le paragraphe précédent. Quant à l'*évaluation* des voisins, elle peut toujours être effectuée de manière incrémentale. Mais si cela n'est pas possible, deux alternatives sont proposées dans ParadisEO-MO :

1. évaluation complète d'un voisin par modification de la solution initiale.
2. évaluation complète d'un voisin par copie de la solution initiale.

Ces deux évaluations utilisent la fonction d'évaluation classique d'une solution. L'évaluation par modification applique le mouvement correspondant à un voisin sur la solution initiale afin d'évaluer ce dernier. Mais l'on doit être capable de définir le mouvement inverse afin de restaurer la solution initiale. Si cela n'est pas possible, on utilise alors l'évaluation par copie qui consiste à appliquer le mouvement sur une copie temporaire de la solution initiale afin d'évaluer le voisin. Ces trois évaluations ont toutes un héritage commun dans la hiérarchie de classe. En terme de performance, l'évaluation incrémentale est préférable à l'évaluation complète par modification qui est elle même préférable à une évaluation complète par copie.

Cette nouvelle conception est plus adéquate pour modéliser le schéma générique d'une recherche locale qui consiste à itérer les étapes suivantes sur une solution courante jusqu'à ce qu'un critère d'arrêt soit satisfait :

- générer un ensemble de voisins à partir d'un voisinage,
- sélectionner un voisin dans cet ensemble,

- déterminer si on remplace la solution courante par la solution voisine correspondant au voisin précédemment sélectionné.

Un modèle générique d’un algorithme de recherche locale a donc été proposé et implémenté (voir algorithme 1).

---

**Algorithme 1** Modèle générique d’un algorithme de recherche locale

---

1. **Initialisation.** Démarrer avec une solution initiale  $S$ .
  2. **Évaluation.** Évaluer la solution  $S$ .
  3. **Initialisation des paramètres.** Initialiser les paramètres de recherche, les opérateurs de contrôle (critères d’arrêt, calcul de statistique, etc.).
  4. **Exploration.** Effectuer une exploration du voisinage  $V$  de la solution  $S$ . Sélectionner un voisin  $v$  dans  $V$ .
  5. **Critère d’acceptation.** Vérifier si le voisin sélectionné lors de l’exploration du voisinage  $V$  est accepté. Si oui, appliquer son mouvement associé sur la solution  $S$ .
  6. **Mise à jour des paramètres.** Mettre à jour les paramètres de recherche.
  7. **Critère d’arrêt.** Appeler tous les opérateurs de contrôle. Si aucun critère d’arrêt n’est vérifié, aller à l’étape 4.
  8. **Terminaison.** Appeler la terminaison de tous les opérateurs de contrôle. Retourner la meilleure solution trouvée au cours de la recherche.
- 

Ainsi, pour définir une nouvelle recherche locale dans ParadisEO-MO, il suffit d’implémenter un explorateur. Voici une liste des recherches pouvant être construites grâce aux explorateurs déjà implémentés dans la nouvelle version :

- méthodes de descente (4 différentes : Hill-Climbing simple (meilleure amélioration), Hill-Climbing avec choix du meilleur aléatoire parmi les meilleurs, Hill-Climbing neutre qui permet de visiter une solution de même qualité si il n’y a pas d’amélioration, Hill-Climbing au premier meilleur)
- marches aléatoires (3 différentes : marche aléatoire, marche neutre aléatoire, Metropolis-Hasting)
- recuit simulé
- recherche tabou
- recherche locale itérée
- recherche à voisinage variable

Ces méthodes incluent les recherches déjà disponibles auparavant ainsi que des nouvelles. De plus, la décomposition modulaire des explorateurs permet de faciliter la réutilisation de ces derniers. Par remplacement d’un ou plusieurs “composants” permettant de construire ces explorateurs, il est facile de changer certaines stratégies ou juste fournir une implémentation plus adéquate à certains problèmes.

De nouveaux critères d’arrêts et des méthodes de contrôles ont été rajoutés. Leurs conceptions reprennent celles de ParadisEO-EO, mais interviennent dans des algorithmes à solution unique et non pas à une population de solutions comme cela est fait dans ParadisEO-EO. Les méthodes de contrôles regroupent donc ces critères d’arrêts ainsi que des outils utiles à l’analyse de paysage comme l’autocorrélation des fitness, la corrélation entre les valeurs de fitness et les distances ainsi que des statistiques sur le voisinage :

- valeurs minimale, maximale, moyenne, écart-type, médiane des valeurs de fitness
- degré de neutralité
- probabilité d’amélioration de la fitness

Trois voisinages indexés sont disponible dans la nouvelle version :

- parcours ordonné du voisinage
  - parcours aléatoire du voisinage avec la possibilité de parcourir plusieurs fois un même voisin
  - parcours aléatoire du voisinage sans la possibilité de parcourir plusieurs fois un même voisin
- Pour utiliser ce genre de voisinage, il doit être possible de définir un mapping qui à partir d’une “clé”, détermine le voisin correspondant.

Afin de faciliter l’utilisation du module, des voisinages classiques sur les représentations standards (chaîne de bits et permutation) sont mis à disposition.

Le module ParadisEO-MO répond encore mieux aux objectifs définis au début du chapitre. L’ensemble de la partie ParadisEO-MO a été revu et amélioré afin de la rendre plus flexible. L’intégralité a été implémenté et mis à disposition de la communauté. La plateforme a été testée sur différents problèmes (TSP, QAP, ...) afin de valider expérimentalement les développements.

### 3.3 ParadisEO-MOEO

Un travail plus important a été fourni dans ParadisEO-MOEO, ce travail a été fait dans le cadre de la thèse d’Arnaud Liefoghe.

#### 3.3.1 Etat de l’art

Un nombre relativement élevé de plateformes logicielles dédiées à la conception de métaheuristiques ont été proposées jusqu’à présent. Cependant, la plupart d’entre elles sont uniquement dédiées aux algorithmes évolutionnaires, et très peu sont capables de résoudre des problèmes multiobjectif. Quelques unes fournissent tout de même des éléments pour un sous-ensemble particulier des stratégies, parmi lesquelles on peut citer ECJ<sup>4</sup>, JavaEVA [162] ou Open BEAGLE [60]. Le tableau 3.2 donne une liste non exhaustive de comparaison entre un certain nombre de plateformes logicielles existantes pour l’implémentation de métaheuristiques pour l’optimisation multiobjectif.

#### 3.3.2 Contributions

##### Unification des méthodes

Un travail d’abstraction et d’unification à propos de la conception de métaheuristiques pour l’optimisation multiobjectif a été réalisé dans la thèse d’Arnaud Liefoghe [111]. Dans ce travail, nous avons mis l’accent sur les questions principales liées à la convergence (et à l’affectation des valeurs de fitness), à la préservation de la diversité et à l’élitisme. De plus, une classification des stratégies existantes pour chacune de ces étapes a été proposée. Nous nous sommes ensuite concentrés sur deux types de méthodologie particuliers que nous avons étudiés de manière approfondie : les algorithmes évolutionnaires, et les algorithmes de recherche locale. Pour chacun d’eux,

---

4. <http://cs.gmu.edu/~eclab/projects/ecj/>

TABLE 3.2 – Caractéristiques principales des plateformes logicielles existantes pour le développement de métaheuristiques pour l’optimisation multiobjectif.

plateforme	problème		outils stat.		hybrid.	para.	boîte	lang.	licence
	cont.	comb.	off-line	on-line					
jMetal	X	X	X	-	X	-	blanche	java	libre
MOEA for Matlab	X	-	-	-	-	X	noire	matlab	comm.
MOMHLib++	X	X	-	-	X	-	blanche	c++	libre
PISA	X	X	X	-	X	-	noire	-	libre
Shark	X	-	-	-	X	-	blanche	c++	libre
ParadisEO	X	X	X	X	X	X	blanche	c++	libre

nous avons identifié les principaux composants communs. Puis, nous avons proposé un modèle de conception unifié, basé sur une décomposition fine, et pour lequel bon nombre d’algorithmes existants peuvent être vus comme de simples spécifications d’un même cadre conceptuel.

**Métaheuristiques multiobjectif** Quel que soit le problème à résoudre, les principaux composants prenant part dans la conception d’un algorithme évolutionnaire multiobjectif sont les suivants :

1. Concevoir une **représentation**.
2. Concevoir une stratégie d’**initialisation de la population**.
3. Concevoir une fonction d’**évaluation**.
4. Concevoir des **opérateurs de variation** appropriés.
5. Déterminer une stratégie d’**affectation d’une valeur de fitness**.
6. Déterminer une stratégie de **préservation de la diversité**.
7. Déterminer une stratégie de **sélection**.
8. Déterminer une stratégie de **remplacement**.
9. Déterminer une stratégie de **gestion de l’archive**.
10. Déterminer une **condition d’arrêt**.

Lors de la conception d’une métaheuristique, on peut toujours distinguer les composants spécifiques au problème traité et les composants génériques. Ainsi, les quatre premiers composants présentés ci-dessus dépendent fortement du problème à résoudre, tandis que les six derniers peuvent être considérés comme indépendants (même si certaines stratégies spécifiques peuvent également être envisagées).

En suivant cette décomposition à grain fin, un modèle de conception unifié pour les algorithmes évolutionnaire multiobjectif est proposé et présenté dans la figure 3.2. Les composants spécifiques au problème apparaissent en gris sur la figure. L’algorithme 2 illustre la trame de haut-niveau d’un algorithme évolutionnaire multiobjectif général.

Dans le tableau 3.3, trois méthodologies, à savoir NSGA-II [44], SPEA2 [188] et IBEA [189], sont considérées comme de simples instances du même modèle. Bien sûr, seuls les composants indépendants du problème sont présentés pour des raisons évidentes.

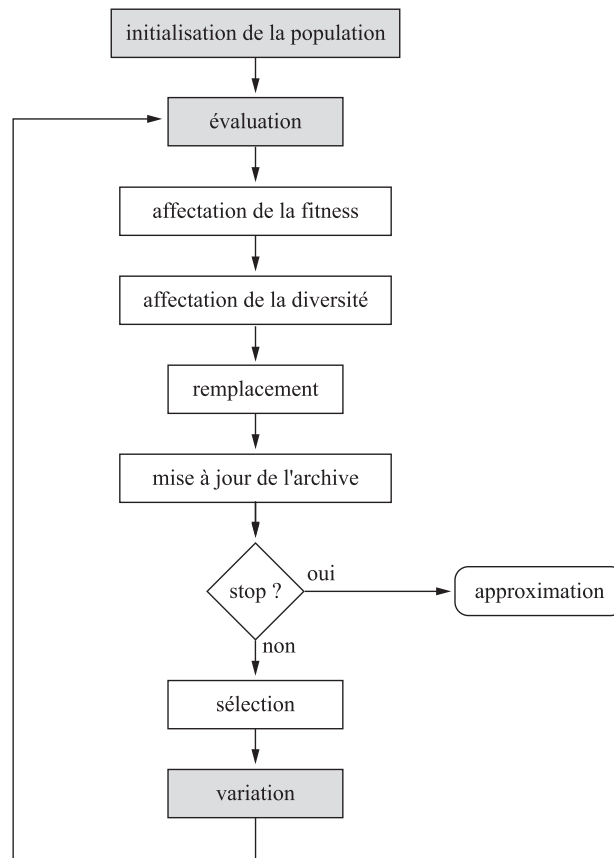


Fig. 3.2 – Étapes générique d'un algorithmes évolutionnaire multiobjectif.

TABLE 3.3 – Algorithmes évolutionnaires multiobjectif existants comme instances du modèle unifié proposé. Trois exemples sont présentés : NSGA-II [44], SPEA2 [188] et IBEA [189].

composants	NSGA-II	SPEA2	IBEA
fitness	profondeur de dominance (dominance Pareto)	compte et rang de dominance (dominance Pareto)	indicateur binaire de qualité
diversité	distance de <i>crowding</i>	plus proche voisin	aucun
sélection	tournoi binaire	élitiste (tournoi binaire)	tournoi binaire
remplacement	remplacement élitiste	remplacement générationnel	remplacement élitiste
gestion de l'archive	aucun	archive de taille fixe	aucun
arrêt	nombre de générations	nombre de générations	nombre de générations

---

**Algorithme 2** Modèle d'un algorithme évolutionnaire multiobjectif
 

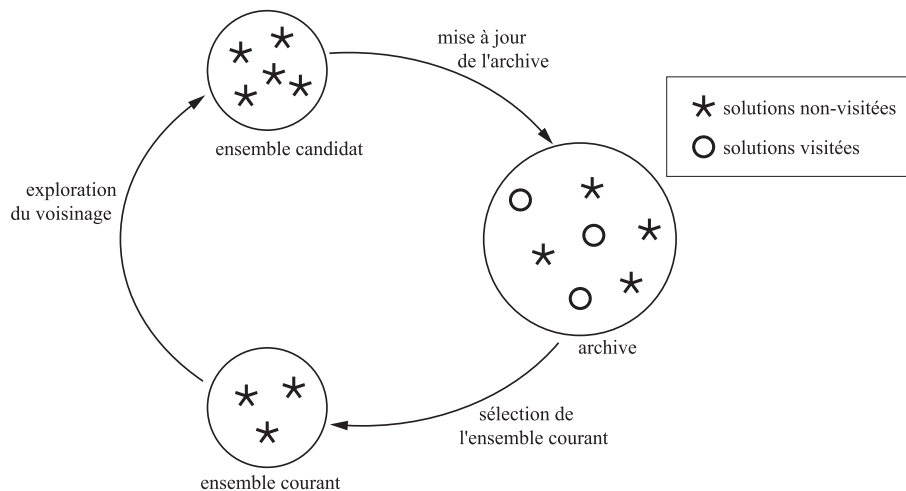
---

1. **Initialisation.** Démarrer avec une population initiale  $P$  de taille  $N$  fournie en paramètre, ou la générer de façon aléatoire.
  2. **Évaluation.** Évaluer les solutions de la population  $P$  dont les valeurs objectif sont inconnues.
  3. **Affectation d'une valeur de fitness.** Calculer les valeurs de fitness pour chaque solution  $x$  de la population  $P$ .
  4. **Affectation d'une valeur de diversité.** Calculer les valeurs de diversité pour chaque solution  $x$  de la population  $P$ .
  5. **Sélection pour le remplacement.** Supprimer  $N'$  solutions de la population  $P$  afin d'obtenir une population de taille  $N$ .
  6. **Mise à jour de l'archive.** Mettre à jour l'archive  $A$  à l'aide des solutions de la population principale  $P$ .
  7. **Condition d'arrêt.** Si une condition d'arrêt est satisfaite, retourner les solutions non-dominées de l'archive  $A$  et/ou de la population principale  $P$ . Stop.
  8. **Sélection pour la reproduction.** Sélectionner  $N'$  individus parmi la population  $P$  et ajouter les solutions sélectionnées dans une population temporaire  $P'$ .
  9. **Variation.** Appliquer les opérateurs de croisement et de mutation aux solutions de la population temporaire  $P'$  et ajouter les solutions résultantes à la population principale  $P$ . Aller à l'étape 2.
- 

En nous basant sur les modèles de conception unifiés présentés, nous avons également proposé de nouvelles approches méthodologiques dédiées à la résolution d'une catégorie très générale de problèmes d'optimisation multiobjectif. Par exemple, un algorithme évolutionnaire élitiste simple, SEEA, a été développé [R4].

**Recherches locales multiobjective** De la même façon que pour les métaheuristiques à population, nous reprenons les principaux composants permettant de concevoir une recherche local multiobjective (textitDMLS Dominance based Multiobjective Local Search) :

1. Concevoir une **représentation**.
2. Concevoir une stratégie d'**initialisation**.
3. Concevoir une fonction d'**évaluation**.
4. Concevoir une **structure de voisinage** appropriée.
5. Concevoir une fonction d'**évaluation incrémentale** (si possible).
6. Déterminer une stratégie de **sélection de l'ensemble courant**.
7. Déterminer une stratégie d'**exploration du voisinage**.
8. Déterminer une stratégie de **gestion de l'archive**.
9. Déterminer une **condition d'arrêt**.



**Fig. 3.3** – Étapes du modèle DMLS.

Parmi les composants présentés ci-dessus, les cinq premiers dépendent fortement du problème à résoudre, alors que les cinq derniers peuvent être considérés comme des composants génériques. Par rapport à l'optimisation monoobjectif, les étapes d'évaluation et d'évaluation incrémentale peuvent être vues comme spécifiques à l'optimisation multiobjectif, car plusieurs fonctions doivent maintenant être calculées. Concernant les composants qui ne sont pas liés au problème, ils sont presque tous spécifiques au cas multiobjectif, à l'exception de quelques stratégies très simples. Trois structures de données s'ajoutent à ces composants afin de stocker le contenu de l'archive, l'ensemble courant, et l'ensemble candidat constitué de solutions susceptibles d'intégrer l'archive à l'étape d'archivage. Dans certains cas, si le voisinage d'une solution est évalué de façon exhaustive, celle-ci peut être marquée comme « visitée » afin d'éviter une réévaluation inutile de son voisinage. Ainsi, l'archive peut contenir à la fois des solutions visitées et non-visitées.

En suivant cette décomposition, un modèle conceptuel est présenté dans la figure 3.3 et dans l'algorithme 3.

Nous décrivons ici succinctement les composants spécifiques aux DMLS, une présentation exhaustive peut être trouvée dans les articles relatifs à ce chapitre [112, 114, 116, 115] ainsi que dans la thèse d'Arnaud Liefoghe [111]. Les composants spécifiques sont :

- Sélection de l'ensemble courant : la phase initiale d'une itération d'un algorithme DMLS porte sur la sélection d'un ensemble de solutions de l'archive à partir desquelles le voisinage est à explorer. Les différentes stratégies sont :
  - une *sélection exhaustive*, où l'ensemble des solutions non-visitées de l'archive est sélectionné ;
  - une *sélection partielle*, où uniquement un sous-ensemble de solutions est sélectionné.
- Exploration du voisinage
  - une *exploration exhaustive du voisinage*, où le voisinage d'une solution est évalué de façon complète et déterministe ;
  - une *exploration partielle du voisinage*, où seul un sous-ensemble de mouvements est appliqué.
    - Voisin aléatoire : Un seul voisin aléatoire de la solution actuelle est proposé comme candidat à l'intégration dans l'archive [99].
    - Premier voisin non-dominé : Pour chaque solution  $x$  de l'ensemble courant, le voisinage



---

**Algorithme 3** Modèle d'un algorithme DMLS
 

---

1. **Initialisation.** Démarrer avec un ensemble de solutions initiales  $P$  fourni en paramètre, ou le générer de façon aléatoire ; initialiser l'archive  $A$  avec les solutions non-dominées de  $P$ .
  2. **Sélection de l'ensemble courant.** Sélectionner un ensemble de solutions non-visitées de l'archive et ajoutez-les à l'ensemble courant  $P_{courant}$ .
  3. **Exploration du voisinage.** Pour chaque solution  $x$  de l'ensemble courant  $P_{courant}$ , explorer le voisinage de  $x$  selon une certaine stratégie et ajouter les solutions non-dominées explorées à l'ensemble candidat  $P_{candidat}$ . Dans le cas d'une exploration exhaustive du voisinage, marquer  $x$  comme visitée.
  4. **Mise à jour de l'archive.** Mettre à jour l'archive  $A$  à l'aide des solutions de l'ensemble candidat  $P_{candidat}$ .
  5. **Condition d'arrêt.** Si les solutions de l'archive  $A$  sont toutes marquées comme visitées ou si une autre condition d'arrêt est satisfaite, retourner les solutions non-dominées de l'archive  $A$ . Stop. Sinon, aller à l'étape 2.
- 

est évalué jusqu'à ce qu'un premier voisin non-dominé, à l'égard de  $x$ , soit trouvé.

- Premier voisin dominant : Pour chaque solution courante  $x$ , les mouvements sont appliqués jusqu'à ce que l'on trouve une solution voisine dominant  $x$  [1]. Ensuite, tous les voisins non-dominés évalués sont proposés comme solutions candidates à l'intégration au sein de l'archive.
- Gestion de l'archive
  - aucune archive ;
  - une archive de taille non-bornée ;
  - une archive de taille bornée ;
  - une archive de taille fixe.

Deux recherches locales Pareto : PLS-1 [140], et PLS-2 [168, 13, 3], ainsi que PAES [99] et moRBC( $|A| : 1 + 1$ )<sup>A</sup> [1] sont pris comme exemples pour montrer comment des algorithmes DMLS provenant de la littérature s'inscrivent facilement dans la vue unifiée proposée. Comme illustré dans le tableau 3.4, ces quatre algorithmes DMLS classiques sont basés sur de simples variantes des composants indépendants du problème présenté précédemment, ce qui valide la grande flexibilité et modularité du modèle d'unification proposé.

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté ParadisEO et les contributions que nous avons apporté dans la plateforme au sein des modules ParadisEO-MO et ParadisEO-MOEO. Il s'agit d'une plateforme boîte blanche et orientée-objet dédiée à l'implémentation flexible et réutilisable de métaheuristiques pour l'optimisation mono-objectif et multiobjectif. Dans un soucis de lisibilité, elle est basée sur une séparation conceptuelle claire entre les méthodes de résolution et le problème qu'elles sont destinées à résoudre, conférant donc une réutilisabilité maximale du code et de la conception. Cette plateforme a été validée expérimentalement par la résolution d'une large étendue de problèmes d'optimisation multiobjectif, à la fois académiques et réels, et de différents

TABLE 3.4 – Algorithmes de recherche locale multiobjectif existants comme instances du modèle DMLS. Quatre exemples sont présentés : PLS-1 [140], PLS-2 [168, 13, 3], PAES [99], and moRBC( $|A| : 1 + 1$ )<sup>A</sup> [1].

composants	PLS-1	PLS-2	PAES	moRBC( $ A  : 1 + 1$ ) <sup>A</sup>
relation de dominance	dominance Pareto	dominance Pareto	dominance Pareto	dominance Pareto
sélection de l'ensemble courant	<i>partielle</i> 1 solution aléatoire	<i>exhaustive</i> toutes les solutions	<i>partielle</i> $\mu$ solutions	<i>partielle</i> 1 solution diversifiée
exploration du voisinage	<i>exhaustive</i> tous les voisins	<i>exhaustive</i> tous les voisins	<i>partielle</i> $\lambda$ voisins aléatoires	<i>partielle</i> 1 voisin dominant
gestion de l'archive	<i>non-bornée</i>	<i>non-bornée</i>	<i>bornée</i> hyper-grille	<i>bornée</i> distance de <i>crowding</i>
condition d'arrêt	<i>naturelle</i> solutions visitées	<i>naturelle</i> solutions visitées	<i>définie par l'utilisateur</i>	<i>naturelle</i> solutions visitées

types (continus, combinatoires). La mise en place de ces différents modules va nous permettre de réaliser sur la plateforme des modèles hybrides de métaheuristique aussi bien en mono-objectif qu'en multiobjectif.



## Deuxième partie

# Coopération de méthodes dans le cadre déterministe



## Chap. 4

# Coopération métaheuristiques-métaheuristiques

### Sommaire

---

<b>4.1</b>	<b>Etat de l'art</b> . . . . .	<b>30</b>
4.1.1	Notion de coopération . . . . .	30
4.1.2	Niveaux de Coopération . . . . .	31
4.1.3	Modes de Coopération . . . . .	31
4.1.4	Classes hiérarchiques . . . . .	31
<b>4.2</b>	<b>Méthodes coopératives multiobjectives</b> . . . . .	<b>35</b>
<b>4.3</b>	<b>Nouvelles métaheuristique coopératives</b> . . . . .	<b>38</b>
4.3.1	Une approche de coopération multiobjectif co-évolutionnaire de bas niveau	38
4.3.2	Une approche de coopération multiobjectif de haut niveau en mode relais	45
<b>4.4</b>	<b>Conclusion</b> . . . . .	<b>50</b>

---

Ce chapitre a donné lieu aux publications suivantes :

- A. Liefoghe and L. Jourdan and E-G. Talbi *Metaheuristics and cooperative approaches for the bi-objective ring star problem* Computers & Operations Research, vol. 37, n. 6, pp. 1033-1044, 2010.
- J-C. Boisson, L. Jourdan, E-G. Talbi and D. Horvath *Single- and Multi-Objective Cooperation for the Flexible Docking Problem*, Journal of Mathematical Modelling and Application, Vol. 9, n° 2, pp. 195-208, 2010
- J-C. Boisson, L. Jourdan, E-G. Talbi et D. Horvath. *Parallel multi-objective algorithms for the molecular docking problem*, Conference in Computational Intelligence in Bioinformatics and Bioengineering (CIBCB), 2008, USA.

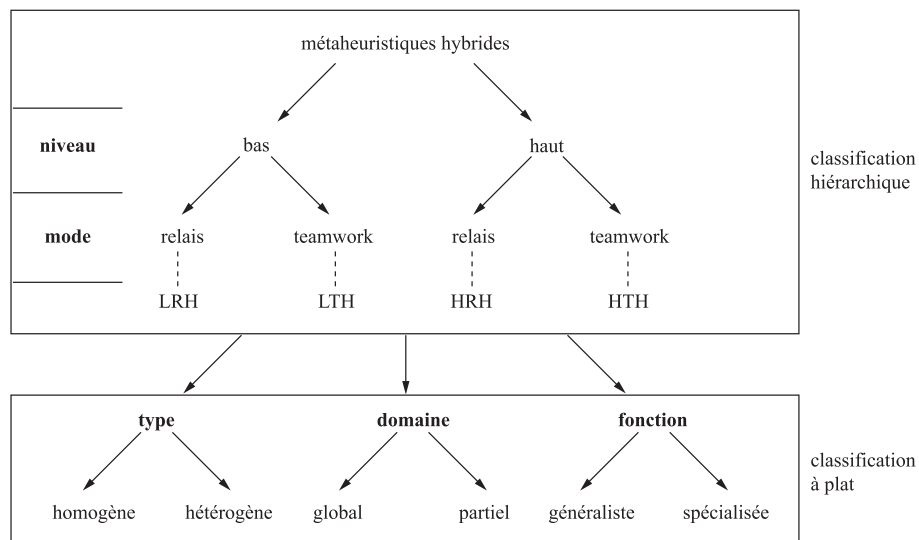
Dans ce chapitre nous présentons la taxonomie qui servira de base à la présentation des méthodes coopératives impliquant des métaheuristiques et nous l'illustrerons par deux exemples de métaheuristiques multiobjectives coopératives, dans un cadre d'optimisation statique, issus des thèses de Jean-Charles Boisson et de Arnaud Liefoghe. L'intérêt est ici de montrer que la coopération de métaheuristiques peut se mettre en œuvre aussi bien en monoobjectif ce qui est devenu classique qu'en multiobjectif.

## 4.1 Etat de l'art

Nous présentons dans cette partie les différents types de coopération existants ainsi que la manière de les appliquer.

### 4.1.1 Notion de coopération

Pour résoudre un problème d'optimisation donné, une méthode qui paraît bien adaptée est choisie parmi celles existantes (méthode exacte, heuristiques, ...). Ensuite, des essais d'amélioration sont apportés en travaillant sur ses paramètres afin d'obtenir la méthode la plus efficace possible. Si c'est possible, sa qualité est évaluée en la comparant à d'autres méthodes proposées pour le problème étudié. Malheureusement, d'après les *No Free Lunch Theorems* [183], il n'existe pas de méthode d'optimisation qui, pour un problème donné, sera meilleure que toutes les autres sur toutes les instances possibles. Dans la pratique, il existera toujours des instances pour lesquelles une méthode sera mise en défaut par rapport à une autre.

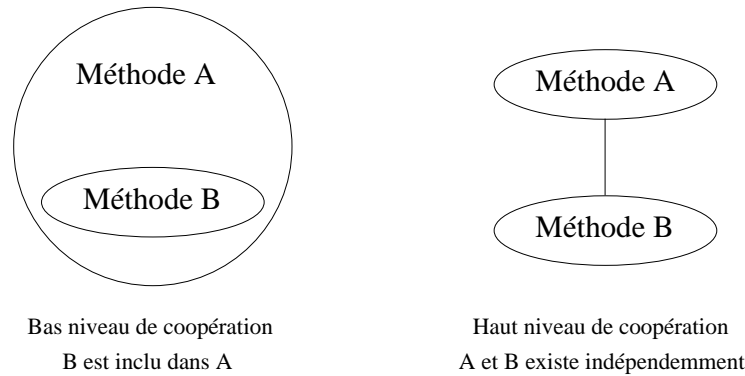


**Fig. 4.1** – Taxonomie servant pour la description de la coopération de méthodes.

Quelque soit la méthode choisie, elle présente des avantages et des inconvénients. La coopération de méthodes peut permettre de trouver de bons compromis où les avantages et défauts de chaque méthode se compensent. Comme le montre la figure 4.1, nous nous baserons sur une taxonomie hiérarchisée présentée dans [164] pour classer les différents types de coopération. Dans celle-ci,

les critères hiérarchiques pour discriminer les différentes coopérations sont le niveau et le mode de coopération.

#### 4.1.2 Niveaux de Coopération



**Fig. 4.2** – Les différents niveaux de coopération.

Dans la taxonomie choisie, le premier point pour classifier les coopérations est le niveau de coopération. Il peut être bas (*Low-Level*) ou haut (*High-Level*). Dans le niveau bas de coopération, une méthode devient un bloc fonctionnel d'une autre méthode qui l'englobe. A l'inverse, dans le niveau haut de coopération, les méthodes, même si elles peuvent communiquer entre elles, gardent leur intégrité propre. La figure 4.2 schématise la notion de niveau de coopération.

#### 4.1.3 Modes de Coopération

Le second point de classification est le choix du mode. Dans le mode relais, les méthodes d'optimisation sont appelées les unes à la suite des autres, le résultat de la précédente servant de point de départ à la méthode courante. On peut parler aussi d'une exécution séquentielle des méthodes. En mode co-évolutionnaire, les différentes méthodes d'optimisation fonctionnent en parallèle pour généralement pouvoir explorer des zones différentes de l'espace de recherche. La figure 4.3 schématise la notion de mode de coopération.

#### 4.1.4 Classes hiérarchiques

Selon le niveau et le mode de coopération choisis, quatre classes de coopération sont possibles : relais de bas niveau, relais de haut niveau, co-évolution de bas niveau et co-évolution de haut niveau.

##### Coopération relais de bas niveau (*Low-level Relay Hybrid*)

Cette classe de coopération (**LRH**) regroupe les méthodes à base de solution unique dans lesquelles une autre méthode d'optimisation est insérée ce qui par conséquent en modifie le comportement. Dans ce type de coopération, les deux (ou éventuellement plus) méthodes utilisées perdent leur intégrité pour produire un nouvel algorithme. Un exemple de ce type de coopération est proposé dans [123] pour résoudre le problème du voyageur de commerce (*Traveling Salesman*



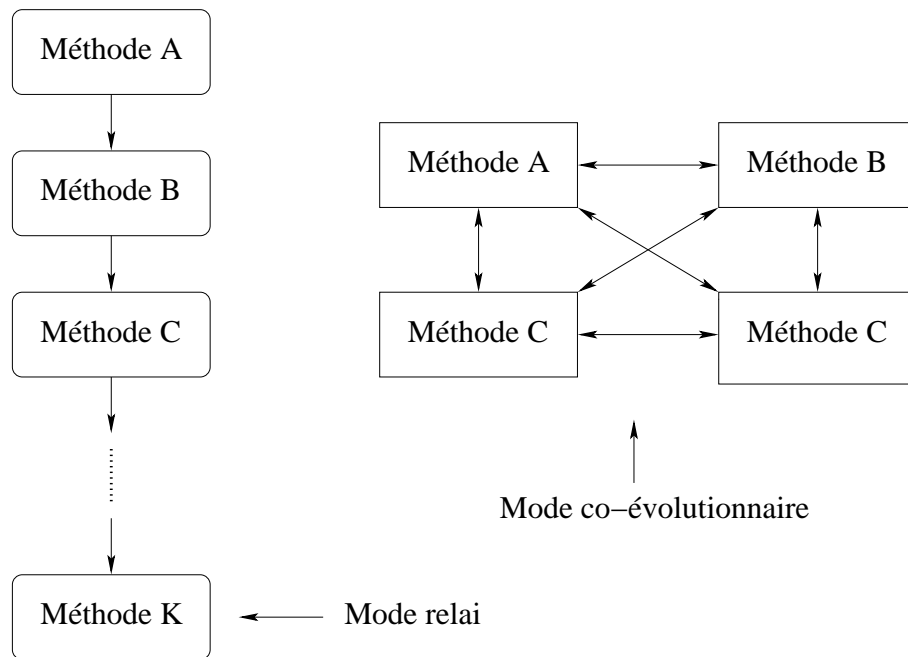


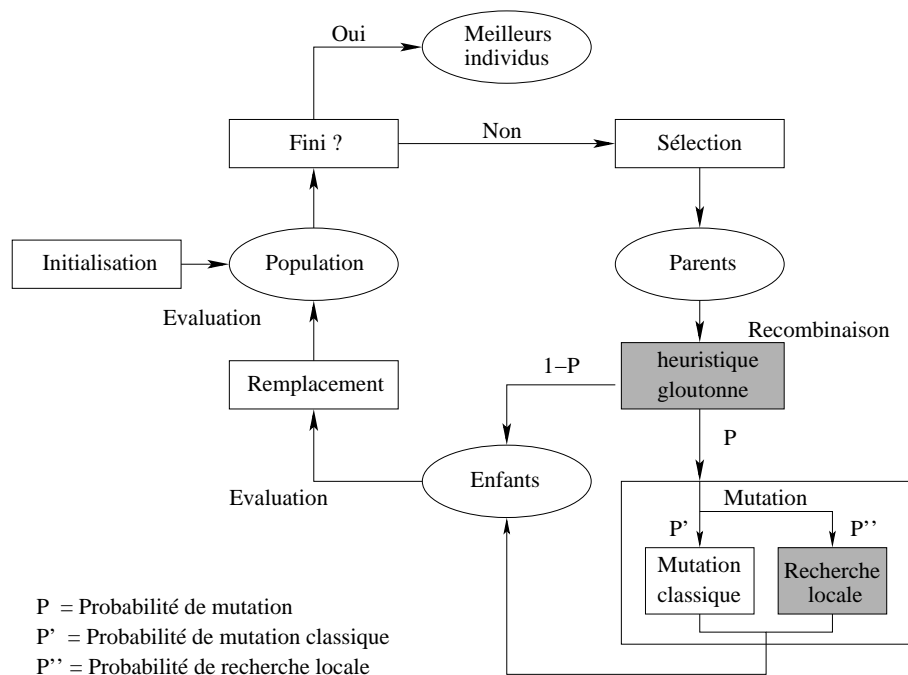
Fig. 4.3 – Les différents modes de coopération.

*Problem*). Un recuit simulé est combiné avec une méthode par descente pour pouvoir générer à partir d'un optimum local courant, un nouvel optimum local différent qui soit un candidat pour le recuit simulé. Cela consiste en une perturbation (un *kick* dans [123]) de l'optimum local pour pouvoir lui appliquer par la suite la recherche par descente pour obtenir le nouvel optimum local. Afin de s'assurer de ne pas retomber sur le même optimum local, il peut être nécessaire d'utiliser une perturbation fonctionnant dans un autre voisinage que celui utilisé par le recuit simulé. Il est à remarquer que ce type de coopération rappelle le principe des recherches locales à voisinage multiple telle que la recherche à voisinage variable (*VNS*).

#### Coopération co-évolutionnaire de bas niveau (*Low-level Teamwork Hybrid*)

Dans cette classe de coopération (**LTH**), on regroupe les métaheuristiques à base de population dans lesquelles un (ou plusieurs) opérateur(s) est remplacé par une (ou des) méthode(s) d'optimisation. Les algorithmes mimétiques en sont un bon exemple de ce type de coopération. Il s'agit d'algorithmes évolutionnaires dans lequel un opérateur est remplacé par une métaheuristique à base de solution unique. L'avantage de ce type de coopération est de combiner le pouvoir d'exploration des algorithmes à base de population de solutions et celui d'intensification des méthodes à base de solution unique. En effet, les algorithmes à base de population de solutions permettent de mieux appréhender les grands espaces de recherche mais n'ont pas un fort pouvoir d'intensification et peuvent avoir tendance à converger lentement. Inversement, les métaheuristiques à base de solution unique ont une très grande capacité d'intensification mais cela peut entraîner le blocage de la méthode dans des optima locaux suivant la forme du paysage. Par conséquent la coopération de ces deux familles de métaheuristiques permet d'obtenir des algorithmes ayant un grand pouvoir d'exploration et d'intensification. Dans ce type de coopération, généralement

l'opérateur de mutation est remplacé, soit par une recherche par descente [163], soit par une recherche tabou [56] ou encore par un recuit simulé [25]. Il existe aussi des coopérations où l'opérateur de recombinaison est remplacé par une méthode d'optimisation telle qu'un algorithme glouton [69].

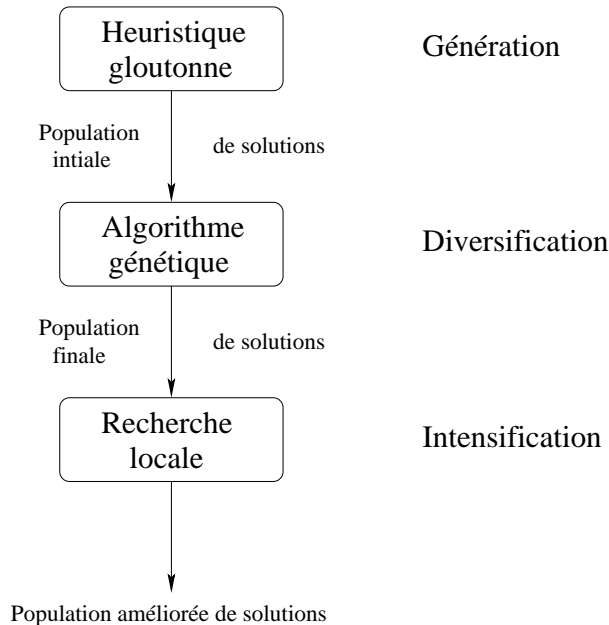


**Fig. 4.4** – Schéma d'une coopération de type co-évolutionnaire de bas niveau (*LTH*) : un algorithme génétique est hybridé avec une heuristique gloutonne (recombinaison) et une recherche locale (mutation).

### Coopération relais de haut niveau (*High-level Relay Hybrid*)

Au sein de cette classe de coopération (**HRH**), les méthodes d'optimisation gardent leur intégrité et sont utilisées de manière séquentielle. La (ou les) solution(s) finale(s) de la première méthode devient(deviennent) la (ou les) solution(s) initiale(s) de la méthode suivante. Cet enchaînement de méthodes peut notamment permettre de procéder à une phase de diversification pour couvrir une grande partie de l'espace de recherche (algorithmes génétiques) avant d'arriver à une phase d'intensification pour améliorer les solutions obtenues (recherches locales). Dans [167], un algorithme génétique est suivi d'une Recherche Tabou afin d'améliorer les solutions obtenues. Il est aussi répandu d'utiliser des méthodes rapides et peu coûteuses de type glouton pour générer des solutions initiales de bonne qualité à une autre méthode plus coûteuse de type métaheuristique

à base de population de solutions. La figure 4.5 montre un exemple d'enchaînement de trois méthodes d'optimisation différentes.



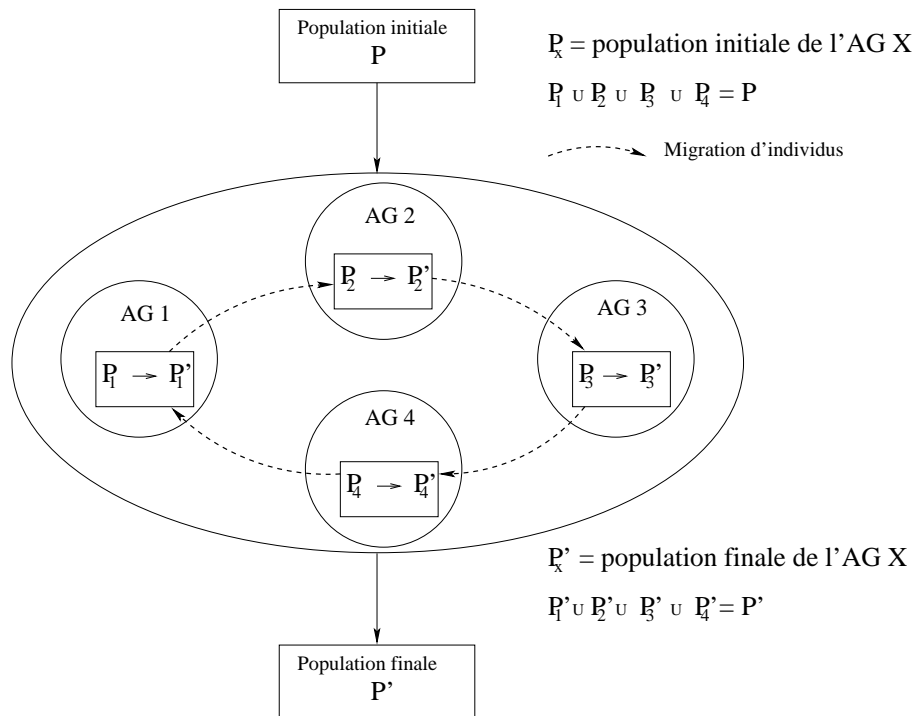
**Fig. 4.5** – Schéma d'une coopération de type relais de haut niveau (*HRH*) : une heuristique glouton génère la population initiale d'un algorithme génétique et ensuite la population finale de l'AG est améliorée grâce à l'application d'une recherche locale sur chaque individu de la population.

### Coopération co-évolutionnaire de haut niveau (*High-level Teamwork Hybrid*)

Cette classe de coopération (**HTH**) correspond à l'exécution en parallèle de méthodes d'optimisation qui peuvent communiquer entre elles durant leur exécution. Un exemple de ce type de coopération est l'exécution d'un algorithme génétique en utilisant un modèle insulaire de parallélisation. Dans ce modèle, la population initiale est divisée en sous-population qui sont réparties entre différentes *îles* sur lesquelles un algorithme génétique est lancée. Ainsi, chaque île correspond à une recherche de solutions dans une partie de l'espace de recherche. De manière périodique, un certain nombre d'individus migre entre les îles afin que les sous-populations des différentes îles contribuent toutes à produire les meilleures solutions possibles. En effet, de l'information extérieure peut permettre, à un algorithme génétique qui n'évoluait plus, de pouvoir être relancé afin de tendre vers les meilleures solutions possibles. La figure 4.6 détaille un modèle insulaire d'exécution d'AG dans le lequel la migration d'individus se fait de manière circulaire entre les îles.

Les travaux basés sur cette classe de coopération utilisent aussi bien des algorithmes génétiques [171], des recuits simulés [41], des Recherches Tabou [42], de la programmation génétique [102] ou encore des stratégies évolutionnistes [180].

La coopération de métaheuristiques permet d'obtenir des algorithmes hybrides de grande effica-



**Fig. 4.6** – Schéma d'une coopération co-évolutionnaire de haut-niveau (*HTH*) : un ensemble d'algorithmes génétiques coopèrent selon un modèle insulaire incluant une migration circulaire des individus.

citée mais la mise au point de telles méthodes a un coût. Il faut mettre au point les métaheuristiques, choisir le mode de coopération le plus adapté ou en tester plusieurs. Par la suite, on peut vouloir changer de métaheuristiques. C'est un travail long et fastidieux. Mais quand on regarde de plus près, un grand nombre de caractéristiques liées à la mise en place de métaheuristiques et à leur coopération est tout à fait générique et non dédié à un problème particulier. C'est sur ce principe que la plateforme ParadisEO a été développée : permettre à l'utilisateur de ne s'intéresser qu'aux parties de ses méthodes de résolution tout en ayant accès à toutes les métaheuristiques actuelles.

## 4.2 Méthodes coopératives multiobjectives

En optimisation multiobjectif, la taxinomie présentée ci-dessus s'applique aisément, même si quelques ajustements mineurs sont souvent nécessaires. Ils sont dus au fait que le but de la recherche est désormais de fournir un ensemble de solutions non-dominées, et non une solution unique.

Dans leur étude sur les métaheuristiques hybrides dédiées à la résolution de problèmes d'optimisation multiobjectif, Erghott et al. identifient trois catégories de méthodes d'hybridation entre un algorithme à base de population et un algorithme de recherche locale [52] : une hybridation pour rendre une méthode plus agressive, une hybridation pour diriger une méthode, et une hybridation pour exploiter des forces complémentaires. Les deux premières catégories peuvent être vues

comme appartenant à la classe LTH, la différence étant que la deuxième regroupe les hybridations dont la fonction est spécialisée. Enfin, la troisième catégorie rassemble les métaheuristiques coopératives de la classe HRH, où plusieurs méthodes (généralement deux) sont exécutées en séquence.

Les quatre classes de métaheuristiques hybrides appliquées dans le cadre de l'optimisation multiobjectif sont discutées ci-dessous.

