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 :
- Examinateur : PIERRE BOULET, Professeur des Universités, Lille I
- Examinateur : 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'enseignent-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 perservérer, 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

I Introduction générale **1**

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

3 ParadisEO **13**
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

II Coopération de méthodes dans le cadre déterministe **27**

4 Coopération métaheuristiques-métaheuristiques **29**
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

5 Coopération métaheuristiques-méthodes exactes 51
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

6 Coopération métaheuristiques-extraction de connaissances 65
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

III Vers la coopération de méthodes dans le cadre stochastique 79

7 Résolution de problèmes dans un cadre incertain 85
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 incertaine (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

8 Méthodes coopératives pour l'optimisation dynamique 105
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

IV Conclusion Générale et Perspectives 121

V Annexes 143

Annexes 145

9 CV Détaillé 145
9.1 Publications . . . 155
9.1.1 Invited tutorial . . . 164

10 Publications Majeures 165

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 [1] dont je suis actuellement la coordinatrice avec Nicolas Jozefowiez (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. `htpp://www.lifl.fr/PM2O/` (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 [2].

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 Liefooghe 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 Edhimbourg, 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 Jozefowiez (LAAS, INSA Toulouse, France)
- Thomas Legrand (INRIA Lille Nord Europe, France)
- Arnaud Liefooghe (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 Materiaux 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 cøeur 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 Ω et une fonction objectif $f : \Omega \to \mathbb{R}$ associant une valeur à chaque solution admissible. La résolution du problème consiste à déterminer la (ou les) solution(s) de Ω 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.

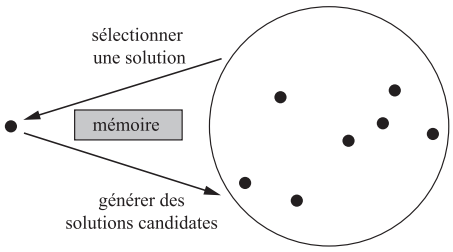


Fig. 2.1 – Illustration des principes généraux d'une métaheuristique à base de solution unique (S-META).

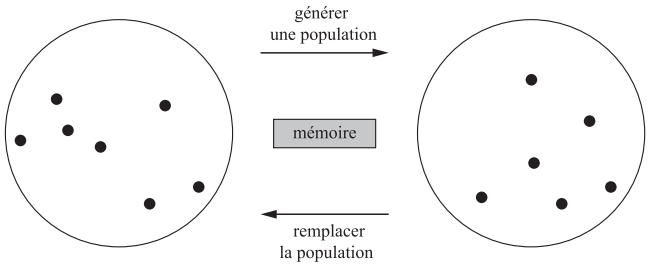


Fig. 2.2 – Illustration des principes généraux d'une métaheuristique à base de population (P-META).

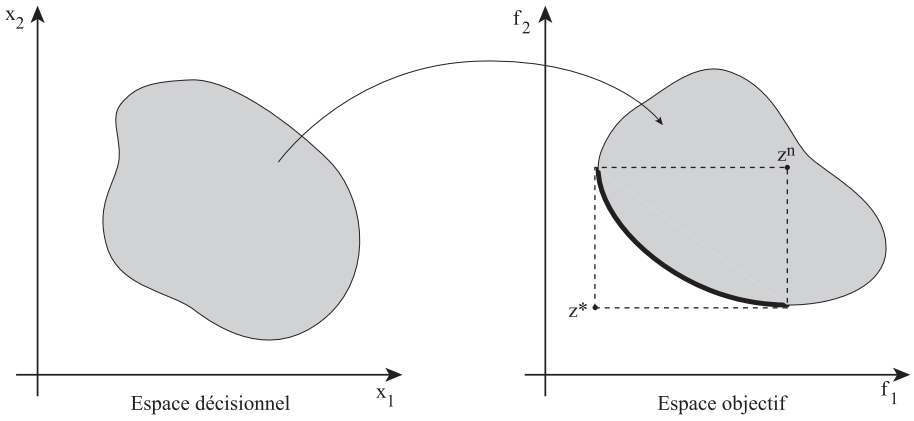


Fig. 2.3 – Espace décisionnel et espace objectif d'un problème d'optimisation multiobjectif (exemple avec deux variables de décision X_1 et X_2 et deux fonctions objectif f_1 et f_2).

Définition d'un problème d'optimisation combinatoire multiobjectif

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

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

où n est le nombre d'objectifs ($n \geq 2$), X est l'ensemble des solutions admissibles dans l'*espace décisionnel*, et $x = (x_1, x_2, \ldots, x_k) \in X$ 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 $f : X \to Z$ tel que $z = (z_1, z_2, \ldots, z_n) = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$, 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 supposerons 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 $\forall i \in \{1, 2, \ldots, n\}$, $z_i \leq z'_i$ et $\exists j \in \{1, 2, \ldots, n\}$ 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 $\forall z' \in Z$, $z' \not\succ z$.

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

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

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

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 ϵ [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, $Z_N^\star$ 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 $S^\star$ 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 $S^\star$.
L'avantage de cette mesure est qu'elle possède une complexité temporelle faible.

ϵ-indicator 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* ($I^1_{\epsilon+}$) peut être défini comme suit.

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

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.

$$I^-_H(A) = I_H(R) - I_H(A). \tag{2.2}$$

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^R_1 est R_1 quand un ensemble de référence est utilisé et permet alors d'avoir un ordre total sur les ensembles d'approximation.

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

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

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

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 monoobjectif qu'en multiobjectif. Après un positionnement sur les métaheuristiques coopératives, nous présenterons différentes cooperations 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. Le 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

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**

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 Liefooghe.

3.1 Contexte

3.1.1 Description

ParadisEO [1] 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*) [2] [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 [3]. 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.

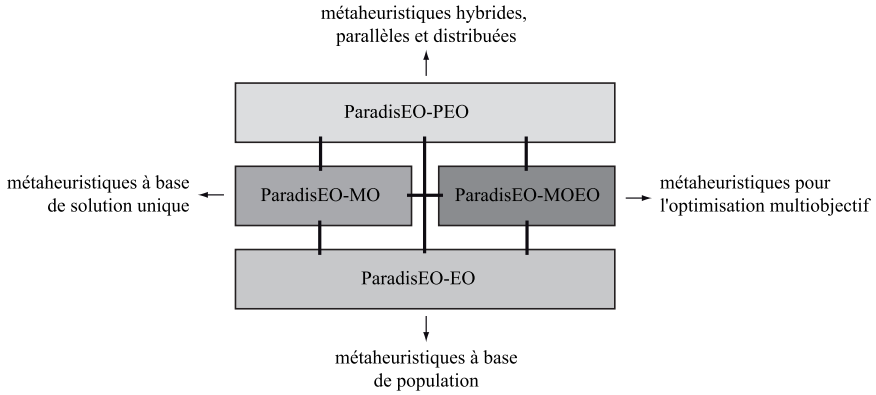


Fig. 3.1 – Les modules de la plateforme ParadisEO.

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. Liefooghe 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 intiale 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 faîte, 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 implementée indépendemment.
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 selectionné.

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 verifié, 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 recherce.

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 : Hll-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-Climing 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 stantards (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 Liefooghe.

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 [4], 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 Liefooghe [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.

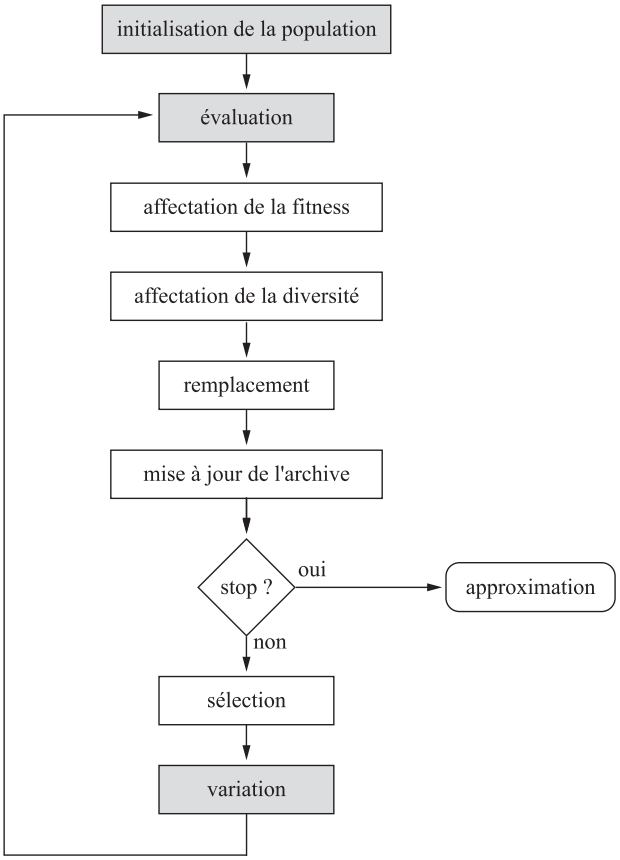


Fig. 3.2 – Étapes générique d'un algorithmes évolutionnaire multiobjectif.

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 *crowding*	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 le 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**.

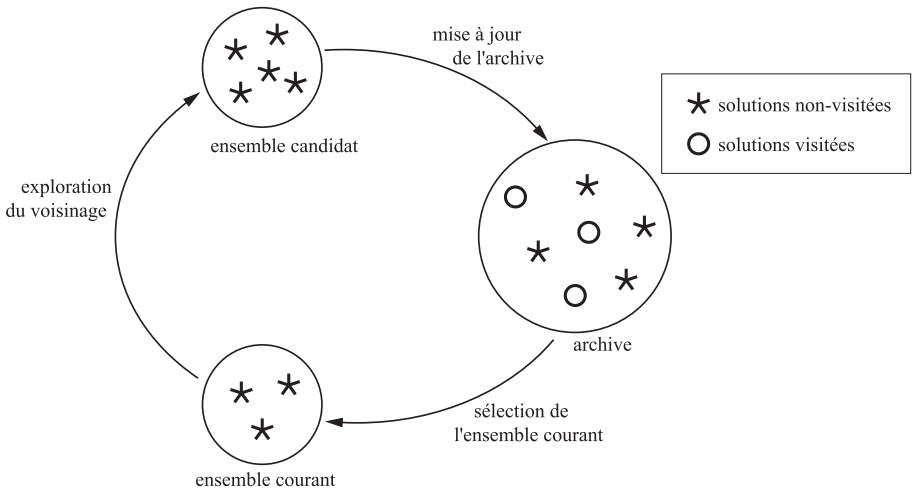


Fig. 3.3 – Étapes du modèle DMLS.

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 chapitreé [112, 114, 116, 115] ainsi que dans la thèse d'Arnaud Liefooghe [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($|A| : 1 + 1)^A$ [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 boite 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($|A| : 1 + 1)^A$ [1].

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

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**

Ce chapitre a donné lieu aux publications suivantes :

- A. Liefooghe 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 Liefooghe. 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.

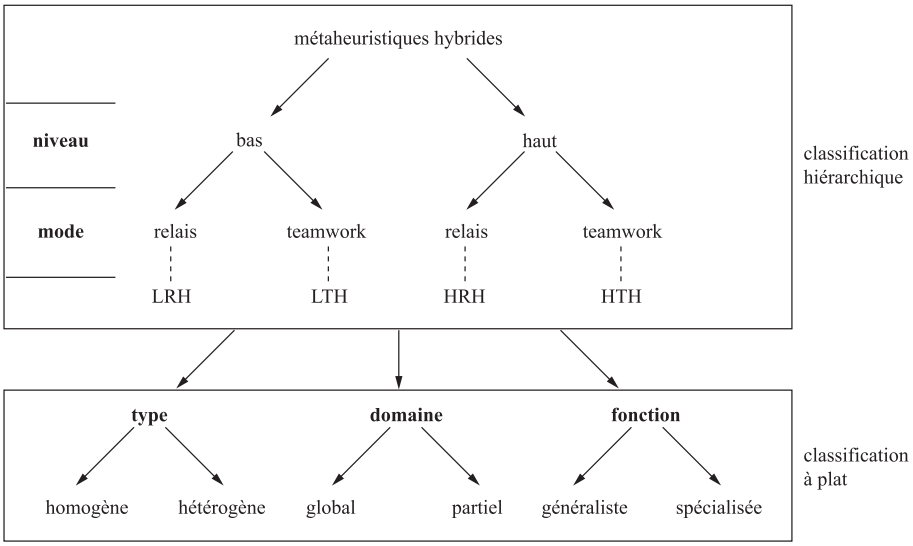


Fig. 4.1 – Taxonomie servant pour la description de la coopération de méthodes.

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 classifier 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

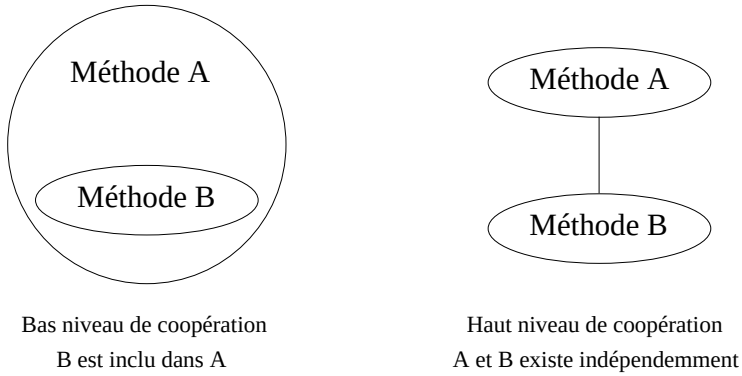


Fig. 4.2 – Les différents niveaux de coopération.

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*

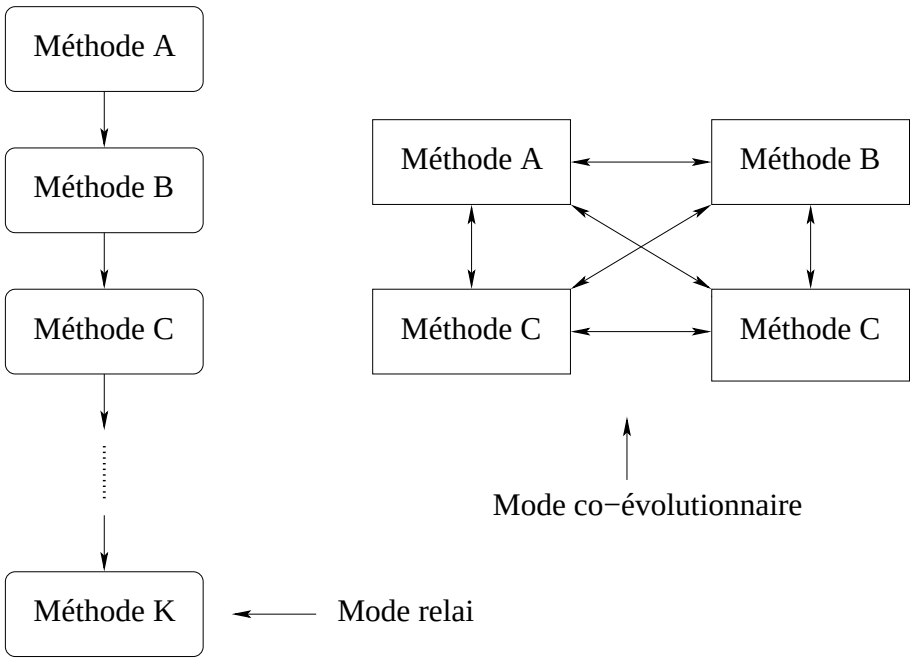


Fig. 4.3 – Les différents modes de coopération.

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'algorithme évolutionnaire 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].

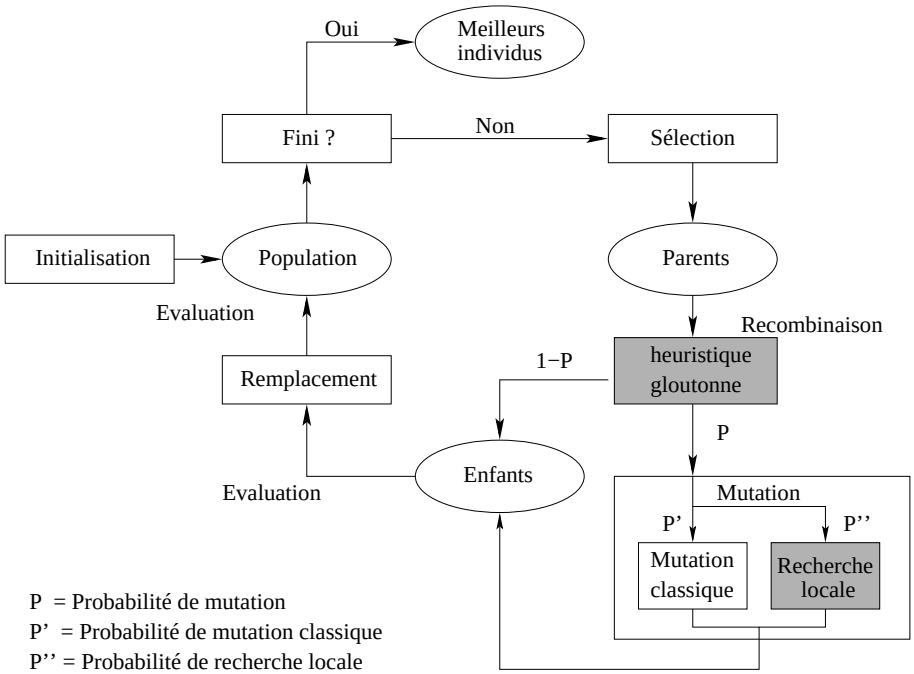


Fig. 4.4 – Schéma d'une coopération de type co-évolutionnaire de bas niveau (*LTH*) : un algorithme génétique est hybridé avec une heuristique gloutonne (recombinaison) et une recherche locale (mutation).

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.

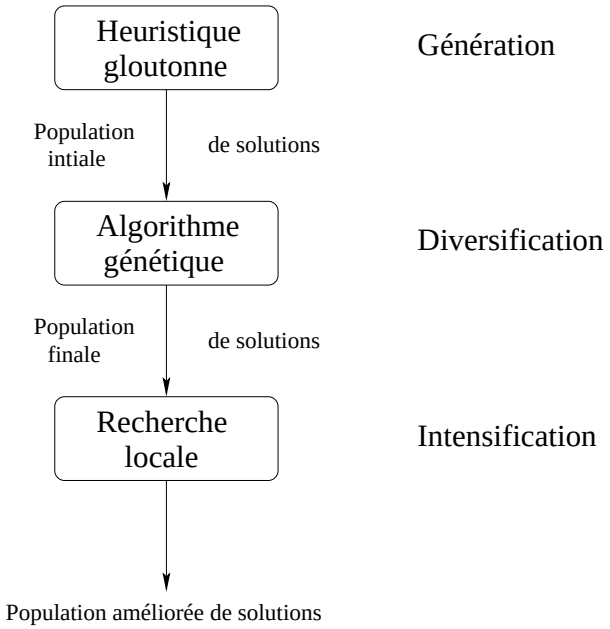


Fig. 4.5 – Schéma d'une coopération de type relais de haut niveau (*HRH*) : une heuristique glouton génère la population initiale d'un algoritme génétique et ensuite la population finale de l'AG est améliorée grâce à l'application d'une recherche locale sur chaque individu de la population.

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-

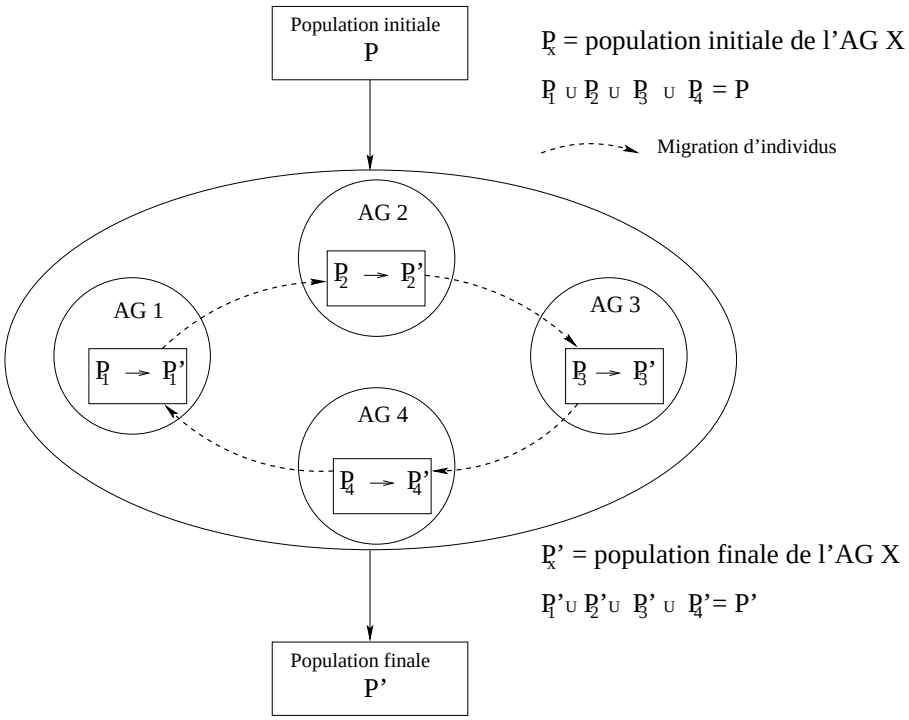


Fig. 4.6 – Schéma d'une coopération co-évolutionnaire de haut-niveau (HTH) : un ensemble d'algorithmes génétiques coopèrent selon un modèle insulaire incluant une migration circulaire des individus.

cité 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 dûs 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 recombinées, 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

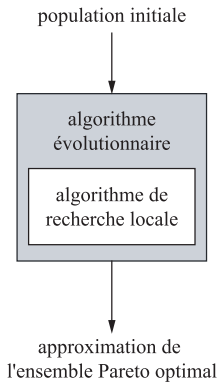


Fig. 4.7 – L'algorithme MOGLS : un exemple d'hybridation LTH.