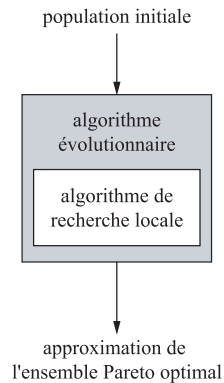
**LRH (*Low-level relais Hybrid*).** Il existe très peu d'exemples de coopération LRH dans la littérature, puisque les métaheuristiques à base de solution unique ne sont pas adaptées à l'approximation de l'ensemble Pareto optimal.

**LTH (*Low-level Teamwork Hybrid*).** Cette classe est probablement la plus représentée, un grand nombre de métaheuristiques hybrides LTH ayant été appliquées à des problèmes d'optimisation multiobjectif. Ces méthodes consistent à coupler une métaheuristique à base de population à une métaheuristique à base de solution unique, telle qu'une recherche locale, un recuit simulé ou encore une recherche tabou, qui sont des méthodes d'optimisation efficaces en terme d'exploitation. Néanmoins, afin d'adapter la métaheuristique à base de solution unique au cas multiobjectif, ce type de méthode requiert généralement la formulation d'une préférence pour guider la recherche locale. La méthode correspondante est donc une méthode de type scalaire, où une quelconque agrégation (typiquement, une somme pondérée) a été réalisée afin de réduire le problème original en un problème monoobjectif.

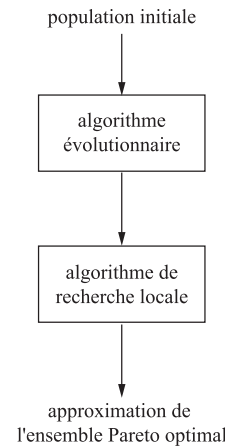
Ainsi, les algorithmes mimétiques sont très populaires dans la communauté multiobjectif [101]. Le principe de base consiste à incorporer un algorithme de recherche locale au sein d'un algorithme évolutionnaire. Cette étape de recherche locale est généralement incluse en lieu et place de l'étape de mutation, même si elle peut également être ajoutée juste après l'étape de mutation, ou encore à chaque itération de l'algorithme général. Un exemple consiste en l'algorithme MOGLS (*multiobjective genetic local search*) proposé par [78, 82]. Il est constitué d'un algorithme évolutionnaire multiobjectif où l'étape de mutation est remplacée par un raffinement local (Fig. 4.7). Une somme pondérée des fonctions objectif est aléatoirement définie afin d'obtenir une fonction fitness scalaire. Des solutions de la population courante sont ainsi sélectionnées selon cette valeur scalaire, puis recombinaées, et enfin localement améliorées à l'aide d'une recherche de voisinage. D'autres exemples plus évolués guident l'étape de recherche locale vers les régions de l'approximation de la frontière Pareto contenant une grande densité de points, ou vers des régions non explorées [52]; la direction de la recherche locale étant définie à l'aide d'une fonction de scalarisation.

**HRH (*High-level relais Hybrid*).** Cette classe de coopération regroupe les hybridations où des métaheuristiques autonomes sont exécutées en séquence. La plupart des méthodes existantes consistent à exécuter une étape d'intensification (recherche locale, path relinking) sur une approximation obtenue à l'aide d'une métaheuristique à base de population. Par exemple, une recherche locale basée sur la dominance Pareto peut être appliquée à une approximation trouvée par un algorithme évolutionnaire multiobjectif [168]. La figure 4.8 montre un exemple typique d'une telle hybridation.

D'autres stratégies consistent à initialiser la population d'une métaheuristique à base de population à l'aide de l'exécution multiple d'une méthode à base de solution unique connue pour être



**Fig. 4.7** – L’algorithme MOGLS : un exemple d’hybridation LTH.



**Fig. 4.8** – Un exemple typique d’hybridation HRH.

efficace pour une variante monoobjectif du problème. Plusieurs exemples sont donnés par [52]. Lors de cette première phase, la résolution est basée sur une somme pondérée des fonctions objectif, et est répétée en faisant varier les poids qui leur sont associés.

**HTH (*High-level Teamwork Hybrid*).** Enfin, dans le cadre de l’optimisation multiobjectif, les hybridations de type HTH impliquent plusieurs métaheuristiques autonomes exécutées en parallèle et coopérant les unes avec les autres afin de trouver une approximation de l’ensemble Pareto optimal. Cette classe d’hybridation est étroitement liée au modèle de parallélisation coopérative au niveau algorithmique, où un ensemble de métaheuristiques autonomes et parallèles échangent de l’information concernant la recherche [165]. Ici, le but de la parallélisation n’est pas tant d’accélérer la recherche que de trouver de meilleures solutions et de rendre la méthode globale plus robuste.

Ainsi, la finalité est de paralléliser la tâche consistant à trouver la totalité de l’ensemble Pareto optimal parmi tous les algorithmes participants. Les différentes métaheuristiques impliquées peuvent posséder différentes structures, ou encore leurs propres opérateurs ou paramètres. Par ailleurs, elles peuvent éventuellement évaluer en sous-ensemble de fonctions objectif, utiliser différents vecteurs d’agrégation pour chaque algorithme, ou diviser l’espace de recherche ou l’espace objectif en plusieurs régions. De façon générale, une ressource de calcul, le processus maître, se charge de la distribution ou de la division de la population, de l’espace de recherche, ou de l’espace objectif, ainsi que de la récupération du résultat final de chacun des algorithmes afin de construire une approximation globale de l’ensemble Pareto optimal.

Les méthodes existantes de la classe HTH se basent presque exclusivement sur le modèle en îles, où une métaheuristique indépendante, utilisant une population distincte, est exécutée sur chaque processeur (Fig. 4.9). Typiquement, il s’agit de plusieurs instances d’une même métaheuristique à base de population (généralement un algorithme évolutionnaire) qui évoluent en parallèle [13, 166]. Ces différents algorithmes coopèrent les uns avec les autres en s’échangeant régulièrement des solutions de bonne qualité au sein de leurs sous-populations (migrations). La plupart des approches basées sur le modèle en îles ne divisent pas directement l’espace (de recherche ou

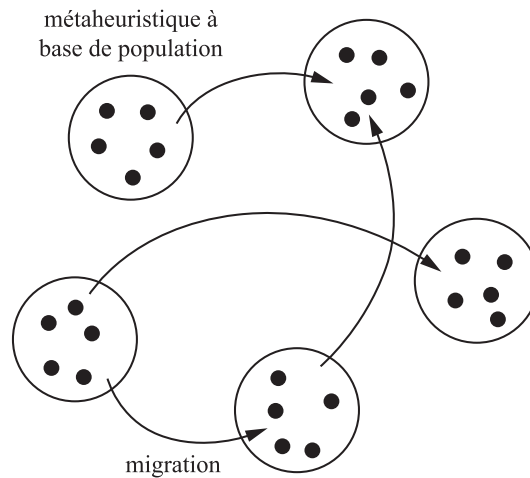


Fig. 4.9 – Le modèle en îles : un exemple d'hybridation HTH.

objectif) en différentes régions, mais résultent en la division implicite d'une population de très grande taille en sous-populations. Différentes variantes peuvent être élaborées en fonction des stratégies de communication définies entre les îles. Tout d'abord, différentes topologies d'échange d'informations peuvent être définies : en anneau, en maille, en hypercube, ou à l'aide d'un graphe complet. Ensuite, ces stratégies de communication peuvent soit se baser sur les sous-populations des différentes îles, sur des archives de solutions non-dominées distribuées sur les différents nœuds, ou sur une combinaison des deux [127].

Aussi, une autre question essentielle à se poser lors du développement de ce type de métaheuristique hybride et parallèle concerne la gestion de l'archive [14]. Celle-ci peut prendre la forme d'une *archive centralisée*, représentée à l'aide d'une structure de donnée centralisée construite au fur et à mesure du processus de recherche. Au contraire, elle peut prendre la forme d'*archives distribuées* parmi les métaheuristiques, de sorte que les algorithmes travaillent avec un ensemble local de solutions non-dominées qui devront, d'une façon ou d'une autre, être combinées à la fin de la recherche. Comme souligné par [14], une archive centralisée pure n'est en pratique jamais utilisée pour des questions évidentes d'efficacité. Toutes les approches centralisées de la littérature combinent en réalité des phases distribuées, où des solutions non-dominées locales sont considérées, et des phases de reconstruction de l'archive centrale.

### 4.3 Nouvelles métaheuristique coopératives

Nous présentons deux approches coopératives issues de nos travaux afin d'illustrer deux types de coopération différents.

#### 4.3.1 Une approche de coopération multiobjectif co-évolutive de bas niveau

Dans ce travail, la mutation de l'algorithme génétique multiobjectif est remplacée par une recherche par descente monoobjective. C'est un cas typique de coopération de type co-évolutive de bas niveau **LTH**. Ce type de coopération est fréquente en monoobjectif et améliore signifi-

cativement les résultats obtenus. Ce modèle assez générique permet notamment d'améliorer les bornes du front Pareto. Ce modèle a été expérimenté sur une application de docking moléculaire, ce travail a été effectué dans le cadre de l'ANR *Dock*.

### Problème illustratif : le docking moléculaire

Le problème du docking moléculaire consiste à prédire le complexe pouvant être formé par deux molécules données. Généralement, il s'agit d'une plus petite molécule, le ligand, qui se lie à une plus grosse, le récepteur. Dans le cadre des travaux présentés dans ce mémoire, nous nous intéressons plus particulièrement au docking dit "flexible", c'est-à-dire que la structure du complexe peut légèrement bouger pour accueillir le ligand.

**Modèle multiobjectif** Dans le cadre des travaux de l'ANR *Dock* nous avons proposé plusieurs modèles multiobjectifs. Nous reprenons ici le modèle tri-objectifs basé sur le critère d'énergie (Equation 4.1 composé de l'énergie liée (Equation 4.2) et de l'énergie non liée (Equation 4.3), la notion de surface (Surface accessible au solvant SAS) dont nous ne détaillerons pas le calcul et la notion de "robustesse" ou stabilité 4.4.

$$f_1 : E_{total} = E_{bonded\_atoms} + E_{non\_bonded\_atoms} \quad (4.1)$$

avec

$$E_{bonded\_atoms} = \sum_{bonds} K_b(b - b_0)^2 + \sum_{angles} K_\theta(\theta - \theta_0)^2 + \sum_{torsions} K_\phi(1 - \cos n(\phi - \phi_0)) \quad (4.2)$$

$$E_{non\_bonded\_atoms} = \sum_{Van\ der\ Waals} \frac{K_{ij}^a}{d_{ij}^{12}} - \frac{K_{ij}^b}{d_{ij}^6} + \sum_{Coulomb} \frac{q_i q_j}{4\pi\epsilon d_{ij}} + \sum_{desolvation} \frac{K q_i^2 V_j + q_j^2 V_i}{d_{ij}^4} \quad (4.3)$$

La surface accessible au solvant a été définie par Lee [108]. Dans ce modèle, la molécule est considérée comme étant dans un solvant et ce solvant est modélisé par une sphère *roulant* sur le pourtour de la molécule dont on veut connaître la surface. Pour la surface accessible au solvant, ce parcours suit le mouvement du centre de la molécule symbolisant le solvant. Cette molécule de solvant a généralement un rayon de 1.4 Å (1 angstrom(Å)= 0.1 nanomètre) car elle doit être capable de contenir une molécule d'eau qui est un des solvants standard.

Le dernier critère est le critère de "robustesse" ou de stabilité :

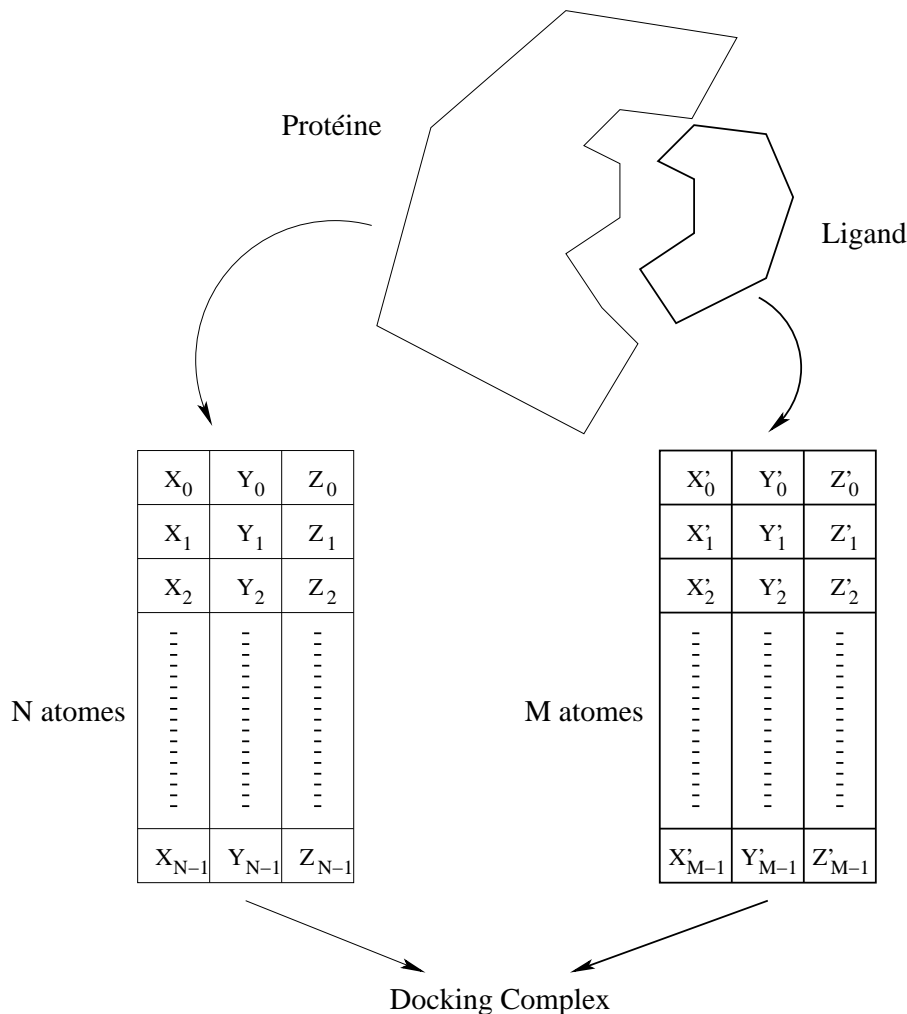
$$f_3 : Robustesse = -\frac{1}{\beta} * \ln \sum_{i=1}^{nbConf} -\beta * Energy[i] \quad (4.4)$$



**Algorithme de résolution**

La méthode de résolution proposée se base sur un algorithme multiobjectif. Les détails de ses composants sont donnés ci-dessous :

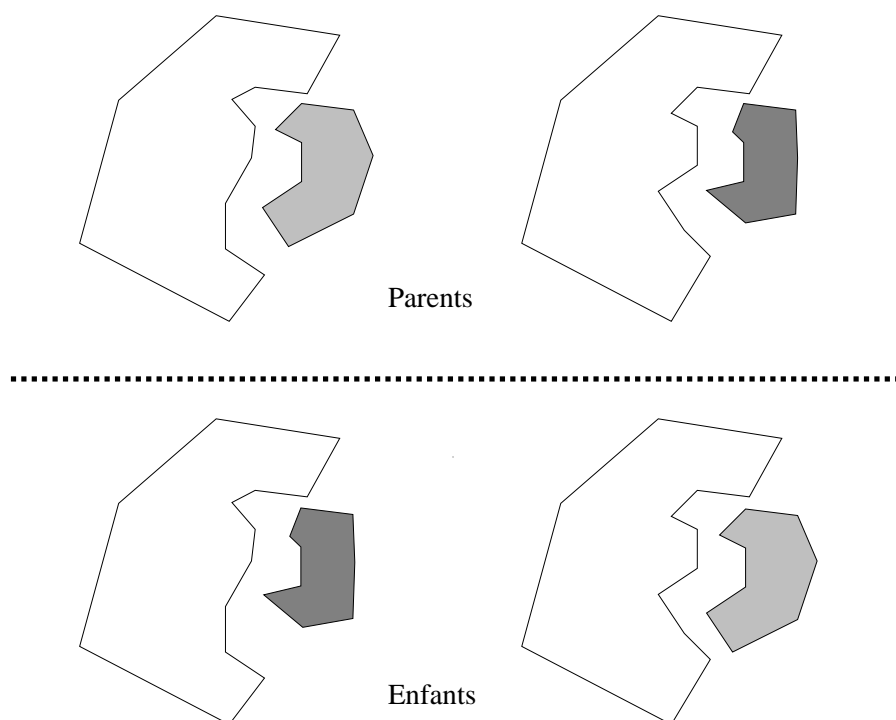
**Représentation** Une solution sera représentée par deux vecteurs de flottants correspondant aux coordonnées atomiques du site et du ligand. La figure 4.10 montre un exemple de ce codage.



**Fig. 4.10** – Représentation d'une solution dans l'algorithme génétique. N et M correspondent respectivement au nombre d'atomes composant le site et le ligand.

**Opérateurs** Des opérateurs de recombinaison et de mutations adaptés à la représentation du problème ont été créés pour l'algorithme génétique.

*Recombinaison* Ici, un mécanisme ne nécessitant pas de vérification de la viabilité des solutions obtenues a été privilégié. La recombinaison basée sur un échange de ligand entre deux sites a été choisie (voir figure 4.11).



**Fig. 4.11** – Opérateur de recombinaison de notre algorithme génétique.

*Mutation* Différents types de mutation ont été réalisées, certaines sont classiques et d'autres spécifiques au problème. Les mutations standards pour le problème de docking flexible sont au nombre de trois :

1. Rotation de conformation : dans cette mutation, une des deux conformations subit une rotation. Le centre de rotation est un point fictif positionné au centre de la conformation, l'angle de rotation est déterminé de manière aléatoire sur les 3 axes X, Y et Z. Dans notre cas, le docking ligand/site, nous ferons des rotations sur le ligand.
2. Translation de conformation : une translation est appliquée sur une des deux conformations selon un vecteur tri-dimensionnel généré aléatoirement. Comme pour la rotation, c'est sur le ligand que va s'appliquer cette mutation.
3. Rotation d'un angle de torsion : cette dernière mutation standard va initier la flexibilité dans le docking. En effet, on va autoriser de modifier un angle de torsion en lui appliquant une rotation. Cela va engendrer une modification plus ou moins importante de la conformation suivant la place de la torsion choisie dans la conformation.

Ces trois premières mutations sont à la base d'un docking moléculaire flexible.

De plus trois nouvelles mutations qui ont chacune un but particulier ont été proposées :

1. Mutation *reverse* : cette mutation consiste à appliquer une rotation à  $180^\circ$  sur une des deux conformations. Dans notre cas, il va s'agir du ligand. Le but de cette mutation est de permettre un redémarrage de la recherche soit dans le cas où une solution reste dans un optimum local et n'arrive plus à être améliorée, soit lorsque la solution est formée d'un ligand qui serait *rentré* dans le mauvais sens à l'intérieur du site ce qui par conséquent a un impact sur le RMSD (Mean Square Deviation) final (voir section résultats, équation 4.5).

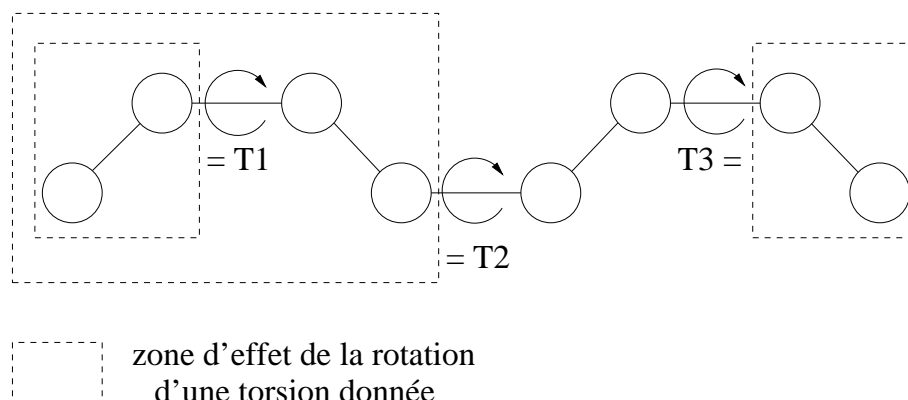
2. Rotations *big* and *small* : cette mutation correspond à une extension de la rotation de conformation. Le but est d'avoir des rotations qui font de grands mouvements et d'autres de plus petits mouvements. Dans la réalité, cela correspond à avoir des rotations de ligand dans l'intervalle  $]0^\circ;10^\circ]$  pour les petites rotations et dans  $[30^\circ;60^\circ]$  pour les grandes rotations. Cela permet d'avoir deux types de mutation qui n'auront pas la même utilité au même moment.
3. Mutation SMO : SMO signifie *Several Mutation in One time*. Il s'agit d'une mutation qui permet de faire plusieurs mutations en une seule fois. Elle est basée sur les limites de chaque mutation. Elle peut faire penser à la notion de voisinage variable, l'idée est de changer la forme du voisinage en permettant l'application de plusieurs mutations en une seule fois. L'intérêt est de tendre plus rapidement vers des solutions de meilleur compromis ou de s'extraire de puits énergétiques importants. Cependant, ce type de mutation, non contrôlée, peut aussi amener à des solutions non réalisables ou de très mauvaise qualité.

### Hybridation

L'objectif de l'hybridation est d'améliorer les résultats obtenus dans nos précédents travaux. Une des hybridations les plus classiques est d'hybrider une métaheuristique à population et une recherche locale. Dans notre cas, la métaheuristique à population est multiobjectif. Nous proposons une hybridation de bas niveau en remplaçant l'opérateur de mutation par différentes recherche par descente. Nous allons tester deux recherche par descente dont le but sera de faire tendre le complexe courant vers un complexe de plus basse énergie par rotation d'une torsion en particulier ou par rotation du ligand :

- Rotation du ligand : dans des fenêtres données, de nouveaux complexes sont générés par rotation du ligand. La taille de ces fenêtres (intervalles de valeurs pour les angles de rotation selon les axes x, y et z) a un impact sur le nombre de complexes générés. Cela représente une sphère de taille donnée où toutes les rotations possibles du ligand dans un espace donné sont réalisées. Le but est de tendre vers le complexe de plus basse énergie le plus proche. Il est clair que pour des raisons de coût, l'intervalle entre les valeurs minimales et maximales d'angle de rotation est relativement restreint (quelques degrés avec un pas de l'ordre du dixième de degré).
- Rotation d'une torsion : le but est de trouver pour une torsion donnée, son meilleur angle pour minimiser l'énergie de la conformation courante. Cette mutation correspond à une intensification d'une partie de la conformation. Comme pour la rotation de ligand, cette mutation se base sur la génération d'un voisinage de taille donnée selon l'intervalle de valeurs pour l'angle de rotation de la torsion. L'impact sur la conformation en son entier dépend de l'endroit où la torsion se trouve. Cela est schématisé par la figure 4.12 dans laquelle un exemple fictif de molécule à trois torsions est proposé. Quelle que soit la torsion choisie, c'est le plus petit morceau de la molécule qui est transformé durant la rotation de torsion ou la partie gauche (choix purement arbitraire ici pour l'exemple) quand les différents morceaux sont équivalents. Comme on peut le remarquer sur la figure, la rotation des torsions T1 et T3 va avoir un impact moindre sur la conformation que la torsion T2. La modification d'énergie sera proportionnelle.

L'utilisation de recherches par descente dans un algorithme génétique correspond à une coopération co-évolutionnaire de bas niveau comme cela était présenté dans la section 4.1.4. L'utilisation de recherches monoobjectif par descente va avoir un impact sur le principe de mutation de l'al-



**Fig. 4.12** – Impact de la rotation d'une torsion dans la modification de conformation d'une molécule donnée. Dans notre exemple, chaque rotation de torsion a un impact sur le plus petit morceau de la molécule (à gauche pour T1 et à droite pour T3) ou sur la partie gauche (T2)

gorithme génétique. Ces recherches locales sont concentrées sur la minimisation de l'énergie de complexe quitte à dégrader les autres objectifs. Le but de cette mutation est d'essayer de pousser le complexe vers une conformation stable du point de vue énergétique. Ainsi dès qu'une importante modification entraîne une montée de l'énergie, ce type de mutation permet d'essayer de trouver le minimum local énergétique associé.

## Résultats

Nous présentons ici une courte synthèse des résultats obtenus, plus de détails sont disponibles de la thèse de Jean-Charles Boisson [23] et l'article [R3] donné en annexe. Dans ce travail, nous avons repris des algorithmes multiobjectifs classiques de la littérature : NSGA-II et IBEA. Ils ont été comparés sur différentes instances (6rsa, 1mbi, 2tsc, 1htf et 1dog). Ces algorithmes seront comparés par rapport au RMSD qui sera à minimiser. Ce critère est pour les problèmes de docking le seul qui peut-être appliqué pour comparer les différents modèles et méthodes existantes à partir du moment où les mêmes instances sont utilisées. RMSD signifie **M**ean **S**quare **D**eviation (RMSD). Selon [173], le RMSD est défini de la manière suivante :

$$RMSD = \sqrt{\frac{\sum_{i=1}^n (dx_i^2 + dy_i^2 + dz_i^2)}{n}} \quad (4.5)$$

Dans l'équation 4.5,  $n$  correspond au nombre total d'atomes (sans considérer les hydrogènes).  $dx_i$ ,  $dy_i$  et  $dz_i$  sont les différences de coordonnées atomiques entre le ligand dans sa position obtenue par l'algorithme et sa position considérée comme optimale (obtenue par cristallographie).

Le RMSD permet de définir la distance entre une conformation donnée et une conformation objectif. Pour le docking, ce critère correspond à évaluer la position du ligand d'une solution finale par rapport à sa position dans la solution considérée comme optimale.

Le tableau 4.1 confirme les résultats selon lesquels IBEA serait meilleur que NSGA-II sur le problème du docking moléculaire pour les instances considérées. Pour chaque instance, IBEA amène à des RMSD meilleures ou au moins égales à celles de NSGA-II.

TABLE 4.1 – Meilleurs résultats pour chaque instance avec les algorithmes NSGA-II et IBEA, pour 30 exécutions. Pour chaque algorithme, la meilleure RMSD ainsi que la déviation standard associée sont données.

Instance	NSGA-II		IBEA	
	RMSD (Å)	Déviatiion standard	RMSD (Å)	Déviatiion standard
6rsa	1,66	<b>1,04</b>	<b>1,32</b>	1,3
1mbi	5,2	<b>0,4</b>	<b>4,16</b>	0,8
2tsc	<b>2,19</b>	2,75	<b>2,19</b>	<b>2,68</b>
1htf	2,88	2,64	<b>2,59</b>	<b>1,33</b>
1dog	4,38	0,99	<b>2,44</b>	<b>0,56</b>

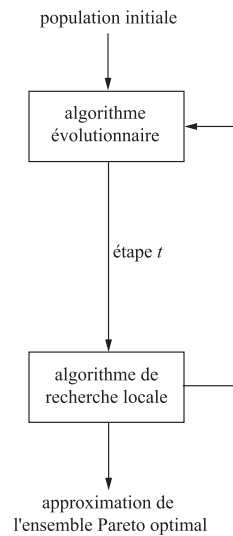
TABLE 4.2 – Les 6 profils testés pour le docking flexible. SR : rotation et translation du ligand, TorsRot rotation d’une torsion, SBRot rotation avec une petite et une grande valeur pour l’angle de la rotation, SMO plusieurs mutations en même temps et les deux modèles hybrides LRHC avec une recherche locale sur la rotation du ligand et TRHC qui réalise une recherche locale sur la rotation d’une torsion.

Config	SR	TorsRot	Reverse	SBRot	SMO	LRHC	TRHC
P1	X	X					
P2	X	X	X				
P3	X	X		X			
P4	X	X			X		
P5	X	X				X	
P6	X	X					X

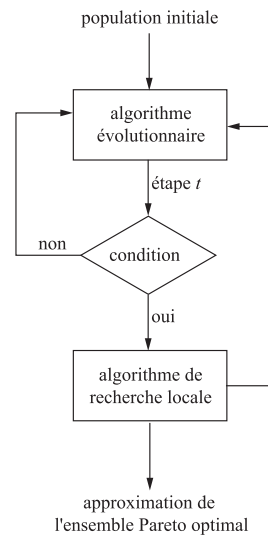
TABLE 4.3 – Synthèse des résultats par profil toutes instances confondues.

	P1	P2	P3	P4	P5	P6
<b>Minimum</b>	2	2,4	2	1,9	2,2	2,4
<b>1<sup>er</sup> quartile</b>	3,2	3,3	3,4	3,3	3,825	3,25
<b>Médiane</b>	<b>4,05</b>	4,5	<b>4,1</b>	4,35	<b>4,1</b>	4,6
<b>Moyenne</b>	<b>4,343</b>	4,729	5,126	5,169	4,733	4,776
<b>3<sup>ème</sup> quartile</b>	5,075	6	6,3	6,325	<b>4,775</b>	6
<b>Maximum</b>	10,5	10,8	11	12,6	14	8,5
<b>Déviatiion standard</b>	1,651	1,937	2,524	2,865	2,29	1,794

Concernant les profils, si on observe les résultats données tableau 4.3, ce sont les profils P1, P5 et P6 qui proposent les meilleurs couples RMSD/stabilité. Par conséquent, la gestion de la flexibilité mais aussi l’utilisation de recherches locales amènent à de meilleurs résultats que les configurations classiques rigides.



**Fig. 4.13** – Une approche de coopération périodique (PCS).



**Fig. 4.14** – Une approche de coopération adaptative (ACS).

### 4.3.2 Une approche de coopération multiobjectif de haut niveau en mode relais

L'approche proposée réalise une collaboration entre des métaheuristiques de types différents (un algorithme évolutionnaire et un algorithme de recherche locale), ceci afin de profiter de leurs comportements respectifs. L'idée générale de notre schéma de coopération consiste à utiliser l'algorithme évolutionnaire multiobjectif comme processus principal, et de lancer régulièrement une méthode de recherche locale afin d'alterner la recherche entre phases d'exploration et phases d'exploitation.

Partant de ce principe, nous pouvons facilement imaginer deux versions de la métaheuristique coopérative : une version périodique, dans laquelle la recherche locale serait lancée à chaque étape de l'algorithme, et une version adaptative, dans laquelle la recherche locale ne serait lancée que si une certaine condition, en relation avec la vitesse de convergence de l'algorithme évolutionnaire, est vérifiée. Ces deux approches sont respectivement dénotées PCS (*periodic cooperative search*) et ACS (*adaptive cooperative search*), et sont illustrées dans les figures 4.13 et 4.14.

Ainsi, la stratégie autoadaptative de ACS doit décider par elle-même, et en cours de recherche, s'il s'avère intéressant de lancer la recherche locale à une étape donnée du processus global. La condition que nous allons utiliser se base sur la convergence de l'approximation courante (contenue dans l'archive ou dans la population principale) d'une phase à une autre. Une possibilité est donc de mesurer la qualité de l'approximation en cours  $A^t$  en comparaison à celle de l'étape précédente  $A^{t-1}$ . Différents indicateurs existent pour évaluer la qualité d'une approximation par rapport à une autre, que ce soit en termes de convergence ou de diversité.

Par rapport aux méthodes de la littérature, l'originalité de ce schéma de coopération est (i) de ne faire intervenir aucun mécanisme de scalarisation des fonctions objectif pour convertir le problème d'optimisation multiobjectif en un problème monoobjectif, et (ii) d'alterner entre

recherche évolutionnaire et recherche locale, là où les méthodes existantes se contentent généralement d'exécuter ces deux méthodologies l'une à la suite de l'autre, avant de stopper le processus de recherche. En outre, l'une des variantes que nous proposons, ACS, détecte automatiquement le moment présumé idéal pour démarrer une phase de recherche locale, ceci en fonction du scénario de l'optimisation.

### Cadre d'étude : le problème du Ring-Star

Le problème de Ring-Star (RSP) peut être défini comme suit. Soit  $G = (V, E, A)$  un graphe mixte complet où  $V = \{v_1, v_2, \dots, v_n\}$  est un ensemble de sommets,  $E = \{\{v_i, v_j\} | v_i, v_j \in V, i < j\}$  est un ensemble d'arêtes, et  $A = \{(v_i, v_j) | v_i, v_j \in V\}$  est un ensemble d'arcs. Le sommet  $v_1$  représente le dépôt. À chaque arête  $\{v_i, v_j\} \in E$  est associé un coût-anneau (*ring cost*) non nul, noté  $c_{ij}$ , et à chaque arc  $(v_i, v_j) \in A$  est associé un coût d'affectation (*assignment cost*) non nul, noté  $d_{ij}$ . Le problème de Ring-Star bi-objectif consiste à identifier un cycle sur un sous-ensemble de nœuds du graphe  $V' \subset V$  (avec  $v_1 \in V'$ ) tout en (i) minimisant la somme des coûts-anneau des arêtes du cycle, et en (ii) minimisant la somme des coûts d'affectation des arcs allant de chaque nœud non visité vers un nœud visité (de sorte que le coût d'affectation associé soit minimum). La première fonction objectif, le coût-anneau, est définie comme suit.

$$\sum_{[v_i, v_j] \in E} c_{ij} b_{ij} \quad (4.6)$$

où  $b_{ij}$  est une variable binaire égale à 1 si et seulement si l'arête  $\{v_i, v_j\}$  appartient au cycle. Le second objectif, le coût d'affectation, est défini ainsi :

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \quad (4.7)$$

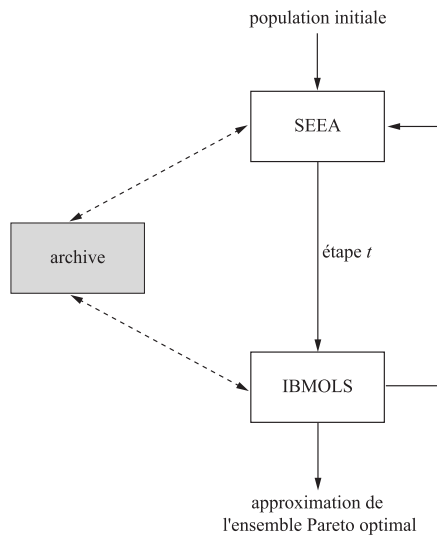
### Mise en œuvre

Une instanciation du schéma de coopération présenté auparavant est réalisé et appliqué au problème de Ring-Star, en utilisant comme algorithme évolutionnaire SEEA [R4] et comme recherche locale d'une variante de l'algorithme IBMOLS [11, I7].

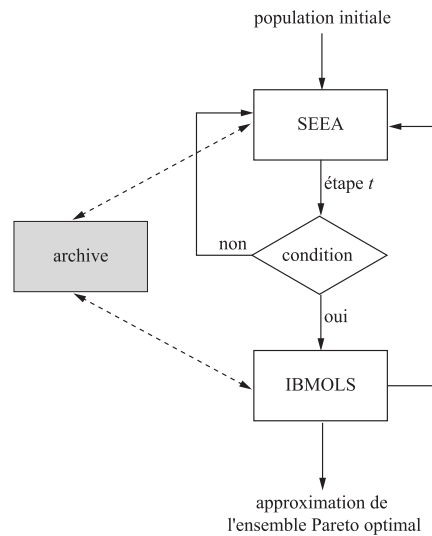
Les figures 4.15 et 4.16 résument les schémas d'hybridation rencontrés. Nous rappelons que ces hybridations appartiennent à la classe LTH, et peuvent donc être toutes deux dénotées par LTH(SEEA + IBMOLS) selon la grammaire proposée par [164].

Les tableaux 4.4 et 4.5 fournissent une comparaison des résultats obtenus par SEEA, I-IBMOLS, PCS et ACS selon la métrique  $I_H^-$  et la métrique  $I_{\epsilon+}^1$ , respectivement.

Ces deux algorithmes hybrides ont été testés sur un problème biobjectif du RingStar et ont permis une amélioration statistiquement prouvée des résultats sur un grand nombre d'instances, en particulier les instances de grande taille. Par ailleurs, en comparaison des approches de résolution du problème de Ring-Star monoobjectif de la littérature, les algorithmes hybrides proposés pour la contrepartie biobjectif du même problème semblent fournir des solutions de bonne qualité. Toutefois, la différence d'efficacité entre PCS et ACS est presque négligeable. Ceci peut être expliqué par le fait que la stratégie autoadaptative de ACS consacre un temps de calcul important dans le but de déterminer si la recherche locale doit être lancée ou non à chaque étape de



**Fig. 4.15** – Instanciation du modèle PCS à l'aide des algorithmes SEEA et IBMOLS.



**Fig. 4.16** – Instanciation du modèle ACS à l'aide des algorithmes SEEA et IBMOLS.

l'algorithme. En effet, au cours de nos expériences, nous avons observé que la différence entre le nombre moyen de fois où IBMOLS est lancé au cours du processus de recherche de PCS et ACS est relativement mince. Ceci peut être expliqué par le fait que (i) IBMOLS consomme plus de temps de calcul à trouver des solutions intéressantes en partant d'une population de moins bonne qualité, ce qui est le cas pour PCS en comparaison à ACS (au moins pour le premier lancement), et (ii) une partie du temps d'exécution alloué à ACS est utilisé pour calculer les valeurs de contribution afin de vérifier la condition d'application de la recherche locale, là où PCS consacre la totalité du temps qui lui est accordé à la recherche. Ces deux aspects ont probablement conduit au fait que le nombre de fois où IBMOLS est lancé au sein de PCS et ACS est, finalement, plus ou moins équilibré entre les deux méthodes de coopération.



TABLE 4.4 – Comparaison des algorithmes selon l'indicateur  $I_H^-$ . Pour chaque instance, soit l'algorithme situé sur une ligne donnée domine significativement l'algorithme situé sur une colonne donnée ( $\succ$ ), soit il est significativement dominé ( $\prec$ ), soit il n'existe pas de différence significative entre les deux algorithmes ( $\equiv$ ). La valeur entre parenthèses correspond à la  $I_H^-$ -valeur moyenne ( $\times 10^{-3}$ ).

			I-IBMOLS	SEEA	PCS	ACS
<i>eil51</i>	PCS	(4.751)	$\equiv$	$\equiv$	.	$\equiv$
	ACS	(4.482)	$\equiv$	$\equiv$	$\equiv$	.
<i>st70</i>	PCS	(2.691)	$\succ$	$\succ$	.	$\equiv$
	ACS	(2.865)	$\equiv$	$\succ$	$\equiv$	.
<i>kroA100</i>	PCS	(3.738)	$\equiv$	$\succ$	.	$\equiv$
	ACS	(3.326)	$\succ$	$\succ$	$\equiv$	.
<i>bier127</i>	PCS	(3.071)	$\equiv$	$\succ$	.	$\succ$
	ACS	(3.693)	$\equiv$	$\succ$	$\succ$	.
<i>kroA150</i>	PCS	(2.792)	$\succ$	$\succ$	.	$\equiv$
	ACS	(2.624)	$\succ$	$\succ$	$\equiv$	.
<i>kroA200</i>	PCS	(2.247)	$\succ$	$\succ$	.	$\equiv$
	ACS	(2.260)	$\succ$	$\succ$	$\equiv$	.
<i>pr264</i>	PCS	(1.342)	$\succ$	$\succ$	.	$\equiv$
	ACS	(1.404)	$\succ$	$\succ$	$\equiv$	.
<i>pr299</i>	PCS	(1.277)	$\equiv$	$\succ$	.	$\equiv$
	ACS	(1.293)	$\equiv$	$\succ$	$\equiv$	.
<i>pr439</i>	PCS	(0.348)	$\succ$	$\succ$	.	$\succ$
	ACS	(0.733)	$\succ$	$\equiv$	$\succ$	.
<i>pr1002</i>	PCS	(2.449)	$\succ$	$\succ$	.	$\succ$
	ACS	(0.707)	$\succ$	$\succ$	$\succ$	.

TABLE 4.5 – Comparaison des algorithmes selon l'indicateur  $I_{c+}^1$ . Pour chaque instance, soit l'algorithme situé sur une ligne donnée domine significativement l'algorithme situé sur une colonne donnée ( $\succ$ ), soit il est significativement dominé ( $\prec$ ), soit il n'existe pas de différence significative entre les deux algorithmes ( $\equiv$ ). La valeur entre parenthèses correspond à la  $I_{c+}^1$ -valeur moyenne ( $\times 10^{-3}$ ).

			I-IBMOLS	SEEA	PCS	ACS
<i>eil51</i>	PCS	(9.561)	$\equiv$	$\equiv$	.	$\equiv$
	ACS	(9.363)	$\equiv$	$\equiv$	$\equiv$	.
<i>st70</i>	PCS	(6.328)	$\succ$	$\equiv$	.	$\equiv$
	ACS	(7.064)	$\equiv$	$\equiv$	$\equiv$	.
<i>kroA100</i>	PCS	(8.963)	$\equiv$	$\equiv$	.	$\equiv$
	ACS	(7.533)	$\succ$	$\succ$	$\equiv$	.
<i>bier127</i>	PCS	(8.114)	$\equiv$	$\succ$	.	$\succ$
	ACS	(9.818)	$\equiv$	$\succ$	$\succ$	.
<i>kroA150</i>	PCS	(5.450)	$\succ$	$\succ$	.	$\equiv$
	ACS	(5.587)	$\succ$	$\succ$	$\equiv$	.
<i>kroA200</i>	PCS	(5.057)	$\succ$	$\succ$	.	$\equiv$
	ACS	(5.702)	$\succ$	$\succ$	$\equiv$	.
<i>pr264</i>	PCS	(4.242)	$\succ$	$\equiv$	.	$\equiv$
	ACS	(4.317)	$\succ$	$\equiv$	$\equiv$	.
<i>pr299</i>	PCS	(4.501)	$\equiv$	$\succ$	.	$\equiv$
	ACS	(4.048)	$\equiv$	$\succ$	$\equiv$	.
<i>pr439</i>	PCS	(2.760)	$\succ$	$\succ$	.	$\succ$
	ACS	(5.553)	$\succ$	$\succ$	$\succ$	.
<i>pr1002</i>	PCS	(5.391)	$\succ$	$\succ$	.	$\equiv$
	ACS	(4.309)	$\succ$	$\succ$	$\equiv$	.

## 4.4 Conclusion

Dans ce chapitre, nous avons présenté la taxonomie qui servira de base à la suite du document. Nous avons ensuite présenté deux exemples de coopération afin d'illustrer nos apports en terme de coopération entre métaheuristiques : une de type bas niveau co-évolutionnaire et une de type haut niveau co-évolutionnaire. Chacune a été appliquée à un problème spécifique issu dans un cas de la biologie et dans le deuxième de la logistique. Nous avons montré que ce type de coopération permettait d'améliorer sensiblement la qualité des résultats obtenus quelque soit l'application choisie. Les perspectives de ce chapitre sont accès sur l'automatisation de la réalisation de l'hybridation de métaheuristiques multiobjectives. L'hybridation de métaheuristiques pose deux questions dont la réponse est difficile à trouver pour le spécialiste et encore plus pour le non spécialiste : quelles métaheuristiques hybrider et comment paramétrer les métaheuristiques ainsi que le schéma d'hybridation. En ce sens, une perspective intéressante est de définir de façon adaptative les mécanismes de coopération afin de sélectionner dynamiquement la méthode la plus adaptée selon tel ou tel critère de convergence, ou de diversité. Une analyse de paysage multiobjectif *a priori* permettrait notamment un choix judicieux des méthodes mais une connaissance *online* permettrait de mieux s'adapter au processus de recherche.

## Chap. 5

# Coopération métaheuristiques-méthodes exactes

### Sommaire

---

<b>5.1</b>	<b>Extension de la taxonomie</b> . . . . .	<b>52</b>
5.1.1	Les classes hiérarchiques . . . . .	52
5.1.2	Classification à plat . . . . .	56
5.1.3	Implémentation . . . . .	58
<b>5.2</b>	<b>Etat de l'art des coopérations Méta/Exacte</b> . . . . .	<b>59</b>
5.2.1	Grammaire et vocabulaire de la classification . . . . .	59
5.2.2	Classification . . . . .	60
<b>5.3</b>	<b>Conclusion</b> . . . . .	<b>63</b>

---

Ce chapitre a donné lieu aux publications suivantes :

- L. Jourdan, M. Basseur, and E.-G. Talbi. *Hybridizing exact methods and metaheuristics : A taxonomy*. European Journal of Operational Research (EJOR), **199**(3) : pages 620–629, 2009 december.
- M. Basseur, L. Jourdan and E-G. Talbi, *Towards parallel design of hybrids between Metaheuristics and exact methods*. In Parallel combinatorial Optimization, chapter 7, pp. 163-186, ISBN : 0-471-72101-8, November 2006.

L'objectif de ce chapitre est de montrer que l'hybridation de métaheuristiques peut également se réaliser avec des méthodes exactes. Nous présentons ici une extension de la taxonomie présentée au chapitre précédent et l'illustrons à travers différents exemples. Puis nous présentons une classification de différents travaux suivant une grammaire permettant de classifier facilement les coopérations entre les différents types de méthodes.