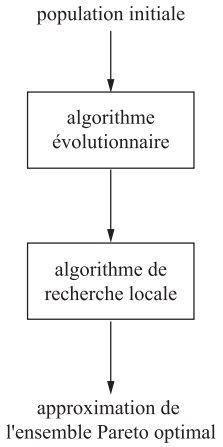


Fig. 4.8 – Un exemple typique d'hybridation HRH.

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

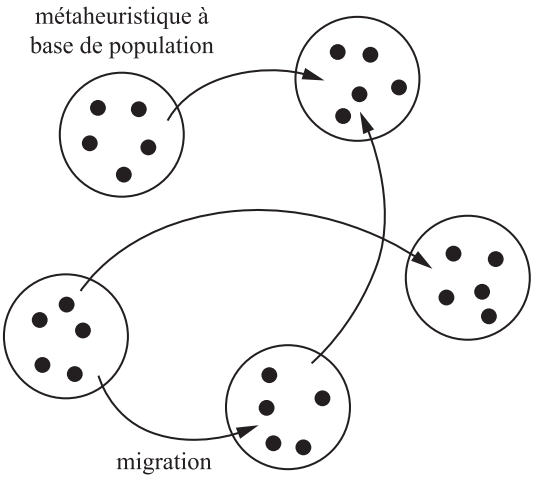


Fig. 4.9 – Le modèle en îles : un exemple d'hybridation HTH.

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-évolutionnaire 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-évolutionnaire 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.

$$f_1 : E_{total} = \quad E_{bonded_atoms} + E_{non_bonded_atoms} \tag{4.1}$$

avec

$$\begin{aligned} E_{bonded_atoms} = \quad & \textstyle\sum_{bonds} K_b(b-b_0)^2 \ + \ \sum_{angles} K_\theta(\theta-\theta_0)^2 \\ & + \ \textstyle\sum_{torsions} K_\phi(1-cos\ n(\phi-\phi_0)) \end{aligned} \tag{4.2}$$

$$\begin{aligned} E_{non_bonded_atoms} \ = \ & \textstyle\sum_{Van\ der\ Waals} \frac{K^a_{ij}}{d^{12}_{ij}} - \frac{K^b_{ij}}{d^6_{ij}} \ + \\ & \textstyle\sum_{Coulomb} \frac{q_i q_j}{4\pi\epsilon d_{ij}} \ + \\ & \textstyle\sum_{desolvation} \frac{Kq_i^2 V_j + q_j^2 V_i}{d^4_{ij}} \end{aligned} \tag{4.3}$$

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 :

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

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.

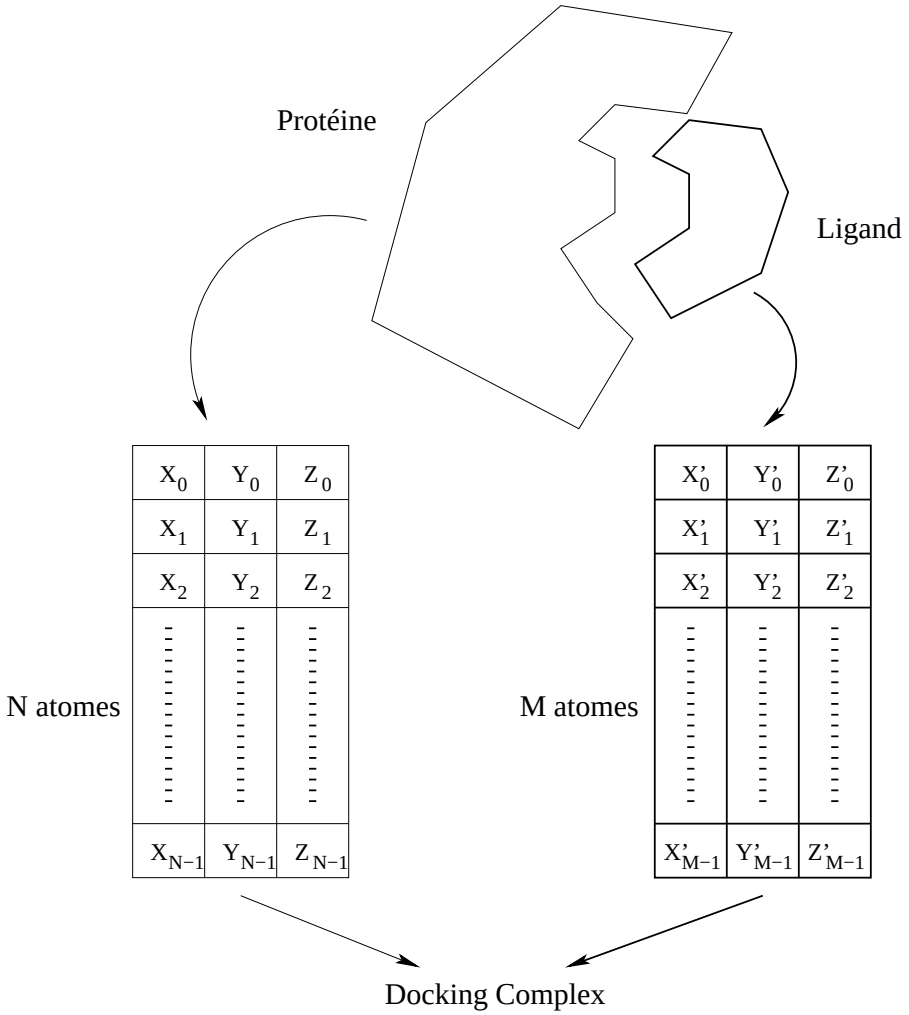


Fig. 4.10 – Représentation d'une solution dans l'algorithme génétique. N et M correspondent respectivement au nombre d'atomes composant le site et le ligand.

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).

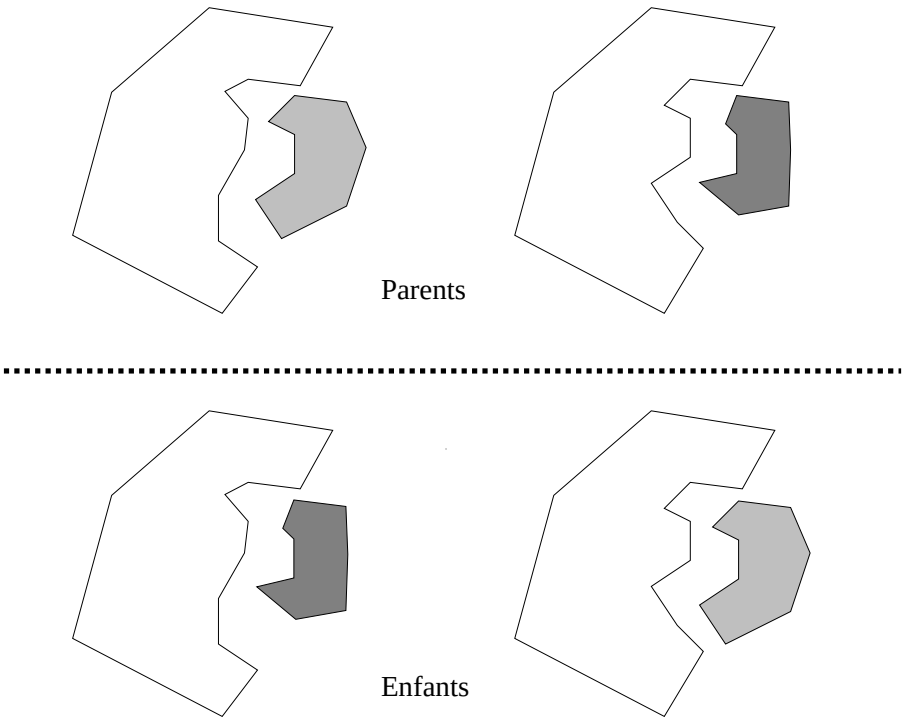


Fig. 4.11 – Opérateur de recombinaison de notre algorithme génétique.

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°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°;10°] pour les petites rotations et dans [30°;60°] 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 ***S**everal **M**utation in **O**ne 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-

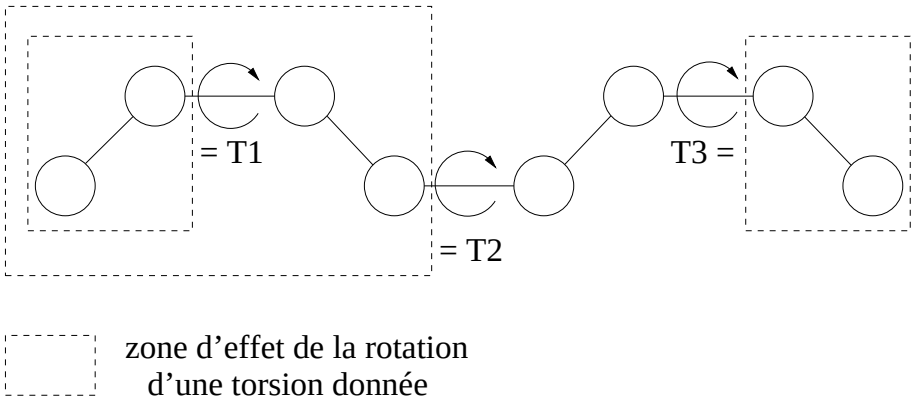


Fig. 4.12 – Impact de la rotation d'une torsion dans la modification de conformation d'une molécule donnée. Dans notre exemple, chaque rotation de torsion a un impact sur le plus petit morceau de la molécule (à gauche pour T1 et à droite pour T3) ou sur la partie gauche (T2)

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 :

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

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.

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

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**
Minimum	2	2,4	2	1,9	2,2	2,4
1er quartile	3,2	3,3	3,4	3,3	3,825	3,25
Médiane	**4,05**	4,5	**4,1**	4,35	**4,1**	4,6
Moyenne	**4,343**	4,729	5,126	5,169	4,733	4,776
3ème quartile	5,075	6	6,3	6,325	**4,775**	6
Maximum	10,5	10,8	11	12,6	14	8,5
Déviation standard	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.

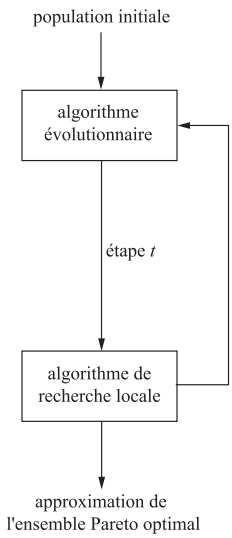


Fig. 4.13 – Une approche de coopération périodique (PCS).

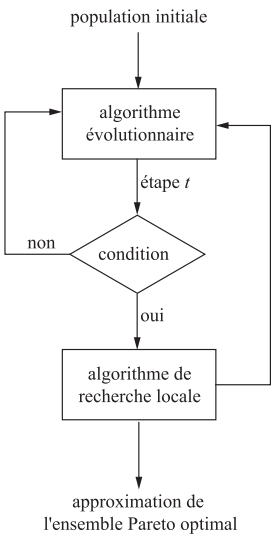


Fig. 4.14 – Une approche de coopération adaptative (ACS).

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 $V = \{v_1, v_2, \ldots, v_n\}$ 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.

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

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 :

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \tag{4.7}$$

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 $\mathrm{I}_{\epsilon+}^1$, 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

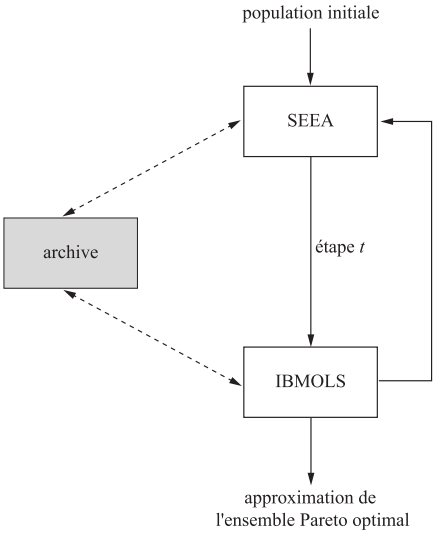


Fig. 4.15 – Instanciation du modèle PCS à l'aide des algorithmes SEEA et IBMOLS.

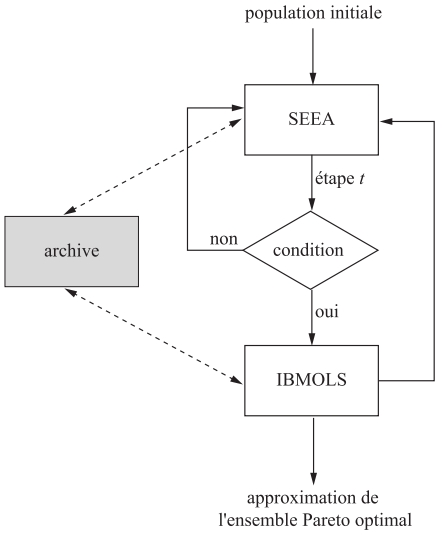


Fig. 4.16 – Instanciation du modèle ACS à l'aide des algorithmes SEEA et IBMOLS.

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

TABLE 4.5 – Comparaison des algorithmes selon l'indicateur $I^1_{\epsilon+}$. 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 $I^1_{\epsilon+}$-valeur moyenne ($\times 10^{-3}$).

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

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**

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 taxinomie 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 taxinomie présentée dans le chapitre précédent.

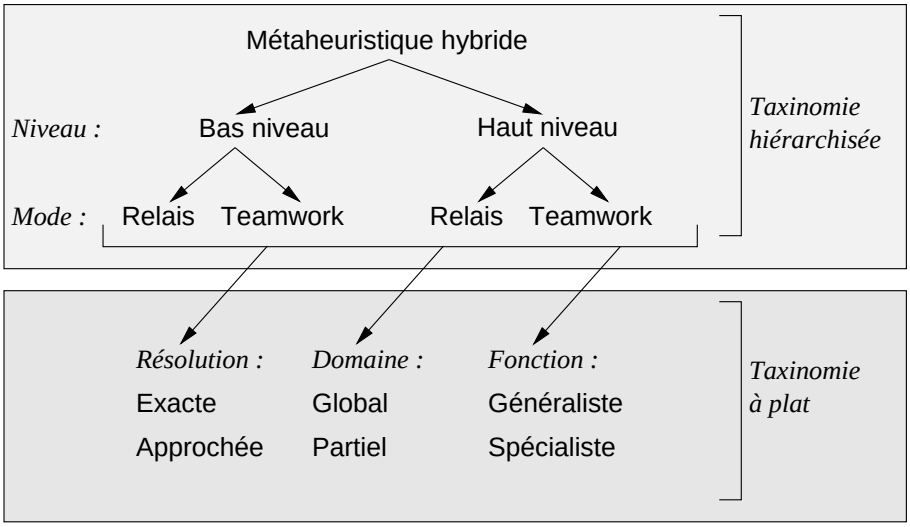


Fig. 5.1 – La taxonomie étendue pour l'hybridation entre métaheuristiques et méthodes exactes.

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 $B\&B$.
Jahuira *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

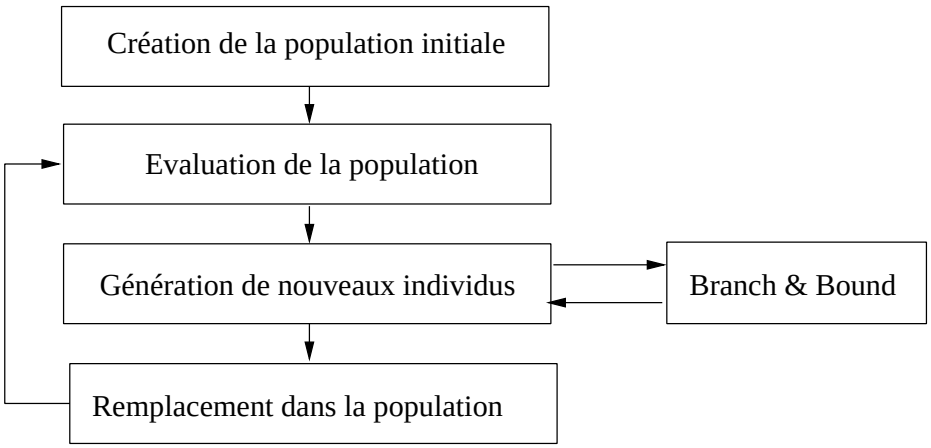


Fig. 5.2 – Schéma général de l'*AG* hybride de Cotta *et al.* [37] : un exemple de coopération de type **LTH**.

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 "α Branch & Bound" ($\alpha B\&B$) et un *AG* hybridé avec un algorithme

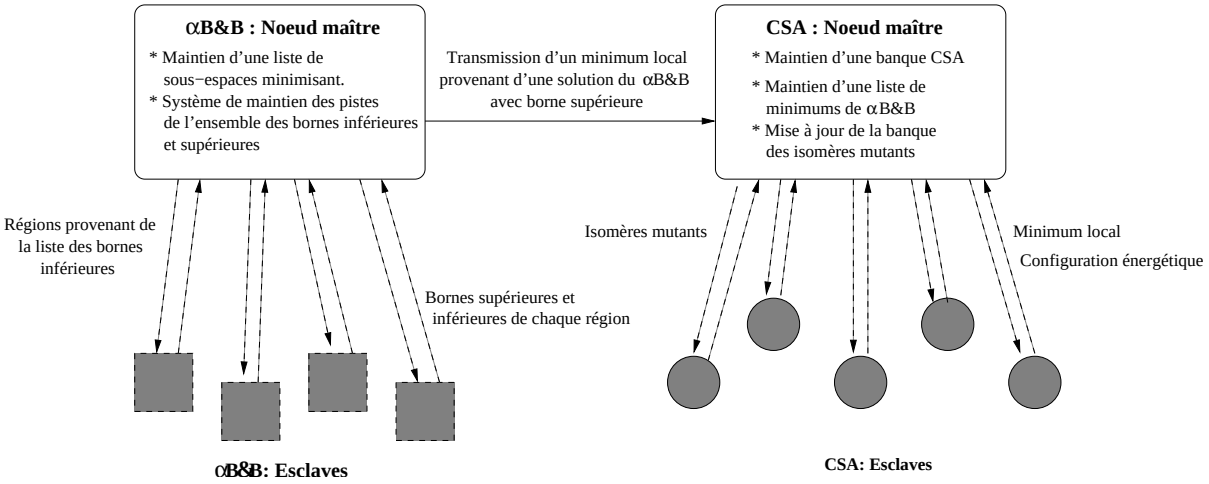


Fig. 5.3 – Exemple de coopération **HRH** pour la prédiction de structure de protéines [98].

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 coopérent en parallèle en se procurant mutuellement diverses informations, comme la montre le

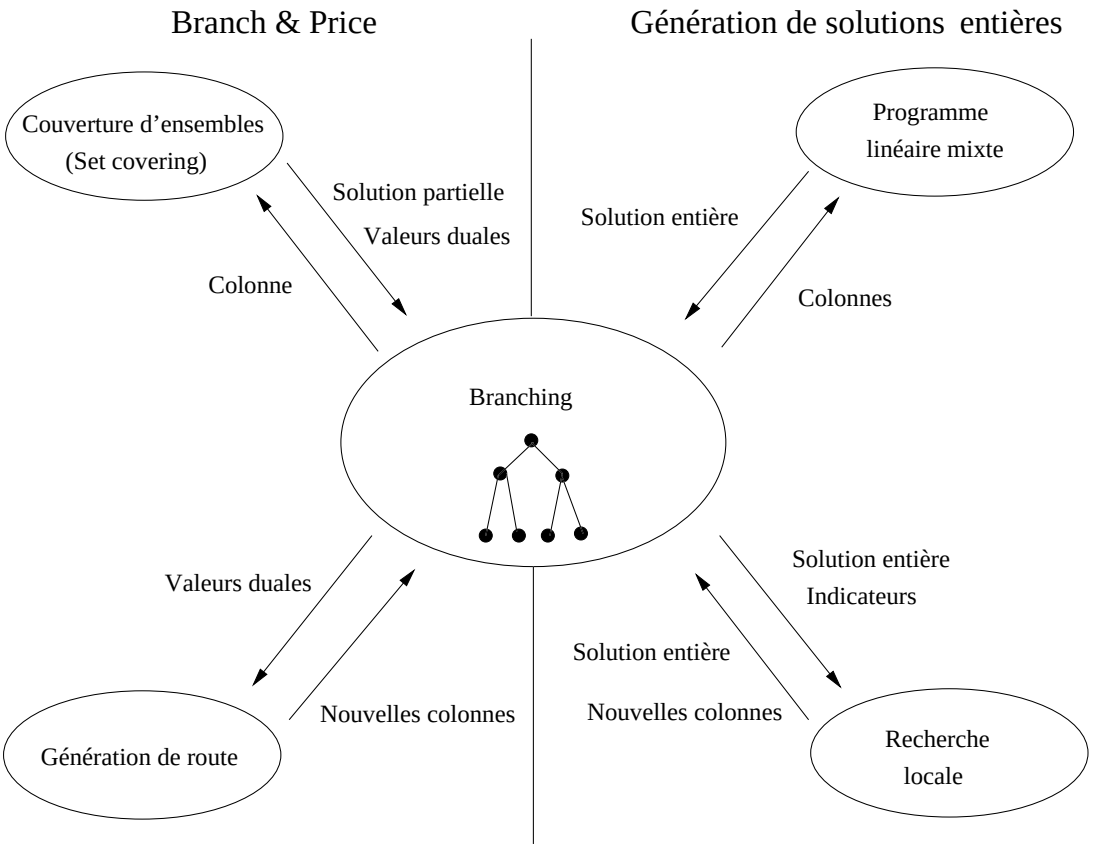


Fig. 5.4 – Exemple de coopération **HTH** pour le problème de tournées de véhicules [32].

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 *Mimausa*, 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'ordonnancement 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 résolvable 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 débuter 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 hydrides 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 classifier 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 taxonomie 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 métaheuristiques hybridées) ; par contre l'approche de résolution (approchée ou exacte), y est notamment ajoutée.