## 5.1 Extension de la taxonomie

De plus en plus de travaux proposent des coopérations entre métaheuristiques et méthodes exactes en optimisation mono-objectif et progressivement en optimisation multiobjectif. Les méthodes proposées sont généralement efficaces, car la coopération permet d'associer leurs avantages afin d'obtenir de bons résultats.

Parmi ces méthodes, il existe une grande diversité de techniques mises en œuvre afin de faire coopérer les différents types de méthode d'optimisation. Dans cette section, nous étendons la taxonomie présentée dans le chapitre précédent afin de pouvoir mieux prendre en compte le caractère particulier de ce type de coopération. En effet, dans le cadre d'une coopération entre métaheuristiques et méthodes exactes, la nature de la méthode obtenue peut être différente en terme de type de méthode de résolution. Dans certains cas, nous pouvons avoir une méthode qui fournit des solutions exactes au problème alors que dans d'autres cas, la résolution est de type approchée. Une nouvelle catégorie apparaît donc dans la classification à plat : le type de résolution fourni par l'hybridation. La figure 5.1 présente l'extension de la taxonomie présentée dans le chapitre précédent.

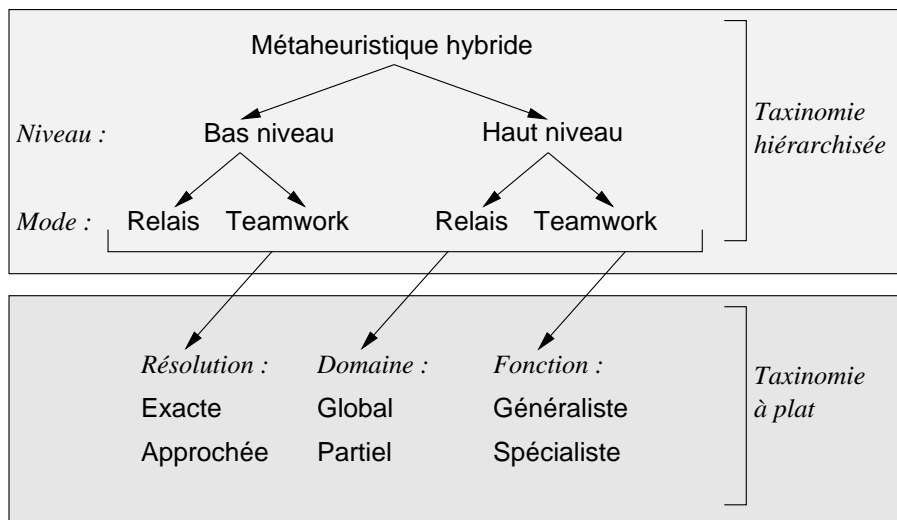


Fig. 5.1 – La taxonomie étendue pour l'hybridation entre métaheuristiques et méthodes exactes.

### 5.1.1 Les classes hiérarchiques

Selon la classification hiérarchique, nous obtenons, à partir des deux propriétés discriminantes que sont le niveau et le mode de coopération, quatre classes détaillées ci-dessous. Nous aurions pu

développer le classement hiérarchique en fonction d'autres discriminants (ceux de la classification à plat) mais nous préférons l'arrêter là, pour éviter d'augmenter le nombre de classes. Nous proposons pour chaque classe des exemples de coopération trouvés dans la littérature pour la coopération entre métaheuristiques et méthodes exactes.

#### **La classe LRH - *Low-level Relay Hybrid***

La classe de coopération de bas niveau en mode relais (**LRH**) regroupe les coopérations constituées d'une méthode d'optimisation à solution unique dans laquelle est insérée une (autre) méthode d'optimisation. Les métaheuristiques utilisant une population de solutions ne sont pas vraiment adaptées à ce type de coopération en mode relais (bien que cela reste possible, si celle-ci remplace un mécanisme d'une méthode d'optimisation à solution unique).

Ce type de coopération est plus facilement envisageable en ayant une méthode heuristique au service d'une méthode exacte. Un exemple de ce type de coopération est proposé par Augerat *et al.* [6]. Dans cette étude, un algorithme de *Branch & Cut* est proposé pour résoudre un problème de tournées de véhicules avec contraintes de capacité. La génération de plans de coupe est une partie cruciale déterminant l'efficacité des algorithmes de *B&C*. Les auteurs partent de la remarque que les inégalités du programme linéaire traitant des contraintes de capacités sont celles qui sont à l'origine des meilleures plans de coupes, ce qui permet les meilleures améliorations de la borne lorsque ces contraintes sont ajoutées à la formulation du programme linéaire relaxé. Trois approches heuristiques, une constructive, une gloutonne, et une Tabou, sont envisagées pour extraire un ensemble pertinent de contraintes de capacité violées du problème relaxé.

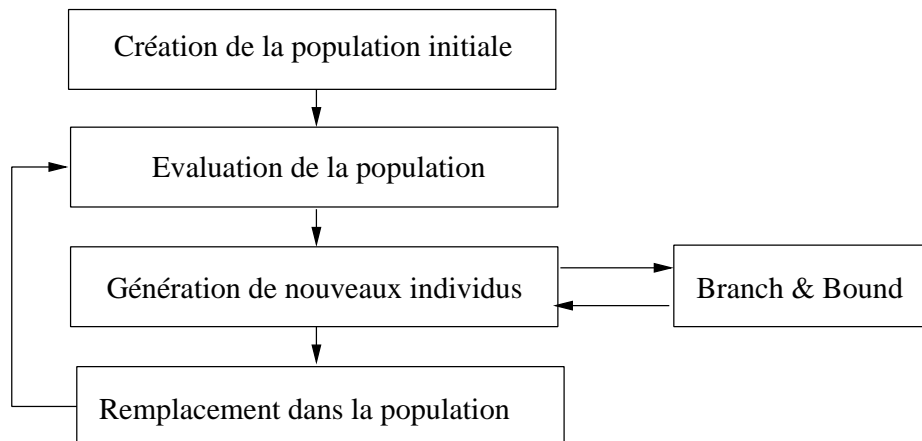
#### **La classe LTH - *Low-level Teamwork Hybrid***

La classe de coopération de bas niveau en mode *teamwork* (**LTH**) regroupe les coopérations constituées d'une méthode d'optimisation à population de solutions pour laquelle un opérateur agissant sur les solutions de manière individuelle ou globale est remplacé par une méthode d'optimisation.

Dans cette classe de coopération, diverses approches sont proposées. D'une part, des approches de type "algorithmes mimétiques" peuvent être rencontrées dans la littérature. Ainsi Cotta *et al.* proposent une approche basée sur la coopération entre *AGs* et *B&B* [37]. L'algorithme exact est incorporé dans l'*AG* pour servir d'opérateur de recombinaison. L'opérateur résultant explore de manière intelligente les solutions potentielles issues de la recombinaison de deux parents, afin d'en extraire la plus intéressante grâce à un algorithme de *B&B*.

Jahura *et al.* proposent différentes coopérations entre *AGs* et méthodes exactes appliquées au problème du voyageur de commerce [79, 80]. La coopération est introduite dans les opérateurs génétiques. L'opérateur de recombinaison est remplacé par un algorithme de *B&B*, d'arbre de recouvrement minimal et de méthode de *backtracking* (méthode utilisée pour reconstruire le chemin emprunté pour la résolution du problème en partant de la fin). De plus, dans cette étude, la population initiale est engendrée à l'aide d'un arbre de recouvrement minimal engendré de manière exacte.

Des algorithmes hybrides différents de ceux de type "algorithme mimétique" ont été proposés, comme l'approche de type **LTH**, proposée par Kostikas & Fragakis [103]. Un algorithme de Programmation Génétique (*PG*) est incorporé dans un algorithme de *B&B* basé sur un problème de Programmation Linéaire Mixte (*PLM*). L'architecture coopérative employée ici utilise la *PG* pour générer l'expression dédiée à la sélection des noeuds à explorer. L'algorithme de *PG*



**Fig. 5.2** – Schéma général de l'AG hybride de Cotta *et al.* [37] : un exemple de coopération de type **LTH**.

exploite les caractéristiques du *PLM* étudié pour faire évoluer la méthode de sélection de nœud. La méthode générée remplace alors l'opérateur de sélection par défaut de l'algorithme de *B&B* pour le reste de l'exécution.

Les algorithmes de recherche sur un voisinage large (*large neighborhood search*) sont des algorithmes de descente qui utilisent des voisinages larges pour améliorer l'efficacité de l'algorithme. La méthode d'exploration du voisinage peut être aussi bien heuristique qu'exacte. Un tour d'horizon de ces méthodes est proposé dans [2]. Les algorithmes de recherche sur large voisinage, faisant intervenir des méthodes exactes pour l'exploration du voisinage, sont des algorithmes typiquement **LTH**. La méthode exacte intervient durant la recherche pour trouver la (les) meilleure(s) solution(s) dans un sous espace de l'espace de recherche global. Plusieurs études utilisent ce principe dont celle de Bent et Van Hentenryck pour le problème de voyageur de commerce asymétrique [17], ou celle de Shaw pour le problème de routage de véhicules [160].

#### **La classe HRH - High-level Relay Hybrid**

Dans les coopérations de haut niveau en mode relais, les heuristiques et/ou méthodes exactes hybridées conservent leur intégrité. Les méthodes coopératives de type **HRH** sont exécutées en séquence. Par exemple, on sait que les algorithmes à base de population de solutions ne parviennent pas à ajuster finement les solutions proches des bons optima. Mais, à l'inverse, leur force réside dans la capacité de trouver rapidement des régions de bonne qualité, même pour des espaces de recherche très vastes ou très complexes. Une fois ces régions repérées, il peut être intéressant de poursuivre la recherche en affinant les solutions performantes qui s'y trouvent ; pour cela, on applique une recherche itérative à solution unique sur la (les) solution(s) trouvées par la première méthode. De même, on utilise également très souvent des heuristiques à convergence très rapide (souvent de type gloutonnes) afin de fournir de bonnes solutions initiales à un algorithme d'optimisation.

Cette classe de coopération inclut notamment les algorithmes exacts utilisant des heuristiques afin d'obtenir des solutions initiales, ou simplement des bornes. Kleipeis *et al.* proposent une coopération entre un " $\alpha$  Branch & Bound" ( $\alpha B&B$ ) et un AG hybridé avec un algorithme

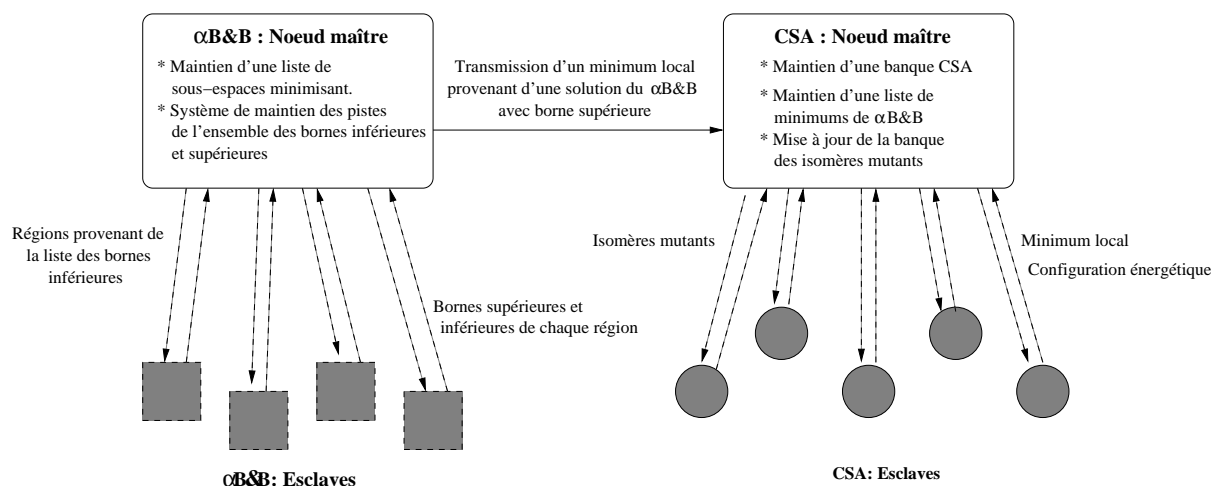


Fig. 5.3 – Exemple de coopération **HRH** pour la prédiction de structure de protéines [98].

de *Conformational Space Annealing (CSA)* pour la prédiction de structure de protéine [98]. L'algorithme de  $\alpha B\&B$  est un algorithme général d'optimisation basé sur le *B&B*, qui peut être appliqué à une large classe de problèmes d'optimisation non-linéaires ayant deux fonctions différentiables. L'algorithme de *CSA* est une méthode stochastique employant différents éléments des *AGs* et du recuit simulé. Dans cet algorithme, les auteurs alternent des exécutions de  $\alpha B\&B$  et de *CSA*. Cet algorithme hybride a été parallélisé dans un modèle maître-esclave (figure 5.3). Dans [17], Bent et Van Hentenryck proposent un algorithme hybride en deux phases pour le problème de tournées de véhicules avec fenêtres de temps. Dans un premier temps l'algorithme minimise le nombre de véhicules par un algorithme de recuit simulé. Puis le coût de chaque tournée est minimisé par une recherche exacte sur un large voisinage pouvant relocaliser un grand nombre de clients.

Un autre exemple de ce type de coopération est proposé par Portmann *et al.* [144] pour résoudre un problème de Flow-shop hybride.

### La classe **HTH** - *High-level Teamwork Hybrid*

Dans les hybridations de haut niveau *teamwork*, la structure interne des métaheuristiques hybridées n'est pas modifiée. Ces dernières sont exécutées simultanément et coopèrent pour résoudre le problème.

Un exemple d'hybridation **HTH** est le modèle des algorithmes génétiques en îles. La coopération peut également se réaliser entre différentes méthodes d'optimisation, et même entre une méthode exacte et une métaheuristique.

Ce type de coopération est assez difficile à mettre en œuvre entre une méthode exacte et une heuristique. En effet, les deux approches ne permettent pas de résoudre les mêmes types de problèmes. Donc, lors d'une coopération de ce type, il paraît indispensable que les deux approches traitent des parties différentes du problème, tout en étant indépendantes l'une envers l'autre.

Dans [32], Chabrier *et al.* proposent une coopération entre recherche locale et algorithme de génération de colonnes pour le problème de tournées de véhicules. Les méthodes d'optimisation coopèrent en parallèle en se procurant mutuellement diverses informations, comme la montre le



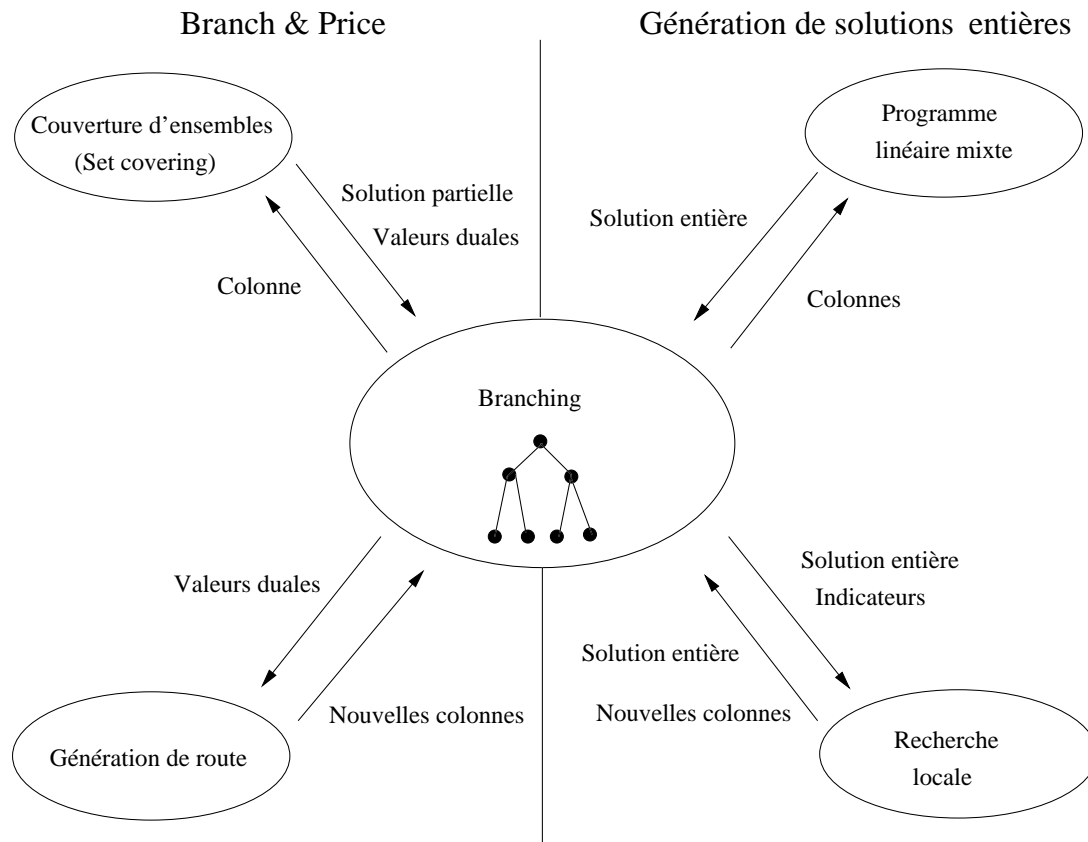


Fig. 5.4 – Exemple de coopération **HTH** pour le problème de tournées de véhicules [32].

schéma de la figure 5.4.

### 5.1.2 Classification à plat

Les critères discriminants retenus pour la classification à plat des coopérations entre métaheuristiques proposée par Talbi *et al.* [164] sont : l'homogénéité des méthodes hybridées, leurs domaines d'application et la nature (généraliste ou spécialiste) de leurs fonctions. Ici nous retiendrons les mêmes discriminants, tout en supprimant celui concernant l'homogénéité des méthodes d'optimisation, qui indiquait si les méthodes d'optimisation hybridées étaient les mêmes ou non. D'une part, pour les coopérations de type méta/méta, seules celles de type parallèle sont concernées. D'autre part, ce type de coopération n'est pas envisageable pour les coopérations de type méta/exacte. De plus, il est intéressant de savoir si l'approche coopérative dans sa globalité est une approche exacte ou heuristique. Nous avons donc ajouté un discriminant *résolution*, qui indique si l'approche générale de la méthode coopérative est exacte ou approchée.

#### Approche de résolution

Ce discriminant concerne essentiellement les coopérations de type méta/exacte. En effet, les coopérations méta/méta sont toutes des méthodes de résolution approchées.

Les approches coopératives exactes utilisent les heuristiques afin d'accélérer l'énumération des solutions en trouvant de bonnes bornes, en offrant des solutions initiales, en définissant des plans de coupes prometteurs...

Par exemple, dans l'approche proposée par Chabrier *et al.* [32], une recherche locale intervient dans un algorithme de *Branch & Cut* afin de générer des nouvelles colonnes permettant de définir des plans de coupes pertinents. Malgré l'utilisation d'une heuristique dans l'approche coopérative, l'approche globale reste exacte.

Dans [29], Burke *et al.* décrivent une méthode coopérative méta/exacte approchée. Dans cette étude, la méthode exacte est incluse dans un algorithme de recherche locale pour explorer de manière exacte les régions prometteuses de l'espace de recherche ; l'approche globale est donc heuristique.

### Domaine d'application

Le domaine d'application des heuristiques coopératives permet de différencier deux classes de coopération : les coopérations globales et les coopérations partielles.

On appelle coopération globale une coopération pour laquelle toutes les méthodes hybridées sont appliquées à la totalité de l'espace de recherche. L'objectif de ce type de coopération est d'explorer l'espace de recherche plus intensément. Dans le cadre de la coopération méta/exacte, Portmann *et al.* proposent une coopération où une heuristique calcule des solutions initiales dans le but d'offrir de bonnes bornes pour le lancement de la méthode exacte [144]. Les deux méthodes coopératives travaillent donc sur le même espace de recherche.

Pour la coopération partielle, le problème à résoudre est décomposé en sous-problèmes, chacun ayant un espace de recherche propre. Ainsi, chaque métaheuristique et/ou méthode exacte de la coopération résout un sous-problème dans son propre espace de recherche. Cependant, il est généralement difficile de découper un problème en sous-problèmes indépendants sans lever quelques contraintes. Ainsi, pour que la coopération partielle produise une solution réalisable, les méthodes hybridées doivent communiquer pour tenter de respecter les contraintes levées par la décomposition.

Une part importante des coopérations de type méta/exacte est partielle étant donné que l'espace exploré est trop large pour la méthode exacte seule. Mais des coopérations globales sont réalisées en vue d'accélérer une méthode exacte qui est déjà "praticable" sur les problèmes étudiés.

Pour le problème d'affectation quadratique, la méthode *Mimosa*, mise en oeuvre par Mautor et Michelon [124], construit à chaque itération un sous-problème au problème initial qu'il résout par *B&B* (coopération partielle). De même, Palpant *et al.* proposent une coopération partielle pour la résolution du problème d'ordonnement de projet avec contraintes de ressources (*Resource-Constrained Project Scheduling Problem*). Dans cette étude, un sous-problème est généré à chaque itération de recherche locale, puis résolu de manière exacte. Cette solution générée sert alors de point de départ pour la construction du reste de la solution.

T'kindt *et al.* proposent de résoudre un problème de Flow-shop bi-objectif à 2 machines par une approche méta/exacte [174]. Les deux objectifs sont le *makespan* ( $C_{max}$ ), et la somme des dates de complétions ( $\sum C_i$ ), à résoudre dans un ordre lexicographique ( $C_{max}$ , puis  $\sum C_i$ ). Le premier objectif est résoluble en temps polynomial par l'algorithme de Johnson [87]. Un algorithme de colonies de fourmis optimise le second objectif en ne considérant que les solutions optimales pour le *makespan*. Ici, l'espace exploré par la métaheuristique est plus restreint que celui exploré par la méthode exacte.

Une autre approche partielle est proposée par Maniezzo, qui propose de résoudre le problème d'affectation quadratique par un algorithme coopératif **LRH** également à base de colonies de fourmis [122]. Il propose d'utiliser une borne inférieure du problème afin de rendre plus efficace cet algorithme. La borne inférieure est calculée en résolvant de manière exacte le problème linéaire en nombre réels associé (calcul de bornes pour le *Branch & Cut & Price*). D'une manière générale, beaucoup d'approches transforment le problème initial afin de pouvoir appliquer une méthode exacte et de fournir des informations à la métaheuristique. On considère dans ce cas que ces approches sont de type partiel.

Les coopérations méta/exacte globales sont généralement des approches exactes. Le cas le plus couramment rencontré dans la littérature est celui où la métaheuristique permet de fournir des bornes et/ou des solutions initiales pour débiter un algorithme de *B&B*, par exemple [144].

Nous n'avons pas trouvé de référence de coopération proposant une approche exacte de type partielle. Cela paraît difficilement imaginable puisque, pour résoudre de manière exacte un problème, la méthode exacte doit opérer dans tout l'espace de recherche. Il faut donc imaginer un algorithme où la méthode approchée travaille sur un espace restreint de l'espace de recherche, cas peu courant, surtout si l'approche de résolution doit être exacte.

### Uniformité du problème traité

On différencie également deux types de coopérations selon que les méthodes d'optimisation traitent le même problème d'optimisation ou non. Pour les coopérations généralistes, toutes les heuristiques (et/ou méthodes exactes) hybridées traitent le même problème d'optimisation. À l'inverse des coopérations généralistes, les coopérations spécialistes combinent des méthodes qui s'attaquent à des problèmes différents.

La plupart des approches coopératives trouvées dans la littérature sont généralistes. Cependant on peut trouver quelques approches spécialistes.

Pour les coopérations méta/exactes, nous avons vu dans la partie précédente, que l'algorithme de T'kindt *et al.* [174] proposait une coopération de type partielle, la métaheuristique n'étant appliquée que sur les solutions optimales pour le premier objectif du problème, trouvées par la méthode exacte. Dans cette étude, la méthode exacte optimise un objectif ( $C_{max}$ ), tandis que l'algorithme de colonies de fourmis optimise un second objectif (la somme  $C_i$ ). Les problèmes traités par ces deux méthodes d'optimisation sont donc différents et l'approche est de type spécialiste.

### 5.1.3 Implémentation

Dans la plupart des cas, les travaux proposant des métaheuristiques hybrides sont des programmes séquentiels. Cependant, les algorithmes hybrides parallèles deviennent très intéressants lorsque les problèmes traités sont de grandes tailles. Ainsi, il existe quand même un certain nombre d'études portant sur des algorithmes hybrides parallèles (voir la taxinomie de Talbi [164]). Mais les algorithmes de coopérations méta/exacte parallèles sont peu nombreux. En effet, les grandes différences entre les deux types d'algorithmes rendent difficile la mise en oeuvre d'une architecture parallèle. Nous avons trouvé deux approches de coopérations méta/exactes parallèles, proposées par Cotta et Troya [38] et par Kleipeis *et al.* [98].

Nous avons donc défini plusieurs critères permettant de classer les méthodes coopératives, classification qui fait l'objet des deux sections suivantes.

## 5.2 Etat de l'art des coopérations Méta/Exacte

Les études de coopérations méta/exacte étant moins courantes, leur classification restait jusqu'à récemment assez inutile. Mais leur nombre grandissant la rend aujourd'hui plus utile afin de voir les différents types d'approches proposées. Un état de l'art de ces coopérations a d'ailleurs été proposé récemment par Stützle et Dumitrescu [49]. Ils distinguent cinq types d'approches des coopérations méta/exacte et détaillent une application trouvée dans la littérature pour chaque type d'approche. La classification proposée n'est pas de type 'taxinomie', mais sépare les approches selon la méthode utilisée pour la coopération. Ces classes sont :

- Utilisation d'un algorithme exact pour explorer des larges voisinages dans un algorithme de recherche locale.
- Utilisation des solutions de bonne qualité afin de réduire l'espace de recherche de la méthode exacte.
- Exploitation des bornes de la méthode exacte pour une heuristique constructive.
- Utilisation des informations fournies par les relaxations des problèmes linéaires pour orienter un algorithme de recherche locale ou constructif.
- Utilisation d'une méthode exacte pour une fonction spécifique de la métaheuristique.

L'état de l'art de Stützle et Dumitrescu survole les grandes 'branches' de la coopération méta/exacte, mais les exemples proposés utilisent, dans la plupart des cas, des recherches locale ou Tabou pour métaheuristique.

Notre taxinomie propose une approche plus générale de la classification, qui permet de regrouper les coopérations méta/méta et méta/exacte. De plus, la classification proposée dans cette section permet d'inclure facilement des nouveaux schémas de coopération.

Après avoir défini la grammaire de notre classification, ainsi que certaines notations, nous proposerons un tableau récapitulatif des coopérations méta/exacte trouvées dans la littérature.

### 5.2.1 Grammaire et vocabulaire de la classification

Nous avons repris la taxinomie du chapitre précédent issue de [164], puis nous l'avons modifiée pour permettre de prendre en compte les approches méta/exacte. En particulier, la grammaire détaillée ici ne distingue pas les modèles homogènes des modèles hétérogènes (inutile pour les coopérations méta/exacte - et cette information peut se retrouver dans la grammaire en regardant les deux métaheurstiques hybridées) ; par contre l'approche de résolution (approchée ou exacte), y est notamment ajoutée.

Pour les différentes métaheurstiques qui sont rencontrées dans cette taxinomie, nous utilisons les abréviations suivantes :

- RS : Recuit simulé (*Simulated Annealing*)
- AG : Algorithme génétique (*Genetic Algorithm*)
- AM : Algorithme mimétique (*Memetic Algorithm*)
- SE : Stratégie évolutionnaire (*Evolution Strategy*)
- PG : Programmation génétique (*Genetic Programming*)
- RN : Réseau de neurones (*Neural Network*)
- AD : Algorithme de descente (*Descent Walk*)
- RL : Recherche Locale (*Local Search*)
- RLI : Recherche Locale Itérée (*Iterated Local Search*)
- RT : Recherche Tabou (*Tabu Search*)

```

< algorithme coopératif > → < conception > < implémentation >
< conception > → < hiérarchique > < plat >
< hiérarchique > → < LRH > | < LTH > | < HRH > | < HTH >
< LRH > → LRH (< métaheuristique > (< exact >) | (< exact > (< métaheuristique >)))
< LTH > → LTH (< métaheuristique > (< exact >) | (< exact > (< métaheuristique >)))
< HRH > → HRH ((< métaheuristique > + < exact >) | (< exact > + < métaheuristique >))
< HTH > → HTH ((< métaheuristique > + < exact >) | (< exact > + < métaheuristique >))
< plat > → (< résolution >, < optimisation >, < fonction >)
< résolution > → exacte | approchée
< optimisation > → globale | partielle
< fonction > → générale | spécialiste
< implémentation > → séquentielle | parallèle < type parallèle >
< type parallèle > → statique | dynamique | adaptative
< métaheuristique > → RS | AG | AM | SE | PG | RN | AD | RL | RLI | RT | HG | CF | RD | HS |
CSA | < algorithme coopératif >
< exact > → B&B | αB&B | B&C | B&P | B&C&P | PL | PLM | PD | AS | RB | MS |
< algorithme coopératif >

```

Fig. 5.5 – Grammaire pour la coopération de méthodes d’optimisation (exactes ou approchées).

- HG : Heuristique gloutonne (*Greedy Heuristic*)
- CF : Colonies de fourmis (*Ant Colonies*)
- RD : Recherche par dispersion (*Scatter Search*)
- HS : Heuristique Spécifique
- CSA : *Conformational Space Annealing*

Pour les méthodes exactes, nous utiliserons les abréviations suivantes :

- B&B : *Branch & Bound*
- αB&B : α *Branch & Bound*
- B&C : *Branch & Cut*
- B&P : *Branch & Price*
- B&C&P : *Branch & Cut & Price*
- PL : Programmation Linéaire (*Linear Programming*) (exemple : solveur CPLEX, ...)
- PLM : Programmation Linéaire Mixte (*Linear Programming*) (exemple : solveur CPLEX, ...)
- PD : Programmation Dynamique (*Dynamic Programming*)
- PC : Programmation par Contraintes (*Constraint Programming*)
- AS : Algorithme du Simplexe (*Simplex Search*)
- NMS : *Nelder-Mead Simplex*
- RB : Recherche par But (*Search Goal*)
- MS : Méthode spécifique au problème

### 5.2.2 Classification

Voici un tableau récapitulatif des coopérations méta/exacte de la littérature. Notons que les coopérations peuvent être multiples.

TABLE 5.1 – Bibliographie annotée des coopérations méta/exacte.

Référence	Design	Problème d'optimisation
[4]	LTH(PD(RLI)) (approchée, partielle, générale) séquentielle	Voyageur de Commerce
[5]	HRH(B&P + HS) (approchée, partielle, générale) séquentielle	Partitionnement large échelle
[29]	LRH(RL(PD)) (approchée, partielle, générale) séquentielle	Voyageur de commerce asymétrique
[16]	HRH(RT + PL) (approchée, partielle, générale) séquentielle	Coupes irrégulières
[27]	HRH(HG + LTH(RT(B&B))) (approchée, partielle, générale) séquentielle	Design réseaux de vols directs
[8]	HRH(HRH(HG + RL) + B&C) (exacte, globale, générale) séquentielle	Routage de véhicules avec fenêtres de temps
[17]	HRH(RS + LTH(RL(B&B))) (approchée, partielle, générale) séquentielle	Routage de véhicules avec fenêtres de temps
[37]	LTH(AG(B&B)) (approchée, partielle, générale) parallèle statique	Voyageur de commerce
[35]	HRH(RL + PL) (approchée, partielle, générale) séquentielle	Problème de fabrication de câbles à fibre optique (ordonnancement)
[32]	HTH(B&P+HRH(PLM+RL)) (exacte, globale, spécialiste) séquentielle	Routage de véhicules
[36]	LTH(RL(PD)) (approchée, partielle, générale) séquentielle	Flow-shop à une machine (somme pondérée des retards)
[33]	HRH(RT + RD) (approchée, partielle, générale) séquentielle	Fonctions continues à multi-minima
[34]	HRH(RT + NMS) (approchée, partielle, générale) séquentielle	Fonctions continues à multi-minima
[38]	LTH(AG(B&B)) (approchée, partielle, générale)	Emploi du temps
<i>suite à la page suivante</i> →		

TABLE 5.1 : Bibliographie annotée des coopérations méta/exacte (suite).

Référence	Design	Problème d'optimisation
	parallèle statique	
[46]	HRH(AG + NMS) (approchée, partielle, générale) séquentielle	Fonctions continues
[40]	LTH(RT(B&B)) (approchée, partielle, générale) séquentielle	Tournées de véhicules
[48]	HTH(CF+HTH(B&B + HS)) (approchée, partielle, générale) séquentielle	Design d'accès au réseau local
[54]	HRH(PL + AG) (approchée, partielle, générale) séquentielle	Affectation généralisée
[58]	LRH(B&B + AG) (exacte, globale, générale) séquentielle	Max-SAT
[76]	HRH(HS + B&B) (approchée, partielle, générale) séquentielle	Flow-shop
[79]	LRH(MST(LTH(AG(B&B ACM)))) (approchée, globale, générale) séquentielle	Voyageur de commerce
[91]	HTH(AG + B&C) (approchée, partielle, générale) séquentielle	Tournée de véhicules multi-objectif
[103]	LTH(PLM(PG)) (exacte, globale, spécialiste) séquentielle	Jeux de données MIPLIB3
[97]	HRH(PL + AM) (approchée, partielle, générale) séquentielle	Problème de l'arbre de Steiner
[98]	HRH( $\alpha$ B&B + CSA) (approchée, partielle, spécialiste) parallèle statique	Prédiction de structure de protéines
[122]	LRH(CF(PL)) (approchée, globale, spécialiste) séquentielle	Affectation quadratique
[124]	HRH(B&B + RT) (approchée, partielle, générale) séquentielle	Affectation quadratique
[137]	HTH(RL+PC) (approchée, partielle, générale)	Ordonnancement de projets avec contrainte
<i>suite à la page suivante</i> →		

TABLE 5.1 : Bibliographie annotée des coopérations méta/exacte (suite).

Référence	Design	Problème d'optimisation
	séquentielle	de ressources
[143]	HTH(RL+PC) (approchée, partielle, générale) séquentielle	Problème du voyageur de commerce
[141]	HRH(AD + B&B) (exacte, globale, générale) séquentielle	Affectation quadratique
[144]	HRH(HS + (HTH(B&B+AG))) (exacte, globale, générale) séquentielle	Flow-shop hybride
[155]	HRH(HS+B&B) (approchée, partielle, générale) séquentielle	Problème <i>p-median</i>
[160]	LTH(RL(B&B)) (approchée, partielle, générale) séquentielle	Routage de véhicules
[161]	HRH(B&B + RN) (approchée, partielle, générale) séquentielle	Ordonnancement de diffusion d'informations bi-objectif
[174]	HRH(MS + CF) (approchée, partielle, spécialiste) séquentielle	Flow-shop biobjectif sur 2 machines
[176]	LRH(RLI(AS)) (approchée, globale, spécialiste) séquentielle	Variante du problème de découpe à une dimension
[178]	HRH(AS + RT) (approchée, partielle, générale) séquentielle	0-1 Sac à dos multi-dimensionnel
[12]	HRH(AG+B&B) (exacte, partielle, général) parallèle	Flowshop bi-objectif

### 5.3 Conclusion

Dans ce chapitre, nous avons présenté l'extension de la taxonomie présentée dans le chapitre précédent aux coopérations entre métaheuristiques et méthodes exactes. L'objectif est ici de montrer le potentiel de ce type de coopération par des exemples de la littérature.

Dans le cadre de mes travaux, j'ai commencé à m'intéresser à ce type de coopération pour le problème du ringstar multiobjectif lors du Master recherche de Matthieu Quatreuvre. La mise au point de méthodes exactes multiobjectif est déjà en soit un travail de recherche assez complexe et nous n'avons malheureusement pas mis un œuvre de méthode collaborative intéressante.

Les perspectives de ce chapitre sont donc de mettre en œuvre ce type de coopération sur les



problématiques multiobjectif que nous rencontrons dans le cadre de nos projets, notamment le docking et les problèmes de transport.

## Chap. 6

# Coopération métaheuristiques-extraction de connaissances

### Sommaire

---

<b>6.1</b>	<b>Brève introduction à l'extraction de connaissances . . . . .</b>	<b>67</b>
6.1.1	Classification supervisée . . . . .	67
6.1.2	Catégorisation ou Clustering . . . . .	68
6.1.3	Recherche de règles d'association . . . . .	68
6.1.4	Sélection d'attributs . . . . .	68
<b>6.2</b>	<b>Synthèse sur les coopérations entre métaheuristiques et extraction de connaissances . . . . .</b>	<b>68</b>
6.2.1	Hybridation de bas niveau relais . . . . .	69
6.2.2	Hybridation de bas niveau co-évolutionnaire . . . . .	70
6.2.3	Hybridation de haut niveau relais . . . . .	70
6.2.4	Hybridation de haut niveau co-évolutionnaire . . . . .	71
<b>6.3</b>	<b>LEMMO . . . . .</b>	<b>72</b>
6.3.1	LEM . . . . .	72
6.3.2	Adaptation de LEM au cas multiobjectif . . . . .	72
6.3.3	Différents modèles pour LEMMO . . . . .	73
6.3.4	Cadre d'étude de LEMMO : design de systèmes hydrauliques . . . . .	74
<b>6.4</b>	<b>Conclusion et Perspectives . . . . .</b>	<b>78</b>

---

Ce chapitre a donné lieu notamment aux publications suivantes :

- E. Alba, J. Garcia-Nieto, L. Jourdan, E-G. Talbi. *Sensitivity and Specifity Based Multiobjective Approach for Feature Selection : Application to Cancer Diagnosis*. Information Processing Letters, Volume 109, Issue 16, 31 July 2009, Pages 887-896.
- E.-G. Talbi, L. Jourdan, J. Garcia-Nieto and E. Alba. *Comparison of population based meta-heuristics for feature selection : Application to microarray data classification*. *IEEE/ACS International Conference on Computer Systems and Applications, 2008. AICCSA 2008.*, Pages 45 - 52. IEEE, 2008.

- E. Alba, J. García-Nieto, L. Jourdan and E.-G. Talbi. *Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms*. *IEEE Congress on Evolutionary Computation CEC-07*, Pages 284–290. IEEE, Singapore, 2007 Sep.
- L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Using datamining techniques to help metaheuristics : A short survey*. *Hybrid Metaheuristics (HM'06)*, LNCS 4030, 57–69. Springer Berlin / Heidelberg, 2006.
- L. Jourdan, D. Corne, S. Savic, and G. Walters. *Preliminary investigation of the 'learnable evolution model' for faster/better multiobjective water systems design*. *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, LNCS 3410, 841–855. 2005.
- L. Jourdan, D. Corne, S. Savic, and G. Walters. *Hybridising rule induction and multi-objective evolutionary search for optimising water distribution systems*. I. C. Press, *Fourth International Conference on Hybrid Intelligent Systems, IEEE HIS 2004*, 434–439. 2004. ISBN 0-7695-2291-2.

Dans ce chapitre, après une petite introduction aux méthodes d'extraction de connaissances, nous rappelons les principales idées de la synthèse sur l'hybridation entre les métaheuristiques et l'extraction de connaissances que j'ai publiée dans [I18]. Nous montrerons que la classification présentée dans le chapitre 4 s'adapte à ce type de coopération. Nos travaux hybridant les métaheuristiques et l'extraction de connaissances sont évoqués dans cette synthèse.

Nous exposons ensuite un exemple de coopération que nous avons réalisé entre des métaheuristiques et des algorithmes d'extraction de connaissances appelé LEMMO et qui est une extension des travaux de Michalski et al. [129, 130] dans le cadre multiobjectif. Nous montrons les résultats de LEMMO pour le design de systèmes hydrauliques. Nous concluons sur les perspectives de ce travail.

## 6.1 Brève introduction à l'extraction de connaissances

L'extraction de connaissances à partir de données (ECD) ou data mining (DM) est un processus automatique d'exploration de gros volume de données pour en extraire de la connaissance intéressante souvent présentée sous forme de modèle (pattern). Les techniques utilisées afin de réaliser l'extraction de connaissances sont issues majoritairement des statistiques et du "machine learning".

Différentes tâches peuvent être réalisées en fonction de l'objectif du processus, ces tâches peuvent être classifiées en 3 grands ensembles : les tâches non-supervisées, les tâches supervisées et les tâches de réduction de données.

Dans le cadre de notre synthèse, nous focaliserons les descriptions de quatre catégories de méthodes : la classification supervisée, la catégorisation (clustering), la recherche de règle d'associations et la sélection d'attributs. Une vue synthétique des tâches et des algorithmes associés est proposée figure 6.1.

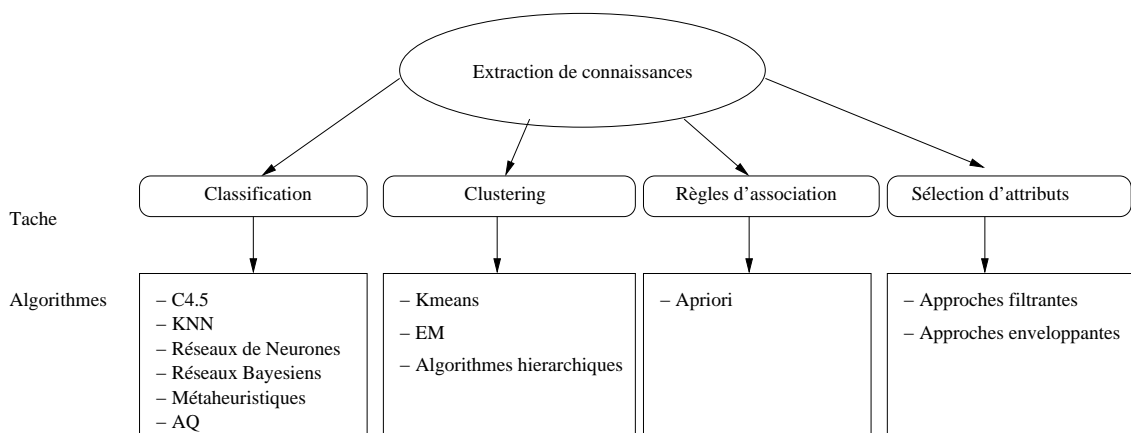


Fig. 6.1 – Vue synthétique des tâches et des algorithmes associés en data mining.

### 6.1.1 Classification supervisée

Dans la classification supervisée, des connaissances sur la description des exemples et leur classe (groupe d'appartenance) sont disponibles. Il s'agit ici de trouver des modèles qui, pour de nou-

veaux exemples, nous donnent leur groupe d'appartenance. Les algorithmes classiques pour réaliser cette tâche sont C4.5 [147], Id3, AQ, KNN ...

Une application typique est l'aide au diagnostic médical. Les algorithmes de classification donne par exemple des arbres de décision permettant de classer des nouveaux patients afin de pouvoir cibler une maladie.

### 6.1.2 Catégorisation ou Clustering

La catégorisation, partitionnement ou clustering, consiste à trouver des groupes homogènes au sein d'une population. Le partitionnement est une tâche d'apprentissage "non supervisée" car on ne dispose d'aucune autre information préalable que la description des exemples. Le clustering a été utilisé dans de nombreux contextes et par de nombreux chercheurs dans différentes disciplines en tant que processus d'analyse exploratoire de données. Les principaux algorithmes rencontrés sont K-means [120], les méthodes hiérarchiques (single link, complete link ...)

### 6.1.3 Recherche de règles d'association

L'étude des règles d'association a été initialement utilisée dans le domaine de la grande distribution pour analyser le panier de la ménagère. Les algorithmes de recherche de règles d'association fournissent un ensemble de règles de type "SI ... ALORS" ("IF ... THEN ..."). Elles permettent ainsi de découvrir des informations utiles, cachées dans des grandes bases de données. L'algorithme pour extraire des règles d'association le plus utilisé est l'algorithme Apriori.

### 6.1.4 Sélection d'attributs

Le processus de sélection d'attributs permet de réduire la taille des données en sélectionnant un sous-ensemble de données permettant d'obtenir des modèles d'apprentissage robustes ou de diminuer le temps de traitement. Classiquement, la sélection d'attributs est définie comme le fait de sélectionner un sous-ensemble de  $M$  attributs à partir d'un ensemble  $N$ , tel que  $M$  soit beaucoup plus petit que  $N$  et que la fonction critère choisie soit optimale sur le sous-ensemble de taille  $M$  choisi. Les algorithmes de sélection d'attributs peuvent être classés en deux catégories selon que la sélection d'attributs est faite indépendamment ou non de l'algorithme d'apprentissage supervisé sur lequel est basé le classifieur : on parlera dans le premier cas de méthode filtrante et dans le second cas de méthode enveloppante.

Après cette brève description des différentes tâches d'extraction de connaissances, nous allons montrer dans la section suivante comment et pourquoi une hybridation peut être réalisée entre cette classe de méthode et les métaheuristiques.

## 6.2 Synthèse sur les coopérations entre métaheuristiques et extraction de connaissances

La synergie entre le data mining (extraction de connaissances) et la recherche opérationnelle est de plus en plus mise en avant par les deux communautés [126, 125, 135]. La figure 6.2 nous montre l'interaction possible entre l'extraction de connaissances et la recherche opérationnelle.

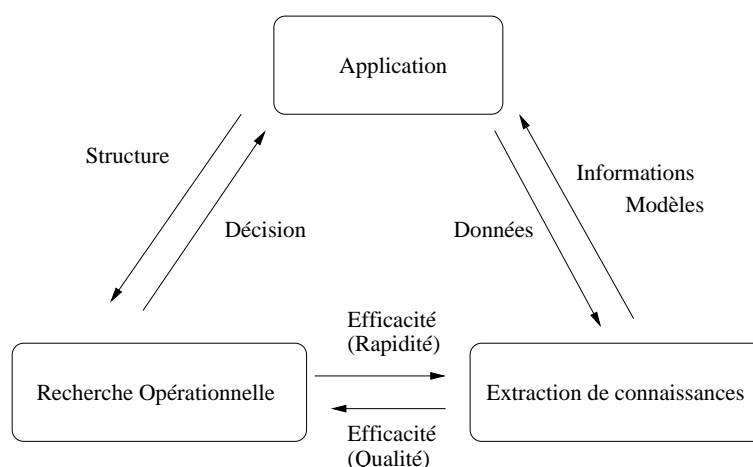


Fig. 6.2 – Interaction entre l'extraction de connaissances et recherche opérationnelle.

Nous restreindrons ici notre étude à l'imbrication entre les métaheuristiques et l'extraction de connaissances, et, plus particulièrement à leur hybridation. Nous ne nous intéresserons pas à la modélisation de problèmes d'extraction de connaissances en problèmes d'optimisation et leurs résolutions ne faisant pas intervenir d'hybridations entre les deux types de méthodes.

L'intégration de l'extraction de connaissances dans les métaheuristiques peut globalement être classifiée selon trois critères :

- **selon quand a lieu l'extraction de la connaissance** : l'extraction peut se faire avant la recherche et on parlera alors de stratégie "off-line"/"a priori" ou alors celui-ci peut se faire dynamiquement au cours de la recherche et on parlera dans ce cas de stratégie "online".
- **selon l'objectif de l'hybridation** : l'hybridation peut avoir comme objectif d'améliorer l'efficacité de la métaheuristique en réduisant son temps de calcul ou alors d'améliorer la qualité de la recherche en conduisant la métaheuristique vers des zones intéressantes de l'espace de recherche. Certaines hybridations réalisent en même temps une amélioration de l'efficacité et de la qualité.
- **selon le composant impliqué** : l'hybridation peut là aussi se faire dans tous les composants de la métaheuristique (son encodage, son initialisation, ses opérateurs etc.). Elle peut également intervenir pour fixer les paramètres de la métaheuristique ou encore de définir le problème à résoudre (comme par exemple en étant une fonction objectif).

Dans la suite de cette synthèse, nous reprendrons la taxonomie qui nous sert de base dans ce document issue de [164] et nous montrerons pour chaque classe la sémantique pour ce type particulier d'hybridation.

### 6.2.1 Hybridation de bas niveau relais

Dans ce premier cas, l'hybridation a lieu *a priori* et concerne les métaheuristiques à solution unique. Nous pouvons distinguer différents cas :

- utilisation de la connaissance extraite pour transformer le problème d'optimisation par exemple pour ajouter des contraintes
- aide au paramétrage des métaheuristiques à solution unique, on parle alors de "learning metaheuristics", un bon exemple est l'application de F-RACE à l'ILS [21].

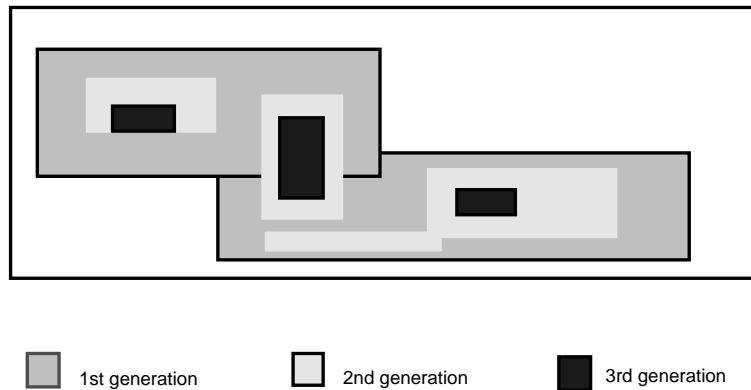


Fig. 6.3 – LEM : exemple de la réduction de l'espace de recherche.

### 6.2.2 Hybridation de bas niveau co-évolutionnaire

Cette classe d'hybridation est fréquemment rencontrée dans la littérature et constitue les fondements de base de certaines métaheuristiques comme les EDAs [105], les algorithmes "culturels", les algorithmes "civilisés" [159]. La connaissance acquise durant la recherche des métaheuristiques est utilisée comme base des méthodes de data mining. Différents cas d'hybridation permettent notamment :

- d'aider au paramétrage des métaheuristiques à population, F-RACE a été appliqué également à des métaheuristiques à population [21],
- de remplacer un opérateur des métaheuristiques à population par une méthode d'extraction de connaissances. Ainsi, pour les problèmes de catégorisation ou "clustering", l'algorithme de K-moyenne est fréquemment utilisé comme recherche locale notamment lors de la phase d'initialisation [53, 73, 179] et de mutation. Cela peut également permettre de trouver des schémas constitutifs des bonnes solutions comme dans les travaux de Handa [71, 72] pour la représentation binaire ou LEM [129, 130] dans un cadre plus général.
- On peut notamment apprendre *a priori* la fonction objectif grâce à un réseau de neurones [63], [151, 149, 149, 150, 152] ou élire des représentants de solutions à évaluer (utilisation d'algorithmes de clustering) [185, 86, 96]
- Un cas fréquent est le remplacement de la fonction objectif par le résultat d'un algorithme d'extraction de connaissances comme dans nos travaux sur la sélection d'attributs [R5].

### 6.2.3 Hybridation de haut niveau relais

Dans ce cas, on utilisera à partir de connaissances acquises *a priori*. Différents cas sont possibles :

- utilisation d'algorithmes de clustering pour découper les clients d'un problème de VRP et ainsi créer des sous-problèmes [9]
- utilisation d'une approximation de la fonction objective qui peut être coûteuse notamment dans les problèmes de design [83]. Cette approximation est faite *a priori* avant de lancer la métaheuristique.

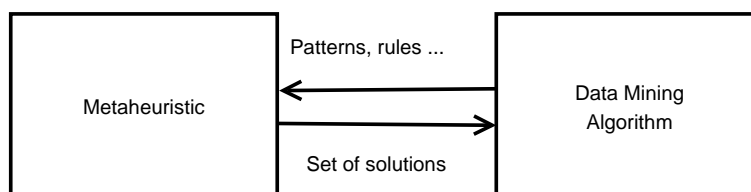


Fig. 6.4 – Algorithme hybride de haut niveau relais.

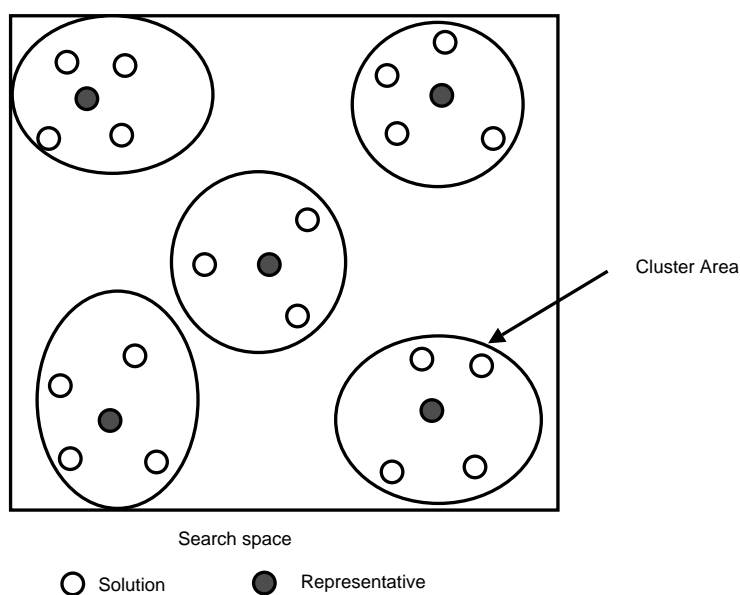


Fig. 6.5 – Exemple d'utilisation de représentants pour la phase d'évaluation en utilisant un algorithme de clustering.

#### 6.2.4 Hybridation de haut niveau co-évolutionnaire

Dans ce type d'approche, la connaissance est acquise durant le processus d'évolution de la métaheuristique. Différentes approches peuvent être trouvées dans la littérature mais le plus fréquent est l'intervention de méthodes d'extraction de connaissances dans le management de la population. Par exemple, dans CIGAR [119], un système de raisonnement à base de cas fourni périodiquement à l'algorithme génétique des solutions qui proviennent d'autres instances et qui pourraient apporter à l'évolution de l'algorithme.

Dans cette synthèse, nous avons vu que nous pouvons classer les hybridations entre les métaheuristiques et les méthodes d'extractions de connaissances selon la sémantique de la classification de [164]. Une partie des travaux que j'ai réalisé depuis mon doctorat sont basés sur ce type de coopération. Je détaille dans la section suivante un modèle innovant d'utilisation de l'extraction de connaissances *on-line* pour améliorer la recherche d'une métaheuristique. Nous nous situons dans ce cas dans un cadre d'hybridation de bas niveau co-évolutionnaire.



### 6.3 LEMMO : Un exemple d'hybridation entre un algorithme génétique multiobjectif et un algorithme de recherche de règles d'association

Dans cette section nous présentons LEMMO qui est basé sur les principes de LEM. Ce travail a été effectué avec D. Corne, D. Savic et G. Walters. Nous ne présentons ici qu'une synthèse des méthodes développées et des résultats obtenus, pour plus de détails le lecteur se référera aux articles [120, I19].

#### 6.3.1 LEM

LEM (Learnable Evolution Model) [129, 130] est une approche intégrant un composant d'apprentissage symbolique dans un algorithme évolutionnaire ; le composant recherche des règles expliquant les différences entre les bons et les mauvais individus de la population et génère des nouveaux individus en se basant sur les schémas obtenus à partir de ces règles. La méthodologie inhérente à LEM a prouvé son efficacité sur de nombreux problèmes. Son principe est relativement simple et consiste en des phases d'évolution entrecoupées par des phases d'apprentissage. Durant la phase d'évolution, l'algorithme évolutionnaire fonctionne normalement et les individus générés sont stockés (tous ou seulement un sous ensemble) ainsi que leur qualité. Un ensemble de règles est ensuite généré à partir de cet ensemble pour permettre de différencier les bonnes solutions des moins bonnes. La génération de nouveaux individus prendra alors en compte ces règles en fixant certains gènes.

#### 6.3.2 Adaptation de LEM au cas multiobjectif

LEMMO (Learnable Evolution Model for Multiobjective Optimization) comme LEM se veut indépendant de l'algorithme évolutionnaire utilisé. La seule contrainte est que l'algorithme multiobjectif utilisé maintienne une archive Pareto à chaque génération. Dans LEMMO, nous allons rechercher les règles pour différencier les bonnes solutions des mauvaises solutions d'un point de vue multiobjectif et les utiliser pour aider à la convergence de l'algorithme en terme de qualité de la recherche et de rapidité. Alors que dans LEM, la différenciation entre bonnes et mauvaises solutions est associée à la qualité de la solution et utilise un seuil pour faire la différence, dans le cas multiobjectif, cette différenciation est moins binaire car une solution n'est pas uniquement bonne ou mauvaise. Cela nous a conduit à explorer plusieurs possibilités afin de déterminer la meilleure méthode et voir si l'amélioration entre la version avec et sans hybridation se faisait en qualité mais aussi en rapidité.

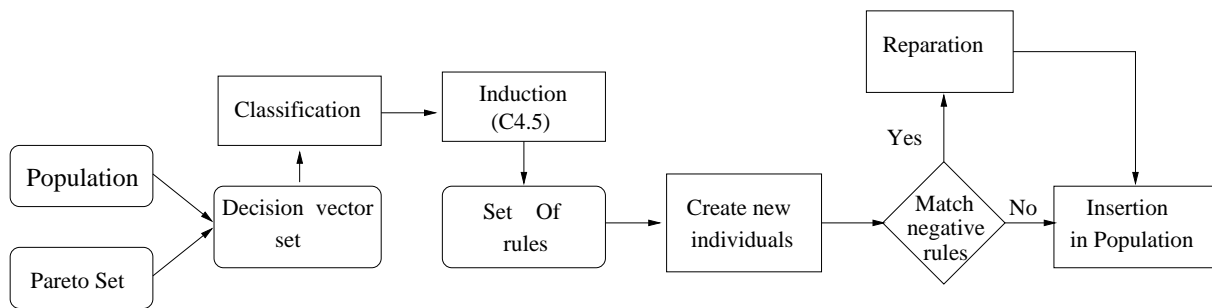
LEM est basé sur l'algorithme AQ [131] dans sa version la plus basique alors qu'il existe différentes versions plus robustes par rapport aux données d'apprentissage, notamment AQ15 [132], mais malheureusement leur code n'est pas public. Dans notre cas, nous avons utilisé C4.5 [147] pour la phase d'induction qui se montre plus robuste que AQ par rapport aux choix des ensembles d'apprentissage et qui permet de générer facilement des règles à partir de l'arbre de décision obtenu. Un exemple de règles obtenues par C4.5 est montré figure 6.6 pour l'application traité par la suite qui fixe le diamètre des tubes d'un réseau hydraulique.

A partir d'un ensemble de règles généré par C4.5, LEMMO va comme LEM générer des nouveaux

```

R1: A3 <= 0 AND A7 <= 6 AND A18 <= 4
-> class bad
R2: A19 <= 4 AND A21 <= 1
-> class bad
R3: A7 > 6 AND A10<=5
-> class good
R4: A18 > 4
-> class good
    
```

**Fig. 6.6** – Exemple des règles produites par C4.5 avec des attributs continus. Ai représente le ième attribut ce qui correspond à la ième composante du codage choisi.



**Fig. 6.7** – Création de nouveaux individus grâce aux règles extraites.

individus qui vont prendre en compte les schémas extraits pouvant donner des solutions de bonne qualité. Ainsi un individu avec \*\*\*\*\*9\*\*3\*\*\*\*\* va satisfaire à la règle R3. Nous voyons bien que seule une petite partie du génotype va être fixée grâce à cette méthode, le reste étant fixé avec les mêmes opérateurs que dans un algorithme génétique multiobjectif classique (dans notre cas un crossover un-point). Dans LEMMO nous fixons des parties des génotypes en fonction des règles extraites sur les "bons" individus mais nous évitons également d'avoir des individus qui vérifient des règles qui qualifient les mauvais individus. La figure 6.7 modélise tout le processus de LEMMO.

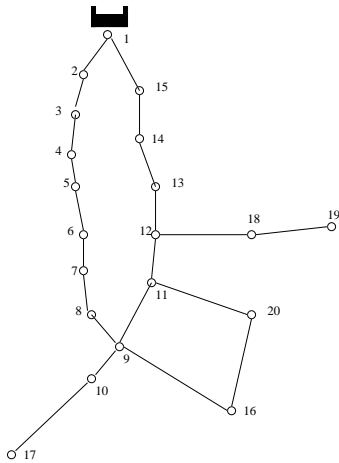
### 6.3.3 Différents modèles pour LEMMO

La question épineuse est : comment qualifier un bon individu et quand réaliser l'hybridation. Pour cela nous avons exploré différents modèles d'hybridation possibles.

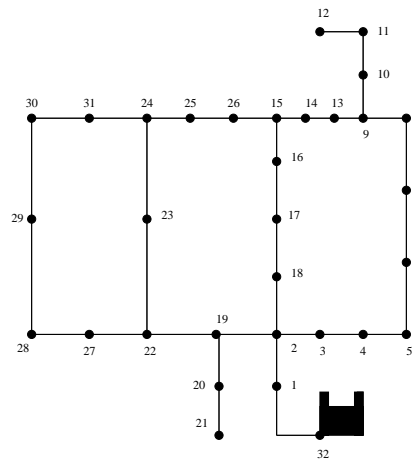
La classification entre les individus "bons" et "mauvais" est refaite avant chaque lancement de l'algorithme d'induction.

Les différents modèles explorés sont :

- LEMMO-1 : lance la phase d'apprentissage quand il n'y a plus de changement dans le front pareto entre deux générations.
- LEMMO-fix1 : lance la phase d'apprentissage toutes les 10 générations en indiquant comme



**Fig. 6.8** – Le problème de la ville de New York.



**Fig. 6.9** – Le problème de la ville de Hanoi.

"mauvais" les individus de la population initiale et "bons" les individus de l'ensemble Pareto final.

- LEMMO-fix2 : lance la phase d'apprentissage toutes les 10 générations en indiquant comme "bons" les 20 derniers individus rentrés dans le front Pareto et mauvais les individus non Pareto de la population.
- LEMMO-fix3 : lance la phase d'apprentissage toutes les 10 générations, en utilisant un ensemble d'individus du front Pareto choisi aléatoirement comme l'ensemble étiqueté "bons" et les reste de la population courante comme "mauvais".
- LEMMO-fix4 : lance la phase d'apprentissage toutes les 10 générations en utilisant 30% des meilleurs individus trouver au cours de l'évolution pour un objectif (l'objectif est choisi au hasard à chaque phase d'apprentissage) et "mauvais" les 30% moins bons par rapport à l'objectif choisi.

### 6.3.4 Cadre d'étude de LEMMO : design de systèmes hydrauliques

LEMMO a été essentiellement testé sur des instances de design de systèmes hydrauliques. Ce type de problème est relativement complexe et les solutions données par les algorithmes sont évaluées par des logiciels permettant de modéliser l'ensemble des systèmes hydrauliques considérés. Dans notre cas, nous avons utilisé EPANET [156] qui est un logiciel libre. Nos tests ont essentiellement porté sur deux instances : le système de la ville de New York que l'on notera NYT pour "New York Tunnels pipe network" et le problème de la ville de Hanoi [59]. Les deux modèles sont détaillés dans l'article [120].

Notre comparaison s'effectuera par rapport à NSGA II [43].

#### Opérateurs et Paramètres

Dans tous les modèles, le codage utilisé est le plus simple : un gène représente le choix du diamètre pour la canalisation correspondante. Les opérateurs utilisés sont le croisement un point et la mutation d'un gène de façon aléatoire.

TABLE 6.1 – Comparaison des différents modèles par rapport à la métrique  $S$  ( $S(\text{Schéma}) - S(\text{NSGAI})/S(\text{Schéma})$ ).

Schéma	S Médiane	S Moyenne	S Min	S Max	S écart type.
LEMMO-1	7.83%	4.9%	7.3%	0.26%	-33.32%
LEMMO-fix1	8.28%	5.44%	3.4%	0.26%	-57.6%
LEMMO-fix2	8.19%	5.5%	2.04%	0.18%	-25.9%
LEMMO-fix3	8.14%	6.75%	8.84%	<b>0.42%</b>	<b>-58.58%</b>
LEMMO-fix4	<b>9.12%</b>	<b>6.8%</b>	<b>7.77%</b>	0.27%	-22.19%

La taille de la population est fixée pour tous les modèles à 100, le taux de croisement à 0.9, le taux de mutation à 0.9 et un nombre maximum d'évaluations est fixé à 250 000. On notera que l'évaluation étant faite par un simulateur, une évaluation est coûteuse et l'objectif ici est de diminuer le nombre d'évaluations sans diminuer la qualité des solutions obtenues.

## Résultats

Dans un premier temps, notre objectif a été d'identifier l'impact des différents paramètres des modèles sur la qualité des résultats et nous avons réalisé cette étude sur le plus petit jeu de données "NYT". Ce jeu de données est plus petit que ce que l'on rencontre dans un cas réel mais il nous a permis de nous faire une première idée afin de sélectionner le meilleur modèle pour des jeux de données plus importants et notamment le problème Hanoi. Pour chacun des cinq modèles et la version basique de NSGA II, 30 expérimentations ont été réalisées. On peut remarquer que, pour ce jeu de données, le front Pareto contient beaucoup de solutions. Le tableau 6.1 présente les statistiques descriptives de la métrique  $S$  en indiquant en pourcentage l'amélioration du modèle par rapport à NSGA II. On peut remarquer que non seulement les modèles introduisant de l'induction donnent des résultats de meilleure qualité mais ils sont également plus robustes que la version standard de NSGA II. Les résultats sur la métrique  $R1_R$  (voir tableau 6.3) nous montrent le même type de résultats.

Nous avons également mesuré la convergence des algorithmes (tableau 6.2) et nous pouvons remarquer que les algorithmes hybrides convergent plus rapidement. La figure 6.10 montre l'évolution de l'hypervolume ( $S_{metric}$ ) au fur et à mesure des générations. Nous pouvons remarquer que les variantes de LEMMO sont non seulement plus rapides pour atteindre une valeur de  $S_{metric}$  définie mais obtiennent également souvent au final une valeur plus élevée. La variante *LEMMO-fix1* a été choisie comme illustration car dès le début de la convergence, elle obtient une meilleure valeur de la métrique  $S$  normalisée.

Le tableau 6.4 mesure la variation en nombre d'évaluations pour atteindre différentes valeurs de  $S_{metric}$  normalisée (valeurs fixées 0.955 et 0.968) pour 30 exécutions. Nous pouvons observer que tous les modèles développés sont plus rapides que NSGA-II et que le modèle LEMMO-fix4 est le moins coûteux et le plus rapide.

Il est à signaler que le modèle multiobjectif LEMMO-fix4 génère à chaque fois dans son front Pareto la meilleure solution monoobjective [134, 158] avec les mêmes caractéristiques que dans [158].

TABLE 6.2 – Statistiques descriptives sur le nombre d'évaluations nécessaires pour trouver le front Pareto sur 30 exécutions (T(Scheme)-T(NSGAI)/T(Scheme)).

Schéma	Médiane	Moyenne	Min	Max	Ecart type
LEMMO-1	+14.07%	+6.02%	+29.8%	-1.84%	+4.11%
LEMMO-fix1	<b>-16.28%</b>	<b>-11.81%</b>	-10.04%	-0.93%	+1.26%
LEMMO-fix2	-8.97%	-4.54%	+20.69%	-0.00%	+5.22%
LEMMO-fix3	-14.69%	-7.38	-14.41%	-0.36%	+8.53%
LEMMO-fix4	-16.14 %	-8.45 %	+25.40%	-0.44%	-21.74%

TABLE 6.3 – Comparaison par rapport à la métrique  $R1_R$  pour le problème de NY.

Modèle	$R1_R$ Médiane	$R1_R$ Moyenne	$R1_R$ min	$R1_R$ max	$R1_R$ écart type.
NSGA-II	0.39172	0.44780	0.37525	0.62271	0.09460
LEMMO-1	0.38323	0.43119	0.35229	0.64671	0.09236
LEMMO-fix1	0.38024	0.41153	0.35229	0.64271	0.09044
LEMMO-fix2	0.38025	0.39476	0.35229	0.51597	0.05348
LEMMO-fix3	0.38124	0.40108	0.35229	0.65469	0.07496
LEMMO-fix4	0.38224	0.39680	0.36527	0.51497	0.04350

TABLE 6.4 – Statistiques sur le nombre d'évaluations nécessaires pour trouver une  $S$  métrique normalisée de valeur supérieure à 0,955 et 0,968 sur 30 exécutions pour les différents modèles (T(modèle)-T(NSGA-II)/T(modèle)) sur le problème de NY.

Modèle	$S_{norm} > 0,955$		$S_{norm} > 0,968$	
	Médiane	Moyenne	Médiane	Moyenne
LEMMO-1	0%	+16,16%	-16%	-6,8%
LEMMO-fix1	+10%	+10%	-10,10%	-5,01%
LEMMO-fix2	+16,66%	+16,66%	-7,8%	+2,6%
LEMMO-fix3	+6,25%	+14,28%	-13,51%	-10,81%
LEMMO-fix4	+10 %	0%	<b>-16,66 %</b>	<b>-20,58%</b>

TABLE 6.5 – Impact des paramètres : la métrique  $R1_R$  par rapport au front de référence de NSGA-II pour le problème de NY.

Modèle (paramètre)	$R1_R$ Médiane	$R1_R$ Moyenne	$R1_R$ Dev.
LEMMO-fix1 (5)	0,38124	0,38503	0,02553
LEMMO-fix1 (10)	0,38024	0,41152	0,09044
LEMMO-fix1 (20)	0,38024	0,42664	0,12099
LEMMO-fix4 (5)	0,40118	0,47329	0,07234
LEMMO-fix4 (10)	0,38224	0,39681	0,0435
LEMMO-fix4 (20)	0,37924	0,39654	0,07272

TABLE 6.6 – Statistiques sur le nombre d'évaluations nécessaires pour trouver une valeur normalisée de la  $Smetric$  plus grande que 0,955 et 0,968 sur 30 runs ((T(Modèle)-T(NSGA-II))/T(Modèle)) pour le problème de NY.

Modèle (parameter)	$S_{norm} > 0,955$		$S_{norm} > 0,968$	
	Médiane	Moyenne	Médiane	Moyenne
LEMMO-fix1 (5)	+21,05%	+21,21%	+2,31	+4,65%
LEMMO-fix1 (10)	+10%	+10%	-10,10%	-5,01%
LEMMO-fix1 (20)	+16,66%	+16,66%	+13,51%	+2,38%
LEMMO-fix4 (5)	+6,25%	+16,66%	-14,28%	-7,89%
LEMMO-fix4 (10)	+10 %	0%	<b>-16,66 %</b>	<b>-20,58%</b>
LEMMO-fix4 (20)	+6,89%	+6,66%	-2,43%	+2,43%

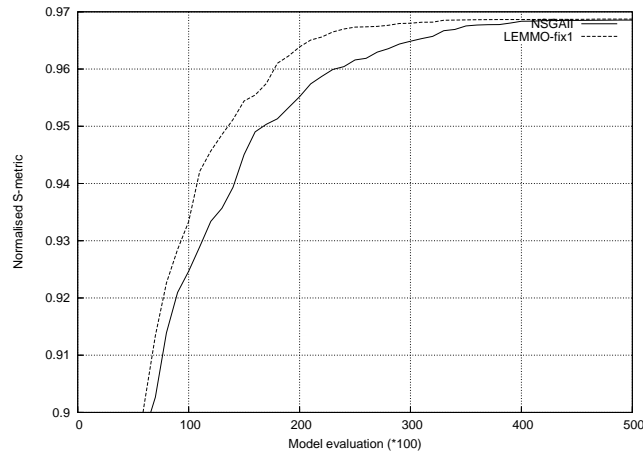
Les tableaux 6.5 et 6.6, nous montrent l'influence des différents paramètres pour chaque modèle. A partir de ces expériences, il a été décidé de fixer le paramètre à 10.

Nous avons remarqué que LEMMO-fix4 semblait être le meilleur modèle et nous l'avons testé sur le problème de Hanoi [59, 57]. Le tableau 6.7 compare NSGA-II et le modèle LEMMO-fix4 par rapport à la métrique  $R1_R$  et nous pouvons de nouveau observer la supériorité de la méthode (cette différence de comportement a été validée par des tests statistiques [51]).

Nous avons fixé différentes valeurs de la métrique  $S$  et nous avons comparé la vitesse pour les atteindre entre LEMMO-fix4 et NSGA-II. Nous avons remarqué que pour obtenir une valeur de 0.711, LEMMO-fix4 requiert en moyenne 27,5% d'évaluations en moins que NSGA II. De plus, LEMMO-fix4 obtient 6 fois sur 10 une valeur de  $S$  métrique égale à 0.7488 alors que NSGA-II ne l'obtient jamais avant la limite de 250,000 évaluations.

TABLE 6.7 – Comparaison de la qualité des fronts par rapport à la métrique  $R1_R$  pour le problème de Hanoi.

Modèle	$R1_R$ Médiane	$R1_R$ Moyenne	$R1_R$ min	$R1_R$ max	$R1_R$ écart-type.
NSGA II	0.05439	0.06871	0.00199	0.02745	0.07335
LEMMO-fix4	0.01247	0.01854	0.00598	0.03992	0.01261



**Fig. 6.10** – Exemple de l'évolution de la métrique  $S$  normalisée par rapport au nombre d'évaluation pour le problème NYT pour la variante *LEMMO-fix1*.

## 6.4 Conclusion et Perspectives

Dans ce chapitre, nous avons tout d'abord montré l'intérêt de l'hybridation entre des métaheuristiques et des méthodes d'extraction de connaissances en réalisant un court état de l'art de ce type d'hybridation suivant la taxinomie servant de base à ce document. Dans un second temps, nous avons montré comment une hybridation de ce type pouvait être réalisée afin d'améliorer les performances d'algorithmes multiobjectif et l'application qui a été faite sur le design de réseaux hydrauliques.

Ces travaux sont une voie prometteuse de recherche et de nombreuses extensions sont réalisables. LEMMO est une approche novatrice et unique dans le cadre des travaux sur le multiobjectif et ses perspectives sont nombreuses. Il serait intéressant de tester son potentiel sur d'autres problématiques qui sont également coûteuses en temps de calcul comme le docking ou d'autres problèmes de design tel que le "mapping sur system on chip" sur lequel je suis en train de collaborer avec une autre équipe projet INRIA. Un autre point intéressant est de tester différents algorithmes d'apprentissage pour réaliser l'extraction de schémas et notamment de voir l'influence de ces algorithmes sur les résultats. Une première étude a été réalisée dans la suite de ces travaux par D. Corne.

Dans le cadre des travaux en logistique et en transport, il serait également intéressant de pouvoir déterminer les ensembles de clients et leur ordre qui ont des conséquences sur la qualité des tournées de véhicules. Ainsi l'étude de méthodes hybrides entre des métaheuristiques et des méthodes de type analyse de séries comme l'algorithme foil [148] pourrait être prometteuse.

## Troisième partie

# Vers la coopération de méthodes dans le cadre stochastique





# Préambule

Les problèmes d'optimisation combinatoire sont, dans la pratique, souvent soumis à des aléas soit parce que leurs données sont incertaines ou non complètement connues à l'avance soit parce que le calcul de la fonction objectif est incertaine. En pratique, résoudre ce type de problème se résume souvent à une transformation du problème original, soumis à incertitude, en un problème déterministe. Pourtant, certaines adaptations en termes de méthodes de résolution peuvent s'avérer utiles. Ce domaine de recherche a reçu un intérêt grandissant ces dernières années en raison de sa difficulté et de son importance en pratique. Dans la troisième partie de ce document, nous nous focalisons sur l'optimisation en environnement stochastique avec un focus plus particulier sur l'optimisation sous incertitude dans le chapitre 7 et pour l'optimisation en environnement dynamique dans le chapitre 8.

Nous allons donner les principaux concepts de l'optimisation en environnement stochastique, nous présenterons deux classifications.

Dans leur état de l'art Bianchi *et al.* proposent une classification des problèmes d'optimisation combinatoire sous incertitude [20]. Les auteurs considèrent que deux aspects sont importants à définir : comment l'information incertaine est modélisée et quelle est la dynamicité du modèle, *i.e.* quand l'information est-elle révélée par rapport au moment de la prise de décision. La figure 6.11 résume cette classification. Les auteurs différencient les problèmes où toute l'information est connue au moment de la décision (optimisation déterministe comme étudiée dans la seconde partie de ce document) des autres types de problèmes.

1. **Problèmes d'optimisation combinatoires stochastiques COPS** : dans cette catégorie sont regroupés les problèmes dont l'information incertaine est modélisée par des variables aléatoires avec des distributions de probabilité connues. Dans cette catégorie, les auteurs différencient les problèmes stochastiques en fonction du moment où il faut prendre la décision : soit avant la connaissance de la véritable valeur des variables, on parle alors d'optimisation *a priori* ou soit après la réalisation d'un événement permettant de connaître la véritable valeur des variables, on parle alors d'optimisation d'un problème *dynamique*<sup>1</sup>. Dans ce dernier cas, l'optimisation se fait souvent par étape car l'information est révélée par étape.
2. **Problème d'optimisation floue** : les valeurs des problèmes appartiennent à des ensembles flous, mais ce type d'approche est relativement rare.
3. **Problème d'optimisation robuste** : les données des problèmes appartiennent à des intervalles
4. **Problème d'optimisation purement online** : les données sont révélées progressivement à l'algorithme d'optimisation sans qu'aucune hypothèse ne permette de prévoir les nouvelles données.

Dans leur étude sur les approches évolutionnaires d'optimisation en environnement incertain [84], Jin et Branke classent les différents types d'incertitude en quatre catégories légèrement différentes de celles proposées par Bianchi *et al.* [20] :

1. **Fonction objectif bruitée**. L'évaluation d'une solution est sujette au bruit. Ce bruit peut provenir de différentes sources (erreur de mesure sensorielle, simulations aléatoires). Pour ce genre de problèmes, chaque appel de la fonction objectif avec les mêmes arguments retourne une valeur différente.

---

1. Ici dynamique aura un sens différent par rapport au reste du manuscrit

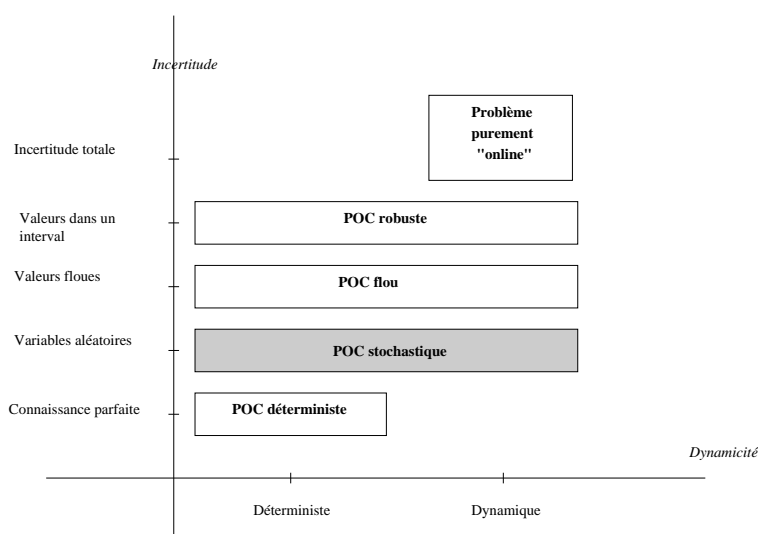


Fig. 6.11 – Classification des problèmes d'optimisation combinatoire (POC) sous incertitude.

2. **Recherche de solutions robustes.** Ici, les variables de décision sont susceptibles d'être soumises à des perturbations ou à des modifications à la suite du processus de recherche. Par conséquent, il est fréquemment exigé qu'une solution reste satisfaisante suite à une légère variation des variables de décision, en raison par exemple de tolérances de fabrication. De telles solutions sont dites *robustes*. Dans l'idéal, une solution robuste doit donc être insensible à de petites variations sur les variables de décision. Dans un tel cas, la fonction objectif est généralement connue et déterministe, l'incertitude étant uniquement introduite après le processus de recherche. Par ailleurs, comme soulevé par les auteurs, il se peut que des perturbations apparaissent sur d'autres paramètres que les variables de décision, comme les paramètres environnementaux. Ces deux types d'incertitudes sont classés dans une seule et même catégorie par [84]. Nous pensons plutôt qu'elles résultent de sources résolument différentes. Ce dernier type est finalement assez proche du cas où la fonction objectif est bruitée, puisque l'incertitude prise en compte ici a une répercussion directe sur le calcul de la fonction objectif. Ainsi, la recherche de solutions robustes peut être traitée à partir de ces deux perspectives. Une solution robuste peut donc être également vue comme insensible à de petites variations sur les paramètres environnementaux.
3. **Fonction objectif approchée.** Lorsque l'évaluation d'une solution est extrêmement coûteuse en temps de calcul, ou qu'une fonction objectif analytique n'est pas disponible, la fonction objectif utilisée est souvent approchée à l'aide de données générées de façon expérimentale ou issue de simulations. En conséquence, l'algorithme doit généralement être en mesure de manipuler deux fonctions : la fonction originale, coûteuse mais exacte, et une fonction de substitution, économique mais imprécise.
4. **Environnement dynamique.** La fonction objectif est ici déterministe à n'importe quel instant, mais est dépendante du temps. En conséquence, l'optimum change également au cours de l'exécution. La méthode mise en place devrait donc être en mesure de continuellement suivre le changement de l'optimum plutôt que de nécessiter un redémarrage complet du processus de recherche.

En pratique, lors du premier cas, la fonction objectif bruitée est souvent estimée à l'aide d'une valeur moyenne explicitement ou implicitement calculée sur un échantillon de valeurs objectif. Lors de la recherche de solutions robustes, la fonction objectif est généralement approchée à l'aide d'une intégration de Monte-Carlo. Ainsi, cette approximation est très proche de celle couramment utilisée en présence de bruit, de sorte que ces deux classes sont étroitement liées l'une à l'autre. En effet, dans tous les cas, la méthodologie la plus classique consiste à établir un échantillon de valeurs objectif, ou de vecteurs de décision ultérieurement transformés en valeurs objectif. Ainsi, peu importe la façon dont l'incertitude est propagée du point de vue de la résolution, les résultats deviennent des ensembles de valeurs, et donc des ensembles de vecteurs objectif dans le cadre de l'optimisation multiobjectif, comme nous le verrons par la suite. En conséquence, la difficulté soulevée concerne la définition d'optimalité en termes d'ensembles.

Jin et Branke [84] remarquent que très peu de travaux sont consacrés à la résolution et au traitement de problèmes d'optimisation multiobjectif en environnement incertain. Ainsi, la première classe d'incertitude, pour laquelle les fonctions objectif sont bruitées, a été indépendamment considérée par [172] et [77]. Par ailleurs, la robustesse de solutions Pareto optimales pour des problèmes d'optimisation multiobjectif où des variables de décision continues sont incertaines a fait l'objet d'une étude approfondie par [45].

Cette partie se divise en deux chapitres. Dans le chapitre 7, après un rapide positionnement sur l'optimisation multiobjectif en environnement incertain, nous présentons deux approches mise en œuvre pour résoudre des problèmes d'optimisation en environnement incertain. Le chapitre 8 se focalise sur une application en optimisation dynamique.

## Chap. 7

# Résolution de problèmes dans un cadre incertain

### Sommaire

---

<b>7.1</b>	<b>Optimisation multiobjectif en environnement incertain . . . . .</b>	<b>87</b>
7.1.1	À propos de robustesse . . . . .	87
7.1.2	Incertitude en optimisation multiobjectif . . . . .	87
7.1.3	Approches métaheuristiques . . . . .	90
<b>7.2</b>	<b>Trouver des solutions par approche multiobjectif à un problème monoobjectif incertain . . . . .</b>	<b>92</b>
7.2.1	Cadre d'étude : problème de tournée de véhicule avec domaine incertain (SVRP) . . . . .	92
7.2.2	Une modélisation pour trouver des solutions robustes . . . . .	92
7.2.3	Algorithmes . . . . .	94
7.2.4	Résultats et discussion . . . . .	94
7.2.5	Protocole . . . . .	94
<b>7.3</b>	<b>Résoudre un problème stochastique multiobjectif . . . . .</b>	<b>98</b>
7.3.1	Problématique . . . . .	98
7.3.2	Modélisation de l'incertitude . . . . .	98
7.3.3	Approches métaheuristiques pour l'optimisation multiobjectif en environnement incertain . . . . .	99
7.3.4	Cadre d'étude : le Flowshop de permutation . . . . .	100
7.3.5	Résultats et discussion . . . . .	102
<b>7.4</b>	<b>Conclusion . . . . .</b>	<b>103</b>

---

Ce chapitre a donné lieu aux publications suivantes :

- D. Sulieman, L. Jourdan and E-G. Talbi. *Using Multiobjective metaheuristics to solve VRP with uncertain demands*, CEC 2010, IEEE 2010.
- A. Liefoghe, L. Jourdan, E-G. Talbi. *Indicator-based approaches for multiobjective optimization in uncertain environments : An application to multiobjective scheduling with stochastic processing times*, 25th Mini-EURO Conference : Uncertainty and Robustness in Planning and Decision Making (URPDM 2010), Coimbra, Portugal, 2010.

- A. Liefoghe, M. Basseur, L. Jourdan and E.-G. Talbi. *Multi-objective optimization for stochastic problems : an application to the flow-shop scheduling problem*. *EMO 2007*, LNCS 4408, 386–400. 2007.
- A. Liefoghe, L. Jourdan, M. Basseur and E.-G. Talbi, *Métaheuristiques pour le flow-shop sous incertitude*, *Revue d'Intelligence Artificielle*, Hermès, vol. 22, n°2, pp. 183-208, 2008. ISBN : 0992-499X

Dans ce chapitre, nous présentons deux approches de résolution de problèmes dans un environnement stochastique. La première propose de modéliser de façon multi-objective un problème monoobjectif présentant des sources d'incertitude et a été réalisée lors du stage de Master recherche de Dalia Sulieman que j'ai co-encadré, la seconde propose de résoudre un problème multi-objectif présentant de l'incertitude et a été traité dans les travaux de thèse de Arnaud Liefoghe.

## 7.1 Optimisation multiobjectif en environnement incertain

### 7.1.1 À propos de robustesse

La notion de *robustesse* étant souvent sujette à discussion, il nous a semblé utile de nous y attarder plus longuement. En effet, il n'existe pas de définition unanime de la robustesse dans la littérature. En règle général, le concept de robustesse se réfère à la capacité du sujet à faire face à des incertitudes. B. Roy définit le terme robuste comme *un qualificatif se rapportant à une aptitude à résister à des « à peu près » ou à des « zones d'ignorances » afin de se protéger d'impacts jugés regrettables* [157]. En effet, l'auteur remarque que la notion de robustesse ne doit pas être uniquement appliquée aux solutions, mais plus généralement à différentes affirmations et recommandations générées au sein d'un système informatique d'aide à la décision. Tout d'abord, nous allons distinguer une *solution robuste* et une *méthode de résolution robuste*, cette dernière se rapportant généralement à son extensibilité. Cependant, une vue commune est partagée : une solution robuste doit être en mesure de bien se comporter sous des conditions (légèrement) différentes, ce qui signifie qu'elle doit, autant que possible, être à l'abri de petites modifications au sein des conditions dans lesquelles elle a été conçue. De façon générale, la prise en compte de la robustesse d'une solution se doit de distinguer les variables de décisions déterministes, les variables de décisions stochastiques, et les paramètres extérieurs.

Ainsi, nous allons considérer qu'une solution à un problème d'optimisation stochastique est robuste si elle conserve une haute performance indépendamment d'une réalisation des données incertaines. Cette notion est donc à la fois liée à la qualité d'une solution ainsi qu'à sa variabilité, ou stabilité. Par ailleurs, la robustesse d'une solution se rapporte habituellement à la *fiabilité* et à la *flexibilité* de cette même solution. La *fiabilité* d'une solution est liée à la probabilité que celle-ci reste réalisable en présence d'incertitude, même s'il est en général difficile de totalement exclure un tel risque. Par ailleurs, il peut s'avérer possible d'ajuster une solution en fonction de tels événements. La *flexibilité* d'une solution peut être définie comme sa capacité à pouvoir être adaptée à la suite d'une réalisation des données incertaines.

### 7.1.2 Incertitude en optimisation multiobjectif

Une approche de résolution très courante en optimisation monoobjectif sous incertitude [84] consiste à établir un échantillon de valeurs objectif associé à chaque solution réalisable. Néanmoins, nous pouvons distinguer le cas où une fonction objectif est intrinsèquement incertaine (présence de bruit, environnement dynamique) du cas où la fonction objectif est une expression fonctionnelle déterministe (robustesse). Lors du passage à l'optimisation multiobjectif, la présence d'incertitude entraîne la comparaison d'ensembles de vecteurs objectif à la place d'un vecteur unique et précis. Ci-dessous, nous tentons d'expliquer la façon méthodologique de pro-

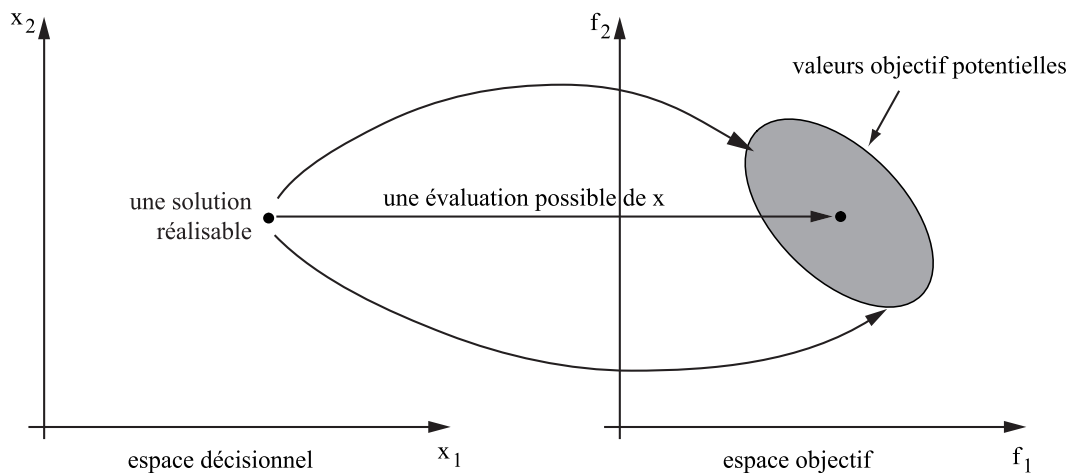


céder afin de générer un échantillon de vecteurs objectif associé à une solution dans le cadre de l'optimisation multiobjectif, ceci pour chacune des classes d'incertitude étudiées.