Pour les différentes métaheuristiques 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*)

$< algorithme\ coopératif > \longrightarrow < conception >< implémentation >$
$< conception > \longrightarrow < hiérarchique >< plat >$
$< hiérarchique > \longrightarrow < LRH > | < LTH > | < HRH > | < HTH >$
$< LRH > \longrightarrow$ LRH $(< métaheuristique >(< exact >)|(< exact >(< métaheuristique >))$
$< LTH > \longrightarrow$ LTH $(< métaheuristique >(< exact >)|(< exact >(< métaheuristique >))$
$< HRH > \longrightarrow$ HRH $((< métaheuristique >+< exact >)|(< exact >+< métaheuristique >))$
$< HTH > \longrightarrow$ HTH $((< métaheuristique >+< exact >)|(< exact >+< métaheuristique >))$
$< plat > \longrightarrow (< résolution >, < optimisation >, < fonction >)$
$< résolution > \longrightarrow$ exacte | approchée
$< optimisation > \longrightarrow$ globale | partielle
$< fonction > \longrightarrow$ générale | spécialiste
$< implémentation > \longrightarrow$ séquentielle | parallèle $< type\ parallèle >$
$< type\ parallèle > \longrightarrow$ statique | dynamique | adaptative
$< métaheuristique > \longrightarrow$ RS | AG | AM | SE | PG | RN | AD | RL | RLI | RT | HG | CF | RD | HS | CSA | $< algorithme\ coopératif >$
$< exact > \longrightarrow$ 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

suite à la page suivante ⟶

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(αB&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

suite à la page suivante ⟶

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 *p-median*
[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 Quatrelivre. 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

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

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 metaheuristics 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.

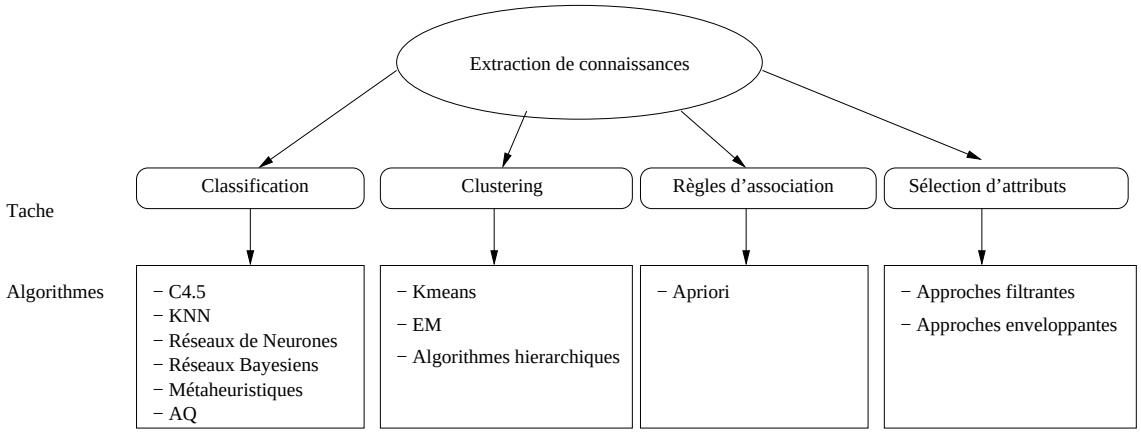


Fig. 6.1 – Vue synthétique des tâches et des algorithmes associés en data mining.

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 diagnostique 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.

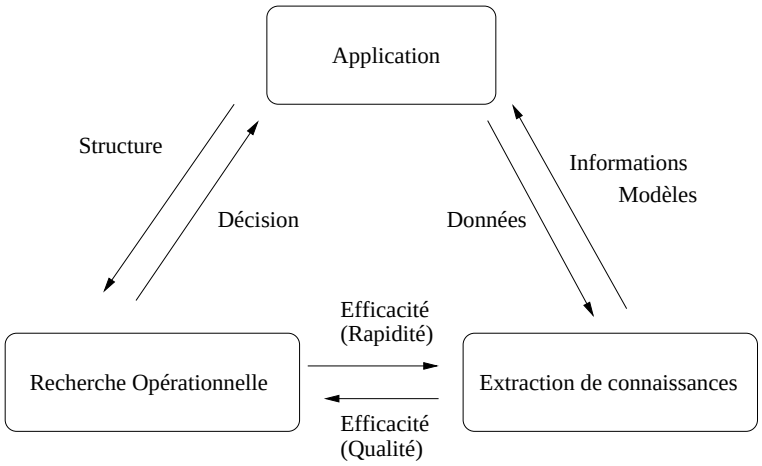


Fig. 6.2 – Interaction entre l'extraction de connaissances et recherche opérationnelle.

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].

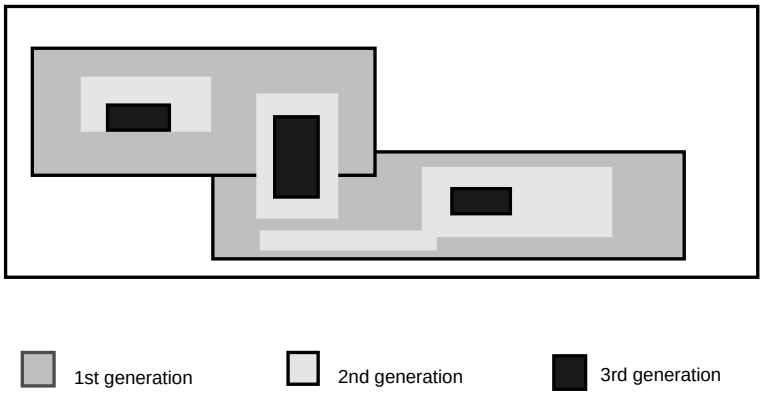


Fig. 6.3 – LEM : exemple de la réduction de l'espace de recherche.

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 métaheurisitques 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.

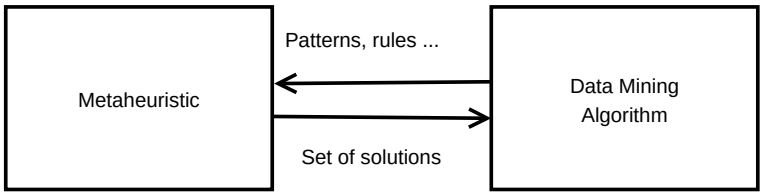


Fig. 6.4 – Algorithme hybride de heut niveau relais.

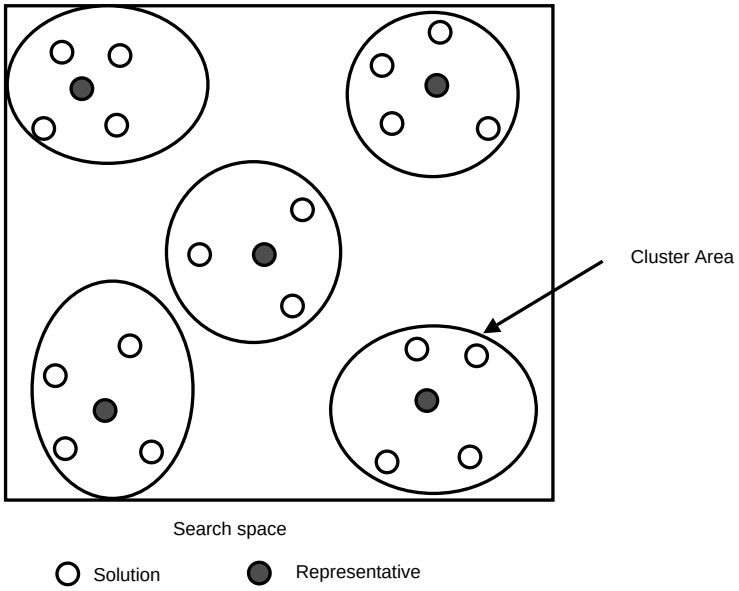


Fig. 6.5 – Exemple d'utilisation de représentants pour la phase d'évaluation en utilisant un algorithme de clustering.

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 classifier 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 [I20, 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 différentiation 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 i ème composante du codage choisi.

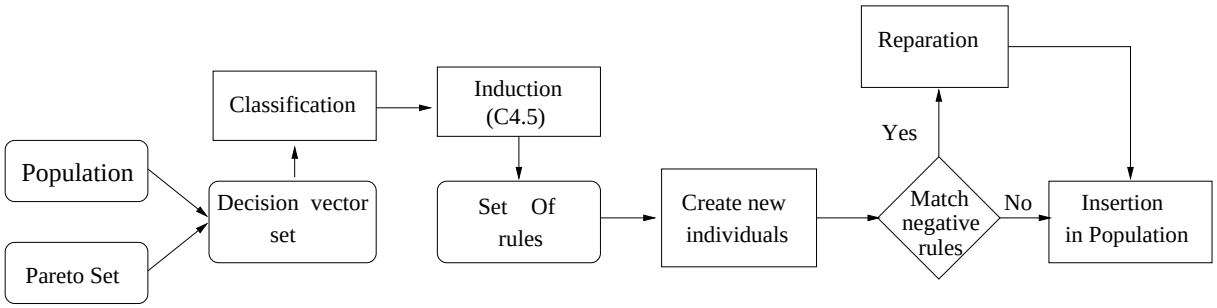


Fig. 6.7 – Création de nouveaux individus grâce aux règles extraites.

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

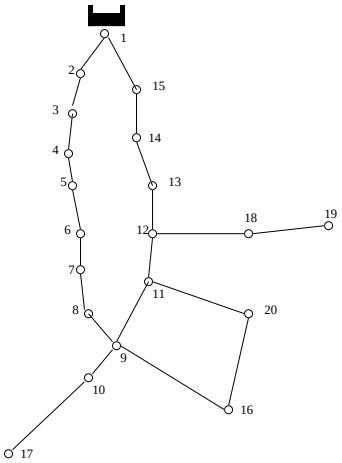


Fig. 6.8 – Le problème de la ville de New York.

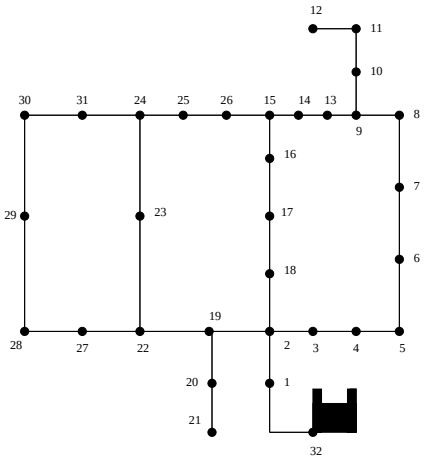


Fig. 6.9 – Le problème de la ville de Hanoi.

"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 [I20].
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(Schéma) - S(NSGAII))/S(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%	**0.42**%	**-58.58**%
LEMMO-fix4	**9.12**%	**6.8**%	**7.77**%	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 un 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 figue 6.10 montre l'évolution de l'hypervolume (*Smetric*) 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 *Smetric* définie mais obtiennent également souvent au final une valeur plus élevée. La variante *LEMMO-fix1* a été choisi 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 *Smetric* 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(NSGAII)/T(Scheme)).

Schéma	Médiane	Moyenne	Min	Max	Ecart type
LEMMO-1	+14.07%	+6.02%	+29.8%	-1.84%	+4.11%
LEMMO-fix1	**-16.28**%	**-11.81**%	-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.

	S_{norm} >**0,955**		S_{norm} > **0,968**	
Modèle	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%	**-16,66 %**	**-20,58%**

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.

	S_{norm} >**0,955**		S_{norm} > **0,968**	
Modèle (parameter)	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%	**-16,66 %**	**-20,58%**
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

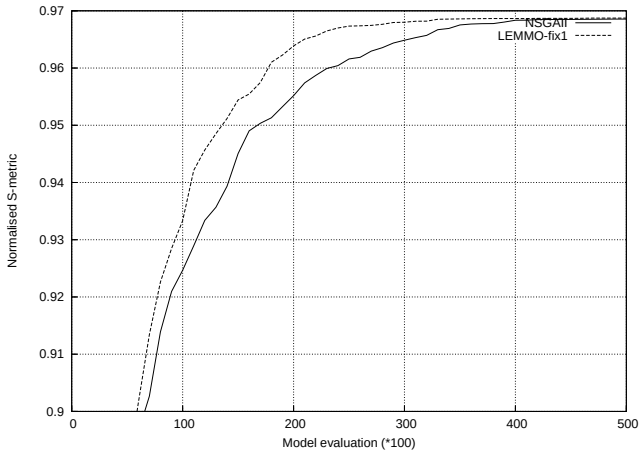


Fig. 6.10 – Exemple de l'évolution de la métrique S normalisée par rapport au nombre d'évaluation pour le problème NYT pour la variante *LEMMO-fix1*.

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 différentient 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 différentient 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 évènement permettant de connaître la véritable valeur des variables, on parle alors d'optimisation d'un problème *dynamique*[1]. 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

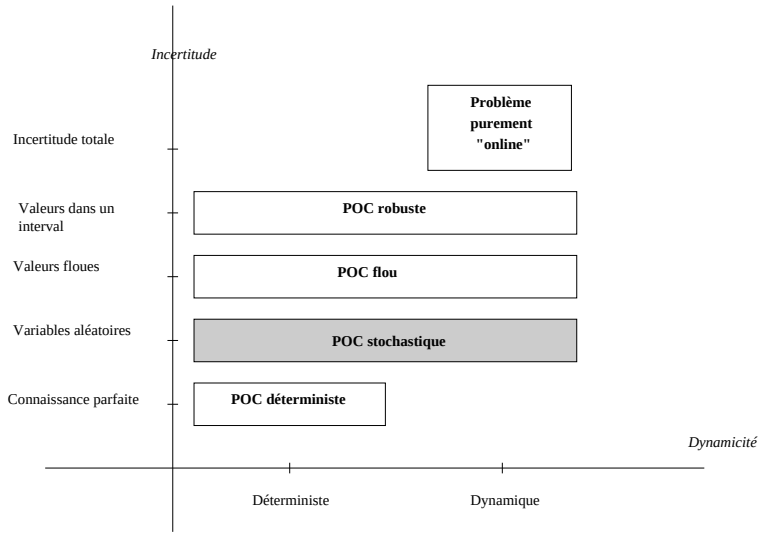


Fig. 6.11 – Classification des problèmes d'optimisation combinatoire (POC) sous incertitude.

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

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 incertaine (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

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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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 Liefooghe.

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.

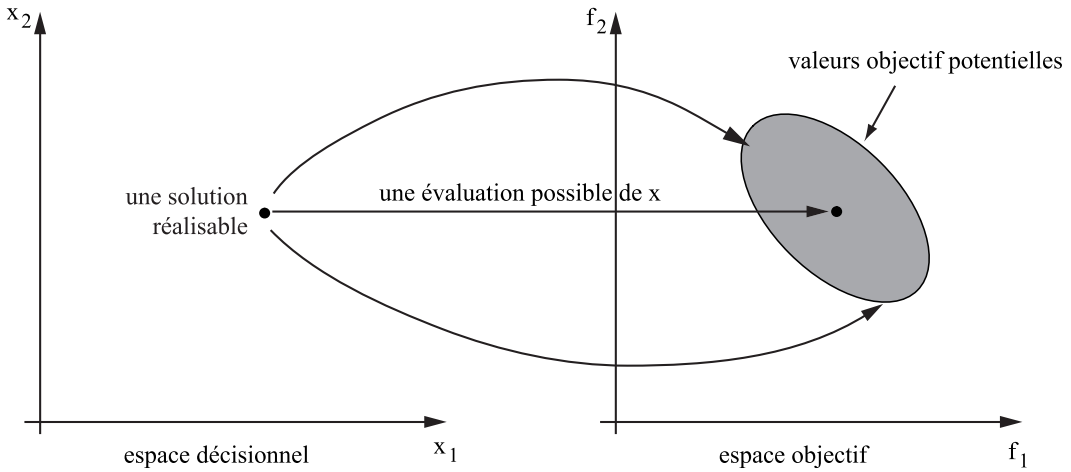


Fig. 7.1 – Prise en compte de l'incertitude à l'aide d'un échantillon de vecteurs objectif pour les fonctions objectif bruitées et pour la recherche de solutions robustes vis-à-vis de paramètres environnementaux incertaines.

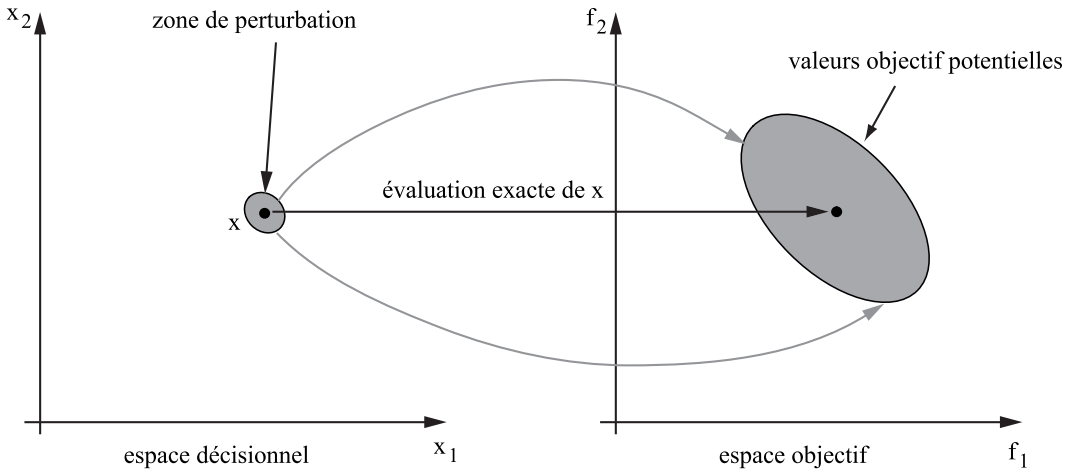


Fig. 7.2 – Prise en compte de l'incertitude à l'aide d'un échantillon de vecteurs objectif pour la recherche de solutions robustes vis-à-vis de variables de décision incertaines.

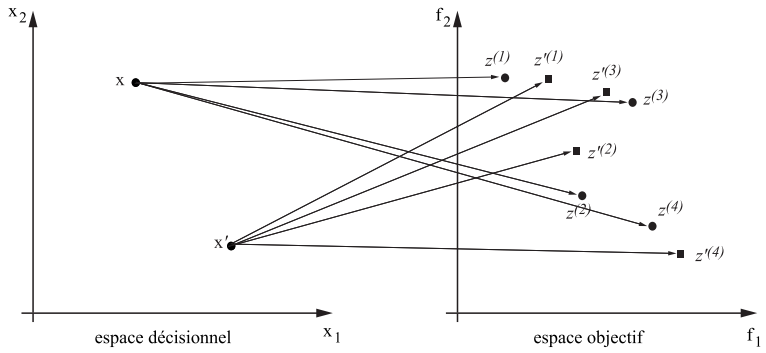


Fig. 7.3 – Illustration de la comparaison de deux ensembles de vecteurs objectif $\{z^{(i)}\}_{i=1}^{4}$ et $\{z'^{(j)}\}_{j=1}^{4}$ obtenus de l'évaluation multiple de deux solutions. Quelle solution est meilleure que l'autre ?

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 réduite 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 une 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. Liefooghe [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 $G = (V, A, D)$, $V = 0, 1, \ldots, n$ 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 ξ_i, $i = 1, \ldots, n$. 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

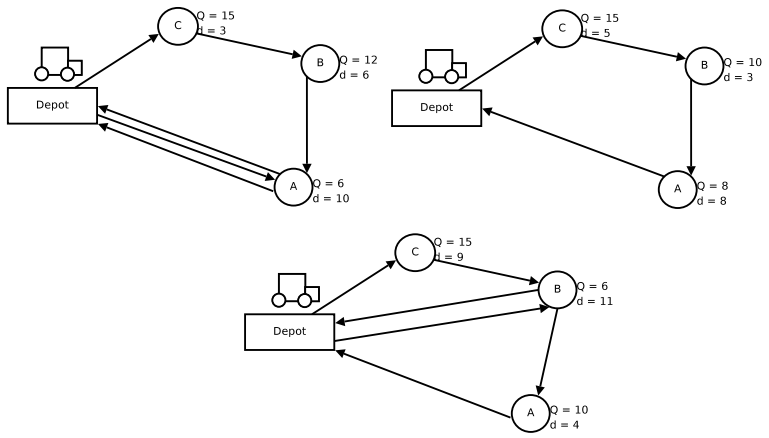


Fig. 7.4 – Exemple de modification de la tournée due au caractère stochastique des demandes

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 μ 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 ϵ 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	**43.56**	63.73
r101	**27.23**	35.35
rc101	41.01	**33.7**
c201	**56.09**	467.42
r201	**72.43**	2949.97
rc201	186.91	**51.85**

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 NSGAII (voir tableaux 7.2 et 7.3).