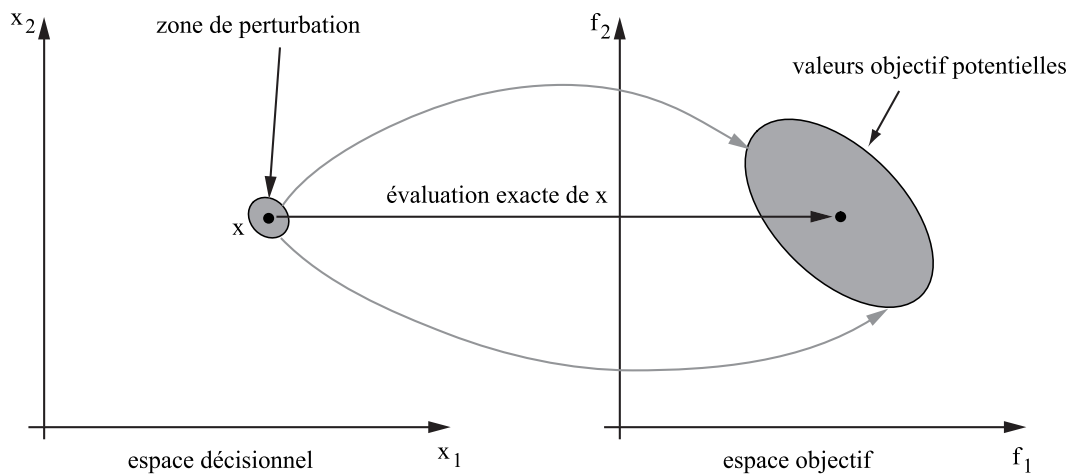
- **Fonctions objectif bruitées.** Dans ce cas, les fonctions objectif sont directement sujettes au bruit, la seule évaluation disponible est donc intrinsèquement stochastique. Ainsi, soit une distribution de probabilité connue ou présumée est considérée, ce qui semble peu probable en pratique, soit il s'avère nécessaire d'évaluer à plusieurs reprises, et de façon indépendante, une même solution en vue d'obtenir un ensemble fini de résultats possibles (Fig. 7.1). C'est cette dernière alternative que nous allons prendre pour la suite. De ce fait, aucune évaluation ne peut être estimée plus probable qu'une autre, et encore moins comme évaluation réelle, ou comme évaluation de référence.
- **Robustesse vis-à-vis des paramètres environnementaux.** Ici, les fonctions objectif ne sont pas intrinsèquement stochastiques. Néanmoins, lors de l'évaluation d'une solution, celles-ci requièrent un certain nombre de paramètres qui peuvent être aléatoires. Pour tout paramètre environnemental incertain, un ensemble de valeurs possibles est considéré, chaque combinaison donnant lieu à une évaluation potentiellement différente pour une même solution. Ainsi, de la même manière que pour le cas précédent, nous allons comptabiliser un ensemble fini de vecteurs objectif possibles (Fig. 7.1). La prise en compte de ce type d'incertitude est donc relativement semblable au cas précédent.
- **Robustesse vis-à-vis des variables de décision.** Ce dernier type d'incertitude se montre légèrement différent des deux autres. En effet, à chaque solution réalisable correspond bien un vecteur objectif unique. Autrement dit, l'évaluation réelle d'une solution peut être calculée. Néanmoins, les variables de décision sont ici susceptibles d'être modifiées à la suite du processus de recherche. Ainsi, la solution trouvée se doit d'être satisfaisante et donc insensible (dans la mesure du possible) à la perturbation des variables de décision dans son voisinage. En pratique, un ensemble fini de solutions peut être défini (aléatoirement ou de façon structurée) autour d'une solution, dans l'espace décisionnel ; ceci selon un opérateur de voisinage donné. Une des difficultés réside donc en la définition d'un opérateur de voisinage pertinent pour le problème sous incertitude à résoudre. A la place d'une seule évaluation par solution, un ensemble fini de vecteurs objectif, correspondant à l'évaluation des solutions voisines, est encore une fois considéré ; même s'il est vrai que l'évaluation réelle et exacte d'une solution donnée est ici connue (Fig. 7.2). L'intention est donc de tenir compte de la sensibilité des valeurs objectif due à l'incertitude des variables de décision dans un contexte multiobjectif.

Il se peut que certains problèmes soient sujets à plusieurs types d'incertitude simultanément, auquel cas un ensemble d'ensembles de vecteurs objectif est dès lors associé à chaque solution réalisable.

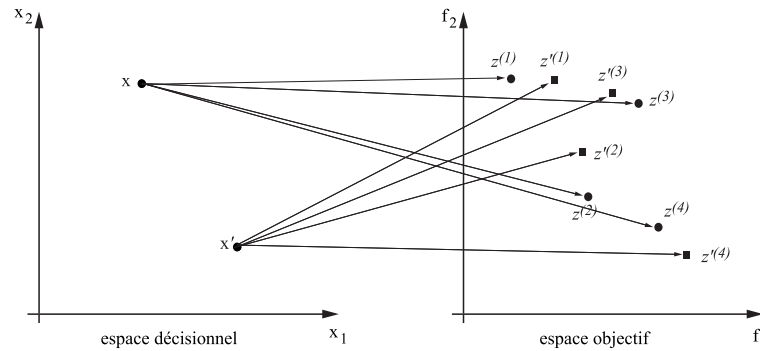
Comme nous venons de le voir, dès que la prise en compte de l'incertitude se base sur un échantillonnage, cela a pour effet de devoir comparer des ensembles de vecteurs objectif plutôt qu'un vecteur objectif unique. Par conséquent, la question fondamentale, et donc le véritable challenge soulevé par ce type de problèmes, se résume à pouvoir comparer et discriminer différentes solutions alternatives en termes de comparaison d'ensembles de vecteurs objectif (Fig. 7.3). Ainsi, cet aspect donne lieu à la principale différence entre de la conception de métaheuristiques pour l'optimisation de problèmes multiobjectif déterministes et stochastiques.



**Fig. 7.1** – Prise en compte de l’incertitude à l’aide d’un échantillon de vecteurs objectif pour les fonctions objectif bruitées et pour la recherche de solutions robustes vis-à-vis de paramètres environnementaux incertains.



**Fig. 7.2** – Prise en compte de l’incertitude à l’aide d’un échantillon de vecteurs objectif pour la recherche de solutions robustes vis-à-vis de variables de décision incertaines.



**Fig. 7.3** – Illustration de la comparaison de deux ensembles de vecteurs objectif  $\{z^{(i)}\}_{i=1}^4$  et  $\{z'^{(j)}\}_{j=1}^4$  obtenus de l'évaluation multiple de deux solutions. Quelle solution est meilleure que l'autre ?

### 7.1.3 Approches métaheuristiques

Dans cette section, nous essayons de fournir un aperçu global des méthodes de résolution en termes de métaheuristiques pour les problèmes d'optimisation multiobjectif sous incertitude. Une première tentative a déjà été faite en ce sens par [170] mais est très focalisée sur les travaux des auteurs et se concentre uniquement sur les algorithmes évolutionnaires.

La classification proposée ici est divisée en deux niveaux : le choix de la valeur ou du vecteur représentatif, et, l'exploitation de cette valeur représentative.

#### Choix de valeurs représentatives

La plupart des techniques visant à déterminer des valeurs représentatives d'un ensemble d'informations sont des approches scalaires, qui consistent à agréger un échantillon de données en une valeur scalaire unique, et qui résultent donc en la conversion de la solution imprécise en un vecteur objectif unique. La comparaison des ensembles est alors réduite à une comparaison de points représentatifs. Ainsi, la formulation résultante du problème devient déterministe et une méthode de résolution usuelle peut être envisagée. Les différents choix de valeurs représentatives sont :

- **Valeur originale** : l'évaluation réelle d'une solution est calculée [45, 68] afin de trouver des solutions robustes vis-à-vis des variables de décision. Afin de ne pas se réduire à étudier le problème déterministe, la valeur originale unique ne peut pas être considérée de façon exclusive, et doit donc être complétée d'un ou plusieurs autres éléments d'information.
- **Mesures basées sur l'espérance mathématique** : une forme typique de scalarisation consiste à approcher l'espérance mathématique de la, ou des données incertaines à l'aide de la valeur moyenne observée sur un échantillon de valeurs [28, 45, 62, 169], on considère alors la valeur moyenne de  $p$  évaluations d'une même solution ce qui permet de générer des solutions de bonne qualité, en moyenne.
- **Mesures basées sur la variance** : ce type de modèle basé sur la moyenne et la variance de valeurs objectif semble particulièrement bien adapté aux distributions normales. Il ne prend en compte que les déviations de la fonction, et ignore généralement le critère de performance. D'autres mesures d'incertitude, comme l'étendue, l'écart interquartile ou l'entropie, peuvent

également être définies.

- **Mesures basées sur les valeurs extrêmes** : le pire et le meilleur des cas correspondent aux valeurs extrêmes de l'intervalle de valeurs obtenues par une solution par rapport à une fonction objectif. Une telle mesure peut être considérée de façon absolue [104], mais également en addition à une autre valeur représentative [68].
- **Échantillon complet** : certaines méthodes de résolution tentent de prendre en compte la totalité de l'échantillon de vecteurs objectif associé à chaque solution, sans déterminer de valeur plus représentative qu'une autre pour chaque fonction objectif.
- **Information additionnelle** : un grand nombre de travaux supposent l'existence explicite d'une loi de probabilité associée sur l'espace objectif à chaque solution réalisable [77, 172].

D'autres types de mesures représentatives peuvent tout aussi bien être envisagés, comme par exemple une mesure dédiée au maintien de la faisabilité d'une solution en environnement incertain, et donc étroitement liée à la fiabilité d'une solution. On pourrait aussi imaginer des mesures de robustesse plus proches, voire spécifiques au problème stochastique étudié. Enfin, à notre connaissance, il n'existe pas d'approche basée sur la logique floue, bien que l'usage de ce type de théories de l'incertitude semble tout à fait envisageable en pratique.

### Exploitation des valeurs représentatives

Ci-dessous sont présentées différentes stratégies permettant d'exploiter l'information contenue au sein des valeurs représentatives désignées pour chaque solution. Nous pensons que ces différentes approches sont caractéristiques de l'état de l'art actuel pour la résolution de problèmes d'optimisation multiobjectif stochastiques en termes de métaheuristiques.

- **Résolution déterministe** : Si l'étape de choix de valeurs représentatives par solution se résume à la sélection d'une valeur scalaire unique par fonction objectif, une méthode classique peut directement être utilisée afin de résoudre le problème étudié [45, 169].
- **Contrainte(s) ou objectif(s) additionnel(s)** : un ensemble de contraintes, ou encore un ensemble de fonctions objectif est ajouté ce qui a donc pour effet d'augmenter le nombre de contraintes ou d'objectifs par rapport au problème initial déterministe. Cette classe d'approches semble surtout populaire afin de traiter de la robustesse vis-à-vis des variables de décision comme par exemple ajouter une contrainte sur la déviation maximale par rapport au vecteur objectif original [45]. La considération d'une fonction objectif additionnelle est une technique relativement courante lors de la résolution d'un problème d'optimisation monoobjectif sous incertitude [153, 85]. Elle consiste généralement à considérer simultanément, et comme objectifs distincts, le vecteur objectif moyen, remplacé par le vecteur objectif déterministe dans le cas de la recherche de solutions robustes, voire la considération des deux [153], ainsi qu'une mesure de robustesse. Cette mesure de robustesse peut par exemple concerner la variance [62] ou encore la valeur dans le pire des cas [68].
- **Relations de dominance probabilistes** : Cet ensemble d'approches correspond à différentes manières de modifier la relation de dominance, telle que la dominance Pareto, afin de prendre l'incertitude en compte par le biais de vecteurs objectif multiples pour chaque solution. Il suffit alors d'utiliser un tel critère au sein d'une métaheuristique basée sur une relation de dominance afin de résoudre le problème incertain considéré [28, 77, 172, 175]. Une discussion plus détaillée sur ce point est présentée dans les travaux de thèse de A. Liefoghe [111].
- **Adaptation des valeurs de fitness** : La dernière catégorie proposée consiste à adapter les valeurs de fitness affectées aux solutions afin de prendre en compte l'incertitude des vecteurs

objectif. Ainsi, la stratégie d'affectation des valeurs de fitness utilisée considère directement l'ensemble des vecteurs objectif associés à chaque solution, ou à un ensemble de valeurs représentatives par objectif (typiquement, une mesure moyenne et une mesure de variance) [7, 10].

## 7.2 Trouver des solutions par approche multiobjectif à un problème monoobjectif incertain

Certaines approches issues de l'optimisation multiobjectif sont parfois considérées en vue de la résolution de problèmes monoobjectif incertains. Dans un tel cas, une fonction objectif proche de l'originale, ainsi qu'un critère de robustesse, sont tous deux considérés simultanément comme des objectifs distincts. Le but est ici d'obtenir de multiples compromis entre performance et robustesse. Par exemple, [85] proposent de considérer à la fois l'espérance mathématique et une mesure de variance d'un ensemble de valeurs observées sur la fonction objectif bruitée originale, obtenant ainsi deux fonctions objectif symbolisant respectivement la qualité et la robustesse d'une solution.

Dans cette partie, basés sur ce principe, nous proposons une approche pour un problème de routage avec demande incertaine. L'objectif est ici de montrer l'importance du critère de robustesse.

### 7.2.1 Cadre d'étude : problème de tournée de véhicule avec domaine incertaine (SVRP)

Le problème statique (ou déterministe) de tournée de véhicule consiste à déterminer en minimisant le coût, un ensemble de tournées pour un nombre limité de véhicules, commençant et finissant à un dépôt, de telle façon que chaque client soit visité exactement une fois par un véhicule, et que la somme des demandes provenant des clients sur une tournée ne dépasse pas la capacité du véhicule qui dessert la route. Nous nous intéressons ici au problème de tournée de véhicules avec demande incertaine. Dans ce cas, la demande du client n'est connue de façon certaine qu'à l'arrivée chez le client. Dans [19], une modélisation mathématique est proposée : Soit un graphe complet  $G = (V, A, D)$ ,  $V = 0, 1, \dots, n$  un ensemble de clients (le client 0 représente le dépôt),  $A = (i, j) : i, j \in V, i \neq j$  un ensemble d'arcs et  $D = d_{ij} : i, j \in V, i \neq j$  sont les coûts des trajets (ou la distance entre les noeuds).  $D$ , la matrice de coût, est symétrique. Chaque véhicule est défini par une capacité  $Q$ . La demande des clients est une variable stochastique  $\xi_i, i = 1, \dots, n$ . Afin de modéliser le problème, nous avons estimé que chaque variable était distribuée suivant une loi de probabilité qui est inconnue pour l'algorithme de résolution [19].

Dans la littérature, deux stratégies sont utilisées pour résoudre ce problème : l'*optimisation a priori* où dans ce cas le véhicule qui devient vide retourne au dépôt afin de servir le client [65] et la stratégie de *prévention du restockage*, qui étudie le restockage lorsqu'un véhicule est proche du dépôt [18, 19, 184]. Dans notre cas, nous nous plaçons dans un cadre d'optimisation *a priori*.

### 7.2.2 Une modélisation pour trouver des solutions robustes

#### Problématique

Comme indiqué précédemment, une solution robuste doit être insensible aux petites variations de la solution et de l'environnement [85].

Dans le cas du SVRP, la demande réelle est inconnue jusqu'à ce que le véhicule arrive chez le

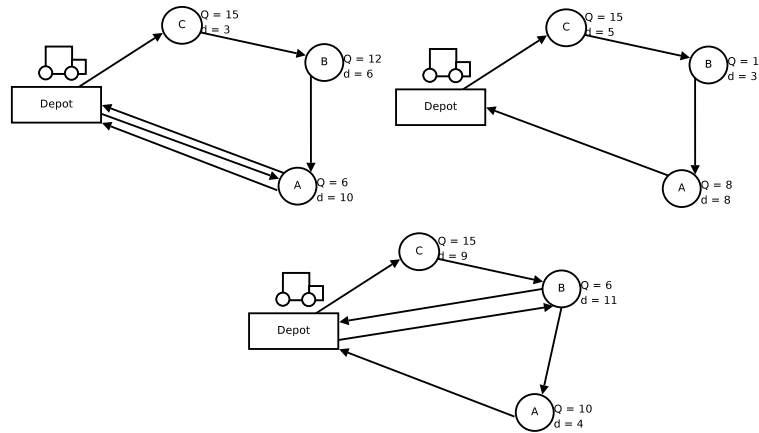


Fig. 7.4 – Exemple de modification de la tournée due au caractère stochastique des demandes

client, lors du calcul de la distance sur une tournée définie *a priori*, trois cas peuvent se présenter (voir figure 7.4) :

1. La demande du client  $i$  est plus petite que la capacité/quantité restante du véhicule  $\xi_i < Q$  alors le véhicule va servir le client et aller chez le client suivant ;
2. La demande est identique à la capacité/quantité restante du véhicule  $\xi_i = Q$ , alors le véhicule va servir le client, retourner au dépôt afin de charger/décharger puis du dépôt va partir vers le client suivant. Dans ce cas, on doit ajouter un aller-retour vers le dépôt pour calculer la longueur de la tournée  $d(Tour)+ = d(i, 0) + d(0, i + 1)$  ;
3. La demande du client est plus grande que la capacité/quantité restante du véhicule  $\xi_i > Q$  t, le véhicule va alors servir ce client avec le restant de sa capacité, puis retourner au dépôt pour charger/décharger puis retourner chez le même client afin de lui donner le restant de sa demande. Dans ce cas, on rajoute à la longueur de la tournée un aller-retour au dépôt  $d(Tour)+ = d(i, 0) + d(0, i)$ .

### Modélisation

Dans ce premier problème, nous avons travaillé sur l'influence du choix de la valeur représentative. Nous allons considérer deux possibilités pour mesurer la robustesse. Nous aurons donc une modélisation bicritère dont le premier critère mesurera la distance moyenne parcourue :

$$f_1(Tour) = \frac{\sum_{j=1..X} d(Tour)}{X}$$

où  $d(Tour) = \sum_{i=0}^{n-1} d_{i,i+1}$  et  $X$  le nombre de scénariis considérés. Pour le deuxième critère qui mesure la robustesse, deux valeurs représentatives vont être comparées :

- **Modèle basé sur la déviation standard (Sigma model)** où le second objectif doit minimiser la variation sur la distance moyenne parcourue :

$$f_2(Tour) = \sqrt{\frac{1}{X} \sum_{i=1}^X (d_i(Tour) - f_1(Tour))^2}$$

- **Modèle basé sur l'entropie** : Dans [181], les auteurs montrent l'intérêt de l'entropie comme mesure de l'incertitude. Ici, nous proposons donc d'utiliser comme second critère la minimisation de l'entropie afin d'obtenir des solutions robustes :

$$f_2(Tour) = \left(\frac{1}{n}\right)^X \ln(n^X) \sum_{i=1}^n d_{i,i+1}$$

Où  $n$  est le nombre de clients et  $X$  le nombre de scénariis considérés.

### 7.2.3 Algorithmes

Du point de vue résolution algorithmique, nous nous baserons sur des algorithmes classiques multiobjectif de la littérature qui ont été implémentés sous Paradiseo-MOEO : IBEA, NSGA II et MOGA. Au niveau des composants communs de ces algorithmes, nous avons repris les travaux classiques sur le VRP, ainsi pour l'encodage, une solution est un ensemble de routes, chaque route contenant une séquence de clients. L'initialisation d'une solution se fait aléatoirement, la mutation est l'échange entre deux clients et le croisement est un croisement deux points.

### 7.2.4 Résultats et discussion

#### Jeux de données et Paramètres

Dans le cadre de ce travail, de nouveaux jeux de données basés sur les jeux de données de Solomon ont été générés. La demande a été modélisée comme une loi de probabilité basée sur une loi normale  $\mathcal{N}(\mu, \sigma)$  avec  $\mu$  la demande du benchmark de Solomon considéré et  $\sigma = 0.15 \times \mu$ . Six jeux de données ont été traités : C101, C201, R101, R201, RC101, RC201. Chacune de ces instances possède 100 clients.

La taille de la population a été fixée expérimentalement à 150, le taux de mutation à 0.25 et le taux de croisement à 0.1. La capacité des véhicules est fixée à  $Q = 20$ .

Pour chaque algorithme et chaque jeu de données, 20 exécutions ont été réalisées.

#### 7.2.5 Protocole

Dans ce premier travail, nous considérerons un ensemble de scénariis indépendants et équiprobables et, à une solution  $x$ , correspondra un ensemble d'évaluation (autant que de scénariis). Dans ce premier travail, nous avons fixé à 20 le nombre de scénarios considérés.

#### Résultats

Nos expérimentations ont été divisées en deux parties en fonction de leur objectif. Afin de réaliser l'analyse des résultats, nous avons utilisé les indicateurs d'hypervolume et  $\epsilon$  de la plateforme PISA.

**Comparaison en terme de robustesse** Nous reprenons une vision classique de la robustesse qui est de mesurer la différence entre le meilleur des cas et le pire des cas, ici, sur la distance parcourue. Plus cette distance est petite, plus la solution est robuste (mais elle n'est pas obligatoirement de bonne qualité par rapport à la minimisation de la distance). A partir des résultats

TABLE 7.1 – Comparaison des modèles pour la robustesse.

Modèle	basé sur l'entropie	basé sur la déviation
c101	<b>43.56</b>	63.73
r101	<b>27.23</b>	35.35
rc101	41.01	<b>33.7</b>
c201	<b>56.09</b>	467.42
r201	<b>72.43</b>	2949.97
rc201	186.91	<b>51.85</b>

obtenus résumés sur les tableaux 7.1, nous remarquons que le modèle basé sur l'entropie donne des résultats généralement plus robustes que le modèle basé sur la déviation standard.

**Comparaison d'algorithmes de résolution** En nous référant au protocole que nous avons utilisé dans la cadre déterministe pour comparer les fronts Pareto, nous remarquons, que, comme pour les modèles déterministes, IBEA est plus performant que MOGA et NSGAI (voir tableaux 7.2 et 7.3).



-		Hypervolume $I_H$			Epsilon $I_\epsilon$		
		IBEA	MOGA	NSGAI	IBEA	MOGA	NSGAI
c101	IBEA	.	≡	∩	.	≡	∩
	MOGA	≡	.	∩	≡	.	≡
	NSGAI	∩	∩	.	∩	≡	.
r101	IBEA	.	≡	∩	.	≡	∩
	MOGA	≡	.	≡	≡	.	∩
	NSGAI	∩	≡	.	∩	∩	.
rc101	IBEA	.	≡	≡	.	≡	≡
	MOGA	≡	.	≡	≡	.	≡
	NSGAI	≡	≡	.	≡	≡	.
c201	IBEA	.	≡	∩	.	≡	∩
	MOGA	≡	.	∩	≡	.	∩
	NSGAI	∩	∩	.	∩	∩	.
r201	IBEA	.	≡	∩	.	≡	≡
	MOGA	≡	.	∩	≡	.	∩
	NSGAI	∩	∩	.	≡	∩	.
rc201	IBEA	.	≡	∩	.	≡	∩
	MOGA	≡	.	≡	≡	.	∩
	NSGAI	∩	≡	.	∩	∩	.

TABLE 7.2 – Comparaison de statistique de la qualité des résultats obtenus par IBEA, MOGA et NSGAI par un test de Mann-Whitney en utilisant les indicateurs d'hypervolume et  $\epsilon$  pour le modèle basé sur l'entropie. La p-valeur est fixée à 0.05.  $\cap$  les résultats sont significativement moins bons,  $\succ$  les résultats sont significativement meilleurs,  $\equiv$  il n'y a pas de différences significatives.

-		Hypervolume $I_H$			Epsilon $I_\epsilon$		
		IBEA	MOGA	NSGAI	IBEA	MOGA	NSGAI
c101	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.
r101	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.
rc101	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.
c201	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.
r201	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.
rc201	IBEA	.	$\succ$	$\succ$	.	$\succ$	$\succ$
	MOGA	$\surd$	.	$\surd$	$\surd$	.	$\surd$
	NSGAI	$\surd$	$\succ$	.	$\surd$	$\succ$	.

TABLE 7.3 – Comparaison de statistique de la qualité des résultats obtenus par IBEA, MOGA et NSGAI par un test de Mann-Whitney en utilisant les indicateurs d'hypervolume et  $\epsilon$  pour le modèle basé sur la déviation standard. La p-valeur est fixée à 0.05.  $\surd$  les résultats sont significativement moins bons,  $\succ$  les résultats sont significativement meilleurs,  $\equiv$  il n'y a pas de différences significatives.

## 7.3 Résoudre un problème stochastique multiobjectif

### 7.3.1 Problématique

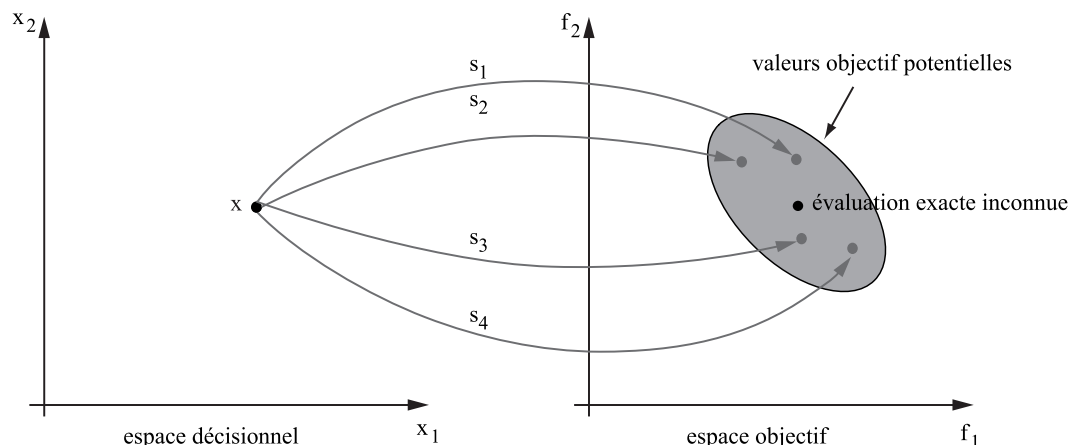
Une des questions fondamentales liée à la résolution d'un problème d'optimisation multiobjectif stochastique concerne le choix d'une taille appropriée de l'échantillon de vecteurs objectif associé à chaque solution, et si cette taille doit être fixe ou adaptée au cours de la recherche, ou encore selon la solution considérée. Aussi, plutôt que de réévaluer une même solution à plusieurs reprises, une autre technique semblable consiste à n'affecter qu'un seul vecteur objectif par solution, tout en manipulant une population de très grande taille au sein de la métaheuristique développée. En effet, lorsque la taille de la population est extrêmement large, beaucoup de solutions identiques sont généralement observées, chacune d'elles ayant vraisemblablement un vecteur objectif différent de par la nature stochastique de la fonction d'évaluation. De cette façon, l'influence de l'incertitude est atténuée, et cette stratégie résulte en un sens à un calcul de moyenne implicite [170]. De même, une autre approche consiste à ne conserver qu'un seul vecteur objectif par solution, mais à réévaluer cette même solution régulièrement, ceci afin de limiter l'impact de solutions dont les valeurs objectif sont aberrantes [26]. Par ailleurs, [28] proposent qu'une nouvelle solution hérite de la valeur des vecteurs objectif de ses parents, ceci dans le cadre d'un algorithme évolutionnaire. Chaque solution est évaluée un petit nombre de fois, et si les vecteurs objectif obtenus diffèrent trop de ceux des parents, elle sera réévaluée plus de fois. Ceci permet de sauver un temps de calcul précieux lorsque la fonction d'évaluation est trop coûteuse à calculer.

### 7.3.2 Modélisation de l'incertitude

Lors de la prise en compte de l'incertitude, à chaque solution est associé un ensemble de vecteurs objectif, chacun d'eux résultant éventuellement en un point différent de l'espace objectif. Nous allons donc considérer que  $f$  ne représente pas une application déterministe de  $X$  vers  $Z$ , mais plutôt qu'un ensemble potentiellement infini de vecteurs objectif est dorénavant associé à une solution réalisable  $x$ . Nous supposons que l'évaluation « réelle » d'une solution est absolument inconnue avant la fin du processus de recherche. Aucune hypothèse n'est formulée sur une quelconque distribution de probabilité associée aux fonctions objectif, aux variables de décisions, ou aux paramètres environnementaux ; une telle distribution étant généralement inconnue à l'avance et étant susceptible de différer pour toute solution.

Ainsi, dans le cas incertain, à chaque solution  $x \in X$  est associé un échantillon de vecteurs objectif. Le degré d'incertitude consiste en une plus grande variance observée sur les vecteurs objectif résultant de l'évaluation multiple de  $x$ . Il s'avère donc nécessaire de déterminer une taille d'échantillon satisfaisante, car l'étape d'échantillonnage peut s'avérer coûteuse en temps de calcul.

Plus formellement, à toute solution  $x \in X$ , nous supposons qu'un ensemble d'évaluations indépendantes et équiprobables est calculé. Ainsi, un échantillon de vecteurs objectif  $\{z^{(i)}\}_{i=1}^p$  est désormais associé à chaque solution. Deux cas peuvent se présenter. Premièrement, pour une évaluation donnée d'une solution arbitrairement considérée, il se peut que la fonction d'évaluation soit strictement indépendante de toutes les évaluations réalisées jusqu'à présent. Dans ce cas, tous les vecteurs objectif de toutes les solutions évaluées sont strictement indépendants les uns des autres, et la taille de l'échantillon de vecteurs objectif peut éventuellement différer d'une solution à l'autre. Une autre alternative justifiable est de considérer un ensemble fini de



**Fig. 7.5** – Illustration de l'évaluation multiple d'une solution  $x$  à partir de quatre scénarios différents  $\{s_1, s_2, s_3, s_4\}$ .

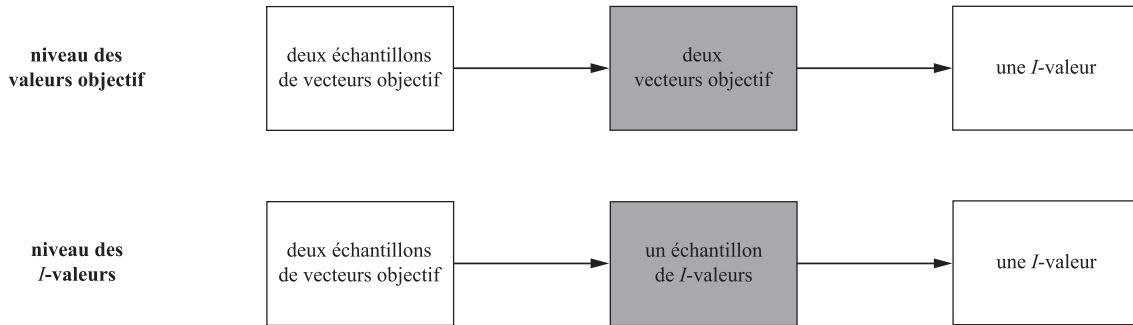
scénarios indépendants et équiprobables  $S = \{s_1, s_2, \dots, s_p\}$  [104].  $S(x) = \{z^{(1)}, z^{(2)}, \dots, z^{(p)}\}$  correspond à l'échantillon d'évaluations indépendantes relatives à une solution  $x \in X$ . Dans ce cas, l'élément  $z^{(i)}$  de l'échantillon associé à  $x$  représente la valeur du vecteur objectif de  $x$  si le scénario  $s_i$  se produit (Fig. 7.5). Par cela, nous supposons que, étant donnés deux solutions  $x$  et  $x' \in X$ , les échantillons de vecteurs objectif  $\{z^{(i)}\}_{i=1}^p$  et  $\{z'^{(j)}\}_{j=1}^{p'}$  sont appariés, et de même taille ( $p = p'$ ). Ainsi, pour un scénario  $s_i$  donné, les vecteurs objectif correspondants  $z^{(i)}$  et  $z'^{(i)}$  sont fondamentalement comparables l'un à l'autre.

### 7.3.3 Approches métaheuristiques pour l'optimisation multiobjectif en environnement incertain

Les algorithmes que nous proposons pour la résolution de problèmes d'optimisation multiobjectif stochastiques s'appuient tous sur la stratégie d'affectation de valeurs de fitness basée sur un indicateur binaire de qualité proposée par [189]. Afin de prendre en compte l'incertitude, nous introduisons un ensemble de huit indicateurs de qualité permettant de manipuler l'incertitude, et pouvant être utilisés au sein de n'importe quelle métaheuristique basée sur un indicateur binaire, comme IBEA [189] ou encore IBMOLS [11]. Ces indicateurs correspondent en quelque sorte à différentes stratégies parmi lesquelles le décideur peut choisir selon ses préférences, ou selon le type de problème à résoudre. Nous supposons tout d'abord qu'un indicateur binaire  $I : Z \times Z \rightarrow \mathbb{R}$ , dédié à la comparaison de deux vecteurs objectif, est défini. Deux exemples ont déjà été donnés dans la première partie du manuscrit à savoir les indicateurs  $I_{\epsilon+}$  et  $I_{HD}$ . Aussi, contrairement au cas déterministe, l'interprétation de ce type d'indicateur depuis l'espace objectif vers l'espace décisionnel n'est plus immédiate. En effet, à une solution de l'espace décisionnel ne correspond plus un vecteur objectif unique, mais plutôt un ensemble de vecteurs objectif.

$$I(x, x') \neq I(f(x), f(x')) \quad (7.1)$$

Les approches proposées consistent donc à définir différentes stratégies d'agrégation de l'information donnée par deux échantillons de vecteurs objectif  $\{z_k^{(1)}, z_k^{(2)}, \dots, z_k^{(p)}\}$  et  $\{z_k'^{(1)}, z_k'^{(2)}, \dots, z_k'^{(p')}\}$ ,



**Fig. 7.6** – Deux classes de métaheuristiques proposées pour la résolution de problèmes multiobjectif stochastiques : les approches basées au niveau des valeurs objectif, et les approches basées au niveau des  $I$ -valeurs.

respectivement associés à deux solutions  $x$  et  $x' \in X$ , en une  $I$ -valeur scalaire unique. En ce sens, ce travail étend donc la contribution de [15] qui ont déjà proposé de tels indicateurs pour traiter de fonctions objectif bruitées, et pour le cas particulier de l'indicateur  $I_{\epsilon+}$ . Les approches présentées ici sont indépendantes de l'indicateur  $I : Z \times Z \rightarrow \mathbb{R}$  choisi. La  $I$ -valeur obtenue peut donc tout naturellement prendre place au sein de la stratégie d'affectation d'une valeur de fitness basée sur un indicateur binaire de qualité. Pour mémoire, le calcul des valeurs de fitness est rappelé ci-dessous.

$$F(x) = \sum_{x' \in P \setminus \{x\}} -e^{-I(x',x)/\kappa} \quad (7.2)$$

Deux types d'indicateur, correspondant à deux niveaux de scalarisation, sont ici proposés (Fig. 7.6). Tout d'abord, les indicateurs situés au niveau des valeurs objectifs se basent directement sur l'échantillon  $\{z_k^{(1)}, z_k^{(2)}, \dots, z_k^{(p)}\}$  associé à une solution  $x \in X$ . Ces approches consistent à transformer l'échantillon associé à une solution en un vecteur objectif unique par le biais d'une valeur, jugée représentative, par fonction objectif. Ces approches situées au niveau des vecteurs objectif sont basées sur des concepts génériques, et peuvent donc être potentiellement appliquées à tout type de méthode pour l'optimisation multiobjectif, et pas uniquement à celles qui sont basées sur un indicateur binaire de qualité. Pour cela, il suffit de considérer le vecteur représentatif unique choisi comme vecteur objectif déterministe. Le deuxième type d'indicateur, situés au niveau des  $I$ -valeurs, consistent à calculer, dans un premier temps, un échantillon de  $I$ -valeurs issu de deux échantillons de vecteurs objectif et associé à une paire de solutions de la population courante. Au sein de ces deux classes d'approches, quatre indicateurs sont proposés. Ils correspondent respectivement à une stratégie dans le meilleur des cas, dans le pire des cas, dans le cas moyen et dans le cas médian (Fig. 7.7).

### 7.3.4 Cadre d'étude : le Flowshop de permutation

#### Modèle et données

**Le flowshop de permutation** Le problème de Flowshop consiste à ordonnancer un ensemble de  $N$  jobs  $\{J_1, J_2, \dots, J_N\}$  sur  $M$  machines  $\{M_1, M_2, \dots, M_M\}$ . Les machines sont des ressources critiques, c'est-à-dire que deux jobs ne peuvent être affectés à une même machine simultanément.

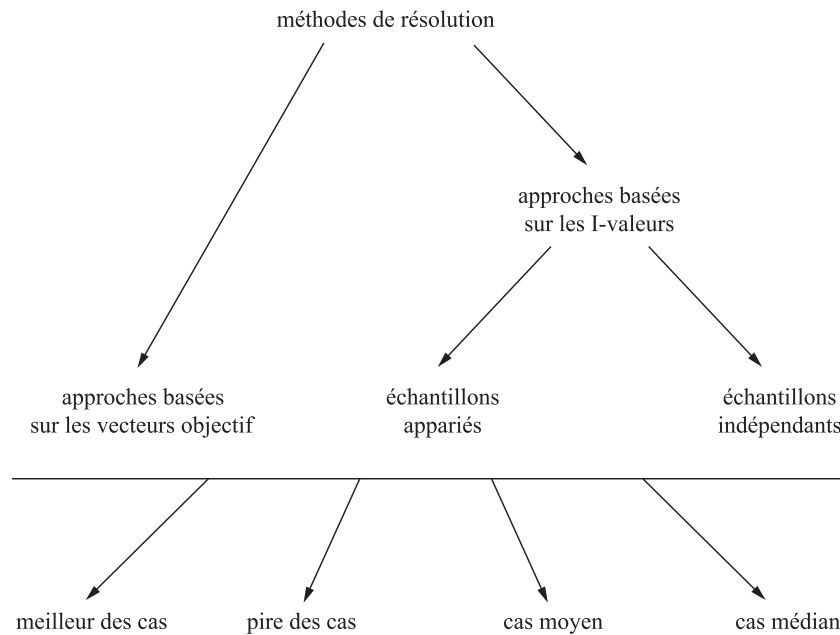


Fig. 7.7 – Résumé des méthodes de résolution proposées.

Un job  $J_i$  est composé de  $M$  tâches consécutives  $\{t_{i1}, t_{i2}, \dots, t_{iM}\}$ , où  $t_{ij}$  représente la  $j$ ème tâche du job  $J_i$  et requiert donc la machine  $M_j$ . A chaque tâche  $t_{ij}$  est associée une durée d'exécution  $p_{ij}$ . Et à chaque job  $J_i$  est associée une date due  $d_i$  (la date de fin souhaitée du job). Un problème de Flowshop à deux fonctions objectif, que nous noterons FSP-2, et qui consiste à minimiser le makespan  $C_{max}$  (la date d'achèvement de l'ordonnancement) et la somme des retards  $\bar{T}$ . Soit  $C_{ij}$  la date de complétion de la tâche  $t_{ij}$ , les fonctions objectif peuvent être définies comme suit.

$$C_{max} = \max_{i \in \{1, \dots, N\}} \{C_{iM}\} \quad (7.3)$$

$$\bar{T} = \sum_{i=1}^N \left\{ \max\{0, C_{iM} - d_i\} \right\} \quad (7.4)$$

**Flowshop multiobjectif stochastique** À notre connaissance, le problème du Flowshop stochastique n'a jamais été étudié de façon multiobjectif. Pourtant, dans un cas réel, dès qu'un historique des temps d'exécution précis et valide des différentes tâches est disponible, il est relativement aisé d'obtenir la distribution de probabilité associée aux données et, ainsi, la loi de probabilité correspondante. Suite à une analyse, nous proposons ici quatre distributions de probabilité générales susceptibles d'être suivies par une durée d'exécution aléatoire. Bien sûr, une analyse statistique rigoureuse, basée sur des données réelles, est impérative pour déterminer la loi de distribution exacte associée à une durée d'exécution pour un problème concret d'ordonnancement. Les lois de probabilités énoncées ci-dessous ont été décidées afin de coller au mieux à la réalité, et donc de fournir une bonne illustration de ce qui peut typiquement être trouvé dans le monde industriel. Une revue du problème du Flowshop monoobjectif avec des durées d'exécutions aléatoires est proposée par [70].

- Loi uniforme : Il se peut qu'une durée d'exécution  $p_{ij}$  soit comprise entre deux valeurs  $a$  et  $b$ , la répartition étant uniforme entre ces deux valeurs.
- Loi exponentielle : Une durée de traitement  $p_{ij}$  est susceptible de suivre une loi exponentielle  $\mathcal{E}(\lambda, a)$ , où  $\lambda$  et  $a$  sont deux paramètres positifs.
- Loi normale : Il s'avère également possible qu'une durée d'exécution  $p_{ij}$  suive une *loi normale*  $\mathcal{N}(\mu, \sigma)$ , où  $\mu$  est l'espérance et  $\sigma^2 > 0$  la variance de  $p_{ij}$ .
- Loi log-normale : La distribution log-normale est couramment utilisée pour modéliser l'influence de variables environnementales incontrôlables. Par exemple, cette modélisation a été utilisée par [121] ainsi que [39].

**Données** Les jeux de données que nous proposons étendent ceux déjà proposés pour le problème de Flowshop multiobjectif<sup>1</sup>. Afin de générer de l'incertitude sur une instance déterministe initiale, quatre distributions de probabilité ont été proposées. L'incertitude portera sur les machines uniquement, en spécifiant, pour chacune d'elles, une distribution de probabilité et ses paramètres ou des proportions liées à sa tendance centrale. Ainsi, à chaque génération de l'incertitude sur une instance déterministe, les durées d'exécutions sont différentes. Nous obtenons alors une réalisation des paramètres environnementaux incertains, correspondant à un scénario, ceci afin de refléter au mieux la réalité du monde industriel.

**Expérimentations** Au cours de nos expérimentations, nous considérons un ensemble de  $(p+q)$  scénarios tout autant plausibles les uns que les autres :  $p$  scénarios pour la phase de recherche, et  $q = 20$  scénarios pour la phase d'évaluation des performances. Deux valeurs de  $p$  sont considérées,  $p = 10$  et  $p = 20$ . Celles-ci correspondent à un échantillon de vecteur objectif de taille 10 (respectivement 20) associé à chaque solution évaluée. A chaque scénario correspond une réalisation des paramètres environnementaux incertains, c'est-à-dire des durées d'exécutions stochastiques pour le cas du problème de Flowshop étudié ici. Les algorithmes que nous avons examinés lors de nos expérimentations correspondent aux huit approches proposées intégrées au sein de l'algorithme IBEA. De plus, nous allons tenir compte également d'une approche naïve qui se base sur un seul et unique scénario (le premier scénario considéré), notée  $z^1$ . La taille de l'échantillon associé à une solution est donc de 1, et l'algorithme résultant se comporte de la même façon que pour le cas déterministe. Nous considérons également l'approche proposée par [15], qui se base sur une estimation de l'espérance mathématique de la  $I$ -valeur associée à une solution pour le cas particulier de l'indicateur  $I_{c+}$ .

### 7.3.5 Résultats et discussion

Les résultats obtenus selon le protocole d'évaluation des performances basé sur une évaluation représentative ne sont pas présentés par soucis de lisibilité mais le lecteur intéressé peut se référer à [111] pour une présentation exhaustive.

Une première remarque cruciale que nous pouvons formuler est que les méthodes prenant en compte l'incertitude se sont avérées globalement plus performantes que la méthode  $z^1$ , pour laquelle un scénario unique est considéré, et ceci pour la quasi-totalité des instances. Les seuls cas où  $z^1$  a obtenu de meilleurs résultats que les autres approches apparaissent lors de la prise en

---

1. Les jeux de données pour le cas déterministe et le cas stochastique sont disponibles à l'URL : <http://www.lifl.fr/~liefooga/benchmarks/>.

compte du meilleur des cas, ceci pour les deux protocoles d'évaluation des performances que nous avons considérés. Preuve que les approches déterministes ne peuvent rivaliser avec les techniques stochastiques, même les plus basiques.

Quant au premier protocole d'évaluation, basé sur des évaluations représentatives, les résultats sont assez variés d'un type de préférence à l'autre, et d'un degré de déviations à l'autre ( $\alpha$ ). Nous pouvons tout de même remarquer les très bonnes performances de l'approche basée sur le vecteur objectif moyen lors de la considération du cas moyen. Globalement, pour une faible incertitude, les approches basées sur les  $I$ -valeurs semblent plus performantes, alors qu'aucune conclusion claire ne peut être dégagée pour un taux d'incertitude plus élevé pour ce protocole d'évaluation (le cas moyen mis à part).

Concernant le protocole d'évaluation basé sur les  $I$ -valeurs, les deux classes de méthodes se comportent relativement aussi bien l'une que l'autre en présence d'incertitude limitée, même si les approches basées sur un vecteur objectif représentatif semblent légèrement plus performantes. Au contraire, dès que l'incertitude augmente ( $\alpha = 20$ ), il apparaît clairement que les approches basées sur les  $I$ -valeurs sont plus performantes.

Enfin, à propos de l'influence de la taille de l'échantillon associé à une solution, il semble que celle-ci soit corrélée au degré d'incertitude considéré. En effet, dans bien des cas, pour une même approche de résolution, un échantillon de taille 10 s'avère généralement plus efficace pour un degré de déviation de 10%, alors qu'un échantillon de taille 20 est plus efficace pour un degré de déviation de 20%.

## 7.4 Conclusion

Dans ce chapitre, nous avons présenté nos premiers travaux sur l'optimisation combinatoire sous incertitude. Après avoir introduit les concepts et les enjeux, nous avons montré l'importance de la définition des critères de robustesse pour la modélisation d'un problème incertain. Nous avons testé de façon expérimentale, sur un problème de tournée de véhicules avec demande stochastique, deux critères de robustesse, l'un basé sur la déviation standard et l'autre sur l'entropie. L'entropie s'est avérée plus efficace pour chercher des solutions robustes. Nous avons également montré l'importance des paramètres et notamment de la taille des échantillons considérés pour résoudre des problèmes multiobjectif incertains. Nous avons également proposé de nouvelles approches méthodologiques dédiées à la résolution très générale de problèmes d'optimisation multiobjectif sous incertitude. Celles-ci se basent sur l'ajout d'un niveau supplémentaire au sein des stratégies d'affectation des valeurs de fitness basées sur un indicateur binaire de qualité. Elles étendent donc l'approche proposée par [189] au cas stochastique en interprétant de diverses façons le degré d'aversion à l'incertitude du décideur ou du praticien. Pour cela, elles se basent sur la prise en compte du meilleur des cas, du pire des cas, du cas moyen et du cas médian.





## Chap. 8

# Méthodes coopératives pour l'optimisation dynamique

### Sommaire

---

<b>8.1</b>	<b>Optimisation dynamique : Généralités . . . . .</b>	<b>106</b>
8.1.1	Définition et contexte . . . . .	106
8.1.2	Classification générale des méthodes . . . . .	106
8.1.3	Mesures de qualité . . . . .	106
<b>8.2</b>	<b>Cadre d'étude : le VRP dynamique . . . . .</b>	<b>107</b>
8.2.1	Le problème statique de tournées de véhicules . . . . .	107
8.2.2	Le problème dynamique de tournées de véhicules . . . . .	107
8.2.3	Intérêts industriels . . . . .	110
<b>8.3</b>	<b>Algorithmes hybrides pour le DVRP . . . . .</b>	<b>110</b>
8.3.1	Représentation et initialisation . . . . .	110
8.3.2	Un algorithme à base d'essaim de particules : APSO . . . . .	110
8.3.3	Un algorithme à voisinage variable : VNS . . . . .	111
8.3.4	Recherche locale basée sur le 2-opt pour l'hybridation . . . . .	113
<b>8.4</b>	<b>Résultats . . . . .</b>	<b>114</b>
8.4.1	Jeux de données . . . . .	114
8.4.2	Paramètres . . . . .	114
8.4.3	Résultats numériques . . . . .	115
<b>8.5</b>	<b>Conclusion . . . . .</b>	<b>116</b>

---

Ce chapitre a donné lieu aux publications suivantes :

- M. Khouadjia, E. Alba, L. Jourdan and E-G. Talbi, *Mutli-Swarm Optimization for Dynamic Combinatorial Problems : a case study on Dynamic Vehicle Routing Problem*, ANTS 2010, LNCS.
- M. Khouadjia, L. Jourdan and E-G. Talbi. *Adaptive Particle Swarm for the Dynamic Vehicle Routing Problem*, AICCSA 2010, IEEE 2010.
- M. Khouadjia, L. Jourdan and E-G. Talbi. *A particle swarm for the resolution of the Dynamic Vehicle Routing*, In International conference on Metaheuristics and Nature Inspired Computing (META'08), 29-31 octobre 2008.

## 8.1 Optimisation dynamique : Généralités

L'optimisation dynamique a été tout d'abord explorée par Golderg et Smith (1987). C'est un domaine qui a relativement peu été étudié et qui reçoit plus d'intérêt pour les problèmes d'optimisation continue (notamment les *moving peaks*). Dans cette partie, nous donnerons les principales définitions et notions essentielles à la compréhension de la problématique.

### 8.1.1 Définition et contexte

Un problème d'optimisation en environnement dynamique est un problème dont la fonction objectif change au cours du temps. De manière informelle, on peut aussi le définir comme un problème où les décisions sont prises à des moments  $t = 1, \dots, T$  avec un horizon  $T$  fini ou infini. Ces problèmes sont fréquents en logistique et en transport où à un instant  $t$  tous les clients ne sont pas toujours connus par exemple.

Pour résoudre ce type de problème, les méthodes d'optimisation doivent être capables de trouver un optimum qui évolue en cours du temps et de le suivre. Deux stratégies sont souvent envisagées pour maintenir cet optimum :

- savoir retrouver très rapidement l'optimum lorsqu'il a été modifié suite à un changement
- suivre effectivement l'optimum lors des changements

### 8.1.2 Classification générale des méthodes

Une première classification des algorithmes évolutionnaires d'optimisation de problèmes dynamiques a été proposée par Branke [84], elle regroupe les méthodes en quatre classes :

- les méthodes qui réagissent au changement. Ces dernières sont utilisées de manière classique et lorsqu'il y a détection d'un changement, des actions extérieures permettent d'augmenter la diversité de la population et facilitent la recherche du nouvel optimum.
- les méthodes qui maintiennent la diversité. Elles permettent de conserver une bonne répartition des individus dans l'espace de recherche et de pouvoir ainsi réagir plus efficacement au changement.
- les méthodes qui utilisent une mémoire permettant de conserver les optima passés.
- Les méthodes qui utilisent plusieurs populations réparties de manière à suivre les optima locaux et à rechercher le nouvel optimum global.

### 8.1.3 Mesures de qualité

Les mesures de qualité pour les problèmes dynamiques ont reçu relativement peu d'attention alors qu'il est relativement difficile de qualifier et quantifier la qualité des résultats obtenus par des algorithmes découvrant les données au cours du temps. Morrison [133] classe les mesures en différentes catégories (certaines s'adaptent surtout au cas continu) :

- mesure qui calcule la différence de performance entre la solution courante et la meilleure solution au problème du pas de temps précédent
- mesure *offline* qui mesure la performance par rapport au meilleur connu à chaque changement de paysage
- mesure de la distance euclidienne d'une solution à l'optimum au temps  $t$
- moyenne en terme de performance des meilleurs individus à chaque pas de l'algorithme

Weicker [182] propose différentes mesures dont les mesures d'*accuracy* (quantifie l'exactitude de la solution, nous garderons le terme anglais plus pertinent que le terme français) et de *stabilité*. L'*accuracy* à l'instant  $t$  pour une fonction à optimiser  $F$  et un algorithme d'optimisation  $A$  est défini comme :

$$accuracy_{F,A}^t = \frac{Min_F^t}{F(best_A^{(t)})} \quad (8.1)$$

où  $best_A^t$  est la meilleure solution de la population au temps  $t$  et  $Min_F^t$  est la meilleure solution connue. L'*accuracy* est donc entre 0 et 1 et lorsque l'*accuracy* vaut 1 l'algorithme a trouvé la meilleure solution possible pour le temps  $t$ .

Comme second objectif, la stabilité est importante en optimisation. Dans le cadre de l'optimisation dynamique, un algorithme est dit stable si des changements dans l'environnement n'ont aucune incidence sévère sur la qualité de l'optimisation. Même en cas de changements drastiques, un algorithme devrait être en mesure de limiter la baisse de qualité. La stabilité à l'instant  $t$  est définie comme :

$$stabilite_{F,A}^t = \max\{0, accuracy(t) - accuracy(t-1)\} \quad (8.2)$$

La stabilité vaut entre 0 et 1. Une valeur proche de 0 indique une grande stabilité.

Nous nous servons de ces mesures pour qualifier la qualité des méthodes développées pour résoudre un problème classique de tournée de véhicules dans un cadre dynamique.

## 8.2 Cadre d'étude : le VRP dynamique

### 8.2.1 Le problème statique de tournées de véhicules

Nous avons brièvement présenté le problème de tournée de véhicules dans le chapitre précédent, section 7.2.1. Nous complétons cette présentation afin de bien pouvoir différencier le problème statique du problème dynamique. Dans le cas d'une planification des tournées de véhicules en mode statique, il est généralement admis que [106] :

- Toute l'information (i.e. données du problème) concernant la planification des routes est supposée connue par le planificateur avant que le processus de planification commence.
- Cette information ne change pas après que les routes aient été construites.

Cette information inclut tous les attributs des clients tels que la position géographique, le temps de service passé sur place, et la demande de chaque client (i.e. quantité à collecter ou à livrer). En plus, les informations concernant les temps de trajets nécessaires pour relier les clients à servir doivent être connues ou calculables par le planificateur. Pour le lecteur, Fisher [55] et Desrosiers & al[47] fournissent une étude étendue sur le VRP.

### 8.2.2 Le problème dynamique de tournées de véhicules

En général, le VRP est résolu de manière statique ; c'est-à-dire en considérant que toutes les informations sont connues à l'avance et quelles sont fiables dans le temps. Cependant en pratique, il est possible qu'un ou plusieurs aléas viennent perturber le routage établi précédemment. Dans ce cas, le problème est traité de façon dynamique. Il se présente alors comme le problème dynamique de tournées de véhicules (DVRP : Dynamic Vehicle Routing Problem). Larsen dans [106] a défini le problème du DVRP comme un problème de VRP statique, mais il se différencie de lui par les spécificités suivantes :

- L'information concernant la planification des tournées n'est pas connue entièrement par le planificateur quand le processus de planification commence. Autrement dit, certaines données du problème dépendent explicitement du temps. Des nouvelles commandes peuvent arriver au cours de la journée de service.
- L'information peut changer après que les tournées initiales aient été construites.

Il est évident que le problème dynamique de tournées de véhicules est plus complexe que le problème statique. Si la classe des problèmes du VRP est dénotée  $P(\text{VRP})$  et que la classe des problèmes du DVRP est dénotée  $P(\text{DVRP})$ , alors  $P(\text{VRP}) \subset P(\text{DVRP})$ . Etant donné que le problème statique du VRP est connu comme un problème NP-complet, à savoir, qu'il n'est pas possible de trouver une solution optimale en un temps de calcul raisonnable. Par conséquent, le problème dynamique de tournées de véhicules DVRP, appartient également à la classe des problèmes NP-complets. Il se résume en la résolution à chaque pas de temps d'un VRP statique. Dans la figure 8.1, nous avons un exemple simple du problème VRP dynamique, où deux véhicules doivent servir un ensemble de clients. Les clients statiques sont représentés par des noeuds noirs, alors que les clients dynamiques (i.e. nouveaux clients) sont représentés par les noeuds blancs. Les arcs en trait fin représentent la route planifiée pour chaque véhicule. Les deux arcs en trait plein épais indiquent les trajets en cours des véhicules lorsque les demandes dynamiques sont reçues. Les nouveaux clients doivent être insérés dans les routes déjà planifiées en tenant en compte les autres clients. Les nouveaux segments de routes sont indiqués en trait tiré.

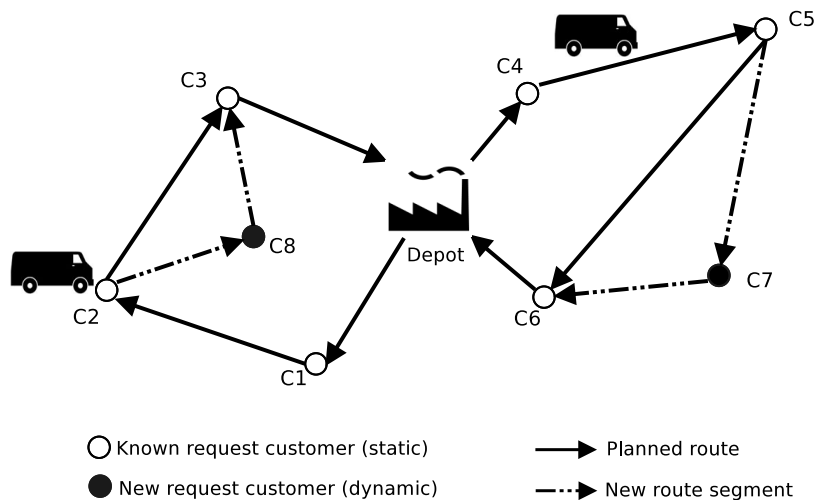


Fig. 8.1 – Problème dynamique de tournées de véhicules (DVRP)

Nous exposerons dans ce qui suit, ce qui différencie le problème statique de tournées de véhicules de sa déclinaison dynamique. Psaraftis liste plusieurs points sur lesquels reposent ces différences [146] :

- **La dimension de temps**

Dans un problème de tournées statiques, la dimension de temps peut avoir ou non de l'importance. Dans le problème dynamique opposé, cette notion est essentielle. Le planificateur doit avoir un minimum de connaissances sur la position de tous les véhicules à chaque instant, et particulièrement lorsqu'une nouvelle demande de service est reçue.

- **Le problème peut être non borné**

Le processus est souvent temporellement borné dans un problème statique. Les tournées sont pré-planifiées avant la journée de service. Dans la version dynamique du problème, le processus peut très bien être non borné, avec par exemple des demandes dynamiques non servies au cours de la journée de service et qui sont reconduites à la journée suivante comme demandes statiques.

– **L'information ultérieure peut être imprécise ou non connue**

Dans un problème statique toutes les informations sont supposées être connues. Dans un problème de tournées dynamiques réel, le futur n'est jamais connu avec certitude. Ceci amène à une situation où une planification *a priori* est incomplète.

– **Les événements à court terme sont plus importants**

Tous les événements ont le même poids dans un problème statique ; en raison de l'uniformité de la qualité de l'information, et du peu de mises à jour sur ces informations. Dans un cadre dynamique, il serait imprudent d'envisager une planification à long terme. Quand un problème dynamique est traité, le planificateur doit faire une planification qui devra mettre l'accent sur des événements à court terme.

– **Des mécanismes de mise à jour de l'information sont essentiels**

Les entrées d'un problème d'acheminement dynamique sont soumises aux changements durant les opérations journalières. Par conséquent, il est essentiel que des mécanismes de mise à jour de l'information soient intégrés dans la méthode de résolution.

– **Le re-séquencement et les décisions de re-assignement sont nécessaires**

Dans le cheminement dynamique, les nouvelles entrées peuvent amener à ce que les décisions prises par le planificateur deviennent sous-optimales. Ceci force le planificateur à rediriger ou même à réassigner les véhicules afin de répondre à la nouvelle situation.

– **Des temps de calcul plus rapides sont nécessaires**

Dans des configuration statiques, le planificateur peut se permettre d'attendre quelques heures afin d'obtenir une solution de bonne qualité, parfois même optimale. Dans des configurations dynamiques ceci n'est pas possible ; le planificateur souhaite connaître la solution du problème actuel aussitôt que possible (de préférence dans des minutes ou des secondes). La contrainte sur la durée des opérations implique que les redirections et les réassignements sont souvent effectués en utilisant une heuristique d'amélioration locale telle que l'insertion.

– **Des demandes peuvent être reportées**

La raison de ce report est le fait que certaines demandes ont des caractéristiques géographiquement défavorables par rapport aux autres demandes.

– **La qualité de service peut être inférieure aux attentes**

Dans les configurations statiques, l'écart de temps entre l'exécution de l'algorithme et l'accomplissement des itinéraires permet généralement des rajustements dans la flotte de véhicules. Cependant, dans un cadre dynamique, les véhicules sont déjà en route alors que les nouvelles demandes arrivent. Il en résulte, que des perturbations dans les temps de services peuvent se produire. Ainsi, quelques clients trouveront la qualité de service inférieure à leurs attentes.

– **La file d'attente peut devenir importante**

Si le taux de demandes des client excède un certain seuil, le système deviendra saturé et les algorithmes peuvent produire des résultats vides de sens.

### 8.2.3 Intérêts industriels

Il existe plusieurs problèmes qui doivent être résolus en temps réel. Ainsi dans le domaine du VRP en temps réel [66], on peut citer quelques exemples d'applications : la gestion dynamique de la flotte, la gestion de réseaux de grande distribution, gestion du courrier longue distance, gestion de sociétés de dépannage et de réparation, gestion de systèmes de transport à la demande (Dial-a-ride systems), gestion de services de secours, gestion de services de Taxis, etc.

## 8.3 Algorithmes hybrides pour le DVRP

Dans cette section, nous présentons les algorithmes développés pour résoudre le DVRP : APSO basé sur une métaheuristique à essaim de particules et VNS une méthode à voisinage. Ces deux métaheuristicues ont une représentation identique que nous décrirons avant de présenter les algorithmes en détail.

### 8.3.1 Représentation et initialisation

La représentation d'une solution est adaptée au caractère dynamique du problème et notamment à la possibilité d'ajouter des clients dans des tournées déjà planifiées. Pour chaque client connu,  $C_i$ , on connaît son état (servi/non servi) et la date de service. Pour chaque véhicule, sont connues sa position courante, sa date de départ du dépôt, sa capacité restante et la distance déjà parcourue. La représentation peut être résumée ainsi :

$$R_i : (C_1, S_1, Ts_1) - \dots - (C_n, S_n, Ts_n) \quad (8.3)$$

Chaque route est servie par un véhicule  $V_i$ , tel que :

$$V_i : ((X_{v_i}, Y_{v_i}), Cap_{v_i}, Dist_{v_i}, Tdep_{v_i}) \quad (8.4)$$

Chaque route  $R_i$  est une liste de clients, commençant et se terminant au dépôt. Pour chaque client  $C_i$ , nous avons l'information suivante :

- $S_i$  : une variable booléenne qui indique que le client a été servi ou non.
- $Ts_i$  : La date à laquelle le client  $C_i$  a été servi.

De plus, pour chaque véhicule  $V_i$  :

- $(X_{v_i}, Y_{v_i})$  : Les coordonnées de chaque véhicule.
- $Cap_{v_i}$  : La capacité restante du véhicule.
- $Dist_{v_i}$  : la distance parcourue par le véhicule.
- $Tdep_{v_i}$  La date de départ du dépôt.

L'initialisation des solutions se fait aléatoirement en générant une permutation aléatoire de clients à traiter.

### 8.3.2 Un algorithme à base d'essaim de particules : APSO

La métaheuristique à essaim de particules (PSO : particle swarm optimization) proposée par Kennedy a été pensée au départ pour traiter les problèmes continus [93, 94]. Elle a également été utilisée pour résoudre des problèmes dynamiques mais essentiellement continus [22, 81, 109, 110]. Une particule de l'essaim est définie par sa position  $\vec{x}_i$ , la position de sa meilleure solution visitée

$\vec{p}_i$ , et sa vitesse  $\vec{v}_i$ . De plus, chaque particule connaît la meilleure solution découverte par une de ses voisins  $\vec{p}_g$ . Différentes topologies de voisinage sont décrites et explorées dans [94], mais le voisinage utilisé le plus couramment est le voisinage global. L'algorithme PSO est itératif et recalcule toutes les vitesses puis toutes les positions en fonction des formules suivantes :

$$\vec{v}_i = \omega \vec{v}_i + c_1 \times r_1 (\vec{p}_g - \vec{x}_i) + c_2 \times r_2 (\vec{p}_i - \vec{x}_i) \quad (8.5)$$

$$\vec{x}_i = \vec{x}_i + \vec{v}_i \quad (8.6)$$

$\omega$  est le coefficient d'inertie et régule la balance entre exploitation et exploration. Les paramètres  $c_1$  et  $c_2$  contrôlent respectivement l'attraction à son meilleur et l'attraction au meilleur global. Enfin,  $r_1$  et  $r_2$  sont des variables aléatoires entre  $[0, 1]$ .

### Une mémoire adaptative

Comme nous l'avons vu dans le bref positionnement de l'optimisation dynamique, il faut être capable de suivre l'optimum. Dans le DVRP, les changements d'environnement amènent l'optimum dans un voisinage des optimum précédents. Afin d'éviter que les particules ne s'éloignent trop des meilleures solutions, un mécanisme de mémoire a été introduit afin de guider les particules éloignées vers de meilleures solutions. Lorsqu'un changement se produit, ce mécanisme permet de redémarrer la recherche à partir des meilleures solutions rencontrées précédemment (à  $t - 1$ ). Ainsi des éléments de ces solutions sont extraits afin de créer de nouvelles solutions intéressantes. Ce concept a été introduit par Rochat et Taillard in [154] puis utilisé pour le DVRP dans une recherche tabou dans [64].

### Algorithme

Le pseudo-code de l'algorithme APSO est présenté figure 4. A chaque pas de temps  $t_{step}$ , la position courante de chaque particule est initialisée à partir des solutions de la mémoire adaptative. Pour chaque particule, les nouveaux clients sont insérés aléatoirement à partir de la position courante du véhicule concerné.

### 8.3.3 Un algorithme à voisinage variable : VNS

La recherche locale à voisinage variable (*VNS Variable Neighborhood Search*) est une S-META qui a été proposée par Hansen et Mladenović [75]. VNS est basée sur le principe de changement systématique de voisinage afin d'échapper à l'optimum local.

Dans un algorithme de type VNS, une succession de voisinage est définie  $\mathcal{N}_1(s), \mathcal{N}_2(s), \dots, \mathcal{N}_n(s)$ , autour d'une solution  $s$  de l'espace de recherche. A chaque itération, une solution  $s'$  est prise du voisinage courant  $\mathcal{N}_k(s)$  et est améliorée par une procédure de recherche locale afin d'obtenir une nouvelle solution  $s''$  qui est un minimum local. Si  $f(s'') < f(s)$ ,  $s''$  est alors choisie comme nouvelle solution courante et la recherche continue avec le voisinage courant  $\mathcal{N}_1(s'')$ ; sinon, la recherche utilise pour la prochaine itération le voisinage suivant  $\mathcal{N}_{k+1}$ .



---

**Algorithm 4** Pseudo-Code d'APSO pour le DVRP
 

---

**INPUT** : VRP instance that corresponds to the set of customers arrived during the last time slice  $T_s$ .

//Reuse the best solutions found by the swarm

**if** *AdaptiveMemory*  $\neq \emptyset$  **then**

**for** Each particle  $i$  **do**

    //Initialize the current position with the solution found in the memory

$\vec{x}_i := \vec{s}_m$ ;

**end for**

**end if**

//PSO Main Algorithm

**for** Each particle  $i$  **do**

  //Insert randomly the new customer requests in the current position

$\vec{x}_i := \text{InsertRandomly}(\vec{x}_i, \text{New customer requests})$

  //Initialize the best position

$\vec{p}_i := \vec{x}_i$ ;

  Evaluate  $f(\vec{p}_i)$

**end for**

**repeat**

**for** Each particle  $i$  **do**

    Update particle position  $\vec{x}_i$  according to Eqs (8.5) and (8.6).

    Evaluate  $f(\vec{x}_i)$

    //Update the Adaptive Memory

    //Update personal best

**if**  $f(\vec{x}_i) \leq f(\vec{p}_i)$  **then**

$\vec{p}_i := \vec{x}_i$ ;

**end if**

    //Update global best

**if**  $f(\vec{x}_i) \leq f(\vec{p}_g)$  **then**

$\vec{p}_g := \vec{x}_i$ ;

**end if**

**end for**

**until** termination criterion reached

---

Pour le problème du DVRP, les voisinages et la recherche locale sont adaptés au problème. Deux voisinage sont proposés

1.  $\mathcal{N}_1(s)$  est l'ensemble de voisins obtenus en échangeant (*swap*) deux clients de la solution  $s$ .
2.  $\mathcal{N}_2(s)$  est l'ensemble des voisins obtenus en réalisant un  $\lambda$ -échange [136], qui sélectionne aléatoirement un client et l'insère sur une autre route de la solution.

Un mécanisme de réparation est effectué avant chaque évaluation si la solution ne respecte pas les contraintes. L'algorithme VNS adapté au DVRP est expliqué figure 5.

---

**Algorithme 5** Pseudo-code du VNS pour le DVRP.

---

```

INPUT : VRP instance which corresponds to the set of customers who are known at  $T_s$ 
if  $T_s < 0$  then {no dynamic orders appeared yet}
   $s := \text{generateRandomInitialSolution}()$ 
else {potential dynamic orders waiting to be scheduled}
   $s := \text{getLastTimeSliceSolution}()$ 
  for all customer  $c_i \notin s$  do
     $s := \text{doBestInsertion}(c_i, s)$ 
  end for
end if
while termination conditions not met do
   $k := 1$ 
  while  $k < k_{max}$  do
    // Select one solution from the current neighborhood
     $s' := \text{pickAtRandom}(N_k(s))$ 
    // Apply local search procedures
    for all local search heuristic  $ls_j$  do
       $s'' := \text{apply}(ls_j, s')$ 
    end for
    // Update current solution and/or neighborhood
    if  $f(s'') < f(s)$  then
       $s := s''$ 
       $k := 1$ 
    else { $n$  is the number of neighborhoods}
       $k := (k + 1) \bmod n$ 
    end if
  end while
end while

```

---

### 8.3.4 Recherche locale basée sur le 2-opt pour l'hybridation

Comme nous l'avons vu précédemment dans le cadre déterministe, l'utilisation de méthodes hybrides peut améliorer significativement la qualité des résultats obtenus. Nous nous plaçons dans un type d'hybridation de bas niveau relai (LRH) en hybridant l'heuristique classique de 2-Opt [118] comme dans une recherche locale pour l'algorithme APSO ainsi que pour l'algorithme VNS. 2-Opt inverse une sous-route d'une route  $R$  en sélectionnant deux arcs  $a = [i_1, i_2]$  et  $b = [j_1, j_2]$  et en les substituant par  $a = [i_1, j_2]$  and  $b = [j_1, i_2]$  (voir l'exemple figure 8.2).

Pour APSO, l'heuristique 2-Opt est appliquée après le mouvements des particules. Pour le VNS, elle est appliquée à chaque production d'une nouvelle solution voisine.

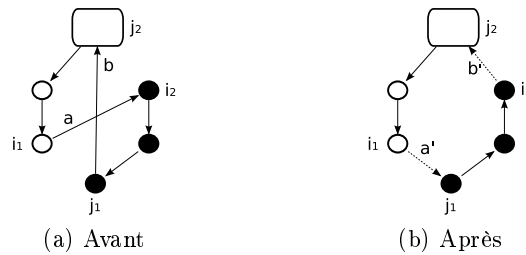


Fig. 8.2 – Exemple de l'application de l'heuristique 2-Opt aux arcs  $a$  et  $b$  sur une route unique.

De plus, VNS a été amélioré en utilisant une recherche locale basée sur 2-Opt\* [145] qui sélectionne deux arcs  $a = [i_1, i_2] \in R_\alpha$  et  $b = [j_1, j_2] \in R_\beta$  et construit deux nouveaux arcs tels que  $a = [i_1, j_2]$  et  $b = [j_1, i_2]$  (voir figure 8.3). L'objectif de cette heuristique est de permettre d'échanger des routes.

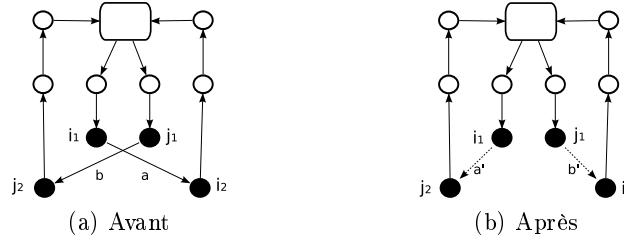


Fig. 8.3 – Exemple de l'application de l'heuristique 2-Opt\* sur les arcs  $a$  et  $b$  appartenant à deux routes différentes.

## 8.4 Résultats

### 8.4.1 Jeux de données

Nous proposons un ensemble de nouveaux jeux de données (*k-series*). L'objectif de ces jeux de données est de standardiser les instances du DVRP et aussi de proposer des instances plus difficiles à résoudre notamment que celles de Kilby [95] (21 instances avec un nombre de clients variant de 50 à 385). Les instances ont été créées par un générateur simulant des données réalistes.<sup>1</sup> Les journées de travail sont découpées en tranche, chaque tranche correspondant à l'arrivée de nouveaux clients. Le degré de dynamisme, fixant la proportion entre les clients déjà connus et les nouveaux clients, est fixé à 0.5.

### 8.4.2 Paramètres

Les paramètres sont :

---

1. disponible sur :  
<http://dolphin.lille.inria.fr>  
<http://neo.lcc.uma.es>

- critère d'arrêt : nombre d'évaluations maximales égale à la moitié de ce qui est nécessaire à APSO dans le cas déterministe (voir tableau 8.1).
- APSO : population initiale 100. Les valeurs des paramètres  $\omega$ ,  $c_1$ ,  $c_2$  ont été fixés expérimentalement (voir tableau 8.2).

TABLE 8.1 – Critère d'arrêt : le nombre d'évaluation ( $n_{ts} = 25$ )

Instances	Déterministe	Dynamique par tranche de temps	Dynamique sur une journée
<i>k100</i>	1000	500	$25 \times 500 = 12500$
<i>k250</i>	1100	550	$25 \times 550 = 13750$
<i>k500</i>	2300	1150	$25 \times 1150 = 28750$

TABLE 8.2 – Valeurs des paramètres d'APSO pour chaque instance traitée

Instances	Paramètres		
	$\omega$	$c_1$	$c_2$
<i>k100</i>	1.0	0.5	0.5
<i>k250</i>	1.0	0.5	0.5
<i>k500</i>	1.0	0.7	0.3

### 8.4.3 Résultats numériques

Les algorithmes ont été comparés aussi bien dans le cas déterministe que dans le cas dynamique. Le tableau 8.3 synthétise les résultats obtenus. L'algorithme APSO obtient des meilleurs résultats que l'algorithme VNS de base sur les instances *k100* et *k250* mais on ne mesure aucune différence significative sur *k500*. Lorsque les algorithmes sont hybridés, on peut remarquer que l'apport est plus important pour des grandes instances. VNS profite d'avantage de cette hybridation avec *2-Opt*. Les tableaux 8.4 et 8.5 montrent les mesure d'*accuracy* et de stabilité pour trois instances des *k-series* par tranche de temps et pour la journée complète. Ces résultats sont synthétisés sur la figure 8.5. On peut remarquer que les algorithmes se comportent de façon très différente sur les différentes instances. Lorsque l'on observe le critère d'*accuracy*, on remarque encore que le comportement des algorithmes dépend de la taille des instances ainsi pour *k100* APSO a une *accuracy* plus élevée (avec ou sans hybridation) alors que sur *k500*, APSO<sub>2-Opt</sub> est compétitif en début de recherche mais VNS d'adapte plus rapidement aux changements. Au niveau de la stabilité, VNS est en général moins stable que APSO, la différence est plus significative sur les petites instances.

TABLE 8.3 – Comparaison statistique des résultats obtenus.

Dynamisme	Instance	Algorithm	Résultats statistiques			
			<i>APSO</i>	<i>APSO</i> <sub>2-Opt</sub>	<i>VNS</i>	<i>VNS</i> <sub>2-Opt</sub>
Static	<i>k100</i>	<i>APSO</i>	–	–	+	–
		<i>APSO</i> <sub>2-Opt</sub>	–	–	+	–
		<i>VNS</i>	+	+	–	+
		<i>VNS</i> <sub>2-Opt</sub>	–	–	+	–
	<i>k250</i>	<i>APSO</i>	–	–	+	–
		<i>APSO</i> <sub>2-Opt</sub>	–	–	+	–
		<i>VNS</i>	+	+	–	+
		<i>VNS</i> <sub>2-Opt</sub>	–	–	+	–
	<i>k500</i>	<i>APSO</i>	–	+	+	+
		<i>APSO</i> <sub>2-Opt</sub>	+	–	+	–
		<i>VNS</i>	+	+	–	+
		<i>VNS</i> <sub>2-Opt</sub>	+	–	+	–
Dynamic	<i>k100</i>	<i>APSO</i>	–	–	+	+
		<i>APSO</i> <sub>2-Opt</sub>	–	–	+	+
		<i>VNS</i>	+	+	–	–
		<i>VNS</i> <sub>2-Opt</sub>	+	+	–	–
	<i>k250</i>	<i>APSO</i>	–	–	+	+
		<i>APSO</i> <sub>2-Opt</sub>	–	–	+	+
		<i>VNS</i>	+	+	–	+
		<i>VNS</i> <sub>2-Opt</sub>	+	+	+	–
	<i>k500</i>	<i>APSO</i>	–	+	–	+
		<i>APSO</i> <sub>2-Opt</sub>	+	–	+	+
		<i>VNS</i>	–	+	–	+
		<i>VNS</i> <sub>2-Opt</sub>	+	+	+	–

## 8.5 Conclusion

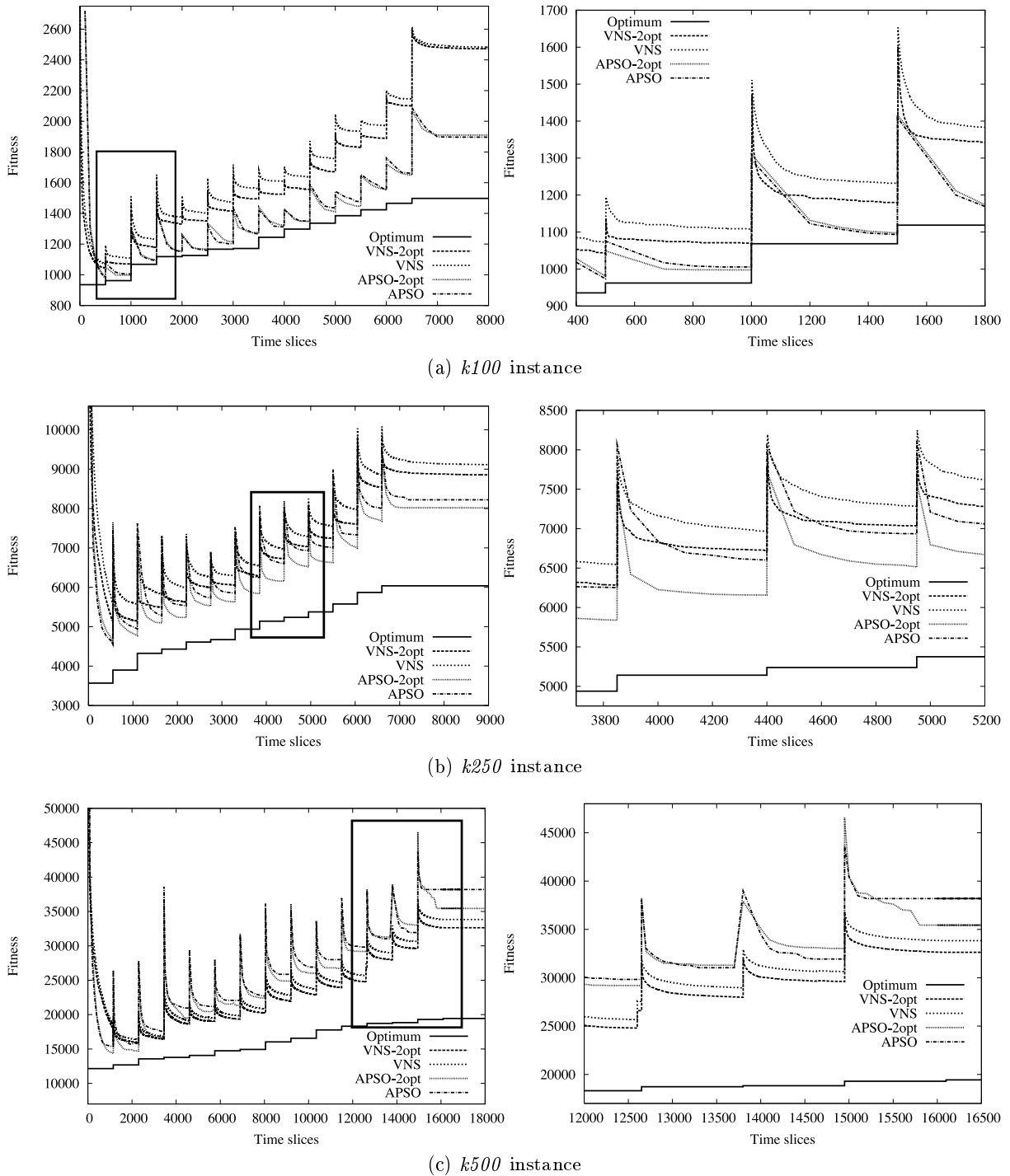
Dans ce chapitre, nous avons présenté le problème de routage de véhicule dynamique et deux algorithmes pour le résoudre. Nous avons montré qu'une hybridation avec une heuristique classique, le 2-opt, permet d'améliorer significativement les résultats obtenus aussi bien dans le contexte déterministe que dans le cadre dynamique. Les questions à se poser pour l'hybridation des métaheuristiques dans l'environnement dynamique sont sensiblement les mêmes qu'en environnement déterministe. Les mêmes schémas d'hybridation que ceux présentés dans la première partie peuvent être mis en oeuvre. Une des perspectives de ce chapitre est donc de continuer les travaux de la première partie et de mettre en œuvre des schémas de coopération plus complexes notamment entre *APSO* et *VNS*. Le mode co-évolonnaire est plus compliqué car il nécessite une synchronisation parfaite des environnements. Nous sommes en train d'améliorer *APSO* en utilisant le principe des *multiswarms* qui évite la convergence prématurée des algorithmes à essaim de particules. Enfin, un des aspects importants de l'optimisation dynamique, est la mesure de la performance. Comme indiqué dans la première partie de ce chapitre, peu de travaux proposent de mesures adéquates pour l'évaluation de performance et il serait intéressant de travailler sur cet aspect.

TABLE 8.4 – Accuracy de APSO et VNS sur les instances dynamiques sur les différentes tranches de temps.

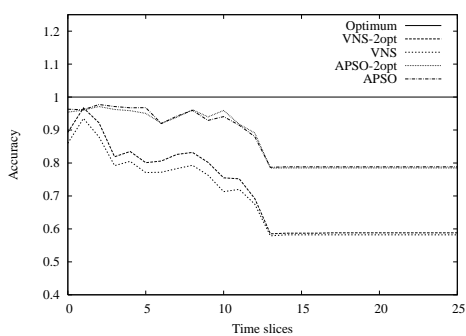
Instance	$T_{step}$	Accuracy			
		APSO		VNS	
		<i>Canonical</i>	<i>2-Opt</i>	<i>Canonical</i>	<i>2-Opt</i>
<i>k100</i>	0	<b>0.961</b>	0.954	0.845	0.867
	5	<b>0.970</b>	0.960	0.805	0.835
	10	0.939	<b>0.957</b>	0.763	0.802
	15	<b>0.790</b>	0.784	0.582	0.587
	20	<b>0.790</b>	0.784	0.582	0.588
	25	<b>0.790</b>	0.784	0.582	0.588
	Avg	<b>0.867</b>	0.865	0.686	0.705
<i>k250</i>	0	0.785	<b>0.787</b>	0.620	0.703
	5	0.798	<b>0.830</b>	0.733	0.763
	10	0.761	<b>0.797</b>	0.720	0.756
	15	0.734	<b>0.753</b>	0.619	0.668
	20	0.734	<b>0.753</b>	0.618	0.669
	25	0.734	<b>0.753</b>	0.618	0.669
	Avg	0.756	<b>0.783</b>	0.673	0.720
<i>k500</i>	0	0.792	<b>0.846</b>	0.585	0.659
	5	0.668	0.690	0.680	<b>0.758</b>
	10	0.614	0.627	0.658	<b>0.760</b>
	15	0.509	0.548	0.543	<b>0.632</b>
	20	0.509	0.548	0.543	<b>0.632</b>
	25	0.509	0.548	0.543	<b>0.632</b>
	Avg	0.589	0.620	0.607	<b>0.698</b>

TABLE 8.5 – Stabilité de APSO et VNS sur les instances dynamiques au cours des différentes tranches de temps.

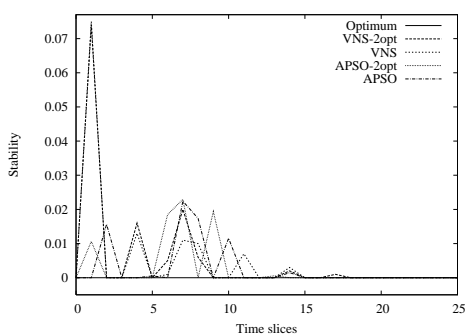
Instance	$T_{step}$	Stabilité			
		APSO		VNS	
		<i>Canonical</i>	<i>2-Opt</i>	<i>Canonical</i>	<i>2-Opt</i>
<i>k100</i>	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	5	<b>0</b>	<b>0</b>	0.013	0.016
	10	0.008	0.010	<b>0</b>	<b>0</b>
	15	<b>0</b>	<b>0</b>	0.003	0.002
	20	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	25	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Avg	<b>0.003</b>	<b>0.003</b>	0.005	0.005
<i>k250</i>	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	5	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	10	<b>0</b>	<b>0</b>	<b>0</b>	0.006
	15	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	20	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	25	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Avg	<b>0.002</b>	0.003	<b>0.002</b>	<b>0.002</b>
<i>k500</i>	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	5	<b>0.003</b>	0.012	0.021	0.023
	10	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	15	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	20	<b>0</b>	<b>0</b>	<b>0</b>	0.001
	25	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Avg	<b>0.002</b>	<b>0.002</b>	0.003	0.004



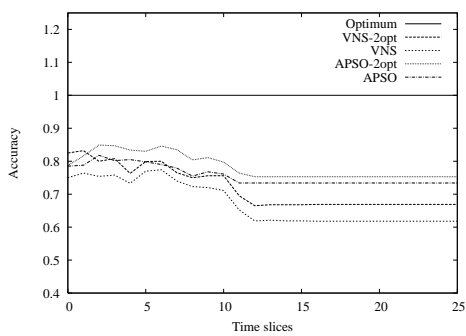
**Fig. 8.4** – Évolution de la moyenne de chaque algorithme pour chaque instance présentée ; chaque figure montre la valeur optimum de la tranche de temps sur le problème déterministe.



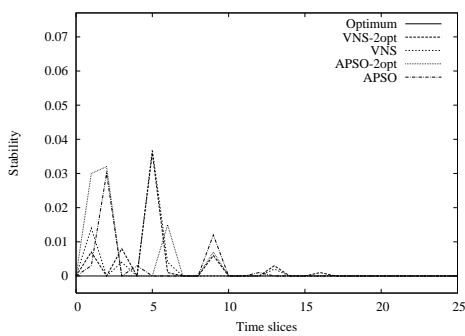
(a) Accuracy over 25 time slices for  $k100$ .



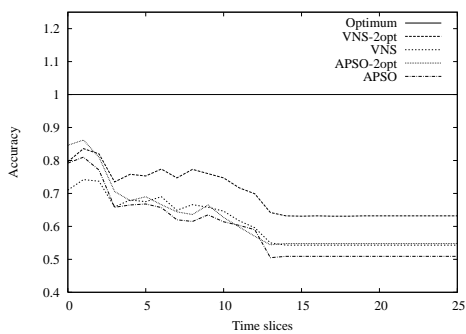
(b) Stability over 25 time slices for  $k100$ .



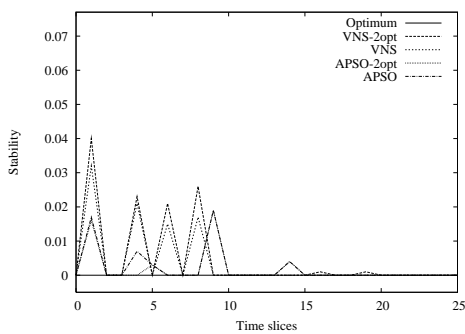
(c) Accuracy over 25 time slices for  $k250$ .



(d) Stability over 25 time slices for  $k250$ .



(e) Accuracy over 25 time slices for  $k500$ .



(f) Stability over 25 time slices for  $k500$ .

**Fig. 8.5** – Evolution de l'accuracy et la stabilité pour chaque instance.





Quatrième partie

Conclusion Générale et Perspectives



# Conclusion générale

Dans ce document, nous avons montré nos contributions à la résolution de problèmes d'optimisation combinatoire en environnements déterministe et stochastique. Les contributions se situent à différents niveaux.