-		Hypervolume I_H			Epsilon I_ϵ		
	-	IBEA	MOGA	NSGAII	IBEA	MOGA	NSGAII
c101	IBEA	-	$\equiv$	$\succeq$	-	$\equiv$	$\succeq$
	MOGA	$\equiv$	-	$\succeq$	$\equiv$	-	$\equiv$
	NSGAII	$\preceq$	$\preceq$	-	$\preceq$	$\equiv$	-
r101	IBEA	-	$\equiv$	$\succeq$	-	$\equiv$	$\succeq$
	MOGA	$\equiv$	-	$\equiv$	$\equiv$	-	$\succeq$
	NSGAII	$\preceq$	$\equiv$	-	$\preceq$	$\preceq$	-
rc101	IBEA	-	$\equiv$	$\equiv$	-	$\equiv$	$\equiv$
	MOGA	$\equiv$	-	$\equiv$	$\equiv$	-	$\equiv$
	NSGAII	$\equiv$	$\equiv$	-	$\equiv$	$\equiv$	-
c201	IBEA	-	$\equiv$	$\succeq$	-	$\equiv$	$\succeq$
	MOGA	$\equiv$	-	$\succeq$	$\equiv$	-	$\succeq$
	NSGAII	$\preceq$	$\preceq$	-	$\preceq$	$\preceq$	-
r201	IBEA	-	$\equiv$	$\succeq$	-	$\equiv$	$\equiv$
	MOGA	$\equiv$	-	$\succeq$	$\equiv$	-	$\succeq$
	NSGAII	$\preceq$	$\preceq$	-	$\equiv$	$\preceq$	-
rc201	IBEA	-	$\equiv$	$\preceq$	-	$\equiv$	$\preceq$
	MOGA	$\equiv$	-	$\equiv$	$\equiv$	-	$\preceq$
	NSGAII	$\succeq$	$\equiv$	-	$\succeq$	$\succeq$	-

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

-		Hypervolume I_H			Epsilon I_ϵ		
	-	IBEA	MOGA	NSGAII	IBEA	MOGA	NSGAII
c101	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\preceq$
	NSGAII	$\preceq$	$\succeq$	-	$\preceq$	$\succeq$	-
r101	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\succeq$
	NSGAII	$\preceq$	$\succeq$	-	$\preceq$	$\preceq$	-
rc101	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\succeq$
	NSGAII	$\preceq$	$\succeq$	-	$\preceq$	$\preceq$	-
c201	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\preceq$
	NSGAII	$\preceq$	$\succeq$	-	$\succeq$	$\preceq$	-
r201	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\preceq$
	NSGAII	$\preceq$	$\succeq$	-	$\preceq$	$\succeq$	-
rc201	IBEA	-	$\succeq$	$\succeq$	-	$\succeq$	$\succeq$
	MOGA	$\preceq$	-	$\preceq$	$\preceq$	-	$\succeq$
	NSGAII	$\preceq$	$\succeq$	-	$\preceq$	$\preceq$	-

TABLE 7.3 – Comparaison de statistique de la qualité des résultats obtenus par IBEA, MOGA et NSGAII par un test de Mann-Whitney en utilisant les indicateurs d'hypervolume et ϵ pour le modèle basé sur la déviation standard. La p-valeur est fixée à 0.05. $\preceq$ les résultats sont significativement moins bons, $\succeq$ 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

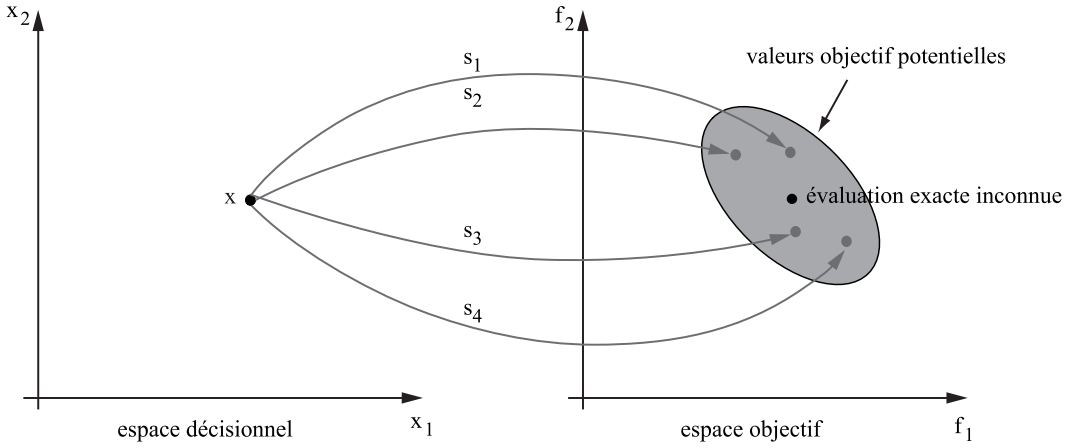


Fig. 7.5 – Illustration de l'évaluation multiple d'une solution x à partir de quatre scénarios différents $\{s_1, s_2, s_3, s_4\}$.

scénarios indépendants et équiprobables $S = \{s_1, s_2, \ldots, s_p\}$ [104]. $S(x) = \{z^{(1)}, z^{(2)}, \ldots, z^{(p)}\}$ 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 $I : Z \times Z \to \mathbb{R}$, 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.

$$I(x, x') \neq I(f(x), f(x')) \tag{7.1}$$

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

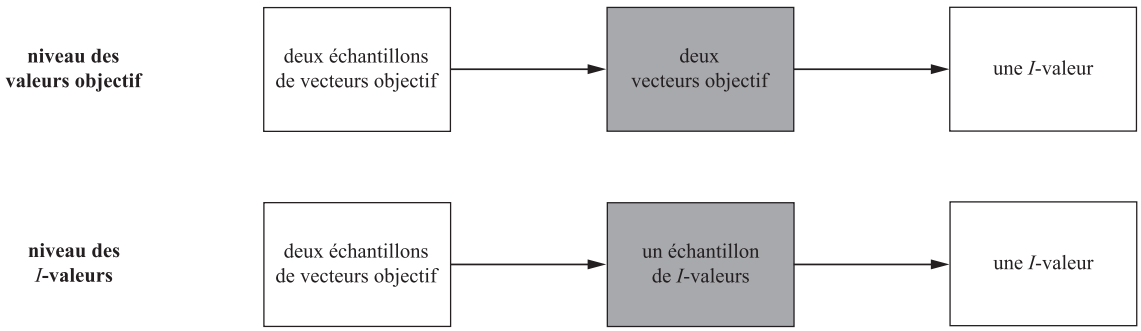


Fig. 7.6 – Deux classes de métaheuristiques proposées pour la résolution de problèmes multiobjectif stochastiques : les approches basées au niveau des valeurs objectif, et les approches basées au niveau des I-valeurs.

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 $I : Z \times Z \to \mathbb{R}$ 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.

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

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 $\{z_k^{(1)}, z_k^{(2)}, \ldots, z_k^{(p)}\}$ 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 $\{J_1, J_2, \ldots, J_N\}$ sur M machines $\{M_1, M_2, \ldots, M_M\}$. Les machines sont des ressources critiques, c'est-à-dire que deux jobs ne peuvent être affectés à une même machine simultanément.

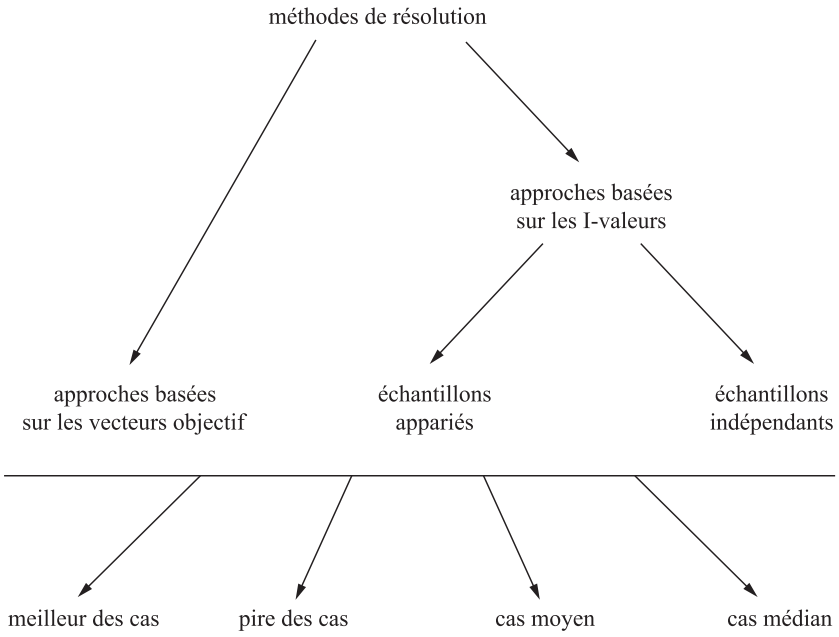


Fig. 7.7 – Résumé des méthodes de résolution proposées.

Un job J_i est composé de M tâches consécutives $\{t_{i1}, t_{i2}, \ldots, t_{iM}\}$, 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 $\overline{T}$. Soit C_{ij} la date de complétion de la tâche t_{ij}, les fonctions objectif peuvent être définies comme suit.

$$C_{max} = \max_{i \in \{1,\ldots,N\}} \{C_{iM}\} \tag{7.3}$$

$$\overline{T} = \sum_{i=1}^{N} \Big\{ \max\{0, C_{iM} - d_i\} \Big\} \tag{7.4}$$

Flowhsop 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 λ 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 μ 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 [1]. 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 $I_{\epsilon+}$.

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 (α). 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

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

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 $t = 1, ..., T$ 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 :

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

où $best_t^A$ est la meilleur 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 :

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

La stabilité vaut entre 0 et 1. Une valeur proche de 0 indique une grande stabilité.
Nous nous servirons 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 différentier 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(VRP) et que la classe des problèmes du DVRP est dénotée P(DVRP), alors P(VRP)⊂P(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.

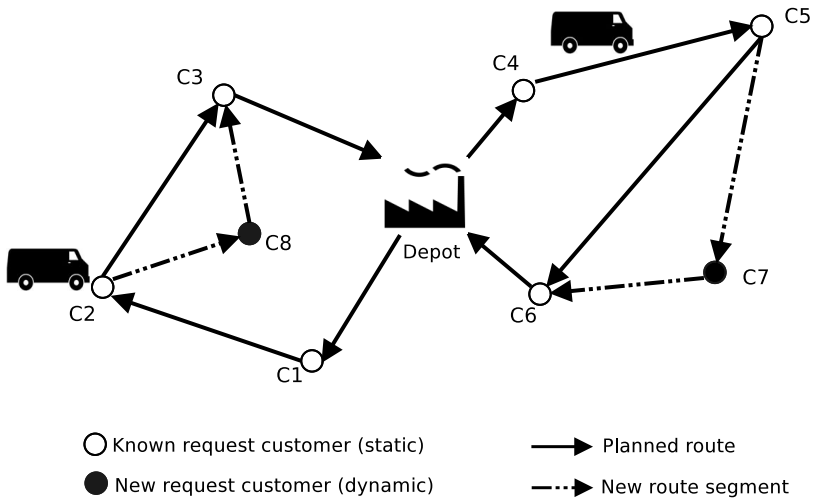


Fig. 8.1 – Problème dynamique de tournées de véhicules (DVRP)

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 métaheuristiques ont une représentation identique que nous décrirons avons de présenter les algorithmes en détail.

8.3.1 Représentation et initialisation

La représentation d'une solution est adapté 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 :

$$R_i : (C_1, S_1, Ts_1) - ... - (C_n, S_n, Ts_n) \tag{8.3}$$

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

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

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 dynamiquse mais essentiellement continus [22, 81, 109, 110]. Une particule de l'essaim est définie par sa position $\overrightarrow{x_i}$, la position de sa meilleure solution visitée

$\overrightarrow{p_i}$, et sa vélocité $\overrightarrow{v_i}$. De plus, chaque particule connait la meilleure solution découverte par une de ses voisins $\overrightarrow{p_g}$. 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 vélocité puis toutes les positions en fonction des formules suivantes :

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

$$\overrightarrow{x}_i = \overrightarrow{x}_i + \overrightarrow{v}_i \tag{8.6}$$

ω 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 meilleurs 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 client 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 $\mathcal{N}_1(s), \mathcal{N}_2(s), \ldots, \mathcal{N}_n(s)$, 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 amélioré 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 $\mathcal{N}_{k+1}$.

Algorithme 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
    $\overrightarrow{x}_i := \overrightarrow{s}_m$;
  end for
end if
//PSO Main Algorithm
for Each particle i **do**
  //Insert randomly the new customer requests in the current position
  $\overrightarrow{x}_i := InsertRandomly(\overrightarrow{x}_i, \textit{New customer requests})$
  //Initialize the best position
  $\overrightarrow{p}_i := \overrightarrow{x}_i$;
  Evaluate $f(\overrightarrow{p}_i)$
end for
repeat
  for Each particle i **do**
    Update particle position $\overrightarrow{x_i}$ according to Eqs (8.5) and (8.6).
    Evaluate $f(\overrightarrow{x_i})$
    //Update the Adaptive Memory
    //Update personal best
    if $f(\overrightarrow{x_i}) \leq f(\overrightarrow{p_i})$ **then**
      $\overrightarrow{p}_i := \overrightarrow{x}_i$;
    end if
    //Update global best
    if $f(\overrightarrow{x_i}) \leq f(\overrightarrow{p_g})$ **then**
      $\overrightarrow{p}_g := \overrightarrow{x}_i$;
    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 λ-échange [136], qui sélectionne aléatoirement un client et l'insère sur une autre route de la solution.

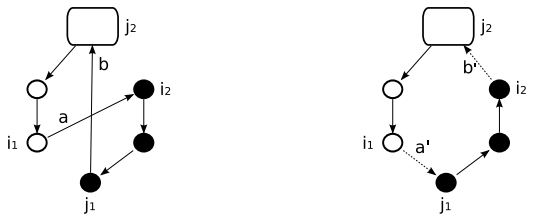
Un mécanisme de réparation est effectué avant chaque évaluation si la solution ne respecte pas les contraintes. L'algorithme VNS adapté au DVRP est expliqué figure 5.