Nous avons tout d'abord présenté nos contributions en terme de métaheuristiques, en proposant une vue unifiée de la conception de métaheuristiques à solution unique et de métaheuristiques pour l'optimisation multiobjectif. Nous avons identifié les concepts communs partagés par un grand nombre d'approches, en délimitant les composants spécifiques au problème traité de la partie invariante impliquée dans cette classe de méthodes de résolution. Cette étude a permis de retravailler la plateforme ParadisEO et notamment les modules ParadisEO-MO et ParadisEO-MOEO afin d'offrir plus de flexibilité et de polyvalence. Dans la partie optimisation déterministe du manuscrit, nous avons tout d'abord montré que la taxonomie de coopération entre métaheuristiques introduite dans [164], s'applique à la coopération entre différentes classes de méthodes et les métaheuristiques. Ainsi, la taxonomie de la coopération entre métaheuristiques et méthodes exactes peut se décrire selon le même schéma et nous avons proposé une extension qui permet de qualifier le type de coopération obtenu, c'est à-dire, si la méthode résultant de la coopération offre une résolution approchée ou une résolution exacte du problème d'optimisation attaqué. Sur ce domaine, nous avons également proposé un survol de la littérature permettant de mieux appréhender les possibilités offertes par ce type de coopération. Nous avons également montré que l'on pouvait décrire grâce à la taxonomie générale les coopérations entre des métaheuristiques et les méthodes d'extraction de connaissances.

Au niveau algorithmique, différentes méthodes coopératives ont été produites. Nous avons tout d'abord proposé une approche coopérative composée d'un algorithme génétique multiobjectif et d'une recherche monoobjectif afin d'intensifier la recherche sur un critère donné.

Ensuite, deux approches coopératives multiobjectif en mode relais, *PCS* et *ACS*, hybridant toutes deux des métaheuristiques de haut niveau, ont été proposées. Dans ces approches, un algorithme évolutionnaire et un algorithme de recherche locale sont exécutés séquentiellement. La première approche, *PCS* opère, à chaque étape de l'algorithme principal, une transition systématique depuis l'algorithme évolutionnaire vers l'algorithme de recherche locale. Au sein de la deuxième variante, *ACS*, cette même transition a lieu de façon adaptative, uniquement si une condition liée à la convergence de l'algorithme est vérifiée. Au cours de nos expérimentations, nous avons montré l'intérêt de ce type d'hybridation, même si la condition de transition adaptative ne s'est pas révélée plus performante que l'approche plus naïve, probablement en raison du temps de calcul plus élevé qu'elle nécessite.

Nous avons également introduit *LEMMO* qui est le premier algorithme tirant parti de l'hybridation entre des métaheuristiques et un algorithme d'extraction de règles d'association pour

l'optimisation multiobjectif. L'hybridation, testée sur une application en hydrologie, a permis une amélioration par rapport aux algorithmes classiques multiobjectif, sur deux plans : la vitesse de convergence et la qualité des résultats obtenus.

Enfin, nous avons montré que les méthodes coopératives pouvaient résoudre des problèmes d'optimisation dans différents domaines en améliorant significativement les résultats obtenus et nous avons montré quelques exemples en protéomique, en logistique et en transport. Par chaque problème, nous avons proposé une modélisation en problème d'optimisation et une métaheuristique coopérative pour le résoudre.

Dans la partie sur l'optimisation dans un contexte stochastique, nous nous sommes intéressés à deux types de problèmes d'optimisation stochastique : les problèmes sous incertitude et les problèmes dynamiques.

En optimisation sous incertitude, nous nous sommes intéressés à différentes problématiques : comment modéliser l'incertitude dans un problème d'optimisation combinatoire, comment évaluer la qualité des solutions et comment réaliser des algorithmes multiobjectif spécifiques à des problèmes d'optimisation combinatoire sous incertitude.

Pour modéliser la robustesse, nous avons proposé deux critères l'un basé sur la déviation standard et l'autre basé sur la notion d'entropie. Expérimentalement, nous avons montré que la modélisation de la robustesse influait sur les résultats obtenus. Dans ce manuscrit, nous avons traité l'incertitude à l'aide d'un échantillonnage nous avons mis en avant que le concept de base se traduisait par la comparaison d'ensembles, depuis l'espace décisionnel vers l'espace objectif. Pour cela, différentes approches de résolution ont été proposées. Un certain nombre d'entre elles se basent sur la transformation de l'ensemble de vecteurs objectif associé à une solution en un point unique. D'autres opèrent cette transformation à un niveau plus fin, ceci au sein d'une stratégie d'affectation des valeurs de fitness basée sur un indicateur de qualité. Pour chacune de ces deux classes, diverses techniques proposent de prendre position en fonction du meilleur des cas, du pire des cas, du cas moyen et du cas médian.

En optimisation dynamique, nous avons proposé pour le problème de routage de véhicule dynamique deux métaheuristiques adaptées au problème par le codage de solution et les opérateurs mis en œuvre. La première, *APSO*, est un algorithme basé sur les essaims de particules augmenté d'une mémoire adaptative. La seconde est un algorithme basé sur la recherche par voisinage variable. Ces deux métaheuristiques ont été hybridées avec une heuristique classique pour le VRP, le 2-opt et une variante le 2-opt\*. Cette hybridation s'est montrée intéressante pour les deux métaheuristiques en améliorant deux critères classiques permettant de qualifier la qualité des solutions obtenues : l'*accuracy* et la stabilité.

# Perspectives

Les perspectives de ce travail sont nombreuses. La combinaison de métaheuristiques fournit généralement des algorithmes de recherche de haute performance pour de nombreuses classes de problèmes académiques et industriels. Malheureusement, la multiplication de mécanismes évolués a tendance à complexifier la démarche de conception et à augmenter significativement le nombre de paramètres, dont le réglage s'avère souvent délicat. En ce sens, une perspective intéressante serait de définir de façon adaptative les mécanismes de coopération afin de sélectionner dynamiquement la méthode la plus adaptée selon tel ou tel critère de convergence, ou de diversité. Ce type de coopération est en partie réalisé au sein de l'approche auto-adaptative *ACS*, mais nécessite d'être généralisée. Nous pensons notamment que l'on pourrait combiner analyse de paysage et méthodes auto-adaptatives. Ainsi nous avons commencé à nous intéresser dans la thèse de Marie-Eléonore Marmion à l'analyse de paysage de problèmes mono et multiobjectif. Nous travaillons actuellement à définir l'analyse de paysage pour des algorithmes à population, et pour des algorithmes multiobjectif. L'objectif est de pouvoir créer de nouvelles méthodes qui en fonction de l'exploration de l'espace de recherche effectuée se combinerait de façon automatique. Concernant la combinaison entre des métaheuristiques et des algorithmes d'extraction de connaissances, nous pensons que ce domaine est encore peu exploré et possède énormément de potentiel. A court terme, il serait intéressant de voir si *LEMMO* est intéressant sur d'autres problématiques, notamment dans des problèmes où la fonction d'évaluation est coûteuse. Ainsi nous pensons que *LEMMO* serait intéressant pour des problèmes comme le docking moléculaire où l'algorithme permettrait de trouver les parties des molécules qui sont intéressantes dans une certaine conformation. Pour certains problèmes tel que le transport et la logistique, nous pensons que les composants communs à identifier sont plus des séquences/des suites dans la composition de la solution, il serait donc intéressant de pouvoir réaliser des hybridations non plus avec des règles d'association mais des hybridations entre métaheuristiques et des méthodes de type analyse de séries comme l'algorithme foil [148].

Au niveau de l'optimisation sous incertitude, de nombreuses questions restent ouvertes concernant la modélisation, la résolution et l'analyse des performances. Les métaheuristiques semblent être un choix pertinent pour lequel, avec un effort relatif, il nous est possible de résoudre des problèmes incertains dont les formulations stochastiques sont très variées mais lors de nos expérimentations, nous avons pu constater que la prise en compte de l'incertitude s'avérait très coûteuse en temps de calcul, du fait de l'évaluation multiple des solutions et de l'augmentation de la cardinalité des ensembles à comparer. Il s'avèrerait donc important d'étudier le potentiel du calcul parallèle pour la résolution de tels problèmes, en particulier pour les applications réelles où l'étape d'évaluation requiert déjà en elle-même d'importantes ressources calculatoires. De plus,

les coopérations se sont révélées très efficaces en optimisation déterministe, et il serait intéressant pour améliorer la qualité des résultats de regarder le potentiel des métaheuristiques coopératives.

Le travail en optimisation dynamique offre lui aussi de nombreuses perspectives à plus ou moins long terme. La méthode *APSO* se montre très performante sur de petits jeux de données mais sur de plus gros jeux de données, notamment *k500*, elle présente des problèmes de convergence prématurée. Nous pensons qu'il serait intéressant de mettre en œuvre des méthodes d'anticonvergence telle que le *multiswarm* afin d'améliorer encore ses résultats. Nous souhaitons également réaliser une hybridation entre les deux méthodes afin d'avoir une méthode plus efficace quelque soit le jeu de données traité. Enfin, un point peu étudié en optimisation dynamique qui est pourtant crucial est l'évaluation de performances et il serait intéressant d'étudier de nouveaux critères.

# Bibliographie

- [1] H. Aguirre and K. Tanaka. Random bit climbers on multiobjective MNK-landscapes : Effects of memory and population climbing. *IEICE Trans. Fundamentals*, 88-A(1) :334–345, 2005.
- [2] R. K. Ahuja, O. Ergun, J. B. Orlin, and A. P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1-3) :75–102, 2002.
- [3] E. Angel, E. Bampis, and L. Gourvés. A dynasearch neighborhood for the bicriteria traveling salesman problem. In *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 6, pages 153–176. Springer-Verlag, Berlin, Germany, 2004.
- [4] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Finding tours in the TSP. Technical Report 99885, Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Germany, 1999.
- [5] A. Atamturk, G. Nemhauser, and M. Savelsbergh. A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems. *Journal of Heuristics*, 1 :247–259, 1996.
- [6] P. Augerat, J. M. Belenguer, E. Benavent, A. Corber'n, and D. Naddef. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, 106(2-3) :546–557, April 1998.
- [7] M. Babbar, A. Lakshmikantha, and D. E. Goldberg. A modified NSGA-II to solve noisy multiobjective problems. In *Genetic and Evolutionary Computation Conference (GECCO 2003)*, volume 2723 of *Lecture Notes in Computer Science*, pages 21–27, Chicago, Illinois, USA, 2003. Springer-Verlag.
- [8] J. Bard, G. Kontoravdis, and G. Yu. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36 :250–269, 2002.
- [9] S. Barreto, C. Ferreira, J. Paixão, and B. S. Santos. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3) :968 – 977, 2007.
- [10] C. Barrico and C. H. Antunes. An evolutionary approach for assessing the degree of robustness of solutions to multi-objective models. In S. Yang, Y.-S. Ong, and Y. Jin,



- editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence*, chapter 25, pages 565–582. Springer-Verlag, 2007.
- [11] M. Basseur and E. K. Burke. Indicator-based multi-objective local search. In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3100–3107, Singapore, 2007. IEEE Press.
- [12] M. Basseur, J. Lemesre, C. Dhaenens, and E.-G. Talbi. Cooperation between branch and bound and evolutionary approaches to solve a bi-objective flow shop problem. In C. C. Ribeiro and S. L. Martins, editors, *WEA*, volume 3059 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2004.
- [13] M. Basseur, F. Seynhaeve, and E. Talbi. Adaptive mechanisms for multiobjective evolutionary algorithms. In *Congress on Engineering in System Application (CESA 2003)*, pages 72–86, Lille, France, 2003.
- [14] M. Basseur, E.-G. Talbi, A. Nebro, and E. Alba. Metaheuristics for multiobjective combinatorial optimization problems : Review and recent issues. Research Report RR-5978, INRIA, 2006.
- [15] M. Basseur and E. Zitzler. Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research*, 2(3) :255–272, 2006.
- [16] J. Bennell and K. Dowsland. Hybridising tabu search with optimisation techniques for irregular stock cutting. *Management Science*, 47(8) :1160–1172, 2001.
- [17] R. Bent and P. Van Hentenryck. A two-stage hybrid local search for the vehicle routing problem with time windows. Technical Report Tech Report CS-01-06, Brown University, USA, September 2001.
- [18] D. Bertsimas, P. Chervi, and M. Peterson. Computational approaches to stochastic vehicle routing problems. Working papers 3285-91., Massachusetts Institute of Technology (MIT), Sloan School of Management, 1991.
- [19] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. Metaheuristics for the vehicle routing problem with stochastic demands. In S. B. . Heidelberg, editor, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242, pages 450–460, 2004.
- [20] L. Bianchi, M. Dorigo, L. Gambardella, and W. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, Online first, 2008.
- [21] M. Birattari. *Tuning Metaheuristics : A Machine Learning Perspective*. Springer Publishing Company, Incorporated, 2009.
- [22] T. Blackwell. Particle swarm optimization in dynamic environments. In *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51, pages 29–49. Springer, 2007.

- [23] J.-C. Boisson. *Modélisation et résolution par métaheuristiques coopératives : de l'atome à la séquence protéique*. PhD thesis, Université des Sciences et Technologies de Lille, France, 2008.
- [24] J. C. Boisson, L. Jourdan, and E.-G. Talbi. ParadisEO-MO. Technical report, 2008.
- [25] D. Brown, C. Huntley, and A. Spillane. A parallel genetic heuristic for the quadratic assignment problem. In *3<sup>rd</sup> International Conference on Genetic Algorithms*, pages 406–415, 1989.
- [26] D. Büche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Multi-objective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE Transaction on Systems, Man, and Cybernetics, part C : Applications and Reviews*, 32(4) :2002, 2002.
- [27] K. Büdenbender, T. Grünert, and H.-J. Sebastian. A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem. *Transportation Science*, 34(4) :364–380, November 2000.
- [28] L. T. Bui, H. A. Abbass, and D. Essam. Fitness inheritance for noisy evolutionary multi-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 779–785, New York, NY, USA, 2005. ACM.
- [29] E. Burke, P. Cowling, and R. Keuthen. Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In *Applications of Evolutionary Computing. EvoWorkshops2001*, volume 2037 of *Lecture Note in Computer Science*, pages 203–212, Como, Italy, April 2001. Springer-Verlag.
- [30] S. Cahon, N. Melab, and E.-G. Talbi. ParadisEO : A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics*, 10(3) :357–380, 2004.
- [31] V. Cerny. A thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45 :41–51, 1985.
- [32] A. Chabrier, E. Danna, and C. L. Pape. Coopération entre génération de colonnes sans cycle et recherche locale appliquée au routage de véhicules. In *JNPC*, 2002.
- [33] R. Chelouah and P. Siarry. A hybrid method combining continuous tabu search and nelder mead simplex algorithms for the global optimization of multim minima functions. *European Journal of Operational Research (EJOR)*, 2004.
- [34] R. Chelouah and P. Siarry. A hybrid method combining continuous tabu search and nelder mead simplex algorithms for the global optimization of multim minima functions. *European Journal of Operational Research*, 161(3) :636–654, March 2005.
- [35] D. Clements, J. Crawford, D. Joslin, G. Nemhauser, M. Puttlitz, and M. Savelsbergh. Heuristic Optimization : A hybrid AI/OR approach. In *Workshop on Industrial Constraint-Directed Scheduling*, Schloss Hagenberg, Austria, 1997.
- [36] R. K. Congram, C. N. Potts, and S. L. van de Velde. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*, 14(1) :52–67, 2002.

- [37] C. Cotta, J. F. Aldana, A. J. Nebro, and J. Troya. Hybridizing genetic algorithms with branch and bound techniques for the resolution of the tsp. In D. Pearson, N. Steele, and R. Albrecht, editors, *Artificial Neural Nets and Genetic Algorithms 2*, pages 277–280. Springer-Verlag, 1995.
- [38] C. Cotta and J. M. Troya. Embedding branch and bound within evolutionary algorithms. *Applied Intelligence*, 18 :137–153, 2003.
- [39] S. Dauzère-Pérès, P. Castagliola, and C. Lahlou. Niveau de service en ordonnancement stochastique. In J.-C. Billaut, A. Moukrim, and E. Sanlaville, editors, *Flexibilité et robustesse en ordonnancement*, chapter 5, pages 97–113. Hermès Science, Paris, France, 2005.
- [40] B. De Backer, V. Furnon, P. Kilby, P. Prosser, and P. Shaw. Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics*, 6(4) :501–524, 2000.
- [41] I. de Falco, R. del Balio, and E. Tarantino. An analysis of parallel heuristics for task allocation in multicomputers. *Computing*, 3(59) :259–275, 1995.
- [42] I. de Falco, R. del Balio, E. Tarantino, and R. Vaccaro. Improving search by incorporating evolution principles in parallel tabu search. In *International Conference on Machine Learning*, pages 823–828, 1994.
- [43] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : Nsga-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.
- [44] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2) :182–197, 2002.
- [45] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4) :463–494, 2006.
- [46] J. Denzinger and T. Offermann. On cooperation between evolutionary algorithms and other search paradigms. In *CEC-99*, pages 2317–2324, 1999.
- [47] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Chapter 2 time constrained routing and scheduling. In C. M. M.O. Ball, T.L. Magnanti and G. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35 – 139. Elsevier, 1995.
- [48] C. Duhamel and A. Quilliot. Coupling a metaheuristic with an exact method for the local access network design problem. In *Optimization 2001*, 2001.
- [49] I. Dumitrescu and T. Stützle. Combinations of local search and exact algorithms. In *EvoWorkshops 2003*, pages 211–223, 2003.
- [50] F. Edgeworth. *Mathematical Physics*. P. Keagan, London, 1881.

- [51] E. Edgington. *Randomization tests*. Marcel Dekker Inc., 1980.
- [52] M. Ehrgott and X. Gandibleux. Hybrid metaheuristics for multi-objective combinatorial optimization. In *Hybrid Metaheuristics : An Emerging Approach to Optimization*, volume 114 of *Studies in Computational Intelligence*, chapter 8, pages 221–259. Springer-Verlag, Berlin, Germany, 2008.
- [53] E. Falkenauer. A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, 2(2) :123–144, 1994.
- [54] H. Feltl and G. R. Raidl. An improved hybrid genetic algorithm for the generalized assignment problem. In *Symposium on Applied Computing, SAC'04*, pages 990–995. ACM press, March 2004.
- [55] M. Fischer. Vehicle routing. In *Network Routing*, volume 8, pages 1–33. Elsevier Science, 1995.
- [56] C. Fleurent and A. Ferland. Genetic hybrids for the quadratic assignment problem. *DI-MACS Series in discrete Mathematics and Theoretical Computer Science*, 16 :173–188, 1994.
- [57] K. Formiga, F. Chaudhry, B. Cheung, and L. Reis. Optimal design of water distribution system by multiobjective evolutionary methods. In C. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization (EMO-2003)*, pages 677–691, Springer-Verlag, LNCS 2632, 2003.
- [58] A. P. French, A. Robinson, and J. Wilson. Using a hybrid genetic-algorithm/branch and bound approach to solve feasibility and optimization integer programming problems. *Journal of Heuristics*, 7(6) :551–564, November 2001.
- [59] O. Fujiwara and D. Tung. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resources Research*, 26(4) :539–549, 1990.
- [60] C. Gagné and M. Parizeau. Genericity in evolutionary computation software tools : Principles and case study. *International Journal on Artificial Intelligence Tools*, 15(2) :173–194, 2006.
- [61] J. Garcia-Nieto, E. Alba, L. Jourdan, and E.-G. Talbi. Sensitivity and specificity based multiobjective approach for feature selection : Application to cancer diagnosis. *Information Processing Letters*, 2009. Impact factor : 0.660.
- [62] A. Gaspar-Cunha and J. A. Covas. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications*, 39(1) :75–96, 2008.
- [63] A. Gaspar-Cunha and A. Vieira. A hybrid multi-objective evolutionary algorithm using an inverse neural network. In LNCS, editor, *Hybrid Metaheuristic*, pages 25–30, 2004.
- [64] M. Gendreau, F. Guertin, J. Potvin, and E. Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4) :381–390, 1999.

- [65] M. Gendreau, G. Laporte, and R. SÃ©guin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *TRANSPORTATION SCIENCE*, 29(2) :143–155, 1995.
- [66] G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing : Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151 :1–11, 2003.
- [67] F. Glover. Tabu Search - part i. *ORSA Journal on Computing*, 1(3) :190–206, 1986.
- [68] C. K. Goh and K. C. Tan. Evolving the tradeoffs between Pareto-optimality and robustness in multi-objective evolutionary algorithms. In S. Yang, Y.-S. Ong, and Y. Jin, editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence*, chapter 20, pages 457–478. Springer-Verlag, 2007.
- [69] D. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimisation. In *2<sup>nd</sup> International Conference on Genetic Algorithms*, pages 41–49, 1987.
- [70] M. Gourgand, N. Grangeon, and S. Norre. ModÃ©le du flow-shop de permutation stochastique. In J.-C. Billaut, A. Moukrim, and E. Sanlaville, editors, *FlexibilitÃ© et robustesse en ordonnancement*, chapter 7, pages 135–161. HermÃ©s Science, Paris, France, 2005.
- [71] H. Handa, T. Horiuchi, O. Katai, and M. Baba. A novel hybrid framework of coevolutionary GA and machine learning. *International Journal of Computational Intelligence and Applications*, 2(1) :33–52, 2002.
- [72] H. Handa, T. Horiuchi, O. Katai, T. Kaneko, T. Konishi, and M. Baba. Fusion of coevolutionary ga and machine learning techniques through effective schema extraction. In L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, page 764, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.
- [73] J. Handl and J. Knowles. Improvements to the scalability of multiobjective clustering. In IEEE, editor, *IEEE Congress on Evolutionary Computation*, page ??, 2005.
- [74] M. Hansen and A. Jaskiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report TR-IMM-REP-1998-7, Technical University of Denmark, 1998.
- [75] P. Hansen and N. MladenoviÄ. An introduction to variable neighborhood search. In S. Voss, S. Martello, I. Osman, and C. Roucairol, editors, *Metaheuristics : Advances and trends in local search paradigms for optimization*, chapter 30, pages 433–458. Kluwer Academic Publishers, 1999.
- [76] M. Haouari and T. Ladhari. A branch and bound based local search for the flowshop problem. *Journal of the Operational Research Society*, 54 :1076–1084, 2003.

- [77] E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 329–343, London, UK, 2001. Springer-Verlag.
- [78] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, 28(3) :392–403, 1998.
- [79] C. A. R. Jahuira. Hybrid genetic algorithm with exact techniques applied to tsp. In *Second international workshop on Intelligent systems design and application*, pages 119–124. Dynamic Publishers, 2002.
- [80] C. A. R. Jahuira and E. Cuadros-Vargas. Solving the tsp by mixing gas with minimal spanning tree. In E. Cuadros-Vargas, E. Tejada-Gamero, and A. da Silva Simao, editors, *First International Conference of the Peruvian Computer Society*, pages 123–123, Lima - Perù, January 2003. Peruvian Computer Society.
- [81] S. Janson and M. Middendorf. A Hierarchical Particle Swarm Optimizer for Dynamic Optimization Problems. In *Applications of Evolutionary Computing*, volume 3005, pages 513–524. Springer, Coimbra, Portugal, 2004.
- [82] A. Jaszkiewicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1) :50–71, 2002.
- [83] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 9(1) :3–12, 2005.
- [84] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions on Evolutionary Computation*, 9 :303–317, 2005.
- [85] Y. Jin and B. Sendhoff. Trade-off between optimality and robustness : An evolutionary multiobjective approach. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 237–251, Faro, Portugal, 2003. Springer-Verlag.
- [86] Y. Jin and B. Sendhoff. Reducing fitness evaluations using clustering techniques and neural networks ensembles. In *Genetic and Evolutionary Computation Conference*, volume 3102 of *LNCS*, pages 688–699. Springer, 2004.
- [87] S. M. Johnson. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1 :61–68, 1954.
- [88] L. Jourdan, D. Corne, S. Savic, and G. Walters. Hybridising rule induction and multi-objective evolutionary search for optimising water distribution systems. In I. C. Press, editor, *Fourth International Conference on Hybrid Intelligent Systems , IEEE HIS 2004*, pages 434–439, 2005. Kita Kyushu, Japan. Acceptance rate : 38%.

- [89] L. Jourdan, D. Corne, S. Savic, and G. Walters. Preliminary investigation of the ‘learnable evolution model’ for faster/better multiobjective water systems design. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, LNCS 3410, pages 841–855, 2005. Guanajua, Mexico. Acceptance rate : 51.3%.
- [90] L. Jourdan, C. Dhaenens, and E.-G. Talbi. Using datamining techniques to help metaheuristics : A short survey. In *Hybrid Metaheuristics (HM’06)*, volume LNCS 4030, pages 57–69. Springer Berlin / Heidelberg, 2006. Acceptance rate : 31%.
- [91] N. Jozefowicz, F. Semet, and E.-G. Talbi. Parallel and hybrid models for multi-objective optimization : Application to the vehicle routing problem. In *Parallel Problem Solving from Nature VII*, pages 271–280, 2002.
- [92] M. Keijzer, J.-J. Merelo, G. Romero, and M. Schoenauer. Evolving objects : A general purpose evolutionary computation library. In *Proceedings of the 5th International Conference on Artificial Evolution (EA’01)*, pages 231–244, Le Creusot, France, 2001.
- [93] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks, 1995.*, volume 4, pages 1942–1948., 1995.
- [94] J. Kennedy, R. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001.
- [95] P. Kilby, P. Prosser, and P. Shaw. Dynamic VRPs : A study of scenarios. APES-06-1998, University of Strathclyde, U.K., 1998.
- [96] H.-S. Kim and S.-B. Cho. An efficient genetic algorithms with less fitness evaluation by clustering. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 887–894. IEEE, 2001.
- [97] G. W. Klau, I. Ljubic, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, and R. Weiskircher. Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem. In *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, volume 3102, pages 1304–1315, Seattle, Washington USA, June 2004. Lecture Note in Computer Science.
- [98] J. L. Klepeis, M. J. Pieja, and C. A. Floudas. Hybrid global optimization algorithms for protein structure prediction : Alternating hybrids. *Biophysical Journal*, 4(84) :869–882, February 2003.
- [99] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2) :149–172, 2000.
- [100] J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation (CEC’02)*, IEEE Press, pages 711–716, 2002.
- [101] J. Knowles and D. Corne. Memetic algorithms for multiobjective optimization : Issues, methods and prospects. In *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 313–352. Springer, 2005.

- [102] J. Kosa and D. Andre. Parallel genetic programming on a network of transputers. Technical report, Stanford University, 1995. Technical Report CS-TR-95-1542.
- [103] K. Kostikas and C. Fragakis. Genetic programming applied to mixed integer programming. In *EuroGP 2004*, volume 3003 of *Lecture Notes in Computer Science*, pages 113–124, 2004.
- [104] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Kluwer Academic, 1997.
- [105] P. Larranaga and J. Lozano. *Estimation of distribution algorithms : a new tool for evolutionary computation*. Kluwer Academic Publishers, 2001.
- [106] A. Larsen. *The Dynamic Vehicle Routing Problem*. PhD thesis, Technical University of Denmark, 2000.
- [107] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3) :263–282, 2002.
- [108] B. Lee and F. Richard. The interpretation of protein structures : Estimation of static accessibility. *Journal of Molecular Biology*, 55 :379–400, 1971.
- [109] X. Li, J. Branke, and T. Blackwell. Particle swarm with speciation and adaptation in a dynamic environment. In *GECCO'06 : Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 51–58, New York, NY, USA, 2006. ACM.
- [110] J. Liang and P. Suganthan. Dynamic multi-swarm particle swarm optimizer. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 124–129, 2005.
- [111] A. Liefoghe. *Métaheuristiques pour l'optimisation multiobjectif : Approches coopératives, prise en compte de l'incertitude et application en logistique*. PhD thesis, Université Lille 1 – Sciences et Technologies, France, 2009.
- [112] A. Liefoghe, M. Basseur, L. Jourdan, and E.-G. Talbi. ParadisEO-MOEO : A framework for evolutionary multi-objective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *Lecture Notes in Computer Science (LNCS)*, pages 386–400, Matsushima, Japan, 2007. Springer-Verlag.
- [113] A. Liefoghe, L. Jourdan, M. Basseur, E.-G. Talbi, and E. Burke. Metaheuristics for the bi-objective ring star problem. In LNCS, editor, *Evocop 2008*, volume 4972 of *LNCS*, pages 206–217, 2008. Acceptance rate : 34.8%.
- [114] A. Liefoghe, L. Jourdan, T. Legrand, J. Humeau, and E.-G. Talbi. ParadisEO-MOEO : A software framework for evolutionary multi-objective optimization. In *Advances in multi-objective nature inspired computing*, Studies in Computational Intelligence. Springer-Verlag, 2010.
- [115] A. Liefoghe, L. Jourdan, and E.-G. Talbi. Software framework for multi-objective optimization. In *22nd European Conference on Operational Research (EURO XXII)*, Prague, Czech Republic, 2007.



- [116] A. Liefvooghe, L. Jourdan, and E.-G. Talbi. A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework. In *IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (IEEE MCDM 2009)*, pages 88–95, Nashville, Tennessee, USA, 2009. IEEE Press.
- [117] A. Liefvooghe, L. Jourdan, and E.-G. Talbi. A unified model for evolutionary multiobjective optimization and its implementation in a general purpose software framework : ParadisEO-MOEO. *European Journal of Operational Research*, 2009. (under review).
- [118] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Computer Journal*, 44 :2245–2269, 1965.
- [119] S. Louis and C. Miles. Playing to learn : case-injected genetic algorithms for learning to play computer games. *Evolutionary Computation, IEEE Transactions on*, 9(6) :669–681, Dec. 2005.
- [120] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [121] M. J. Maddox and J. R. Birge. Using second moment information in stochastic scheduling. In *Recent advances in control and optimization of manufacturing systems*, volume 214 of *Lecture Notes in Control and Information Sciences*, pages 99–120, London, UK, 1996. Springer.
- [122] V. Maniezzo. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORM Journal on Computing*, 11(4) :358–369, 1999.
- [123] O. Martin, S. Otto, and E. Felten. Large-step markov chains for the tsp : Incorporating local search heuristics. *Operations Research Letters*, 11(4) :219–224, 1992.
- [124] T. Mautor and P. Michelon. Mimausa : a new hybrid method combining exact solution and local search. In *Second International Conference on Meta-Heuristics (MIC'97)*, 1997.
- [125] S. Meisel and D. Mattfeld. Synergies of operations research and data mining. *European Journal of Operational Research*, In Press, Corrected Proof :–, 2009.
- [126] S. Meisel and D. C. Mattfeld. Synergies of data mining and operations research. In *HICSS '07 : Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 56, Washington, DC, USA, 2007. IEEE Computer Society.
- [127] N. Melab, E.-G. Talbi, S. Cahon, E. Alba, and G. Luque. Parallel metaheuristics : Models and frameworks. In E.-G. Talbi, editor, *Parallel Combinatorial Optimization*, chapter 6, pages 149–162. John Wiley & Sons, Chichester, UK, 2006.
- [128] H. Meunier, E.-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 317–324 vol.1, 2000.

- [129] R. Michalski. Learnable evolution model : Evolutionary processes guided by machine learning. *Machine Learning*, 38(1–2) :9–40, 2000.
- [130] R. Michalski, G. Cervon, and K. Kaufman. Speeding up evolution through learning : Lem. In *Intelligent Information Systems 2000*, pages 243–256, 2000.
- [131] R. Michalski and J. Larson. Selection of most representative training examples and incremental generation of v11 hypothesis : The underlying methodology and the descriptions of programs esel and aq11. Technical Report Report No. 867, Urbana, Illinois : Department of Computer Science, University of Illinois, 1978.
- [132] R. Michalski, I. Mozetic, J. Hong, and N. N. Lavrac. The multipurpose incremental learning system AQ15 and its testing application to three medical domains. In *Proc. of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045. PA : Morgan Kaufmann, 1986.
- [133] R. Morrison. Performance measurement in dynamic environments. In J. Branke, editor, *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pages 5–8, 2003.
- [134] L. Murphy, A. Simpson, and G. Dandy. Pipe network optimisation using an improved genetic algorithm. Technical report, Dept of civil and environment engineering, University of Adelaide, Australia., 1993.
- [135] S. Olafsson, X. Li, and S. Wu. Operations research and data mining. *European Journal of Operational Research*, 187(3) :1429 – 1448, 2008.
- [136] I. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.*, 41(1-4) :421–451, 1993.
- [137] M. Palpant, C. Artigues, and P. Michelon. LSSPER : Solving the resource-constrained project scheduling problem with large neighbourhood search. *Annals of Operations Research*, 31(1) :237–257, 2004.
- [138] C. Papadimitriou. *The complexity of combinatorial optimization problems*. PhD thesis, Princeton, 1976.
- [139] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Prentice-Hall, 1982.
- [140] L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem : An experimental study. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer-Verlag, Berlin, Germany, 2004.
- [141] P. Pardalos, K. Ramakrishnan, M. Resende, and Y. Li. Implementation of a variance reduction based lower bound in a branch and bound algorithm for the quadratic assignment problem. *SIAM Journal on Optimization*, 7 :280–294, 1997.

- [142] V. Pareto. *Cours d'économie politique*. Rouge, Lausanne, 1896.
- [143] G. Pesant and P. Gendreau. A constraint programming framework for local search methods. *Journal of Heuristics*, 5(3) :255–279, 1999.
- [144] M. C. Portmann, A. Vignier, D. Dardihac, and D. Dezalay. Branch and bound crossed with ga to solve hybrid flowshops. *European Journal of Operational Research*, 107(2) :389–400, 1998.
- [145] J. Y. Potvin and J. M. Rousseau. An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, 46 :1433–1446, 1995.
- [146] H. Psaraftis. Dynamic vehicle routing problems. *Vehicle routing : Methods and studies*, 16 :223–248, 1988.
- [147] J. R. Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
- [148] J. R. Quinlan and R. M. Cameron-jones. Foil : A midterm report. In *In Proceedings of the European Conference on Machine Learning*, pages 3–20. Springer-Verlag, 1993.
- [149] K. Rasheed. Guided crossover : A new operator for genetic algorithm based optimization. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzal, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1535–1541, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.
- [150] K. Rasheed. An incremental-approximate-clustering approach for developing dynamic reduced models for design optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 986–993, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 2000. IEEE Press.
- [151] K. Rasheed and H. Hirsh. Using case based learning to improve genetic algorithm based design optimization. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, San Francisco, CA, 1997. Morgan Kaufmann.
- [152] K. Rasheed, S. Vattam, and X. Ni. Comparison of methods for developing dynamic reduced models for design optimization, 2002.
- [153] T. Ray. Constrained robust optimal design using a multiobjective evolutionary algorithm. In *IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 419–424, Honolulu, HI, USA, 2002. IEEE Press.
- [154] Y. Rochat and É. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1) :147–167, 1995.
- [155] C. Rosing and C. ReVelle. Heuristic concentration : Two stage solution construction. *European Journal of Operational Research*, 97 :75–86, 1997.
- [156] L. Rossman. Epanet, users manual. Technical report, U.S. Envi. Protection Agency, Cincinnati, Ohio, 1993.

- [157] B. Roy. à propos de robustesse en recherche opérationnelle et aide à la décision. In J.-C. Billaut, A. Moukrim, and E. Sanlaville, editors, *Flexibilité et robustesse en ordonnancement*, chapter 2, pages 35–50. Hermès Science, Paris, France, 2005.
- [158] D. Savic and G. Walters. Genetic algorithms for least-cost design of water distribution networks. *Journal of Water Resources Planning and Management*, 123(2) :67–77, 1997.
- [159] M. Sebag, M. Schoenauer, and C. Ravisé. Evolution darwinienne vs evolution civilisée ? In *Actes de la Conférence Evolution Artificielle, EA94* editors, 1994.
- [160] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming - CP98, 4th International Conference*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431, Pisa, Italy, 1998. Springer-Verlag.
- [161] H. Shi and L. Wang. A mixed branch-and-bound and neural network approach for the broadcast scheduling problem. In *Design and application of hybrid intelligent systems, HIS'03*, pages 42–49. IOS Press, December 2003.
- [162] F. Streichert and H. Ulmer. JavaEvA : a java based framework for evolutionary algorithms. Technical Report WSI-2005-06, Centre for Bioinformatics Tübingen (ZBIT) of the Eberhard-Karls-University, Tübingen, 2005.
- [163] J. Suh and D. van Gucht. Incorporating heuristic information into genetic search. In *2<sup>nd</sup> International Conference on Genetic Algorithms*, pages 100–107, 1987.
- [164] E.-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(2) :541–564, 2002.
- [165] E.-G. Talbi. *Metaheuristics : from design to implementation*. Wiley, 2009.
- [166] E.-G. Talbi and H. Meunier. Hierarchical parallel approach for gsm mobile network design. *Journal of Parallel and Distributed Computing*, 66(2) :274–290, 2006.
- [167] E.-G. Talbi, T. Muntean, and I. Samarandache. Hybridation des algorithmes génétiques avec la recherche tabou. In *Evolution Artificielle (EA'94)*, 1994.
- [168] E.-G. Talbi, M. Rahoual, M. H. Mabed, and C. Dhaenens. A hybrid evolutionary approach for multicriteria optimization problems : Application to the fow shop. In *First International Conference on Evolutionary Multi-criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 416–428, Zurich, Switzerland, 2001. Springer-Verlag.
- [169] K. C. Tan, C. Y. Cheong, and C. K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177(2) :813–839, 2007.
- [170] K. C. Tan and C. K. Goh. Handling uncertainties in evolutionary multi-objective optimization. In J. M. Zurada, G. G. Yen, and J. Wang, editors, *IEEE World Congress on Computational Intelligence (WCCI 2008)*, volume 5050 of *Lecture Notes in Computer Science*, pages 262–292, Hong Kong, China, 2008. Springer.

- [171] R. Tanese. Parallel genetic algorithms for a hypercube. In *2<sup>nd</sup> International Conference on Genetic Algorithms*, pages 177–183, 1987.
- [172] J. Teich. Pareto-front exploration with uncertain objectives. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Evolutionary Multi-Criterion Optimization, First International Conference (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 314–328, London, UK, 2001. Springer-Verlag.
- [173] R. Thomsen. *Computational Intelligence in Bioinformatics*, chapter Protein-Ligand Docking with Evolutionary Algorithms. 2008.
- [174] V. T'kindt, N. Monmarché, F. Tercinet, and D. Laügt. An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, 142 :250–257, 2002.
- [175] H. Trautmann, J. Mehnen, and B. Naujoks. Pareto-dominance in noisy environments. In *IEEE Congress on Evolutionary Computation (CEC 2009)*, pages 3119–3126, Trondheim, Norway, 2009. IEEE Press.
- [176] S. Umetani, M. Yagiura, and T. Ibaraki. One-dimensional cutting stock problem with a given number of setups : a hybrid approach of metaheuristics and linear programming. In *Hybrid Metaheuristics HM'04, at workshop European Conference on Artificial Intelligence ECAI'04*, pages 101–114, Valencia Spain, August 2004.
- [177] D. Van Veldhuizen and G. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 204–211 vol.1, 2000.
- [178] M. Vasquez and J.-K. Hao. A hybrid approach for the 0/1 multidimensional knapsack problem. In *IJCAI*, pages 328–333, 2001.
- [179] L. Vermeulen-Jourdan, C. Dhaenens, and E.-G. Talbi. Clustering nominal and numerical data : A new distance concept for a hybrid genetic algorithm. In J. Gottlieb and G. R. Raidl, editors, *Evolutionary Computation in Combinatorial Optimization - EvoCOP 2004*, volume 3004 of *LNCS*, pages 220–229, Coimbra, Portugal, 5-7 April 2004. Springer Verlag.
- [180] H.-M. Voigt, J. Born, and I. Santibanez-Koref. Modelling and simulation of distributed evolutionary search processes for function optimization. *Parallel Problem Solving from Nature*, 496 :373–380, 1990. LNCS.
- [181] Y. Wei. Variance, entropy, and uncertainty measure. *Dept. of Statistics, People's Univ. of China*.
- [182] K. Weicker. Performance measures for dynamic environments. In *Parallel Problem Solving from Nature PPSN VII*, pages 64–76. Springer, 2002.
- [183] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, 1996.

- [184] W. Yang, K. Mathur, and R. Ballou. Stochastic vehicle routing problem with restocking. *Transportation Science*, 34(1) :99–112, 2000.
- [185] S.-H. Yoo and S.-B. Cho. Partially evaluated genetic algorithm based on fuzzy c-means algorithm. In LNCS, editor, *PPSN*, pages 440–449, 2004.
- [186] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms : Empirical results. *Evolutionary Computation*, 8(2) :173–195, 2000.
- [187] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of pareto set approximations. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization - Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, chapter 14, pages 373–404. Springer-Verlag, Berlin, Germany, 2008.
- [188] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2 : Improving the strength pareto evolutionary algorithm. TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, 2001.
- [189] E. Zitzler and S.Kunzli. Indicator-based selection in multiobjective search. *Parallel Problem Solving from Nature, PPSN VIII*, 3242 :832–842, 2004.
- [190] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers : An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2) :117–132, 2003.



## Cinquième partie

### Annexes





**Chap. 9**

**CV Détaillé**

Laetitia VERMEULEN-JOURDAN  
 Née le 9 Juillet 1976 (33 ans)  
 à Lille (59)  
 Mariée, 1 Enfant  
 E-mail :  
 laetitia.jourdan@inria.fr

**Chargée de recherche 1ère classe INRIA  
 INRIA Lille Nord Europe**

### Fonctions occupées

---

2008-...	<b>Chargée de recherche 1ère classe INRIA</b> <i>Inria Lille Nord Europe, Equipé Dolphin</i>
2005-2008	<b>Chargée de recherche 2ème classe INRIA</b> <i>Inria Futurs, Team Dolphin</i>
2004-2005	<b>Chargée de recherche en “Fast Optimisation”</b> , ESPRC Council grant, University of Exeter, Devon, Angleterre
2004-2005	<b>Attachée Temporaire d’Enseignement et de Recherche</b> (mi-temps) à l’UFR d’IEEA de l’USTL. – Cours, TD et TP d’Informatique (IUP MIAGE et GMI, Licence S6, Master Pro).
2003-2004	<b>Attachée Temporaire d’Enseignement et de Recherche</b> (plein temps) à l’UFR d’IEEA de l’USTL. – Cours, TD et TP d’Informatique (DEUG, IUP MIAGE, Licence, DESS), – Encadrement de stages et de projets.
2000-2003	<b>Monitrice de l’Enseignement Supérieur</b> <i>Université Lille 1-USTL</i> : enseignante en informatique en premier, deuxième et troisième cycle. – Cours, TD et TP d’Informatique (DEUG, IUP MIAGE, Licence, DESS), – Encadrement de stages et de projets.
1999-2000	<b>Vacataire de l’Enseignement Supérieur.</b> <i>Université Lille 1-USTL</i>

## Formation

---

- 2004-2005 **Postdoctorat** à l'Université d'Exeter (Devon, Angleterre) : Chargée de recherche en "Fast Optimisation" sous la direction de D.W. Corne. (6 mois : Juillet-Janvier)  
Financement : ESPRC Council grant (Uk)
- 2000-2003 **Doctorat d'informatique** à l'Université des Sciences et Technologies de Lille. Allocation Ministerielle (MNERT) + Monitorat CIES  
Laboratoire d'Informatique Fondamentale de Lille (LIFL, 59). Equipe OPAC (Optimisation PARallèle et Collaborative).  
Sujet : "Métaheuristiques pour l'extraction de connaissances : application à la génomique"  
Thèse soutenue le 26 Novembre 2003, Mention Très Honorable avec félicitations du jury  
Jury  
Mme M. CLERBOUT (Professeur, Université de Lille 1, Président)  
Mme M. SEBAG (Directrice de Recherche CNRS, LRI, Orsay, Rapporteur)  
M. G. VENTURINI (Professeur, Ecole Polytechnique de Tours, Rapporteur)  
M. A.A. FREITAS (Lecturer at the Computing Laboratory, University of Kent at Canterbury, UK, Examineur)  
M. C. DINA (Ingénieur, Institut de Biologie de Lille, Pasteur, Examineur)  
Directeurs : Mme C. DHAENENS, M. E-G. TALBI
- 1999-2000 **DEA d'Informatique** de Lille, Mention "Bien", stage réalisé au sein de l'équipe OPAC du LIFL
- 1999 Diplôme d'**Ingénieur maître** de l'*Université Paris IX Dauphine*, Mention "Très Honorable"

## Réalisation de logiciels

---

- ARV** Association Rule Viewer : Opensource, 4 mois, co-auteurs : Damien Delautre, Sébastien Demay, mode de diffusion : téléchargeable sur internet. ARV est un logiciel permettant de visualiser différentes règles d'association suivant plusieurs critères de qualité. <http://www.lifl.fr/~jourdan/download/arv.html>
- Ascq\_Me** Opensource (Cecill), 1 an, co-auteurs : Jean-Charles Boisson, Christian Rolando. Ascq\_Me est un outil permettant l'identification automatiquement de protéines à partir de spectres de masse. [https://www.genopole-lille.fr/logiciel/ascq\\_me/index.html](https://www.genopole-lille.fr/logiciel/ascq_me/index.html).
- ParadisEo-MOEO** Opensource, extension toujours en cours, Co-auteurs : S. Cahon, J. Humeau, T. Legrand, A. Liefoghe, H. Denech. <http://paradiseo.gforge.inria.fr>.