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 := generateRandomInitialSolution()
else {potential dynamic orders waiting to be scheduled}
  s := getLastTimeSliceSolution()
  for all customer c_i ∉ s do
    s := 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' := pickAtRandom(N_k(s))
    // Apply local search procedures
    for all local search heuristic ls_j do
      s'' := 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) mod n
    end if
  end while
end while
```

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

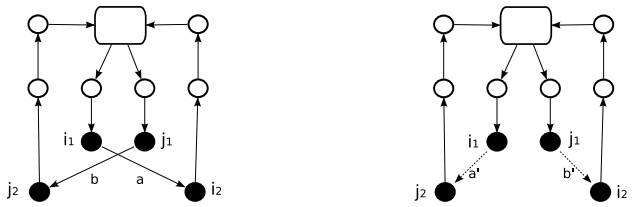
Comme nous l'avons vu précédemment dans le cadre déterministe, l'utilisation de méthodes hybrides peut améliorer significativement la qualité des résultats obtenus. Nous nous plaçons dans un type d'hybridation de bas niveau relai (LRH) en hybridant l'heuristique classique de 2-Opt [118] comme dans une 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.



(a) Avant (b) Après

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.



(a) Avant (b) Après

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.[1] 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 paralètres ω, 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
k100	1000	500	$25 \times 500 = 12500$
k250	1100	550	$25 \times 550 = 13750$
k500	2300	1150	$25 \times 1150 = 28750$

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

Instances \ Paramètres	ω	c_1	c_2
k100	1.0	0.5	0.5
k250	1.0	0.5	0.5
k500	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'algoritme 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_{2-Opt} est compétitif en début de recherche mais VNS d'adpate 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			
			$APSO$	$APSO_{2-Opt}$	VNS	VNS_{2-Opt}
Static	*k100*	$APSO$	−	−	+	−
		$APSO_{2-Opt}$	−	−	+	−
		VNS	+	+	−	+
		VNS_{2-Opt}	−	−	+	−
	k250	$APSO$	−	−	+	−
		$APSO_{2-Opt}$	−	−	+	−
		VNS	+	+	−	+
		VNS_{2-Opt}	−	−	+	−
	k500	$APSO$	−	+	+	+
		$APSO_{2-Opt}$	+	−	+	−
		VNS	+	+	−	+
		VNS_{2-Opt}	+	−	+	−
Dynamic	*k100*	$APSO$	−	−	+	+
		$APSO_{2-Opt}$	−	−	+	+
		VNS	+	+	−	−
		VNS_{2-Opt}	+	+	−	−
	k250	$APSO$	−	−	+	+
		$APSO_{2-Opt}$	−	−	+	+
		VNS	+	+	−	+
		VNS_{2-Opt}	+	+	+	−
	k500	$APSO$	−	+	−	+
		$APSO_{2-Opt}$	+	−	+	+
		VNS	−	+	−	+
		VNS_{2-Opt}	+	+	+	−

8.5 Conclusion

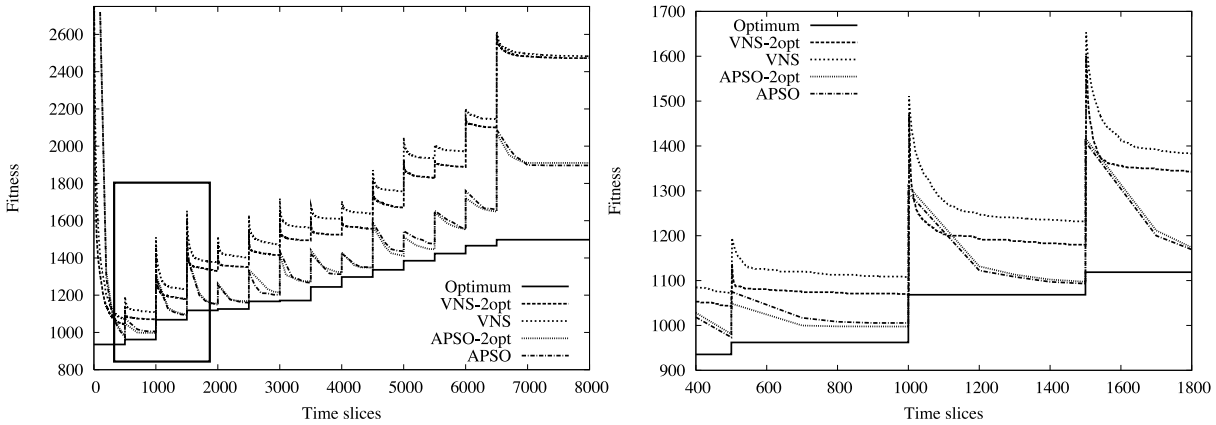
Dans ce chapitre, nous avons présenté le problème de routage de véhicule dynamique et deux algorithmes pour le résoudre. Nous avons montré qu'une hybridation avec une heuristique classique, le 2-opt, permet d'améliorer significativement les résultats obtenus aussi bien dans le contexte déterministe que 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 même 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-évolutionnaire est plus compliqué car il nécessite une synchronisation parfaite des environnement. Nous sommes en train d'améliorer APSO en utilisant le principe des *multiswarms* qui évite la convergence prématuré 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 propose 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.

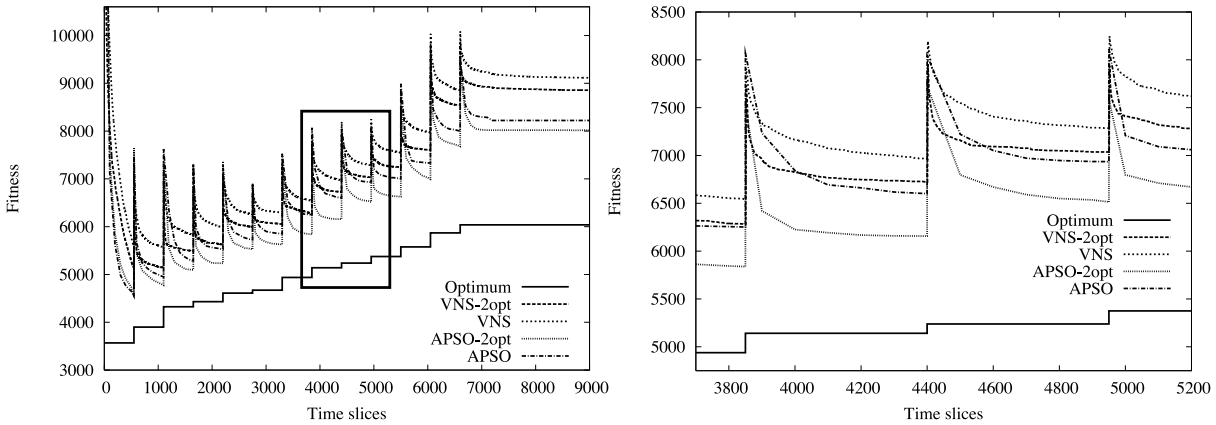
Instance	T_{step}	Accuracy			
		APSO		VNS	
		Canonical	*2-Opt*	*Canonical*	*2-Opt*
k100	0	**0.961**	0.954	0.845	0.867
	5	**0.970**	0.960	0.805	0.835
	10	0.939	**0.957**	0.763	0.802
	15	**0.790**	0.784	0.582	0.587
	20	**0.790**	0.784	0.582	0.588
	25	**0.790**	0.784	0.582	0.588
	Avg	**0.867**	0.865	0.686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

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

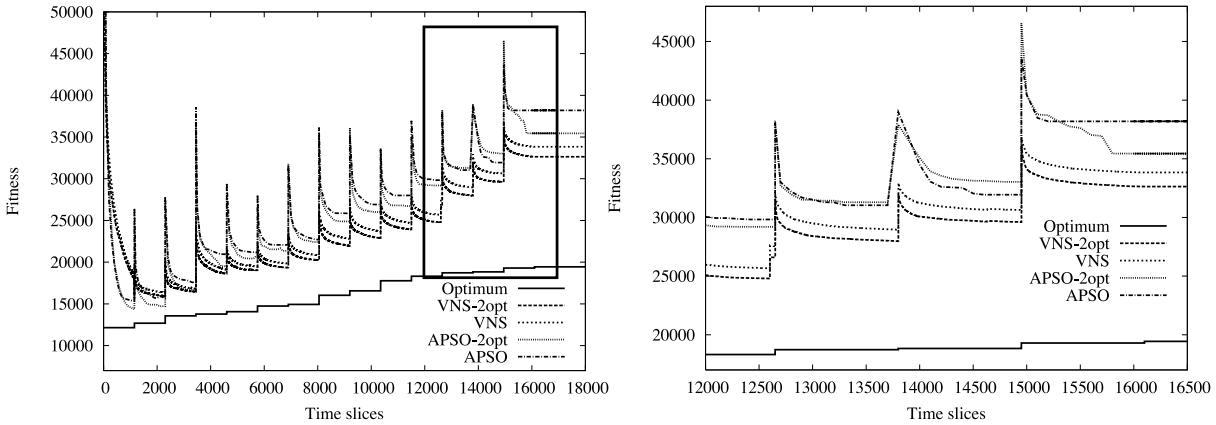
Instance	T_{step}	Stabilité			
		APSO		VNS	
		Canonical	*2-Opt*	*Canonical*	*2-Opt*
k100	0	**0**	**0**	**0**	**0**
	5	**0**	**0**	0.013	0.016
	10	0.008	0.010	**0**	**0**
	15	**0**	**0**	0.003	0.002
	20	**0**	**0**	**0**	**0**
	25	**0**	**0**	**0**	**0**
	Avg	**0.003**	**0.003**	0.005	0.005
k250	0	**0**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



(a) *k100* instance

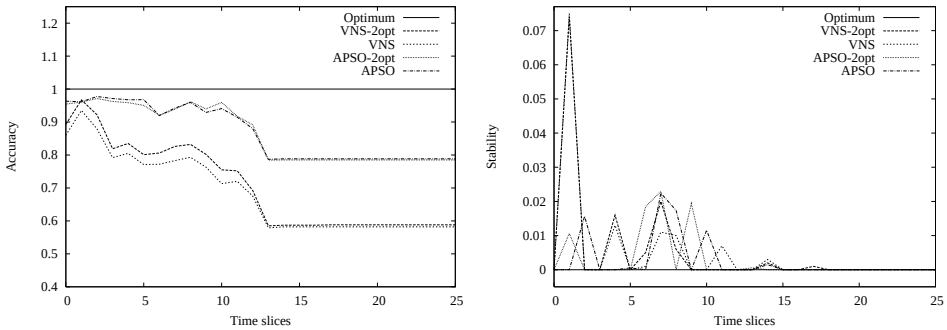


(b) *k250* instance



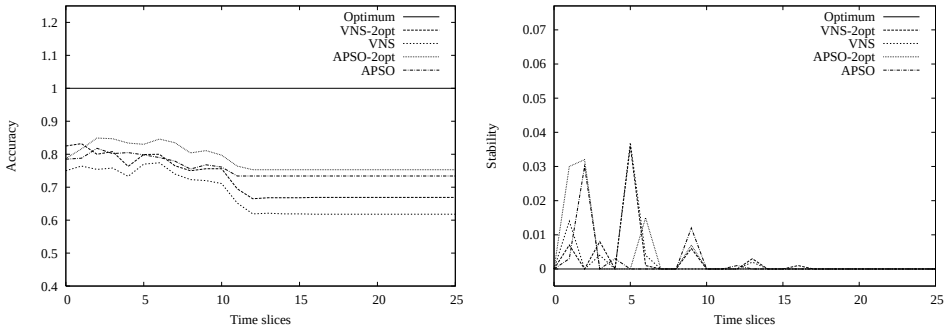
(c) *k500* instance

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.



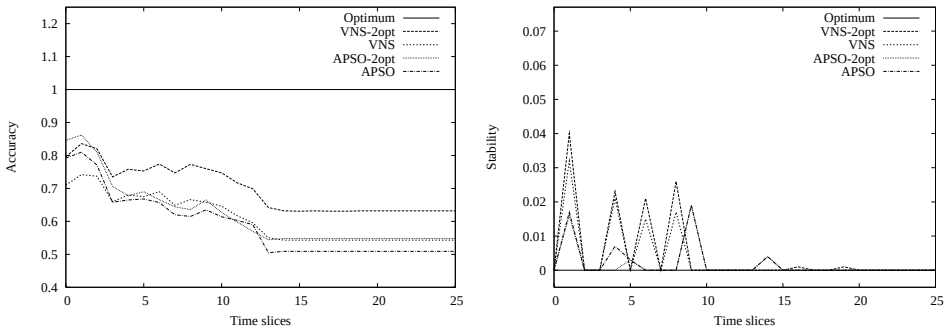
(a) Accuracy over 25 time slices for *k100*.

(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-adaptives. 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 neighbohood 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 3rd *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 approch 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 multiminima 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 multiminima 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. *DIMACS 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 2nd *International Conference on Genetic Algorithms*, pages 41–49, 1987.

[70] M. Gourgand, N. Grangeon, and S. Norre. Modèle du flow-shop de permutation stochastique. In J.-C. Billaut, A. Moukrim, and E. Sanlaville, editors, *Flexibilité et robustesse en ordonnancement*, chapter 7, pages 135–161. Hermès 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. Jaszkiewicz. 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. Mladenović. 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. Jozefowiez, 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 Evolutionnary 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, Standford 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. Liefooghe. *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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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 vl1 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. he 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. Zalzala, 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 civilisee ? In *Actes de la Conférence Evolution Artificielle, EA94editors*, 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 2nd *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 2nd *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 Stastistics, 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 Evolutionnary 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. Slowínski, 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-...	**Chargée de recherche 1ère classe INRIA** *Inria Lille Nord Europe, Equipê Dolphin*
2005-2008	**Chargée de recherche 2ème classe INRIA** *Inria Futurs, Team Dolphin*
2004-2005	**Chargée de recherche en "Fast Optimisation",** ESPRC Council grant, University of Exeter, Devon, Angleterre
2004-2005	**Attachée Temporaire d'Enseignement et de Recherche** (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	**Attachée Temporaire d'Enseignement et de Recherche** (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	**Monitrice de l'Enseignement Supérieur** *Université Lille 1-USTL* : 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	**Vacataire de l'Enseignement Supérieur.** *Université Lille 1-USTL*

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, Examinateur) M. C. Dina (Ingénieur, Institut de Biologie de Lille, Pasteur, Examinateur) 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`.
ParadisEo-MOEO	Opensource, extension toujours en cours, Co-auteurs : S. Cahon, J. Humeau, T. Legrand, A. Liefooghe, 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, JOBIM'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 Liefooghe 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 muultiobjectifs 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 Méditerannée), *Implémentation de composants dans Paradiseo-MOEO.* (50%, avec E-G. Talbi)
- José Manuel Nieto (2007) : Master Université de Malaga (Spain, Projet 3+3 Méditerannée), *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 Liefooghe (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 Méditerannée), *Implémentation de composant dans Paradiseo-MOEO*, Abdelhakim Denech.
  - PHD Université de Malaga (Spain, Projet 3+3 Méditerannée), *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*, Jérome 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 Méditerrannée 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 parrallèles pour des problèmes de transport (plus particulièrement des problèmes de plus court chemin dynamique et des problèmes de tournées enrichis). 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.

	Revues	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	IJFCS	EMO'05		SCBA, CCWI	ROADEF'05
2004		Evocop'04	NIDISC'04, HIS'04		
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.*

Revues internationales 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. Liefooghe, 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 mutli-objective cooperation for the flexible docking problem.* Journal of Mathematical Modelling and Algorithms, **volume 9**(2) : pages 195–208, 2010.

[R4] A. Liefooghe, 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 multifactoral 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. Liefooghe, 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. Liefooghe, 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 Intellignece, 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. Liefooghe, 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 Milti-Criteria Decision-Making, 2009. MCDM '09. IEEE Symposium on*, pages 88–95. 2009.

[I5] A. Liefooghe, 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. Liefooghe, 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. Liefooghe, L. Jourdan, N. Jozefowiez, 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.* Bejing China, 2007.

[I13] A. Liefooghe, M. Basseur, L. Jourdan, and E.-G. Talbi. *Multi-objective optimization for stochastic problems : an application to the flow-shop scheduling ptoblem.* In *EMO 2007*, volume LNCS 4408, pages 386–400. 2007.

[I14] A. Liefooghe, 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 multiobjective 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 Medecine and Biological Sciences (METMBS'2001)*, pages 40–46. 2001 June. Las Vegas, USA.

Conférences Internationales

[p1] S. Verel, L. Jourdan, C. Dhaenens, A. Liefooghe. *Set-based Multiobjective Fitness Landscapes : Definition, Properties.* In *4th workshop on Theory of Randomized Search Heuristics (ThRaSH 2010)* Paris, France, 2010

[p2] A. Liefooghe, 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. Liefooghe, 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 22nd *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 55th *ASMS Conference on Mass Spectrometry.* 2007 June 3-7. Indianapolis, Indiana, USA.

[p8] A. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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. Liefooghe, 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 multicritère 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. Liefooghe, 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. Liefooghe, 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.

5

ParadisEO-MOEO: A Software Framework for Evolutionary Multi-Objective Optimization

Arnaud Liefooghe[1,2], Laetitia Jourdan[1,2], Thomas Legrand[2], Jérémie Humeau[2], and El-Ghazali Talbi[1,2]

[1] Laboratoire d'Informatique Fondamentale de Lille (LIFL), UMR CNRS 8022, Université Lille 1, Bâtiment M3, 59655 Villeneuve d'Ascq cedex, France
[2] INRIA Lille-Nord Europe, Parc Scientifique de la Haute Borne, 40 avenue Halley, 59650 Villeneuve d'Ascq, France
Arnaud.Liefooghe@lifl.fr, Laetitia.Jourdan@lifl.fr, Thomas.Legrand@inria.fr, Jeremie.Humeau@inria.fr, talbi@lifl.fr

Summary. This chapter presents ParadisEO-MOEO, a white-box object-oriented software framework dedicated to the flexible design of metaheuristics for multi-objective optimization. This paradigm-free software proposes a unified view for major evolutionary multi-objective metaheuristics. It embeds some features and techniques for multi-objective resolution and aims to provide a set of classes allowing to ease and speed up the development of computationally efficient programs. It is based on a clear conceptual distinction between the solution methods and the problems they are intended to solve. This separation confers a maximum design and code reuse. This general-purpose framework provides a broad range of fitness assignment strategies, the most common diversity preservation mechanisms, some elitist-related features as well as statistical tools. Furthermore, a number of state-of-the-art search methods, including NSGA-II, SPEA2 and IBEA, have been implemented in a user-friendly way, based on the fine-grained ParadisEO-MOEO components.

5.1 Introduction

A large number of existing real-world optimization problems are characterized by multiple conflicting objective functions. Evolutionary algorithms are commonly used to solve these multi-objective problems since they are particularly well-suited to find a spread set of good-quality solutions. Over the past few years, major contributions have been made in the field of evolutionary multi-objective optimization. In this work, we propose a new software framework for evolutionary multi-objective optimization called ParadisEO-MOEO. Its modular implementation follows a general purpose model based on a fine-grained decomposition. This model is founded on a unified view of evolutionary algorithms for multi-objective optimization where the fundamental issues of fitness assignment, diversity preservation and elitism are

C.A. Coello Coello et al. (Eds.): Adv. in Multi-Obj. Nature Inspired Computing, SCI 272, pp. 87–117.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

involved. ParadisEO[1] is a free open-source C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics. It attempts to simplify and accelerate the development process of efficient solver programs while having a minimal programming effort. It is based on a clear conceptual separation between the problem-specific and the invariant part of the solution method. This separation is expressed at the implementation level, which confers a maximum design and code reuse. ParadisEO is composed of four connected modules: ParadisEO-EO for population-based metaheuristics, ParadisEO-MO for single solution-based metaheuristics, ParadisEO-MOEO for multi-objective metaheuristics and ParadisEO-PEO for parallel and distributed models for metaheuristics and their hybridization. Each module has been validated and successfully applied to solve a wide range of academic and real-world optimization problems of both continuous and combinatorial nature. Historically, ParadisEO was especially dedicated to parallel and distributed metaheuristics and was the result of the PhD work of Sébastien Cahon, supervised by Nouredine Melab and El-Ghazali Talbi [10]. The initial version already contained a few number of features related to evolutionary multi-objective optimization, mainly with regard to elitism. This work has been partially extended and presented in [33]. But since then, the ParadisEO-MOEO module has been completely redesigned in order to confer an even more fine-grained decomposition, and major additional features have been integrated into the framework.

In this chapter, we provide a general presentation of ParadisEO, and a detailed description of the ParadisEO-MOEO module. First, a unified view of evolutionary multi-objective optimization techniques is presented in Sect. 5.2. Then, software frameworks are discussed in Sect. 5.3. Sect. 5.4 is devoted to the design and the implementation of evolutionary multi-objective metaheuristics with ParadisEO-MOEO. A case study on a bi-objective scheduling problem is given in Sect. 5.5. Finally, the last section concludes the chapter.

5.2 Evolutionary Multi-Objective Optimization, a Unified View

This section presents some basic concepts about Evolutionary Multi-objective Optimization (EMO). Next, a couple of related design issues are briefly discussed and a unified model for EMO algorithms is proposed.

5.2.1 Evolutionary Multi-Objective Optimization

A Multi-objective Optimization Problem (MOP) can be defined by a set f of $n \geq 2$ objective functions $f_1, f_2, \ldots, f_n$, a set X of feasible solutions in the *decision space*, and a set Z of feasible points in the *objective space*. Without loss of generality, we here assume that $Z \subseteq \mathbb{R}^n$ and that all n objective functions are to be minimized. To each decision vector $x \in X$ is assigned an objective vector $z \in Z$ on the basis of the vector function $f : X \rightarrow Z$ with $z = f(x) = f_1(x), f_2(x), \ldots, f_n(x)$.

[1] `http://paradiseo.gforge.inria.fr`

Definition 1 (Pareto dominance). *An objective vector $z \in Z$ is said to* dominate *another objective vector $z' \in Z$ if $\forall i \in \{1,2,\ldots,n\}$, $z_i \leq z'_i$ and $\exists j \in \{1,2,\ldots,n\}$ such as $z_j < z'_j$.*

Definition 2 (Non-dominated point). *An objective vector $z \in Z$ is said to be* non-dominated *if there does not exist any other objective vector $z' \in Z$ such that z' dominates z.*

By extension, we say that a decision vector $x \in X$ *dominates* a decision vector $x' \in X$ if $f(x)$ dominates $f(x')$, and that a decision vector $x \in X$ is *non-dominated* (or *efficient*, *Pareto optimal*) if $f(x)$ is non-dominated. Note that other dominance relations exist in the frame of multi-objective optimization and will be discussed later in the chapter. A possible MOP resolution method is to find the minimal set of efficient solutions, *i.e.* one feasible solution per non-dominated point. However, generating the entire efficient set is usually infeasible, due to the complexity of the underlying problem or the large number of optima. Therefore, in many approaches, the overall goal is to identify a good approximation of it. Evolutionary algorithms are commonly used to this end, as they are particularly well-suited to find multiple efficient solutions in a single simulation run. The reader is referred to [11, 13] for more details about EMO.

5.2.2 Design Issues

As pointed out by various authors (see *e.g.* [11, 49]), approximating the efficient set is itself a bi-objective problem. Indeed, the approximation to be found must have both good convergence and distribution properties, as its mapping in the objective space has to be (*i*) close to, and (*ii*) well-spread over the (unknown) Pareto optimal front. As a consequence, the main differences between the design of a single- and a multi-objective metaheuristic in general, and EA in particular, deal with these two goals. As noticed by Zitzler et al. [49], in the EMO literature, initial approaches were mainly focused on moving toward the Pareto front [19, 39]. Afterward, diversity preservation mechanisms quickly emerged [18, 26, 41]. Then, at the end of the twentieth century, the concept of elitism, related to the preservation of non-dominated solutions, became very popular and is now employed in most recent EMO methods [50, 51]. The importance of the issues of *fitness assignment*, *diversity preservation* and *elitism* are commonly approved and are also presented under different names in, for instance, [11, 49]. They are discussed in details below.

Fitness Assignment

In the single-objective case, the fitness value assigned to a given solution is most often its unidimensional objective value. While dealing with MOPs, fitness assignment aims to guide the search toward Pareto optimal solutions for a better convergence. We propose to classify existing fitness assignment schemes into four different classes:

- *Scalar approaches*, where the MOP is reduced to a single-objective optimization problem. A popular example consists in combining the n objective functions into a single one by means of a weighted-sum aggregation. Other examples are ε-constraint or achievement function-based methods, see [36].
- *Criterion-based approaches*, where each objective function is treated separately. For instance, in VEGA (Vector Evaluated GA) [39], a parallel selection is performed where solutions are discerned according to their values on a single objective function, independently to the others. In lexicographic methods [19], a hierarchical order is defined between objective functions.
- *Dominance-based approaches*, where a dominance relation is used to classify solutions. Existing strategies are *dominance-rank* [18], *dominance-count* [51] and *dominance-depth* [21] techniques. And different schemes can also be combined, as for instance in [51]. In the frame of dominance-based approaches, the most commonly used dominance relation is the Pareto-dominance relation given in Definition 1. But some recent techniques are based on other dominance operators such as ε-dominance in [15] or g-dominance in [37].
- *Indicator-based approaches*, where the fitness values are computed by comparing individuals on the basis of a quality indicator I. The chosen indicator represents the overall goal of the search process. Generally speaking, no particular diversity preservation mechanism usually necessary, with regards to the indicator being used. Examples of indicator-based EAs are IBEA (Indicator-Based EA) [48] and SMS-EMOA (S-Metric Selection EMO Algorithm) [6].

Diversity Preservation

As noticed earlier, aiming at approximating the efficient set is not only a question of convergence. The final approximation also has to be well spread over the objective space. However, classical fitness assignment schemes often tend to produce premature convergence by privileging non-dominated solutions, which does not guarantee a uniformly sampled output set. In order to prevent that issue, a diversity preservation mechanism, based on a given distance measure, is usually integrated into the metaheuristic to uniformly distribute the population over the trade-off surface. In the frame of EMO, a common distance measure is based on the euclidean distance between objective vectors. But, this measure can also be defined in the decision space or can even combine both spaces. Popular examples of diversity preservation techniques are *fitness sharing* [22] and *crowding* [25], respectively used in *e.g.* Fonseca and Fleming's MOGA (Multi-Objective GA) [18] and Deb et al.'s NSGA-II (Non-dominated Sorting GA II) [14].

Elitism

Another essential issue about MOP solving is the notion of *elitism*. It mainly consists in maintaining an external set, the so-called *archive*, that allows to store either all or a subset of non-dominated solutions found during the search process. This secondary population mainly aims at preventing the loss of these solutions during

the stochastic optimization process. The update of the archive contents with new potential non-dominated solutions is mostly based on the Pareto-dominance criteria. But, in the literature, other dominance criteria are found and can be used instead of the Pareto-dominance relation. Examples are weak-dominance, strict-dominance, ε-dominance [24], etc. When dealing with archiving, one may distinguished four different techniques depending on the problem properties and on the designed algorithm: (*i*) *no archive*, (*ii*) an *unbounded archive*, (*iii*) a *bounded archive* or (*iv*) a *fixed-size archive*. First, if the approximation set is maintained by, or contained into the population itself, there can be no archive at all. On the other hand, if an archive is maintained, it usually comprises the current non-dominated set approximation, as dominated solutions are removed. Then, an unbounded archive can be used in order to save the whole set of non-dominated solutions found since the beginning of the search process. However, as some continuous optimization problems may contain an infinite number of non-dominated solutions, it is simply not possible to save them all. Therefore, additional operations must be used to reduce the number of stored solutions. Then, a common strategy is to bound the size of the archive according to some fitness and/or diversity assignment scheme(s). Finally, another archiving technique consists of a fixed size storage capacity, where a bounding mechanism is used when there are too many non-dominated solutions, and some dominated solutions are integrated into the archive if the non-dominated set is too small. This is done, for instance, in the frame of SPEA2 [50]. Usually, an archive is used as an external storage only. However, archive members can also be integrated during the selection phase of an EMO algorithm [51].

5.2.3 A Unified Model

An Evolutionary Algorithm (EA) [21] is a search method where a population of solutions is iteratively improved by means of some stochastic operators. EAs belong to the class of population-based metaheuristics. Starting from an initial population, each individual is evaluated in the objective space and a selection scheme is performed to build a so-called parent population. An offspring population is then created by applying variation operators. Next, a replacement strategy determines which individuals will survive in the next EA generation. The search process is iterated until a given stopping criterion is satisfied. As noticed earlier in the chapter, in the frame of EMO, the main expansions deal with the issues of *fitness assignment*, *diversity preservation* and *elitism*. Fitness and diversity informations are necessary to discriminate individuals at to the selection and the replacement steps of the EA. Moreover, the update of the archive contents possibly appears at each EA iteration. As a consequence, whatever the MOP to be solved, the common concepts for the design of an EMO algorithm are the following ones.

1. Design a representation.
2. Design a population initialization strategy.
3. Design a way of evaluating a solution.

4. Design suitable variation operators.
5. Decide a fitness assignment strategy.
6. Decide a diversity preservation strategy.
7. Decide a selection strategy.
8. Decide a replacement strategy.
9. Decide an archive management strategy.
10. Decide a continuation strategy.

When dealing with any kind of metaheuristics, one may distinguish problem-specific and generic components. Indeed, the first four common concepts presented above strongly depend of the MOP at hand, while the six last ones can be considered as problem-independent, even if some problem-dependent strategies can also be envisaged in some particular cases. Note that concepts of representation and evaluation are shared by any metaheuristic, concepts of population initialization and stopping criterion are shared by any population-based metaheuristic, concepts of variation operators, selection and replacement are shared by any EA, whereas concepts of fitness, diversity and archiving are specific to EMO.

5.3 Software Frameworks for Evolutionary Multi-Objective Optimization

In this section, the motivations in using a software framework for metaheuristics are outlined. Next, ParadisEO, a platform dedicated to the reusable design of metaheuristics, and ParadisEO-MOEO, its module devoted to EMO are presented. The main characteristics of ParadisEO are then detailed, and a comparative study of existing software frameworks for EMO is given.

5.3.1 Motivations

In practice, there exists a large diversity of optimization problems to be solved, giving rise to a wide number of possible models to be handled, in the context of a metaheuristic solution method. Moreover, a growing number of general-purpose search methods are proposed in the literature, with evolving complex mechanisms. ¿From a practitioner's point of view, there is a popular demand to provide a set of ready-to-use metaheuristic implementations, allowing a minimum programming effort. On the other hand, an expert generally wants to be able to design new algorithms, to integrate new components into an existing method, or even to combine different search mechanisms. As a consequence, an approved approach for the development of metaheuristics is the use of frameworks. A framework may be defined by a set of components based on a strong conceptual separation of the invariant part and the problem-specific part of metaheuristics. Then, each time a new optimization problem is tackled, both code and design can directly be reused in order to redo as little code as possible.

5.3.2 ParadisEO and ParadisEO-MOEO

ParadisEO[2] is a white-box object-oriented software framework dedicated to the flexible design of metaheuristics for optimization problems of both discrete and combinatorial nature. Based on EO (Evolving Objects)[3] [30], this template-based, ANSI-C++ compliant computation library is portable across both Unix-like and Windows systems. Moreover, it tends to be used both by non-specialists and optimization experts. ParadisEO is composed of four connected modules that constitute a global framework. Each module is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers a maximum code and design reuse to the user. The first module, ParadisEO-EO [30], provides a broad range of components for the development of population-based metaheuristics, including evolutionary algorithms and particle swarm optimization techniques. Second, ParadisEO-MO [8] contains a set of tools for single-solution based metaheuristics, *i.e.* hill climbing, simulated annealing, tabu search, iterated local search and variable neighborhood search. Next, ParadisEO-MOEO is specifically dedicated to the reusable design of metaheuristics for multi-objective optimization. Finally, ParadisEO-PEO [10] provides a powerful set of classes for the design of parallel and distributed metaheuristics: parallel evaluation of solutions, parallel evaluation function, island model and cellular model. In the frame of this chapter, we will exclusively focus on the module devoted to multi-objective optimization, namely ParadisEO-MOEO.

ParadisEO-MOEO provides a flexible and modular framework for the design of metaheuristics for multi-objective optimization. Its implementation is based on the unified model proposed in the previous section and is conceptually divided into fine-grained components. On each level of its architecture, a set of abstract classes is proposed and a wide range of sub-classes, corresponding to different state-of-the-art strategies, are also provided. Moreover, as the framework aims to be extensible, flexible and easily adaptable, all its components are generic in order to provide a modular architecture allowing to quickly and conveniently develop any new scheme with a minimum of code writing. The underlying goal here is to follow new strategies coming from the literature and, if necessary, to provide any additional components required for their implementation. Moreover, ParadisEO-MOEO constantly evolves and new features might be added to the framework regularly in order to provide a wide range of efficient and modern concepts and to reflect the most recent advances of the EMO field.

5.3.3 Main Characteristics

A framework is usually intended to be exploited by a large number of users. Its exploitation could only be successful if a range of user criteria are satisfied. Therefore, the main goals of the ParadisEO software framework are the following ones:

[2] `http://paradiseo.gforge.inria.fr`
[3] `http://eodev.sourceforge.net`

- *Maximum design and code reuse.* The framework must provide a whole architecture design for the metaheuristic approach to be used. Moreover, the programmer should need to redo as little code as possible. This aim requires a clear and maximal conceptual separation of the solution methods and the problem to be solved. The user might only write the minimal problem-specific code and the development process might be done in an incremental way, what will considerably simplify the implementation and reduce the development time and cost.
- *Flexibility and adaptability.* It must be possible to easily add new features or to modify existing ones without involving other components. Users must have access to source code and use inheritance or specialization concepts of object-oriented programming to derive new components from base or abstract classes. Furthermore, as existing problems evolve and new others arise, the framework components must be conveniently specialized and adapted.
- *Utility.* The framework must cover a broad range of metaheuristics, fine-grained components, problems, parallel and distributed models, hybridization mechanisms, etc.
- *Transparent and easy access to performance and robustness.* As the optimization applications are often time-consuming, the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. Moreover, the execution of the algorithms must be robust in order to guarantee the reliability and the quality of the results. Hybridization mechanisms generally allow to obtain robust and better solutions.
- *Portability.* In order to satisfy a large number of users, the framework must support many physical architectures (sequential, parallel, distributed) and their associated operating systems (Windows, Linux, MacOS).
- *Usability and efficiency.* The framework must be easy to use and must not contain any additional cost in terms of time or space complexity in order to keep the efficiency of a special-purpose implementation. On the contrary, the framework is intented to be less error-prone than a specifically developed metaheuristic.

The ParadisEO platform honors all the above-mentioned criteria and aims to be used by both non-specialists and optimization experts. Furthermore, the ParadisEO-MOEO module must cover additional goals related to multi-objective optimization. Thus, in terms of design, it might, for instance, be a commonplace to extend a single-objective optimization problem to the multi-objective case without modifying the whole metaheuristic implementation.

5.3.4 Existing Software Frameworks for Evolutionary Multi-Objective Optimization