Département	Cours	Volume (annuel)	Année
UFR IEEA, Miage 3	Cours FDD : entrepôt de données et fouilles de données	36h	2009-2011
UFR IEEA, Master 1	Cours FDD : entrepôt de données et fouilles de données	18h	2008-2011
SUAIO Miage M1 FC	Mise à niveau FDD et recherche opérationnelle	20h	2008-2011
UFR IEEA M2 IAGL	Cours Intelligence Artificielle	30h	2007-2009
Polytech'Lille (GIS5)	Cours de Datamining	14h	2006-2008
Polytech'Lille (GIS4)	Cours de Recherche Opérationnelle	8h	2006-2008
Polytech'Lille (GIS4)	TD de Recherche Opérationnelle	12h	2006-2008
Polytech'Lille (GIS5)	Cours d'Optimisation avancée	14h	2006-2008
UFR IEEA, Miage M1	Cours de Recherche Opérationnelle	30 h	2005
M2 Bioinformarique	TD de Fouilles de données	6h	2005

TABLE 9.1 – Récapitulatif des cours donnés depuis ma nomination en qualité de chargée de recherche en 2005

## Enseignement (Depuis 2005)

J'ai tenu en tant que chercheur à continuer à enseigner afin de garder le contact avec les étudiants et d'essayer de susciter des vocations. Le tableau 9.1 récapitule les enseignements donnés depuis ma nomination en tant que chercheur INRIA.

## Encadrement

L'encadrement d'étudiants fait intégralement partie de la tâche du chercheur par la formation à la recherche, qu'il s'agisse d'étudiants de 3ème cycle (Thèse de doctorat, de master) ou bien de 2ème cycle (Ecole d'ingénieur, maîtrise...).

### Thèses (5 : 2 Soutenues, 3 en cours)

- Jean-Charles Boisson, *Modélisation et résolution par métaheuristiques coopératives : de l'atome à la séquence protéique* (50%, avec E-G. Talbi) (2005-2008). Soutenue en Décembre 2008.
  - *Résumé* : A travers cette thèse, nous montrons l'importance de la modélisation et de la coopération de métaheuristiques pour la résolution de problèmes réels en bioinformatique. Pour se faire, deux problèmes ont été étudiés : le premier dans le domaine de la protéomique pour l'identification de protéines à partir de données spectrales et le second dans le domaine de l'analyse structurale des molécules pour le problème du docking moléculaire flexible. Ainsi, pour le premier problème, un nouveau modèle basé sur une comparaison directe des bases de données protéiques avec les données expérimentales brutes a été mise en place. L'approche associée a été intégrée au sein d'un moteur d'identification par empreinte de masse peptide appelé ASCQ\_ME. Ce modèle d'identification a permis ensuite de proposer et de valider une modélisation pour le problème de *de novo protein sequencing* qui consiste à retrouver la séquence d'une protéine à partir seulement des données expérimentales. Il s'agit d'un modèle en trois étapes appelé SSO pour *Sequence, Shape et Order*. Après une étude de chacune de ces étapes, SSO a été implémenté et testé à travers trois métaheuristiques collaborant de manière séquentielle. Pour le second problème, une étude de nouvelles modélisations multi-objectif a été menée et a conduit à la définition d'un ensemble de huit modèles différents testés à l'aide d'algorithmes génétiques multi-objectif parallèles. Une douzaine de configuration d'opérateurs génétiques a été testée afin de mettre en évidence l'efficacité de l'hybridation des algorithmes génétiques avec des recherches locales. Pour chacune des parties, l'implémentation et la mise en place des collaborations furent possible grâce à la plateforme ParadisEO et notamment grâce à nos contributions à la partie ParadisEO-MO dédiée aux métaheuristiques à base de solution unique. L'ensemble de ces travaux a été soutenu par le PPF Bio-Informatique de l'Université des Sciences et Technologies de Lille et le projet ANR Dock.
  - *Publications* : HUPO'05, ROADEF'06, Recomb'06, CEC'06, AINA'06, ROADEF'08, JO-BIM'08, CIBCB'08, META'08, ROADEF'09, JMMA (2010)
  - *Devenir* : Jean-Charles Boisson est maître de conférences à l'université de REIMS.
- Arnaud Liefooghe, *Métaheuristiques pour l'optimisation multiobjectif : Approches coopératives, prise en compte de l'incertitude et application en logistique*. (50% avec E-G. Talbi) (2006-2009).
  - *Résumé* : De nombreux problèmes d'optimisation issus du monde réel, notamment dans le domaine de la logistique, doivent faire face à beaucoup de difficultés. En effet, ils sont souvent caractérisés par des espaces de recherche vastes et complexes, de multiples fonctions objectif contradictoires, et une foule d'incertitudes qui doivent être prises en compte. Les métaheuristiques sont des candidates naturelles pour résoudre ces problèmes, ce qui les rend préférables aux méthodes d'optimisation classiques. Toutefois, le développement de métaheuristiques efficaces découle d'un processus de recherche complexe. Le cœur de ce travail réside en la conception, l'implémentation et l'analyse expérimentale de métaheuristiques pour l'op-

timisation multiobjectif, ainsi que leurs applications à des problèmes logistiques de tournées et d'ordonnancement. Tout d'abord, une vue unifiée de ces approches est présentée, puis intégrée dans une plateforme logicielle dédiée à leur implémentation, ParadisEO-MOEO. Ensuite, plusieurs approches de coopération, combinant des métaheuristiques pour l'optimisation multiobjectif, sont proposées. Enfin, la question de la prise en compte de l'incertitude est abordée dans le contexte de l'optimisation multiobjectif.

- *Publications* : EMO'07 (2), Euro XXII, META 2008, Evocop'08, HM'08, RIA'2008, SLS'09, MCDM'09, EURO XXIII, URPDM'10, COR'10, ROADEF'2010,
- *Devenir* : Arnaud Liefoghe est maître de conférences à l'université de Lille 1.
- Marie-Eléonore Marmion, *Métaheuristiques coopératives auto-adaptatives pour environnements incertains*, (50%, avec C. Dhaenens) (2008-...)
  - *Résumé* : Dans le cadre de cette thèse, nous étudions les paysages de problèmes combinatoires mono et multi-objectifs. En particulier, nous nous intéressons aux problèmes de tournées de véhicules et de flowshop.
  - *Publications* : ROADEF'10, AICCSA'10.
- Mostepha-Redouane Khouadjia, *Nouveaux paradigmes de métaheuristiques en environnement incertain et dynamique*, (50% avec E-G. Talbi) (2008-...)
  - *Résumé* : Dans le cadre de cette thèse, nous proposons différents algorithmes basés sur les essaims de particules pour résoudre des problèmes de tournées de véhicules dynamiques.
  - *Publications* : META'08, AICCSA'10, ANTS'10.
- Khedidja Seridi, *Optimisation multiobjectif pour le biclustering* (50% avec E-G. Talbi) (Déc 2009-...)
  - *Résumé* : Dans le cadre de cette thèse, nous proposons de nouveaux modèles pour le biclustering multi-critère. Ce travail est effectué sur des données de déséquilibre de liaison fournies par la société Genesdiffusion.

## Postdoc (2)

- Oliver Schuetze (Post-doc INRIA)(30%, avec E-G. Talbi) (2006-2007)
  - *Résumé* : Le travail a porté sur de nouveaux modèles multiobjectifs pour le design de polymères polyconducteurs. Nous avons également fournis des outils d'aide à la décision permettant un choix plus interactif des solutions intéressants.
  - *Publications* : EMO'07, PAT, JMPP.
  - *Devenir* : Oliver Schuetze est chercheur au CINVESTAV-IPN au Mexique.
- Waldo Cancino (Post-doc INRIA) (50%, avec E-G. Talbi) (2008-2009)
  - *Résumé* : Le travail a porté sur l'amélioration et la parallélisation d'algorithmes multiobjectifs pour l'inférence en phylogénie. Une parallélisation des méthodes en fin grain et gros grain a été réalisée et testée sur différents environnements parallèles (dont GRID'5000).
  - *Publications* : LION'4 (2010), Evobio'2010 (Best paper nominee)
  - *Devenir* : Waldo Cancino est chercheur au Brésil.

## DEA/Master 2 recherche : 8 (MRI Français : 6 (LIFL) et MRI étrangers : 2 (Espagne, Algérie))

- Mohamed Redjedal (2009-2010) : Master Recherche d'Informatique de Lille *optimisation multicritère pour le placement d'applications intensives sur système-sur-puce (SoC)*. (50% avec

Pierre Boulet de l'équipe DART)

- Dalia Sulieman (2008-2009) : Master Recherche d'Informatique de Lille *Solutions robustes pour le VRP avec demande stochastique*. (70%, avec E-G. Talbi).
- Mustapha Redouane Khouadjia (2007-2008) : Master Recherche d'Informatique de Lille, *Nouveaux paradigmes de métaheuristiques pour le problème de transport dynamique*. (50%, avec E-G. Talbi)
- Abdelhakim Denech (2007) : Master université de Constantine (Algérie, Projet 3+3 Méditerranée), *Implémentation de composants dans Paradiseo-MOEO*. (50%, avec E-G. Talbi)
- José Manuel Nieto (2007) : Master Université de Malaga (Spain, Projet 3+3 Méditerranée), *Sélection d'attributs par SVM*. (50%, avec E-G. Talbi)
- Matthieu Quatrelivre (2006-2007) : Master Recherche d'Informatique de Lille, *Méthode exacte multiobjective pour le Ringstar Problem*. (50%, avec E-G. Talbi)
- François le Floc'h (2005-2006) : Master Recherche d'Informatique de Lille, *Coopération online de méthodes d'optimisation pour la sélection d'attributs*. (50%, avec E-G. Talbi)
- Arnaud Liefoghe (2005-2006) : Master Recherche d'Informatique de Lille, *Optimisation multi-objectif sous incertitude*. (50%, avec E-G. Talbi)
- Jean-Charles Boisson (2004-2005) : Master Recherche d'Informatique de Lille, *Méthode d'optimisation pour la protéomique*. (50%, avec E-G. Talbi)
- Vanessa Chantreau (2003-2004) : DEA d'Informatique de Lille : *Approche multi-critère hybride pour les règles d'association : Application l'analyse des données à puces à ADN*. (40%, avec C. Dhaenens et E-G. Talbi)

## Ingénieurs Associés (2)

Dans le cadre de nos travaux sur la plateforme Paradiseo, j'ai encadré deux ingénieurs associés. Thomas Legrand a ensuite été recruté en qualité d'ingénieur de recherche à l'INRIA.

- Jérémie Humeau (2008-2010)
- Thomas Legrand (2006-2008)

## projets de niveaux Bac +4/+5

- 2009-2010 PFE GIS 5 (Polytech'lille), *Intégration de critères environnementaux dans le VRP : un modèle multiobjectif*, Ghizlane Chniber.
- 2008-2009 PFE GIS 5 (Polytech'lille), *Intégration et évaluation d'algorithmes multicritères pour l'optimisation de transport à la demande*, Vincent Dhellemmes et Elisabeth Alias
- 2007-2008 PFE GIS 5 (Polytech'lille), *Validation statistique d'algorithmes multiobjectifs*, Nicaise Tassadong et Hannane Jorti.
- 2006-2007
  - PFE GIS 5 (Lille), *Algorithme multiobjectif pour la sélection d'attributs*, Youness Lemrabet.
  - Master d'informatique (Lille), *Étude et développement d'algorithmes évolutionnaires classiques en optimisation multi-objectif*, Franck Vare.
  - Master université de Constantine (Algérie, Projet 3+3 Méditerranée), *Implémentation de composant dans Paradiseo-MOEO*, Abdelhakim Denech.
  - PHD Université de Malaga (Spain, Projet 3+3 Méditerranée), *Sélection d'attributs par SVM*, José Manuel Nieto.
- 2005-2006



- PFE GIS 5 (Lille), *Algorithme multiobjectif pour les polymères polyconducteurs*, Thomas Legrand.
- PFE GIS 5 (Lille), *Analyse de scores pour la protéomique*, Grégory Vanhuisbergh.
- 2004-2005 Master Bioninformatique (Lille), *Nouvel algorithme pour l'extraction de données en Protéomique*. Nicolas Dolet, Nicolas Gruszczynska.
- 2003-2004
  - IUP GMI 3 (Lille), *Etude et implémentation de critères de clustering*, Yves Delalandes, Gabriel Westrelin.
  - PFE GIS 5 (Polytech'lille), *Challenge ROADEF 2005*, Margaux Roussel.
  - PFE GIS 5 (Polytech'lille), *Hybridation métaheuristique / méthode exacte*, Mathilde Oudin, Lucie Sénéchal.
- 2002-2003
  - IUP GMI 3 (Lille), *Etude des critères de qualité d'une règle d'association*, Longuemart Thomas et Wasson Axelle.
  - PFE GIS 5 (Polytech'lille), *ClusterDing.com*, Hicham Bandali, Julien Gallant, Grégory Tyrou.
  - DESS IAGL (Lille), *Une interface de visualisation de critères de qualité de règles d'association*, Damien Delautre, Sébastien Demay.
- 2001-2002
  - Génie Informatique et Statistique à Polytech'Lille 5ème année, *Un algorithme génétique pour le clustering*, Grégory Legoff et Nassera Nachit.
  - IUP GMI 3 (Calais), *Un algorithme génétique pour l'extraction de connaissances*, Wyngaert Jean-Mickael et Achiba Sophian.
  - Maitrise d'informatique (Lille), *Interface graphique pour la génomique*, Arbelaiz Christophe et Lardjoune Samira.
  - DESS de BioInformatique, *Test et Evaluation d'un outil de datamining pour la recherche d'interaction entre gènes et facteurs d'environnement*, Jérôme Eteve.

## Participation à des projets et administration de la recherche

### Participation à des projets

J'ai participé à différents projets.

- **ANR DOCK** (Docking on Grids) (2006-2009) : collaboration avec l'IBL (Institut de Biologie de Lille) et le CEA (Grenoble). Développement de logiciels de sampling et de docking sur grilles. Je participe plus particulièrement à la partie framework multiobjectif (Paradiseo MOEO) et à la proposition de modèles multiobjectifs du docking.
- **PPF (Bioinformatique)** (2006-2009, 2010-2013) : Programme national avec l'université de Lille (USTL) sur la résolution de problèmes liés à la bioinformatique en utilisant des techniques d'optimisation combinatoire.
- ACI "Masse de données" **Projet GGM "Grille Geno-Medicale"** (2004-2007) en collaboration avec le LIRIS (Lyon) et IRIT (Toulouse).

- **Projet INRIA 3+3 Méditerranée PERFORM** (2006-2012) : Université de Malaga (Espagne), Université de Constantine (Algérie), Université de Tunis (Tunisie).
- **MOMDI - Université de Malaga** : ce projet avec l'équipe du professeur E. Alba traite de problèmes dynamiques et multicritère sur grille de calcul. Les problèmes d'optimisation abordés sont relatifs à des problèmes de routage.
- **Collab - SINTEF (Norway)** (2009-2011) : cette collaboration traite de méthodes collaboratives et parallèles pour des problèmes de transport (plus particulièrement des problèmes de plus court chemin dynamique et des problèmes de tournées enrichies). Les autres partenaires du projet sont : Univ. of Jyväskylä, (Finland), CIRRELT (Montreal, Canada), ITMMA (Univ. of Antwerp, Belgium).

### Participation à des projets industriels

- Société **Genediffusion** <http://www.genesdiffusion.com/> : analyse de données génomiques
- Institut Pasteur de Lille : extraction de connaissances à partir de données médicales sur grille informatique
- Institut de Biologie de Lille : extraction de connaissances à partir de données médicales

### Participation à des groupes de travail

- Co-responsable du groupe PM2O (2007-...) (Programmation Multi-Objective (<http://www.lifl.fr/PM2O/>))
- Secrétaire de l'association Évolution Artificielle (2007-...)(<http://www.lifl.fr/EA/>)
- Membre du groupe META (Metaheuristiques de P. Siarry et E-G. Talbi)

### Administration de la recherche

#### • Participation à des conseils

- Membre élue au Conseil de Laboratoire du LIFL (2001-2005)
- Participation à la commission de recrutement des ingénieurs associés (2007)
- Membre titulaire élue de la CSE 27 de l'université de Lille 1 (2007-2009)
- Membre nommée de la CSE 27 de l'université du Littoral (2007-2009)
- Membre nommée du COST Relations Internationales de l'INRIA (2007-....)
- Membre élu du Comité de Centre INRIA Lille Nord Europe (2008-2012)
- Membre élu du CLHS du centre INRIA Lille Nord Europe (2008-2012)
- Membre du vivier 27 Université de Lille 1 (COS 2009)
- Membre du comité de sélection de l' Université d'Angers (COS 2009)
- Membre du comité de sélection de l'Université de Reims (COS 2010)
- Membre du comité de sélection de la chaire INRIA-Lille 2 (COS 2010)
- Membre nommée de la commission de recrutement des chargés de recherche de l'INRIA Lille Nord Europe (2009, 2010)

- **Tâches d'intérêt collectif** : Correspondant Raweb du centre Lille Nord Europe (le Raweb s'occupe de la rédaction des rapports d'activité des équipes de recherche INRIA), le rôle du correspondant est de coordonner cette rédaction et de répondre aux questions techniques.

- **Organisation de Numéros spéciaux de Journaux et d'ouvrage :**

- Numéro spécial Journal of Mathematical Modelling and Algorithm (JMMA) on Metaheuristics (2008) (volume 7, number 2), avec E-G. Talbi
- Numéro spécial Journal of Mathematical Modelling and Algorithm (JMMA) on Advances on nature bioinspired computing (2010) (volume 9, number 2) , avec E-G. Talbi
- Ouvrage chez Springer avec C. Dhaenens et C. Coello Coello Advances in multi-objective nature inspired computing (Studies in computational intelligence, Vol. 272)

- **Organisation de sessions et de conférences :**

- Organisation d'une session "Metaheuristics and Knowledge discovery", META'10, avec D. Corne et C. Dhaenens (Djerba, Tunisie)
- Organisation de AICCSA 2010 (Hammamet, Tunisie)
- Organisation de META'2010 (Djerba, Tunisie)
- Co-organisation de la 5eme école d'été EA 2010 (Calais, France)
- Organisation de sessions à ROADEF 2010 à Toulouse (groupe PM2O)
- Publicity chair de EA 2009 (Strasbourg, France)
- Organisation d'une Journée satellite à la conférence internationale EMO 2009 (Nantes, France)
- Organisation de sessions à ROADEF 2009 à Nancy (groupe PM2O)
- Organisation de JOBIM'2008 (Lille, France), 450 participants
- Organisation de META'2008 (Hammamet, Tunisie), 100 participants
- Organisation de META'2006 (Hammamet, Tunisie), 90 participants
- Organisation d'une école sur les Métaheuristiques (Lille, France), 45 participants
- Organisation de ROADEF'2006 (Lille, France), 350 participants
- Co-organisation de EGC'2006 (Lille, France), 150 participants.
- Co-organisation de EA'2005 (Villeneuve d'Ascq, France), 70 participants

## Arbitrage

Je suis régulièrement sollicitée pour différents arbitrages :

- **Participation à des comités de lecture** : EA(2005-...), CEC(2004-..., 2005 : Expert topic chair), CIBCB 2007, ICTAI 2006, ROADEF 2006, MOPGP 2006, ISPDC 2005, GECCO 2005, NISCO'07, GECCO 2008, EMO 2009, ROADEF 2009, CEC 2009, ROADEF 2010, GECCO 2010, SEAL 2010, EMO 2010 ...

- **Chairman** : EA, ROADEF, ECCO-CO, META.

- **Arbitrage pour des revues** : BioInformatics, IEEE Transaction on Evolutionary Computation, EJOR European Journal of Operational Research, JPDC Journal of Parallel and Distri-

buted Computing, JOH Journal of Heuristics, RAIRO (special issue), IEEE/ACM Transactions on Computational Biology and Bioinformatics, Knowledge-Based Systems, Journal of Artificial Evolution and Application, 4OR, ITOR ...

## 9.1 Publications

Le tableau 9.2 présente une vue synthétique des publications que j'ai rédigées après ma thèse.

	Revue	LNCS	IEEE	Autres publications en Anglais	Français
2010	COR, JMMA, EJOR, ASoCo	LION'4, Evobio'10 (2), ANTS'10	CEC 2010	URPDM'10, TraSh'10	ROADEF'10 (3)
2009	EJOR, IPL, MMP	SLS'09	MCDM'09		ROADEF'09
2008	PAT, JMMA, RIA (fr)	HM'08, Evocop'08	CIBCB'08, AICCSA'08	META'08	ROADEF'08, JOBIM'08
2007		EMO'07 (2)	CEC'07	GECCO'07, Euro'07, ASMS	JOBIM'07
2006		HM'06	CEC'06, HipComb'06	Ecco/Co'06, META'06	ROADEF'06
2005 2004	IJFCS	EMO'05 Evocop'04	NIDISC'04, HIS'04	SCBA, CCWI	ROADEF'05
Total	8	12	10	8	9

TABLE 9.2 – Synthèse des publications depuis le doctorat par année et par type.



# Bibliographie personnelle

## Edition journaux et livre

- [E1] L. Jourdan and E.-G. Talbi. *Special issue on metaheuristics*. Journal of Mathematical Modelling and Algorithms, **volume 7**(2), 2008.
- [E2] L. Jourdan and E.-G. Talbi. *Special Issue on Recent Developments in Bioinspired Algorithms*. Journal of Mathematical Modelling and Algorithms, **volume 9**(2), 2010.
- [E3] C. Coello Coello, C. Dhaenens and L. Jourdan. *Advances in multi-objective nature inspired computing Studies in computational intelligence, Volume 272, 2010*.

## Revue internationale avec comité de lecture

- [R1] J.-C. Boisson, L. Jourdan, and E.-G. Talbi, *Metaheuristics based De Novo Protein Sequencing : A new approach*, Applied Soft Computing, **Accepted**.
- [R2] A. Liefoghe, L. Jourdan, and E.-G. Talbi *A software framework based on a conceptual unified model for evolutionary multiobjective optimization : ParadisEO-MOEO*, European Journal of Operational Research, **Accepted**.
- [R3] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and D. Horvath. *Single- and multi-objective cooperation for the flexible docking problem*. Journal of Mathematical Modelling and Algorithms, **volume 9**(2) : pages 195–208, 2010.
- [R4] A. Liefoghe, L. Jourdan, and E. G. Talbi. *Metaheuristics and cooperative approaches for the bi-objective ring star problem*. Computers & Operations Research, **volume 37**(6) :pages 1033–1044, 2010. ISSN 0305-0548. Impact factor : 1.366.
- [R5] J. Garcia-Nieto, E. Alba, L. Jourdan, and E.-G. Talbi. *Sensitivity and specificity based multiobjective approach for feature selection : Application to cancer diagnosis*. Information Processing Letters, **volume 109** :pages 887–896, 2009. Impact factor : 0.706.
- [R6] L. Jourdan, M. Basseur, and E.-G. Talbi. *Hybridizing exact methods and metaheuristics : A taxonomy*. European Journal of Operational Research (EJOR), **volume 199**(3) :pages 620–629, 2009 december. Impact factor : 1.627.
- [R7] L. Jourdan, O. Schutze, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. *An analysis of the effect of multiple layers in the multi-objective design of conducting polymer composites*. Journal Materials and Manufacturing Processes, **volume 24**(3) :pages 350–357, 2009 March. Impact factor : 0.777.

- [R8] O. Schuetze, L. Jourdan, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. *New analysis of the optimization of electromagnetic shielding properties using conducting polymers and a multi-objective approach*. Polymers for Advanced Technologies, **volume 19**(7) :pages 762–769, 2008. Impact factor : 2.017.
- [R9] L. Vermeulen-Jourdan, C. Dhaenens, and E. Talbi. *Linkage disequilibrium study with a parallel adaptive ga*. International Journal of Foundation in Computer Science, **volume 16**(2) :pages 241–260, 2005. ISSN : 0129-0541, World Scientific. Impact factor : 0.554.
- [R10] L. Jourdan, C. Dhaenens, E. Talbi, and S. Gallina. *A data mining approach to discover genetic and environmental factors involved in multifactorial diseases*. Knowledge Based Systems, **volume 15**(4) :pages 235–242, 2002 May. ISSN 0950-7051. Impact factor : 0.924.

### Revues nationales avec comité de lecture

- [RN1] A. Liefoghe, L. Jourdan, M. Basseur, and E.-G. Talbi. *Métaheuristiques pour le flow-shop de permutation bi-objectif stochastique*. Revue d'Intelligence artificielle, **volume 22**(2) :pages 183–208, 2008.
- [RN2] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *ASGARD : un algorithme génétique pour les règles d'association*. Extraction de Connaissances et Apprentissage (ECA - Hermès), **volume 16**(6) :pages 657–683, 2002. ISBN : 2-7462-0637-4.

### Chapitres d'ouvrage

- [C1] C. Coello, C. Dhaenens and L. Jourdan. *Advances in Multi-objective Nature Inspired Computing, Studies in Computational Intelligence* , Chapitre Multi-Objective Combinatorial Optimization : Problematic and Context vol. 272, pp. 1–21, Springer, 2010
- [C2] A. Liefoghe, L. Jourdan, T. Legrand, J. Humeau and E-G. Talbi. *Advances in Multi-objective Nature Inspired Computing, Studies in Computational Intelligence* , Chapitre ParadisEO-MOEO : A Software Framework for Evolutionary Multi-objective Optimization volume 272, pp. 87-117, Springer, 2010
- [C3] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *the Encyclopedia of Data warehousing and Mining 2nd Edition*, Chapitre Evolutionary Data Mining for Genomics (revised version). Idea Group, 2008 April.
- [C4] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Computational Intelligence in Bioinformatics*, Chapitre Evolutionary Feature Selection for Bioinformatics, pages 117–139. Gary Fogel, Yi Ping, David Corne, 2007.
- [C5] M. Basseur, L. Jourdan, and E.-G. Talbi. *Parallel combinatorial Optimization*, Chapitre Towards parallel design of hybrids between Metaheuristics and exact methods, pages 163–186. Wiley, 2006 November.

- [C6] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *the Encyclopedia of Data warehousing and Mining*, Chapitre Evolutionary Data Mining for Genomics, pages 482–486. Idea Group, 2005 April.
- [C7] L. Jourdan, M. Khabzaoui, C. Dhaenens, and E.-G. Talbi. *Handbook of Bioinspired Algorithms and Applications*, Chapitre A hybrid metaheuristic for knowledge discovery in microarray experiments, pages 489–505. CRC Press, USA, 2005 October.
- [C8] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Evolutionary Computation in Bioinformatics.*, Chapitre Discovery of Genetic and Environmental Interactions in Disease Data using Evolutionary Computation, pages 297–316. Morgan Kaufmann, 2002.

### Conférences Internationales avec actes et comités de sélection

- [I1] M. Khouadjia, E. Alba, L. Jourdan and E-G. Talbi, *Mutli-Swarm Optimization for Dynamic Combinatorial Problems : a case study on Dynamic Vehicle Routing Problem* In *International Conference on Swarm Intelligence, ANTS 2010, LNCS*.
- [I2] D. Sulieman, L. Jourdan and E-G. Talbi, *Using Multiobjective metaheuristics to solve VRP with uncertain demands* In *CEC 2010, IEEE*.
- [I3] S. Mesmoudi, J. Tavares, L. Jourdan and E-G. Talbi, *Variable Genetic Operator Search for the Molecular Docking Problem* In *Evobio 2010, LNCS*, pages 1-12, 2010.
- [I4] W. Cancino, L. Jourdan, E-G. Talbi and A. Delbem, *Parallel multi-objective approaches for inferring phylogenies* In *Evobio 2010, LNCS*, pages 26-37, 2010.
- [I4] W. Cancino, L. Jourdan, E-G. Talbi and A. Delbem, *A parallel multi-objective evolutionary algorithm for phylogenetic inference* In *LION 4, LNCS*, 2010.
- [I5] A. Liefoghe, L. Jourdan, and E.-G. Talbi. *A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework.* In *Computational Intelligence in Multi-Criteria Decision-Making, 2009. MCDM '09. IEEE Symposium on*, pages 88–95. 2009.
- [I5] A. Liefoghe, S. Mesmoudi, J. Humeau, L. Jourdan, and E.-G. Talbi. *A study on dominance-based local search approaches for multiobjective combinatorial optimization.* In *SLS*, pages 120–124. LNCS 2009.
- [I6] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and D. Horvath. *Parallel multi-objective algorithms for the molecular docking problem.* In *Conference in Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2008.
- [I7] A. Liefoghe, L. Jourdan, M. Basseur, E.-G. Talbi, and E. Burke. *Metaheuristics for the bi-objective ring star problem.* In LNCS, editor, *Evocop 2008*, volume 4972 of LNCS, pages 206–217. 2008.
- [I8] A. Liefoghe, L. Jourdan, N. Jozefowicz, E.-G. Talbi, and E. Burke. *On the integration of a TSP heuristic into an EA for the bi-objective ring star problem.* In *International*



- Workshop on Hybrid Metaheuristics (HM 2008)*, volume 5296 of *LNCS*, pages 117–130. Springer Berlin / Heidelberg, 2008.
- [I9] E.-G. Talbi, L. Jourdan, J. Garcia-Nieto, and E. Alba. *Comparison of population based metaheuristics for feature selection : Application to microarray data classification*. In *IEEE/ACS International Conference on Computer Systems and Applications, 2008. AICCSA 2008.*, pages 45 – 52. IEEE, 2008.
- [I10] E. Alba, J. García-Nieto, L. Jourdan, and E.-G. Talbi. *Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms*. In *IEEE Congress on Evolutionary Computation CEC-07*, pages 284–290. IEEE, IEEE, Singapore, 2007 Sep.
- [I11] J. García-Nieto, L. Jourdan, E. Alba, and E. G. Talbi. *A comparison of pso and ga approaches for gene selection and classification of microarray data*. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*. ACM Press, London, UK, 2007 July.
- [I12] L. Jourdan, T. Legrand, O. Schuetze, E.-G. Talbi, and J.-L. Wojkiewicz. *A multiobjective genetic algorithm to optimize electromagnetic properties of conducting polymer composites in the microwave band*. In *IESM 2007*. Beijing China, 2007.
- [I13] A. Liefoghe, M. Basseur, L. Jourdan, and E.-G. Talbi. *Multi-objective optimization for stochastic problems : an application to the flow-shop scheduling problem*. In *EMO 2007*, volume LNCS 4408, pages 386–400. 2007.
- [I14] A. Liefoghe, M. Basseur, L. Jourdan, and E.-G. Talbi. *ParadisEO-MOEO : a framework for multiobjective optimization*. In *EMO 2007*, volume 4408 of *LNCS*, pages 457–471. 2007.
- [I15] O. Schuetze, L. Jourdan, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. *A multi-objective approach to the design of conducting polymer composites for electromagnetic shielding*. In *EMO 2007*, volume 4403 of *LNCS*, pages 590–603. Matsushima, Japan, 2007.
- [I16] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and C. Rolando. *Preliminary work on evolutionary identification of protein variants and new proteins on grids*. In *20th International Conference on Advanced Information Networking and Applications - HIPcomb 2006*, volume 2, pages 583–587. IEEE, 2006 April.
- [I17] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and C. Rolando. *Protein Sequencing with an Adaptive Genetic Algorithm from Tandem Mass Spectrometry*. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1412–1419. IEEE, 2006 July 16-21.
- [I18] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Using datamining techniques to help metaheuristics : A short survey*. In *Hybrid Metaheuristics (HM'06)*, volume LNCS 4030, pages 57–69. Springer Berlin / Heidelberg, 2006.
- [I19] L. Jourdan, D. Corne, S. Savic, and G. Walters. *Hybridising rule induction and multi-objective evolutionary search for optimising water distribution systems*. In I. C. Press, editor, *Fourth International Conference on Hybrid Intelligent Systems , IEEE HIS 2004*, pages 434–439. 2005.

- [I20] L. Jourdan, D. Corne, S. Savic, and G. Walters. *Preliminary investigation of the 'learnable evolution model' for faster/better multiobjective water systems design*. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, LNCS 3410, pages 841–855. 2005.
- [I21] L. Vermeulen-Jourdan, C. Dhaenens, and E.-G. Talbi. *Clustering nominal and numerical data : a new distance concept for an hybrid genetic algorithm*. In *EvoCop'2004, 4th European Conference on Evolutionary Computation in Combinatorial Optimization*, volume LNCS 3004, pages 220–229. 2004 April.
- [I22] L. Vermeulen-Jourdan, C. Dhaenens, and E.-G. Talbi. *A parallel adaptive genetic algorithm for linkage disequilibrium in genomics*. In *NIDISC'2004 workshop, IEEE IPDPS*, page p167b. 2004 April.
- [I23] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Discovering haplotypes in linkage disequilibrium mapping with an adaptive genetic algorithm*. In *Applications of Evolutionary Computing, EvoWorkshops2003 : EvoBIO*, volume 2611 of LNCS, pages 66–75. Springer-Verlag, 2003 April.
- [I24] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Rules extraction in linkage disequilibrium mapping with an adaptive genetic algorithm*. In *European Conference on Computational Biology (ECCB) 2003*, pages 29–32. 2003. Paris, France.
- [I25] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *A genetic algorithm for feature selection in data-mining for genetics*. In *Metaheuristic International Conference (MIC) 2001*, pages 29–34. 2001. Porto, Portugal.
- [I26] L. Jourdan, C. Dhaenens, E.-G. Talbi, and S. Gallina. *An optimization approach to mine genetic data*. In *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS'2001)*, pages 40–46. 2001 June. Las Vegas, USA.

### Conférences Internationales

- [p1] S. Verel, L. Jourdan, C. Dhaenens, A. Liefoghe. *Set-based Multiobjective Fitness Landscapes : Definition, Properties*. In *4th workshop on Theory of Randomized Search Heuristics (ThRaSH 2010)* Paris, France, 2010
- [p2] A. Liefoghe, L. Jourdan, E.-G. Talbi. *Indicator-based approaches for multiobjective optimization in uncertain environments : An application to multiobjective scheduling with stochastic processing times*. In *25th Mini-EURO Conference : Uncertainty and Robustness in Planning and Decision Making (URPDM 2010)*. Coimbra, Portugal, 2010.
- [p3] E.-G. Talbi A. Liefoghe, L. Jourdan. *Metaheuristics and hybrid metaheuristics for the bi-objective ring star problem*. In *International Conference on Metaheuristics and Nature Inspired Computing (META 2008)*. 2008.
- [p4] E.-G. Talbi, J.-C. Boisson, L. Jourdan and D. Horvath. *A new tri-objective model for the flexible docking problem*. In *International conference on Metaheuristics and Nature Inspired Computing (META'08)*. 2008.

- [p5] M. Khouadjia, L. Jourdan, and E.-G. Talbi. *A particle swarm for the resolution of the dynamic vehicle routing*. In *International conference on Metaheuristics and Nature Inspired Computing (META'08)*. 2008.
- [p6] J.-C. Boisson, E.-G. Talbi, and L. Jourdan. *De Novo Protein Sequencing using Evolutionary Algorithm*. In *22<sup>nd</sup> European Conference on Operational Research (EURO XXII)*, page 98. 2007 July 8-11. Prague, Czech Republic.
- [p7] J.-C. Boisson, O. Guerre, L. Jourdan, E.-G. Talbi, and C. Rolando. *ASCQ\_ME : a New Engine for Protein and Post-Translational Modification Identification from MS and MS/MS Raw Spectrum without Mass List Extraction*. In *55<sup>th</sup> ASMS Conference on Mass Spectrometry*. 2007 June 3-7. Indianapolis, Indiana, USA.
- [p8] A. Liefoghe, L. Jourdan, and E.-G. Talbi. *Software framework for multi-objective optimization*. In *22nd European Conference on Operational Research*. 2007.
- [p8] L. Jourdan, J.-C. Boisson, P. Laurence, E.-G. Talbi, and C. Rolando. *An online engine for peptide mass fingerprinting from spectrum*. In *Recomb 2006, Annual International Conference on Research in Computational Molecular Biology*, page 1. Venezia, Italia, 2006. Poster.
- [p10] L. Jourdan, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. *Mono and multi-objective continuous optimization for conducting polymer composites*. In *ECCO/CO 2006*, page 1. Porto, Portugal, 2006.
- [p11] J.-C. Boisson, L. Jourdan, E.-G. Talbi, C. Cren-Olivé, and C. Rolando. *ASCQ\_ME : a new engine for peptide mass fingerprinting directly from mass spectrum without mass list extraction*. In *HUPO 2005, Human Proteome Organisation Annual World congress*. 2005.
- [p12] L. Jourdan, C. Dhaenens, E.-G. Talbi, S. Gallina, and C. Dina. *A branch and bound procedure to analyse multifactorial diseases*. In *European Mathematical Genetics Meeting*, pages 19–20. 2001. Lille, France.

### Conférences nationales avec comité de sélection

- [n1] A. Liefoghe, J. Humeau, S. Mesmoudi, L. Jourdan, E.-G. Talbi. *Recherche locale basée sur une relation de dominance pour l'optimisation combinatoire multiobjectif*. In *Dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF 2010)*.
- [n2] R. Chevrier, A. Liefoghe, L. Jourdan. *Comparaison de trois algorithmes évolutionnaires Pareto pour le DARP multicritère*. In *Dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF 2010)*.
- [n3] M.-E. Marmion, L. Jourdan, C. Dhaenens. *Une nouvelle mesure de distance pour l'ACVRP*, In *Dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF 2010)*.

- [n4] L. Jourdan, E.-G. Talbi, J.-C. Boisson, S. Mesmoudi. *Paradiseo-mo : une plateforme pour le développement de métaheuristiques à base de solution unique*. In *Dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF 2009)*. 208–209. 2009.
- [n5] J.-C. Boisson, L. Jourdan, G. Even, E.-G. Talbi, and D. Horvath. *Docking moléculaire multi-objectif par algorithme évolutionnaire*. In *JOBIM*. 2008.
- [n6] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and D. Horvath. *Docking moléculaire multi-objectif par algorithme évolutionnaire*. In *Roadef*. 2008.
- [n7] G. Even, L. Jourdan, C. Boutroue, C. Dhaenens, and E.-G. Talbi. *Evolutionary feature selection plugin for base*. In *JOBIM*. 2007.
- [n8] T. Legrand, O. Schütze, L. Jourdan, C. Coello-Coello, and E.-G. Talbi. *Stratégies mimétiques pour les essaims de particules et implémentation sous paradiseo*. In *second séminaire francophone sur l'optimisation par essaim particulaire (OEP)*. 2007.
- [n9] E.-G. Talbi, L. Jourdan, J. García-Nieto, and E. Alba. *Sélection d'attributs de puce à ADN par essaim de particules*. In *second séminaire francophone sur l'optimisation par essaim particulaire (OEP)*. 2007.
- [n10] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and C. Rolando. *Identification de nouvelles protéines et de variants de protéines connues par méthode évolutionnaire*. In *ROADEF 2006*. 2006.
- [n11] A. Liefoghe, M. Basseur, L. Jourdan, and E.-G. Talbi. *Optimisation multi-objectif sous incertitude pour le flow-shop de permutation*. In *META'06*. 2006.
- [n12] J.-C. Boisson, L. Jourdan, E.-G. Talbi, C. Cren-Olivé, and C. Rolando. *ASCQ\_ME : a new engine for peptide mass fingerprinting directly from mass spectrum without mass list extraction*. In *SCBA 2005, Symposium de Chimie et Biologie Analytiques*. 2005.
- [n13] L. Jourdan and D. Corne. *LEMMO : Application au design de réseau de distribution d'eau potable*. In *Sixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision, Roadef 2005*, pages 223–224. 2005. Tours, France.
- [n14] M. Khabzaoui, L. Jourdan, C. Dhaenens, and E. Talbi. *Approche évolutionnaire multi-critère pour les règles d'association en génomique*. In *Roadef 2003*, pages 173–174. 2003. Avignon, France.
- [n15] C. Dhaenens, L. Jourdan, and E.-G. Talbi. *Datamining pour la génomique : approche par métaheuristiques*. In *Journées ouvertes Biologie Informatique Mathématiques (JOBIM 2002)- Poster*, pages 81–82. 2002. St Malo, France.
- [n16] L. Jourdan, C. Dhaenens, and E. Talbi. *Algorithmes génétiques pour l'extraction de connaissances en génomique*. In *Roadef 2002*, pages 170–171. 2002. Paris, France.

### Autres

- [a1] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Algorithme génétique pour l'analyse des maladies multifactorielles*. Journée Evolutionnaire Trimestrielle (JET'6), 2001.
- [a2] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *A genetic algorithm for feature selection in data-mining for genetics*. Mini symposium de BioInformatique, 2000. Vrij Universiteit, De Boelelaan 1083, Amsterdam.
- [a3] J.-C. Boisson, L. Jourdan, E.-G. Talbi, and C. Rolando. *Un nouveau moteur de recherche en protéomique travaillant directement à partir du spectre de masse sans extraction de la liste des pics monoisotopiques*. Forum LABO & Forum BIOTECH 2006.
- [a4] L. Jourdan. *Du datawarehouse au datamining*. Séminaire, 16 Octobre 2003, Béthune.
- [a5] L. Jourdan. *Un algorithme génétique pour les règles d'association : application à la génomique*. Séminaire, 30 Mai 2002, CERIM, France.

### Rapports de Recherche, thèse

- [RR1] A. Liefooghe, L. Jourdan, and E.-G. Talbi. *Metaheuristics and their hybridization to solve the bi-objective ring star problem : a comparative study*. Technical Report RR-6515, INRIA, 2008.
- [RR2] L. Jourdan. *Métaheuristiques pour l'extraction de connaissances : Application à la génomique*. Ph.D. thesis, Université des Sciences et Technologies de Lille, 2003 November.
- [RR3] L. Jourdan. *Data-Mining pour la Bio-Informatique*. Master's thesis, Université des Sciences et Technologies de Lille, 2000.

### Invited tutorial

*Combinatorial optimization in Bioinformatics* with C. Dhaenens PRIB 2010.

### En révision

- [S1] . Liefooghe, J. Humeau, S. Mesmoudi, L. Jourdan and E-G. Talbi *On Dominance-based Local Search : Design, Implementation and Experiments on Multiobjective Scheduling and Traveling Salesman Problems* En revision, Journal of Heuristic.
- [S2] . R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan and E-G. Talbi *A comparative study between adaptive PSO and VNS for the Vehicle Routing Problem with Dynamic Requests* En revision, Soft Computing Journal.

## Chap. 10

# Publications Majeures

Les publications sont présentées en deux groupes : les publications relatives au manuscrit puis quelques publications significatives sur des activités de recherche non présentées dans le manuscrit. L'annexe comporte les articles suivants :

- A. Liefoghe, L. Jourdan, T. Legrand, J. Humeau and E-G. Talbi. *ParadisEO-MOEO : A Software Framework for Evolutionary Multi-objective Optimization* **Advances in Multi-objective Nature Inspired Computing, Studies in Computational Intelligence** . volume 272, pp. 87-117, Springer, 2010
- A. Liefoghe, L. Jourdan, and E. G. Talbi. *Metaheuristics and cooperative approaches for the bi-objective ring star problem.* **Computers & Operations Research**, Vol. **37**(6) : pp. 1033–1044, 2010. ISSN 0305-0548. Impact factor : 1.366.
- J.-C. Boisson, L. Jourdan, E.-G. Talbi, and D. Horvath. *Single- and mutli-objective cooperation for the flexible docking problem.* **Journal of Mathematical Modelling and Algorithms**, Vol. **9**(2) : pp. 195–208, 2010.
- L. Jourdan, M. Basseur, and E.-G. Talbi. *Hybridizing exact methods and metaheuristics : A taxonomy.* **European Journal of Operational Research (EJOR)**, Vol. **199**(3) : pp. 620–629, 2009 december. Impact factor : 1.627.
- L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Using datamining techniques to help metaheuristics : A short survey.* **Hybrid Metaheuristics (HM'06)**, LNCS 4030, pp. 57–69. Springer Berlin / Heidelberg, 2006.
- J. Garcia-Nieto, E. Alba, L. Jourdan, and E.-G. Talbi. *Sensitivity and specificity based multiobjective approach for feature selection : Application to cancer diagnosis.* **Information Processing Letters**, Vol. **109** :pp. 887–896, 2009. Impact factor : 0.706.
- O. Schuetze, L. Jourdan, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. *New analysis of the optimization of electromagnetic shielding properties using conducting polymers and a multi-objective approach.* **Polymers for Advanced Technologies**, Vol. **19**(7) : pp. 762–769, 2008. Impact factor : 2.017.