Many frameworks dedicated to the design of metaheuristics have been proposed so far. However, very few are able to handle MOPs, even if some of them provide components for a few particular EMO strategies, such as ECJ [1], JavaEVA [42] or Open BEAGLE [20]. Table 5.1 gives a non-exhaustive comparison between a number of existing software frameworks for multi-objective metaheuristics, including jMetal [17], the MOEA toolbox for Matlab [45], MOMHLib++ [2], PISA [7]

Table 5.1. Main characteristics of some existing frameworks for multi-objective metaheuristics

Framework	Problems		Statistical tools		Hybrid.	Parallel	Type	Lang.	License
	Cont.	Comb.	Off-line	On-line					
jMetal	yes	yes	yes	no	yes	no	white	java	free
MOEA for Matlab	yes	no	no	no	no	yes	black	matlab	free / com.
MOMHLib++	yes	yes	no	no	yes	no	white	c++	free
PISA	yes	yes	yes	no	no	no	black	any	free
Shark	yes	no	no	no	yes	no	white	c++	free
ParadisEO	yes	yes	yes	yes	yes	yes	white	c++	free

and Shark [3]. Note that other software exists for multi-objective optimization [38], but some of them cannot be considered as frameworks and others do not deal with metaheuristics. The frameworks presented in Table 5.1 are distinguished according to the following criteria: the kind of MOPs they are able to tackle (continuous and/or combinatorial problems), the availability of statistical tools (including performance metrics), the availability of hybridization or parallel features, the framework type (black box or white box), the programming language and the license type (free or commercial).

First, let us mention that every listed software framework is free of use, except for the MOEA toolbox, which requires the commercial software Matlab. They can all handle continuous problems, but only a subset of them are able to deal with combinatorial MOPs. Moreover, some cannot be considered as white-box frameworks since their architecture is not decomposed into components. For instance, to design a new algorithm under PISA, it is necessary to implement it from scratch, as no existing element can be reused. Similarly, even if Shark can be considered as a white-box framework, its components are not as fine-grained as the ones of ParadisEO. On the contrary, ParadisEO is an open platform where anyone can contribute and add his/her own features. Finally, only a few ones are able to deal with hybrid and parallel metaheuristics at the same time. Hence, in opposition to jMetal and MOMHLib++, ParadisEO offers easy-to-use model for the design of parallel and distributed features. Therefore, in comparison to other existing software frameworks dedicated to multi-objective metaheuristics design, ParadisEO is the only one that achieves all the aforementioned goals.

5.4 Design and Implementation of Evolutionary Multi-Objective Metaheuristics with ParadisEO-MOEO

This section gives a detailed description of the base classes provided within the ParadisEO framework to design an EMO algorithm.[4] The flexibility of the framework and its modular architecture, based on the three main multi-objective metaheuristic design issues (fitness assignment, diversity preservation and elitism), allows to implement efficient algorithms in solving a large diversity of MOPs. The granular

[4] The classes presented in this paper are described as in version 1.2 of ParadisEO.

decomposition of ParadisEO-MOEO is based on the unified model proposed in the previous section.

As an EMO algorithm differs of a single-objective one only in a number of points, some ParadisEO-EO components are directly reusable. Therefore, in the following, note that the names of ParadisEO-EO classes are all prefixed by `eo` whereas the names of ParadisEO-MOEO classes are prefixed by `moeo`. ParadisEO is an object-oriented platform, so that its components will be specified by the UML standard [4]. But, due to space limitations, only a subset of the UML diagrams is provided, but the whole inheritance diagram as well as the classes documentation and many examples of use are available at the ParadisEO website. Moreover, a large part of the ParadisEO components are based on the notion of template and are defined as class templates. This concept and many related functions are featured within the C++ programming language and allows the classes to handle generic types, so that they can work with many different data types without having to be rewritten for each one.

In this section, both problem-dependent and problem-independent components are detailed. First, basic elements (representation, evaluation, initialization and stopping criteria) are outlined. Then comes the EMO-specific (fitness, diversity and elitism) and EA-related (variation, selection, replacement) components. Finally, the way to build a whole EMO algorithm is presented and a brief discussion concludes the section.

5.4.1 Basic Components

In this section, basic components are presented: solution representation, evaluation, initialization and stopping criteria.

Representation

Solution representation is the starting point for anyone who plans to design any kind of metaheuristic. Successful applications of metaheuristics strongly requires a proper solution representation. Various encodings may be used such as binary variables, real-coded vectors, permutations, discrete vectors, and more complex representations. Note that the choice of a representation will considerably influence the way solutions will be initialized and evaluated in the objective space, and the way variation operators will be applied. A solution needs to be represented both in the decision space and in the objective space. While the representation in the objective space can be seen as problem-independent, the representation in the decision space must be relevant to the tackled problem. In the single-objective case, a single value is usually used for the representation in the unidimensional objective space. For MOPs, where the objective space is multi-dimensional, a tuple of n values, called *objective vector*, might be used for such a representation. Using ParadisEO-MOEO, the first thing to do is to set the number of objectives for the problem under consideration and, for each one, if it has to be minimized or maximized. This can be done by using the `moeoObjectiveVectorTraits` static class. Then, a class templatized with the later one and inheriting of `moeoObjectiveVector` has to

be created for the representation of an objective vector, as illustrated in Fig. 5.1. Since a big majority of MOPs deal with real-coded objective values, a class modelling real-coded objective vectors is already provided within ParadisEO-MOEO. Note that this class can be used for any MOP without loss of generality.

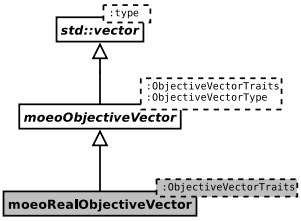


Fig. 5.1. UML diagram for the representation of a solution in the objective space

The class used to represent a whole solution within ParadisEO-MOEO is then templatized within a given objective vector type, and must define its representation in the decision space, which fully depends of the tackled problem. In the implementation-level, the way to do so is to extend the `MOEO` class in order to be used for a specific problem. This modeling is applicable for every kind of problem with the aim of being as general as possible. Nevertheless, ParadisEO-MOEO also provides easy-to-use classes for standard vector-based representations and, in particular, implementations for vectors composed of bits, of integers or of real-coded values, that can thus directly be used in a ParadisEO-MOEO-designed application. These classes are summarized in Fig. 5.2.

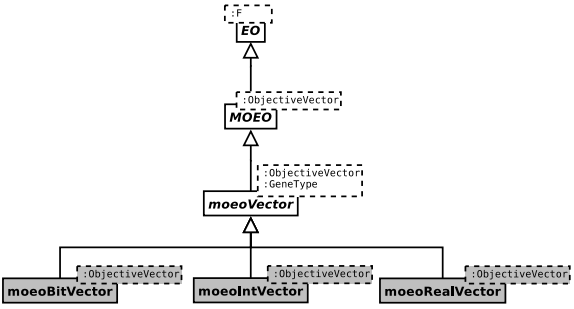


Fig. 5.2. UML diagram for the representation of a solution

Evaluation

The problem at hand is to optimize a set of objective functions simultaneously over a given search space. Then, each time a new solution integrates the population,

its objective vector must be evaluated, *i.e.* the value corresponding to each objective function must be set. ParadisEO-MOEO stores an objective vector within any `MOEO` object, and the way it is computed is ensured by components inheriting of the `eoEvalFunc` abstract class which is illustrated in Fig. 5.3. It basically takes a `MOEO` object and sets its objective vector. Moreover, note that a C++ function can be embedded into an `eoEvalFuncPtr` object in order to apply it to the individual and to set its objective values. Similarly, the `eoExternalEvalFunc` class provides a component able to embed an external evaluation function which is then considered as a black-box function by the problem solver. Finally, the `eoEvalFuncCounter` class allows to count the number of evaluations performed until the end of the algorithm. The resulting counter can either serve as a stopping criteria or provide a statistical resource to the user.

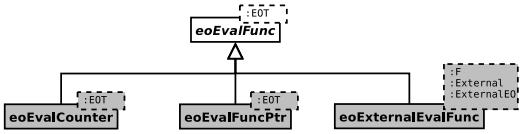


Fig. 5.3. UML diagram for evaluation

Generally speaking, for real-world optimization problems, the evaluation of a solution in the objective space is by far the most computationally expensive step of the chosen metaheuristic approach. A possible way to overcome this trouble is the use of parallel and distributed models, that can largely be simplified in the context of ParadisEO thanks to the ParadisEO-PEO module of the software library. The reader is referred to [10] for more information on how to design parallel and distributed metaheuristics within ParadisEO-PEO.

Initialization

Whatever the algorithmic solution to be designed, a way to initialize a solution (or a population of solutions) is expected. While dealing with any population-based metaheuristic, one has to keep in mind that the initial population must be diversified in order to prevent a premature convergence. This remark is even more true for MOPs where the goal is to find a well-converged and a well-spread approximation. The way to initialize a solution is closely related to the problem under consideration and to the representation at hand. In most approaches, the initial population is generated randomly or according to a given diversity function. A number of initialization schemes already exist in a lot of libraries for standard representations, which is also the case within ParadisEO. But some situations could require a combination of many operators or a specific implementation. Indeed, as shown in Fig. 5.4, the framework provides a range of initializers all inheriting of `eoInit`, as well as an easy way to combine them thanks to an `eoCombinedInit` object.

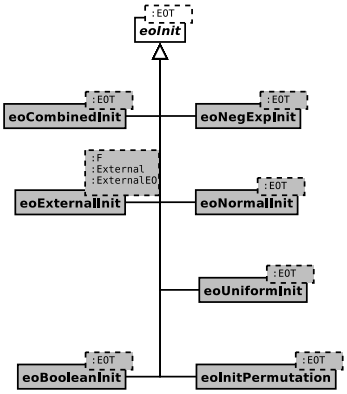


Fig. 5.4. UML diagram for initialization

Stopping Criteria, Checkpointing and Statistical Tools

Since an iterative method computes successive approximations, a practical test is required to determine when the process must stop. As illustrated in Fig. 5.5, in the frame of ParadisEO, many stopping criteria extending `eoContinue` are provided. For instance, the algorithm can stop after a given number of iterations (`eoGenContinue`), a given number of evaluations (`eoEvalContinue`), a given run time (`eoTimeContinue`) or in an interactive way, as soon as the user decides to (`eoCtrlCContinue`). Moreover, note that different stopping criteria can be combined thanks to an `eoCombinedContinue` object, in which case the process stops once one of the embedded criteria is satisfied.

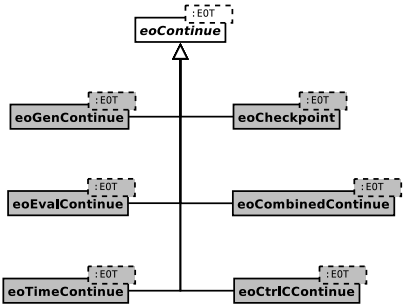


Fig. 5.5. UML diagram for stopping criteria

In addition, many other procedures may be called at each iteration of the main algorithm. The `eoCheckPoint` class allows to perform some systematic actions at each algorithm iteration in a transparent way by being embedded in the global

`eoContinue` object. The checkpointing engine is particularly helpful for fault tolerance mechanisms and to compute statistics. Indeed, some useful statistical tools are also provided within ParadisEO-MOEO. Then, it is for instance possible to save the contents of the current approximation set at each iteration, so that the evolution of the current non-dominated front can be observed or studied using graphical tools such as Guimoo (Graphical User Interface for Multi-Objective Optimization)[5]. Furthermore, as pointed out in Sect. 5.2, an important issue in the EMO field relates to the algorithm performance analysis and to set quality metrics [52]. As shown in Fig. 5.6, a couple of metrics are featured within ParadisEO-MOEO. Unary metrics are used to quantify the quality of a non-dominated set (or of a single solution), while binary metrics are used for pairwise comparisons (between two non-dominated sets or solutions). Thus, the hypervolume metric is available both in its unary [51] and its binary [52] form. Moreover, the entropy metric [5], the contribution metric [35] as well as the additive and the multiplicative ε-indicators [52] are all implemented and can thus be used to compare two sets of solutions. Besides, some implementations for pairwise comparison of solutions (that are then usable within the binary indicator-based fitness assignment schemes, see Sect. 5.4.2) are also proposed. Of course, other metrics can easily be implemented by inheritance. Another interesting feature is the possibility to compare the current archive with the archive of the previous iteration by using a given binary metric, and to print the progression of this measure iteration after iteration.

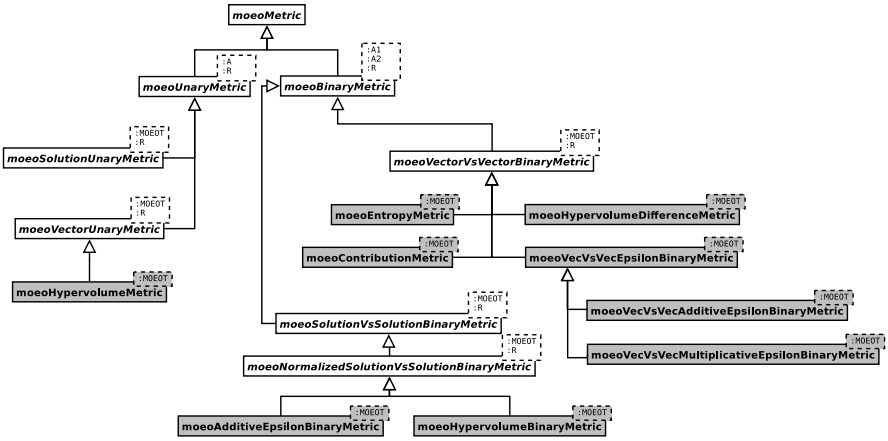


Fig. 5.6. UML diagram for metrics

5.4.2 EMO-Related Components

Here, we give a detailed description of EMO-specific components: fitness, diversity and elitism.

[5] `http://guimoo.gforge.inria.fr/`

Fitness Assignment Schemes

The most common fitness assignment strategies are implemented within ParadisEO-MOEO: scalar approaches, dominance-based approaches and indicator-based approaches. Following the taxonomy introduced in Sect. 5.2, the fitness assignment schemes are classified into four categories, as illustrated in the UML diagram of Fig. 5.7:

- Scalar approaches: `moeoScalarFitnessAssignment`
- Criterion-based approaches: `moeoCriterionBasedFitnessAssignment`
- Dominance-based approaches: `moeoDominanceBasedFitnessAssignment`
- Indicator-based approaches: `moeoIndicatorBasedFitnessAssignment`

A detailed description of existing fitness assignment schemes provided within the framework are listed below. Moreover, note that there also exists a dummy fitness assignment strategy in case it would be useful for some specific implementation.

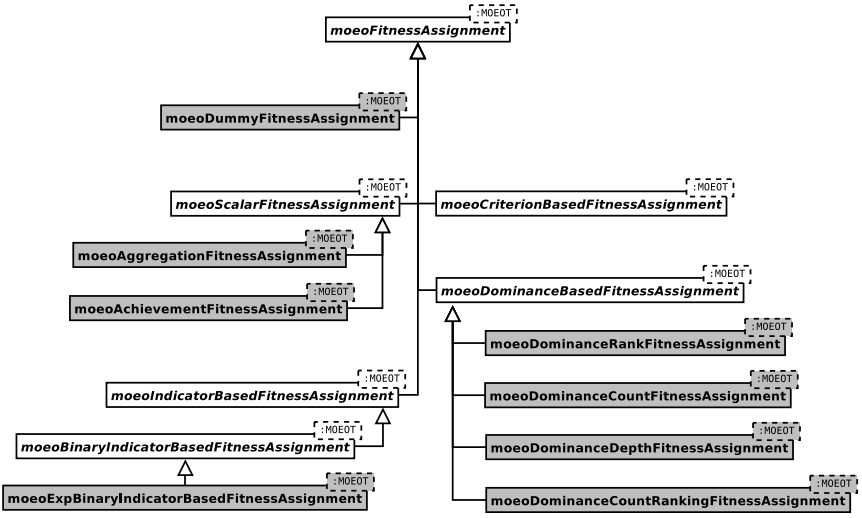


Fig. 5.7. UML diagram for fitness assignment

Achievement Fitness Assignment Scheme

One of the provided fitness assignment schemes is the family of *achievement scalarizing functions*, proposed by Wierzbicki [47]. This scalar approach is based on an arbitrary reference point R, generally given by a decision maker, and consists in projecting R onto the set of Pareto optimal solutions.

Dominance-Rank Fitness Assignment Scheme

In this strategy, the fitness value associated to a given solution x corresponds to the number of population items that dominate x. This scheme has been proposed in [18]

and is, for instance, used in the Fonseca and Fleming MOGA (Multi-Objective GA) [18] and in the Horn et al. NPGA (Niched-Pareto GA) [26].

Dominance-Count Fitness Assignment Scheme

This approach consists in assigning, to a solution x, a fitness value equal to the number of population items that are dominated by x. For instance, it is combined to the dominance rank scheme in the frame of SPEA (Strength Pareto EA) [51] and SPEA2 [50].

Dominance-Depth Fitness Assignment Scheme

Another implemented fitness assignment scheme is the dominance depth approach proposed by Goldberg [21] and used, for instance, in NSGA (Non-dominated Sorting GA) [41] and NSGA-II [14]. This strategy consists in classifying a set of solutions into several classes (or fronts). A solution that belongs to a class does not dominate another one from the same class. Then, individuals from the first front all belong to the best non-dominated set of the population; individuals from the second front all belong to the second best non-dominated set; and so on.

Dominance Count Ranking Fitness Assignment Scheme

In this strategy, the dominance-count and dominance-rank schemes are combined. The fitness value of a solution x corresponds to the sum of ranks of all solutions dominated by x. This technique is used in SPEA2 [50].

Binary Indicator-Based Fitness Assignment Scheme

In this strategy, the fitness values are computed by comparing individuals on the basis of an arbitrary binary quality indicator I (or binary performance metric). Thereby, no particular diversity preservation mechanism is generally necessary, with regards to the indicator being used. The chosen indicator represents the overall goal of the search process. Thus, the fitness value of a solution reflects its usefulness according to this goal. As discussed earlier in the chapter (see Sec. 5.4.1), several binary quality indicators to be used in the frame of this scheme are proposed within ParadisEO-MOEO.

Diversity Assignment Schemes

As illustrated in the UML diagram of Fig. 5.8, diversity preservation strategies must inherit of the `moeoDiversityAssignment` class. In addition to a dummy technique, a number of other diversity assignment schemes are available and are listed below.

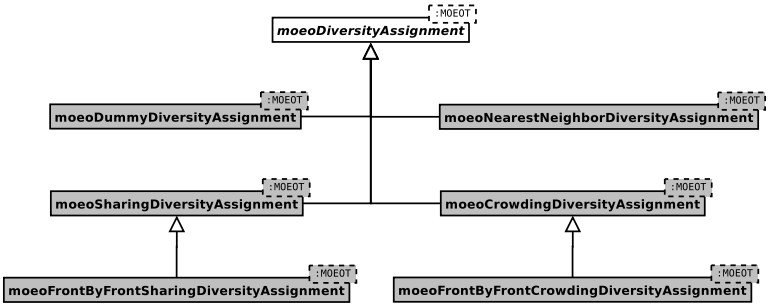


Fig. 5.8. UML diagram for diversity assignment

Sharing Diversity Assignment Scheme

The notion of *sharing* (or *fitness sharing*) was originally suggested by Goldberg and Richardson [22] to preserve diversity among solutions in an EA population. It was first employed by Fonseca and Fleming [18] in the frame of EMO. This *kernel* method consists in estimating the distribution density of a solution using a so-called *sharing function* that is related to the sum of distances to its neighborhood solutions. A sharing distance parameter specifies the similarity threshold, *i.e.* the size of *niches*. The distance measure between two solutions can be defined in the decision space, in the objective space or can combine both. A distance metric partly or fully defined in the parameter space strongly depends of the tackled problem. But, standard normalized distance metrics defined in the objective space are already provided within ParadisEO-MOEO for real-coded objective vectors. Sharing is one of the most popular technique and is commonly used in a large number of EMO algorithms such as MOGA [18], NPGA [26], NSGA [41] and more. Note that a 'front by front' procedure is also proposed as, in some cases [18, 41], sharing only occurs between solutions of same rank.

Nearest Neighbor Diversity Assignment Scheme

The nearest neighbor diversity maintaining strategy computes the distance between a given objective vector and its nearest neighbors in order to estimate the density of its neighborhood. The density estimator is generally based on the volume of the hyper-rectangle defined by these nearest neighbors. For instance, the SPEA2 [50] diversity preservation mechanism is based on this technique.

Crowding Diversity Assignment Scheme

Another diversity assignment scheme is the concept of *crowding*, firstly suggested by Holland [25] and used by De Jong [28] to prevent *genetic drift* [28]. It is employed by Deb et al. [14] in the NSGA-II. Contrary to the sharing diversity preservation scheme, this one allows to maintain diversity without specifying any parameter. It consists in estimating the density of solutions surrounding a particular

point of the objective space. As before, a similar mechanism working on sub-classes of solutions is also provided within ParadisEO-MOEO.

Elitism

As shown in Fig. 5.9, an archive is represented by the `moeoArchive` abstract class and is a population using a particular dominance relation to update its contents. An abstract class for fixed-size archives is given: `moeoFixedArchive`. But implementations of an unbounded archive (`moeoUnboundedArchive`), a general-purpose bounded archive based on a fitness and/or a diversity assignment scheme(s) (`moeoBoundedArchive`) as well as the SPEA2 archive (`moeoSPEA2Archive`) are provided. Generally speaking, the dominance relation used to update the archive contents is the Pareto-dominance relation, which is employed by default. But, other dominance criteria are found in the literature. Therefore, the framework offers the opportunity to use any dominance relation for that purpose by means of a `moeoObjectiveVectorComparator` object. As shown in Fig. 5.10, implemented criteria consist of Pareto-dominance, weak-dominance, strict-dominance, ε-dominance [24], and g-dominance [37]. Usually, an archive is used as an external

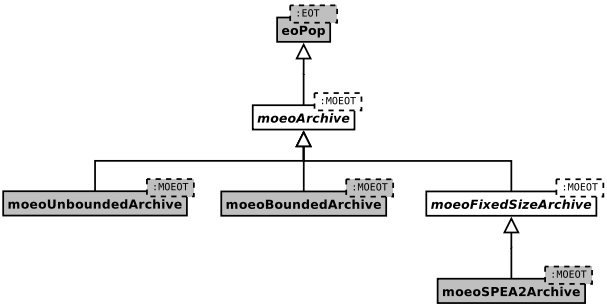


Fig. 5.9. UML diagram for archiving

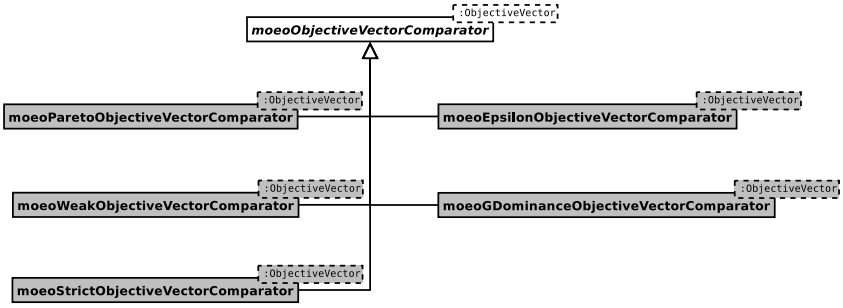


Fig. 5.10. UML diagram for for dominance relation (used for pairwise objective vector comparison)

storage only. However, we will see in the next section that archive members can also be used during the selection phase of an EMO algorithm.

5.4.3 EA-Related Components

EA-related components are presented below: variation operators as well as selection and replacement mechanisms.

Variation

The purpose of variation operators is to modify the representation of solution in order to move them in the search space. Generally speaking, while dealing with EAs, these problem-dependent operators are stochastic. They can be classified according to the number of arguments they use or modify, *i.e.*:

- Variation operators involving two individuals are called *recombination operators*. They can either modify one parent according to the material of the other one, or modify both parents. At the implementation level, the former are called *binary operators* and the latter *quadratic operators*.
- Variation operators involving a single individual are *mutation operators*.

Note that straightforward extensions allow to combine these simple operators. For instance, in a standard proportional combination, a given operator is selected from among a set of operators based on some user-defined rates. Furthermore, other variation operators generating any number of offspring from any number of parents, called *general operators*, can also be defined.

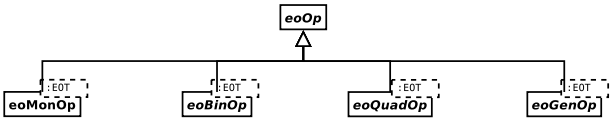


Fig. 5.11. UML diagram for variation

As shown in Fig. 5.11, in terms of implementation, all variation operators must derive from the `eoOp` base class. Four abstract classes inherit of `eoOp`, namely `eoMonOp` for mutation operators, `eoBinOp` and `eoQuadOp` for recombination operators and `eoGenOp` for other kinds of variation operators. Various operators of the same arity can be combined using some helper classes. Note that variation mechanisms for some classical (real-coded, vector-based or permutation-based) representations are already provided in the framework. Moreover, a hybrid mechanism can easily be designed by using a single-objective local mechanism search as a mutation operator, as they both inherit from `eoMonOp`, see ParadisEO-MO [8]. All variation operators designed for a given problem must be embedded into a `eoTranform` object.

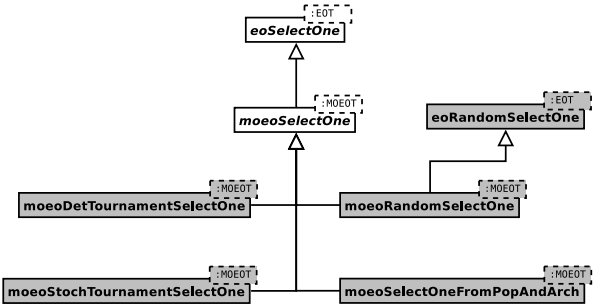


Fig. 5.12. UML diagram for selection

Selection Schemes

The selection step is one of the main search operators of EAs. It consists in choosing some solutions that will be used to generate the offspring population. In general, the better is an individual, the higher is its chance of being selected, so that fitness and/or the diversity value(s) are normally used. There exists a large number of selection strategies in the frame of EMO. Four ones are provided within ParadisEO-MOEO (see Fig. 5.12):

- A *random selection* (`moeoRandomSelectOne`), that consists in selecting a parent randomly among the population members, without taking fitness nor diversity information into account.
- A *deterministic tournament selection* (`moeoDetTournamentSelectOne`), that consists in performing a tournament between m randomly chosen population members and in selecting the best one.
- A *stochastic tournament selection* (`moeoStochTournamentSelectOne`), that consists in performing a binary tournament between randomly chosen population members and in selecting the best one with a probability p or the worst one with a probability $(1-p)$.
- An *elitist selection* (`moeoSelectOneFromPopAndArch`), that consists in selecting a population member based on some selection scheme with a probability p, or in selecting an archive member using another selection scheme with a probability $(1-p)$. So, elite solutions also contribute to the evolution engine by being used as parents. This scheme has been integrated in various elitist EMO algorithms including SPEA [51], SPEA2 [50] or PESA [12].

All these selection methods are of the `moeoSelectOne` type and need to be embedded into an `eoSelect` object to be properly used. Of course, everything is done to easily implement a new selection scheme with a minimum programming effort.

Replacement Schemes

Selection pressure is also affected at the replacement step where survivors are selected from both the current and the offspring population. A large majority

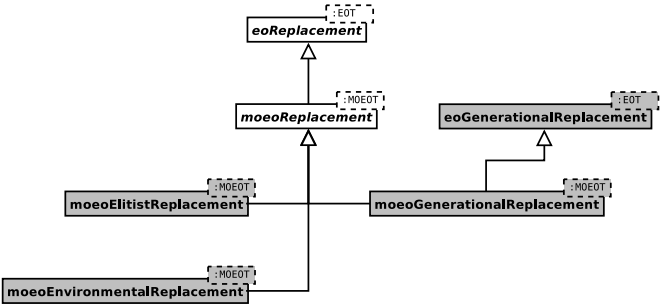


Fig. 5.13. UML diagram for replacement

of replacement strategies depend on the fitness and/or the diversity value(s) and are, somehow, EMO-specific. Three replacement schemes are provided within ParadisEO-MOEO (see Fig. 5.13), but this list is not exhaustive as new ones can easily be implemented due to the genericity of the framework.

- A *generational replacement* (`moeoGenerationalReplacement`), that consists in keeping the offspring population only, while all parents are deleted.
- An *elitist replacement* (`moeoElitistReplacement`), that consists in choosing the N best solutions (where N stands for the population size).
- An *environmental replacement* (`moeoEnvironmentalReplacement`), that consists in deleting one-by-one the worst individuals, and in updating the fitness and the diversity values of the remaining solutions each time there is a deletion. The process ends once the required population size is reached.

5.4.4 Evolutionary Multi-Objective Optimization Algorithms

Now that all the basic, EMO-specific and EA-related components are defined, an EMO algorithm can easily be designed using the fine-grained classes of ParadisEO. As the implementation is conceptually divided into components, different operators can be experimented without engendering significant modifications in terms of code writing. As seen before, a wide range of components are already provided. But, keep in mind that this list is not exhaustive as the framework perpetually evolves and offers all that is necessary to develop new ones with a minimum effort. Indeed, ParadisEO is a white-box framework that tends to be flexible while being as user-friendly as possible. Fig. 5.14 illustrates the use of the `moeoEasyEA` class that allows to define an EMO algorithm in a common fashion, by specifying all the particular components required for its implementation. All classes use a template parameter *MOEOT (Multi-Objective Evolving Object Type)* that defines the representation of a solution for the problem under consideration. This representation might be implemented by inheriting of the `MOEO` class as described in Sect. 5.4.1. Note that archive-related components do not appear in the UML diagram, as we chose to let the use of an archive as optional. The archive update can easily be

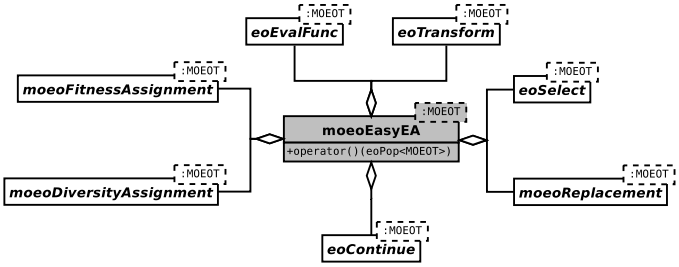


Fig. 5.14. UML diagram for the design of an EMO algorithm

integrated into the EA by means of the checkpointing process. Similarly, the initialization process does not appear either, since an instance of `moeoEasyEA` starts with an already initialized population.

Easy-to-use EMO Algorithms

In order to satisfy both the common user and the more experimented one, ParadisEO-MOEO also provides even more easy-to-use EMO algorithms (see Fig. 5.15). These classes propose different implementations of some state-of-the-art algorithms by using the fine-grained components of ParadisEO. Hence, MOGA [18], NSGA [41], NSGA-II [14], SPEA2 [50], IBEA [48] and SEEA [34] are proposed in a way that a minimum number of problem- or algorithm-specific parameters are required. These easy-to-use algorithms also tend to be used as references for a fair performance comparison in the academic world, even if they are also well-suited for a straight use to solve real-world MOPs. In a close future, other easy-to-use multi-objective metaheuristics will be proposed while new fined-grained components will be implemented into the frame of ParadisEO-MOEO.

5.4.5 Discussion

ParadisEO-MOEO has been used and experimented to solve a large range of MOPs from both academic and real-world fields, which evidences its high flexibility. Indeed, various academic MOPs have been tackled within ParadisEO-MOEO, including continuous test functions (like the ZDT and DTLZ functions family defined in [16]), scheduling problems (permutation flow-shop scheduling problem [32]), routing problems (multi-objective traveling salesman problem, bi-objective ring star problem [34]), etc. Moreover, it has been successfully employed to solve real-world applications in structural biology [9], feature selection in cancer classification [44], data-mining [29], materials design in chemistry [40], etc. Besides, a detailed documentation as well as some tutorial lessons and problem-specific implementations are freely available on the ParadisEO website[6]. And we expect the number of MOP

[6] `http://paradiseo.gforge.inria.fr`

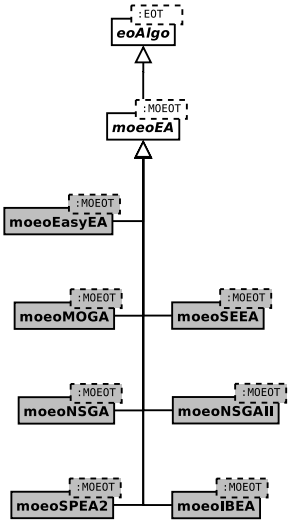


Fig. 5.15. UML diagram for easy-to-use EMO algorithms

contributions to largely grow in a near future. Furthermore, note that the implementation of EMO algorithms is just an aspect of the features provided by ParadisEO. Indeed, the whole framework allows to conveniently design hybrid as well as parallel and distributed metaheuristics, including EMO methods. Hence, hybrid mechanisms can be exploited in a natural way to make cooperating metaheuristics belong to the same or to different classes. Moreover, the three main parallel models are concerned: algorithmic-level, iteration-level and solution-level and are portable on different types of architecture. For instance, in the frame of ParadisEO, hybrid EMO algorithms have been experimented in [34], a multi-objective cooperative island model has been designed in [43], and costly evaluation functions have been parallelized in [9]. The reader is referred to [10] for more information about ParadisEO's hybrid and parallel models.

5.5 Case Study: An EMO Algorithm for a Bi-objective Scheduling Problem

The Flow-shop Scheduling Problem (FSP) is one of the most well-known scheduling problems and has been widely studied in the literature. The majority of works dedicated to the FSP considers it on a single-objective form and mainly aim at minimizing the makespan (*i.e.* the total completion time). However, many objective functions, varying according to the particularities of the tackled problem, may be considered and some multi-objective approaches have also been proposed. For a survey, see for instance [31, 46].

5.5.1 Problem Definition

Solving the FSP consists in scheduling a set of N jobs $J_1, J_2, \ldots, J_N$ on M machines $M_1, M_2, \ldots, M_M$. Machines are critical resources, *i.e.* one machine cannot process more than one job at a time. Each job J_i is composed of M consecutive tasks $t_{i1}, t_{i2}, \ldots, t_{iM}$, where t_{ij} represents the jth task of the job J_i, requiring the machine M_j. A processing time p_{ij} is associated to each task t_{ij}; and a due date d_i is given to each job J_i (the deadline of the job). In this study, we focus on the permutation FSP, where the operating sequences of the jobs are identical and unidirectional on every machine, as illustrated in Fig. 5.16. Many objective functions may be

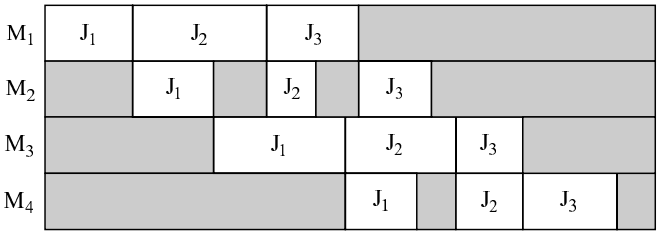


Fig. 5.16. An example of solution for a permutation flow-shop problem where 3 jobs (J_1, J_2, J_3) have to be scheduled on 4 machines (M_1, M_2, M_3, M_4).

tackled while scheduling tasks on several machines. The FSP that we consider here aims at minimizing the makespan (C_{max}) and the total tardiness ($\overline{T}$). These objectives are among the most widely investigated in the literature. For each task t_{ij} being scheduled at the time s_{ij}, they are computed as follows:

$$C_{max} = \max_{i \in \{1,\ldots,N\}} \{s_{iM} + p_{iM}\} \tag{5.1}$$

$$\overline{T} = \sum_{i=1}^{N} \left\{ \max\{0, s_{iM} + p_{iM} - d_i\} \right\} \tag{5.2}$$

According to the Graham et al. notation [23], the problem under consideration can be denoted by $F/perm, d_i/(C_{max}, \overline{T})$.

5.5.2 Implementation

In this section, we focus on the implementation of an EMO algorithm to approximate the efficient set for the FSP presented above. First, the design and the implementation of problem-dependent components are discussed. Then, the choice of problem-independent components is presented. And finally, the implementation of the EMO algorithm is given. Note that this case study is closely related to a ParadisEO-MOEO tutorial available on the website, so that detailed source code can easily be retrieved.

Problem-Dependent Components Design

Below are presented the problem-dependent components designed for the problem under consideration: solution representation, evaluation, initialization and variation operators.

Representation

First of all, let us define the number of objectives for the problem under consideration, and if they are to be minimized or maximized. This is done by specializing the `moeoObjective VectorTraits` class. In our case, let denote the specialized class by `fspObjVecTraits`. Then, the representation in the objective space can be defined as a real-coded objective vector:

```
typedef moeoRealObjectiveVector<fspObjVecTraits> fspObjVec;
```

Now, for the representation in the decision space, we use a permutation-based encoding. So, let us define our solution type `FSP` by a vector of integer:

```
typedef moeoIntVector<fspObjVec> FSP;
```

Evaluation

The evaluation class has to evaluate the values of a given solution for every objective, *i.e.* the makespan and the total tardiness for the problem under consideration. We here define a `fspEval` class inheriting of `eoEvalFunc`:

```
class fspEval : public eoEvalFunc<FSP>
{
public:
    void operator() (FSP & _fsp)
    {
        fspObjVec objVec;
        objVec[0] = makespan(_fsp);
        objVec[1] = tardiness(_fsp);
        _fsp.objectiveVector(objVec);
    }
    // ...
}
```

Initialization

ParadisEO already provides an implementation for initializing permutations. Therefore, let us define a `fspInit` class of the `eoInitPermutation` type:

```
typedef eoInitPermutation<FSP> fspInit;
```

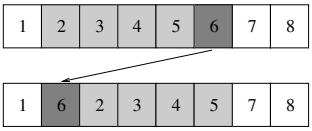


Fig. 5.17. Shift mutation.

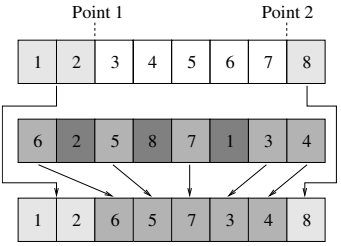


Fig. 5.18. Two-point crossover

Variation

Regarding variation operators, we choose to use a shift mutation and to implement a two-point crossover as described in [27]. These operators are respectivelly illustrated in Fig. 5.17 and Fig. 5.18.

```
typedef eoShiftMutation<FSP> fspMutation;


class fspCrossover : public eoQuadOP<FSP>
{
public:
    bool operator() (FSP & _fsp1, FSP & _fsp2)
    {
        // ...
    }
}
```

Problem-Independent Components Decision

We arbitrary choose to use the following problem-independent components in the frame of the EMO algorithm designed in this section. Of course, different operators can easily be experimented with a very low coding effort.

- *Fitness assignment*: dominance-rank.
- *Diversity assignment*: sharing.
- *Elitism*: unbounded archive.
- *Selection*: deterministic tournament.
- *Replacement*: elitist.
- *Stopping criteria*: maximum number of generations.

EMO Algorithm Design

The source code of the main program file is given below. First, some parameters are given. Then, problem-dependent and problem-independent components are instantiated. Finally, the algorithm is built and is applied to the initialized population.

```
/* parameters */
// population size
int _popSize = 100;
// crossover probability
double _pCross = 0.25;
// mutation probability
double _pMut = 1.0;
// tournament size for selection
int _tourSize = 2;
// maximum number of generations
int _maxGen = 1000;

/* representation-dependent components */
// evaluation
fspEval eval;
// initialization
fspInit init;
// variation operators
fspCrossover cross;
fspMutation mut;
eoSGATransform<FSP> op(cross,_pCross,mut,_pMut);

/* representation-independent components */
// initial population
eoPop<FSP> pop(_popSize,init);
// unbounded archive
moeoUnboundedArchive<FSP> arch;
// fitness assignment
moeoDominanceRankFitnessAssignment<FSP> fitness;
// diversity assignment
moeoSharingDiversityAssignment<FSP> diversity;
// selection
moeoDetTournamentSelect<FSP> select(_tourSize);
// replacement
moeoElitistReplacement<FSP> replace(fitness,diversity);
// stopping criteria
eoGenContinue<FSP> stop(_maxGen);
// checkpoint
eoCheckPoint<FSP> check(stop);
// archive updater
moeoArchiveUpdater<FSP> updater(arch,pop);
check.add(updater);
// algorithm
moeoEasyEA<FSP> algo
      (check,eval,select,op,replace,fitness,diversity);

/* apply the algorithm to the population */
algo(pop);
```

5.6 Conclusion

In this chapter, we first presented a unified view of evolutionary algorithms for solving multi-objective problems of both continuous and combinatorial optimization. The resulting flexible model, based on the fundamental issues of fitness, diversity and elitism, has been used as a starting point for the implementation of a general purpose software framework called ParadisEO-MOEO. Base-class components follow the fine-grained decomposition of the model and allow to design many resolution methods in a modular way, by combining different strategies at each stage of its conception, with a minimum programming effort. Many classical strategies for problem-independent components are already provided. ¿From this set of mechanisms, state-of-the-art algorithms such as NSGA-II, SPEA2 and IBEA have been implemented and are available. Nevertheless, new components and algorithms will be integrated in a near future, as we hope the framework to constantly evolve in order to follow the most recent advances of the literature. Furthermore, a clear conceptual separation of the problem-specific part and problem-independent part of the metaheuristic is provided, so that the representation, the initialization and the evaluation of a solution as well as variation operators are the only components that must be specifically implemented for the problem to be solved. However, the framework also proposes standard techniques for the most common representation encodings, in which case the user only has to implement the objective functions associated to his/her problem. In addition, the platform also includes the most well-known parallel and distributed models for metaheuristics and their hybridization.

A large part of components involved in evolutionary multi-objective optimization are shared by many other search methods. Hence, we plan to generalize the unified view presented in this chapter to additional population-based multi-objective metaheuristics, including local search, scatter search and particle swarm optimization approaches. Afterward, the resulting general purpose model will be implemented in a modular way in order to be integrated into the ParadisEO-MOEO software framework. As well, there is a growing need in the MCDM community to provide a powerful tool devoted to interactive multi-objective optimization. And we believe that such a strong and reliable framework like ParadisEO-MOEO is the ideal platform to provide base-class components to an higher level software where metaheuristics would be involved. Another interesting extension would be to add components to deal with stochastic and dynamic multi-objective optimization problems.

Acknowledgement. This work was supported by the ANR DOCK project. The authors would like to gratefully acknowledge Sébastien Cahon and Nouredine Melab for their work on the preliminary version of ParadisEO-MOEO, as well as Abdel-Hakim Deneche for his precious contribution on the implementation of some components presented in this chapter.

References

1. `http://cs.gmu.edu/~eclab/projects/ecj/`
2. `http://home.gna.org/momh/`
3. `http://shark-project.sourceforge.net/`

4. OMG unified modeling language specification. Object Management Group (2000)
5. Basseur, M., Seynhaeve, F., Talbi, E.G.: Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In: Congress on Evolutionary Computation (CEC 2002), Honolulu, Hawai, USA, vol. 2, pp. 1151–1156 (2002)
6. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
7. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA — a platform and programming language independent interface for search algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
8. Boisson, J.C., Jourdan, L., Talbi, E.G.: ParadisEO-MO. Tech. rep. (2008)
9. Boisson, J.C., Jourdan, L., Talbi, E.G., Horvath, D.: Parallel multi-objective algorithms for the molecular docking problem. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2008), Sun Valley Resort, Idaho, USA (2008)
10. Cahon, S., Melab, N., Talbi, E.G.: ParadisEO: A framework for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics 10(3), 357–380 (2004)
11. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
12. Corne, D., Knowles, J.D., Oates, M.J.: The pareto envelope-based selection algorithm for multi-objective optimisation. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 839–848. Springer, Heidelberg (2000)
13. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
14. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
15. Deb, K., Mohan, M., Mishra, S.: Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. Evolutionary Computation 13(4), 501–525 (2005)
16. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In: Abraham, A., Jain, R., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, ch. 6, pp. 105–145. Springer, Heidelberg (2005)
17. Durillo, J.J., Nebro, A.J., Luna, F., Dorrosoro, B., Alba, E.: jMetal: A java framework for developing multi-objective optimization metaheuristics. Tech. Rep. ITI-2006-10, University of Málaga (2006)
18. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Forrest, S. (ed.) Proceedings of the 5th International Conference on Genetic Algorithms (ICGA 1993), pp. 416–423. Morgan Kaufmann, Urbana-Champaign (1993)
19. Fourman, M.P.: Compaction of symbolic layout using genetic algorithms. In: Grefensette, J.J. (ed.) Proceedings of the 1st International Conference on Genetic Algorithms (ICGA 1985), pp. 141–153. Lawrence Erlbaum Associates, Pittsburgh (1985)
20. Gagné, C., Parizeau, M.: Genericity in evolutionary computation software tools: Principles and case study. International Journal on Artificial Intelligence Tools 15(2), 173–194 (2006)

21. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston (1989)
22. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Second International Conference on Genetic Algorithms and their application, pp. 41–49. Lawrence Erlbaum Associates, Inc., Mahwah (1987)
23. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics 5, 287–326 (1979)
24. Helbig, S., Pateva, D.: On several concepts for ε-efficiency. OR Spektrum 16(3), 179–186 (1994)
25. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Abor (1975)
26. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched pareto genetic algorithm for multiobjective optimization. In: IEEE Congress on Evolutionary Computation (CEC 1994), pp. 82–87. IEEE Press, Piscataway (1994)
27. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Systems, Man and Cybernetics 28, 392–403 (1998)
28. Jong, K.A.D.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D thesis, Ann Arbor, University of Michigan (1975)
29. Jourdan, L., Khabzaoui, M., Dhaenens, C., Talbi, E.G.: A hybrid evolutionary algorithm for knowledge discovery in microarray experiments. In: Olariu, S., Zomaya, A.Y. (eds.) Handbook of Bioinspired Algorithms and Applications, ch. 28, pp. 489–505. CRC Press, Boca Raton (2005)
30. Keijzer, M., Merelo, J.J., Romero, G., Schoenauer, M.: Evolving objects: A general purpose evolutionary computation library. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 231–244. Springer, Heidelberg (2002)
31. Landa Silva, J.D., Burke, E., Petrovic, S.: An introduction to multiobjective metaheuristics for scheduling and timetabling. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation. LNEMS, vol. 535, pp. 91–129. Springer, Berlin (2004)
32. Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.G.: Combinatorial optimization of stochastic multi-objective problems: an application to the flow-shop scheduling problem. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 457–471. Springer, Heidelberg (2007)
33. Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.G.: ParadisEO-MOEO: A framework for evolutionary multi-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 386–400. Springer, Heidelberg (2007)
34. Liefooghe, A., Jourdan, L., Talbi, E.G.: Metaheuristics and their hybridization to solve the bi-objective ring star problem: a comparative study. Tech. Rep. RR-6515, Institut National de Recherche en Informatique et Automatique, INRIA (2008)
35. Meunier, H., Talbi, E.G., Reininger, P.: A multiobjective genetic algorithm for radio network optimization. In: IEEE Congress on Evolutionary Computation (CEC 2000), pp. 317–324. IEEE Press, San Diego (2000)
36. Miettinen, K.: Nonlinear Multiobjective Optimization. International Series in Operations Research and Management Science, vol. 12. Kluwer Academic Publishers, Boston (1999)

37. Molina, J., Santana, L.V., Hernández-Díaz, A.G., Coello Coello, C.A., Caballero, R.: g-dominance: Reference point based dominance for multiobjective metaheuristics. European Journal of Operational Research 197(2), 685–692 (2009)
38. Poles, S., Vassileva, M., Sasaki, D.: Multiobjective optimization software. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 329–348. Springer, Heidelberg (2008)
39. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Grefensette, J.J. (ed.) Proceedings of the 1st International Conference on Genetic Algorithms (ICGA 1985), pp. 93–100. Lawrence Erlbaum Associates, Pittsburgh (1985)
40. Schuetze, O., Jourdan, L., Legrand, T., Talbi, E.G., Wojkiewicz, J.L.: New analysis of the optimization of electromagnetic shielding properties using conducting polymers and a multi-objective approach. Polymers for Advanced Technologies 19(7), 762–769 (2008)
41. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)
42. Streichert, F., Ulmer, H.: JavaEvA: a java based framework for evolutionary algorithms. Tech. Rep. WSI-2005-06, Centre for Bioinformatics Tübingen (ZBIT) of the Eberhard-Karls-University, Tübingen (2005)
43. Talbi, E.G., Cahon, S., Melab, N.: Designing cellular networks using a parallel hybrid metaheuristic on the computational grid. Computer Communications 30(4), 698–713 (2007)
44. Talbi, E.G., Jourdan, L., Garcia-Nieto, J., Alba, E.: Comparison of population based metaheuristics for feature selection: Application to microarray data classification. In: IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2008), pp. 45–52. IEEE, Los Alamitos (2008)
45. Tan, K.C., Lee, T.H., Khoo, D., Khor, E.F.: A multi-objective evolutionary algorithm toolbox for computer-aided multi-objective optimization. IEEE Transactions on Systems, Man and Cybernetics: Part B (Cybernetics) 31(4), 537–556 (2001)
46. T'Kindt, V., Billaut, J.C.: Multicriteria Scheduling: Theory, Models and Algorithms. Springer, Berlin (2002)
47. Wierzbicki, A.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Objective Decision Making, Theory and Application. LNEMS, vol. 177, pp. 468–486. Springer, Heidelberg (1980)
48. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
49. Zitzler, E., Laumanns, M., Bleuler, S.: A tutorial on evolutionary multiobjective optimization. In: Gandibleux, X., Sevaux, M., Swrensen, K. (eds.) Metaheuristics for Multiobjective Optimisation. LNEMS, vol. 535, pp. 3–38. Springer, Heidelberg (2004)
50. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Tech. Rep. 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001)
51. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
52. Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)

Computers & Operations Research 37 (2010) 1033–1044

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Metaheuristics and cooperative approaches for the Bi-objective Ring Star Problem

A. Liefooghe *, L. Jourdan, E.-G. Talbi

Laboratoire d'Informatique Fondamentale de Lille, UMR CNRS 8022, INRIA Lille-Nord Europe, Université Lille 1, Parc Scientifique de la Haute Borne, 40 avenue Halley, 59650 Villeneuve d'Ascq, France

ARTICLE INFO

Available online 17 September 2009

Keywords:
Ring star problem
Multi-objective optimization
Metaheuristic
Cooperative approach

ABSTRACT

This paper presents and investigates different approaches to solve a new bi-objective routing problem called the ring star problem. It consists of locating a simple cycle through a subset of nodes of a graph while optimizing two kinds of cost. The first objective is the minimization of a ring cost that is related to the length of the cycle. The second one is the minimization of an assignment cost from non-visited nodes to visited ones. In spite of its obvious bi-objective formulation, this problem has always been investigated in a single-objective way. To tackle the bi-objective ring star problem, we first investigate different stand-alone search methods. Then, we propose two cooperative strategies that combine two multi-objective metaheuristics: an elitist evolutionary algorithm and a population-based local search. We apply these new hybrid approaches to well-known benchmark test instances and demonstrate their effectiveness in comparison to non-hybrid algorithms and to state-of-the-art methods.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The Bi-objective Ring Star Problem (B-RSP) aims at locating a simple cycle through a subset of nodes of a graph. The first objective consists in minimizing a *ring cost* related to the cycle length. The second objective is to minimize an *assignment cost* from non-visited nodes to visited ones. Two conflicting costs are then considered here, which naturally leads to a bi-objective formulation. But this problem has always been addressed in a single-objective way; either where both costs are combined [1], or where the assignment cost is treated as a constraint [2]. However, from a practitioner point of view, it is often necessary to take multiple objectives into account simultaneously, and the problem investigated in this paper is a good illustration of what can typically be found in the industry. Indeed, as pointed out in [3], a large number of routing problems can be formulated as multi-objective optimization problems, and according to the same paper, the problem at hand is a generalization of the single-objective problems introduced in [1,2]. Besides, the B-RSP is highly combinatorial as, once is decided which nodes are to be visited, a standard traveling salesman problem is still to be solved. Therefore, in practice, large-scale problem instances generally cannot be solved exactly. This is the reason why heuristic search methods are considered in this paper. Note that the current work extends preliminary results presented during the EvoCOP 2008 conference [4]. Nevertheless, a deeper analysis, extra stand-alone methods and novel hybrid approaches are additionally provided in the actual paper.

The main contribution of this work is twofold and can be summarized as follows. First, a set of multi-objective search methods are proposed for a new bi-objective problem, the B-RSP. Second, this paper introduces new general-purpose cooperative schemes for multi-objective combinatorial optimization, and their features and efficiency are investigated and rather discussed on solving the problem at hand. Thus, as an initial step, we investigate four metaheuristics to approximate the set of efficient solutions for the problem under consideration. Hence, IBMOLS, a population-based local search recently proposed in [5], is fitted for the B-RSP resolution. Next, we present slight variations of two well-known multi-objective search methods, namely IBEA [6] and NSGA-II [7]. At last, we propose a general-purpose multi-objective metaheuristic called SEEA and presented in this paper for the first time. We compare all these methods to each other on state-of-the-art benchmark test instances and discuss their respective behaviors. As a second step, we propose two cooperative approaches combining the local search method and SEEA. These methods are based either on a periodic or on an adaptive strategy, each one trying to benefit of the advantages of each algorithm it is compound of. Experiments validate the contribution of these hybrid schemes over non-hybrid approaches by achieving a strict improvement of the generated set of non-dominated solutions.

The remainder of the paper is organized as follows. Section 2 is devoted to the B-RSP. Then, four metaheuristics for

* Corresponding author.
E-mail addresses: Arnaud.Liefooghe@lifl.fr (A. Liefooghe), Laetitia.Jourdan@lifl.fr (L. Jourdan), talbi@lifl.fr (E.-G. Talbi).

0305-0548/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2009.09.004

multi-objective optimization are introduced in Section 3. A general presentation of these methods and their application to the B-RSP is followed by a comparative study. In Section 4, we propose new cooperation schemes to solve multi-objective combinatorial optimization problems. We experiment the resulting search methods on the B-RSP and we compare the obtained computational results to the previous ones. At last, conclusions and perspectives are drawn in the last section.

2. The Bi-objective Ring Star Problem

In this section, we first present some basic concepts, notation and definitions related to multi-objective optimization. Next, we provide a formulation of the ring star problem as a bi-objective problem. Finally, we briefly survey the literature related to the problem at hand and discuss its industrial concerns.

2.1. Multi-objective optimization

A *Multi-objective Optimization Problem* (MOP) aims of optimizing a set of $n \geq 2$ objective functions $f_1, f_2, \ldots, f_n$ simultaneously. Each objective function can be either minimized or maximized. Let X denote the set of feasible solutions in the *decision space*, and Z the set of feasible points in the *objective space*. Without loss of generality, we here assume that $Z \subseteq \Re^n$ and that all n objective functions are to be minimized. To each decision vector $x \in X$ is assigned exactly one objective vector $z \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $z = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$. We will assume, throughout the paper, that objective values are normalized.[1] Therefore, a MOP can be formulated as follows:

$$(MOP) = \begin{cases} \text{‘ min’} & f(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ \text{s.t.} & x \in X \end{cases} \tag{1}$$

Definition 1. An objective vector $z \in Z$ *weakly dominates* another objective vector $z' \in Z$ if and only if $\forall i \in \{1, 2, \ldots, n\}$, $z_i \leq z'_i$.

Definition 2. An objective vector $z \in Z$ *dominates*[2] another objective vector $z' \in Z$ if and only if $\forall i \in \{1, 2, \ldots, n\}$, $z_i \leq z'_i$ and $\exists j \in \{1, 2, \ldots, n\}$ such as $z_j < z'_j$.

Definition 3. An objective vector $z \in Z$ is *non-dominated* if and only if there does not exist another objective vector $z' \in Z$ such that z' dominates z.

A solution $x \in X$ is said to be *efficient* (or *Pareto optimal*, *non-dominated*) if its mapping in the objective space results in a non-dominated point. The set of all efficient solutions is the *efficient* (or *Pareto optimal*) *set*, denoted by X_E. The set of all non-dominated vectors is the *non-dominated front* (or the *trade-off surface*), denoted by Z_N. A possible approach in MOP solving is to find the minimal set of efficient solutions, *i.e.* one solution $x \in X_E$ for each non-dominated vector $z \in Z_N$ such as $f(x) = z$ (in case multiple solutions map to the same non-dominated point). But, generating the entire set of Pareto optimal solutions is usually infeasible due to the complexity of the underlying problem or to the large number of optima. Therefore, the overall goal is often to identify a good approximation of it. Population-based metaheuristics are commonly used to this end as they naturally find multiple and well-spread non-dominated solutions in a single simulation run. The reader could refer to [8,9] for more details about evolutionary multi-objective optimization and to [10] for more details about multi-objective combinatorial optimization.

2.2. Problem definition

The *Ring Star Problem* (RSP) can be described as follows. Let $G = (V, E, A)$ be a complete mixed graph where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, $E = \{[v_i, v_j] | v_i, v_j \in V, i < j\}$ is a set of edges, and $A = \{(v_i, v_j) | v_i, v_j \in V\}$ is a set of arcs. Vertex v_1 is the depot. To each edge $[v_i, v_j] \in E$ we assign a non-negative *ring cost* c_{ij}, and to each arc $(v_i, v_j) \in A$ we assign a non-negative *assignment cost* d_{ij}. The B-RSP consists of locating a simple cycle through a subset of nodes $V' \subset V$ (with $v_1 \in V'$) while (i) minimizing the sum of the ring costs related to all edges that belong to the cycle, and (ii) minimizing the sum of the assignment costs of arcs directed from every non-visited node to a visited one so that the associated cost is minimum. An example of solution is given in Fig. 1, where solid lines represent edges that belong to the ring and dashed lines represent arcs of the assignments.

The first objective is called the *ring cost* and is defined as

$$\sum_{[v_i, v_j] \in E} c_{ij} b_{ij} \tag{2}$$

where b_{ij} is a binary variable equal to 1 if and only if the edge $[v_i, v_j]$ belongs to the cycle. The second objective, the *assignment cost*, can be computed as follows:

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \tag{3}$$

Let us remark that these objective functions are comparable only if we assume that the ring cost and the assignment cost are proportional one to another. But this is rarely the case in practice. Moreover, the fact of privileging a cost compared to the other is closely related to the decision-maker preferences, whereas we deal here with *a posteriori* optimization. However, the B-RSP is a *NP*-hard combinatorial problem because the particular case of visiting the whole set of nodes is equivalent to a traditional traveling salesman problem.

2.3. Related works

The RSP belongs to the class of location–allocation problems aiming at locating structures in a graph (see [11] for a review). It has initially been formulated by Labbé et al. [12] in two different ways. In the first formulation (denoted MCP1 in [12] and more often called ‘ring star problem’), a weighted sum of both objective functions is to be optimized. In the second formulation (MCP2 [12], usually called ‘median cycle problem’), the ring cost is to be minimized while the assignment cost is bounded by a prefixed value. Even if it was not explicitly noticed by the original authors, these two formulations are commonly employed to convert a MOP

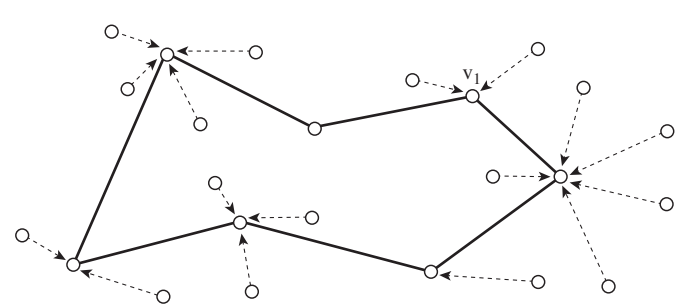


Fig. 1. An example of solution for the ring star problem.

[1] To achieve this, the minimum and the maximum value of each objective function are used in order to adaptively replace each objective function by its corresponding normalized function, so that its values lie in the interval [0, 1].

[2] We will also say that a decision vector $x \in X$ *dominates* a decision vector $x' \in X$ if $f(x)$ dominates $f(x')$.

into a single-objective problem by using scalar approaches. They are, respectively, denoted by *aggregation* and *epsilon-constraint* methods in the multi-objective optimization literature [13]. The first formulation of the problem has been more widely studied in [1]. The authors used a branch-and-bound method and successfully solved TSPLIB and randomly generated instances involving up to 200 nodes in less than two hours. In [2], the same authors solved the second formulation of the problem by a similar method. Finally, one or both versions of the problem have been heuristically tackled by a variable neighborhood tabu search [14], an evolutionary algorithm [15], a multi-start greedy add heuristic [15], and a variable neighborhood tabu search hybridized with a greedy randomized adaptive search procedure [16].

As shown in the survey of Jozefowiez et al. [3], an increasing number of multi-objective routing problems appeared in the literature in recent years. However, in spite of its numerous industrial applications (see Section 2.4), the RSP has never been explicitly investigated in a multi-objective fashion. Nevertheless, Current and Schilling [17] defined two multi-objective variants of a very similar problem: the *median tour problem* and the *maximal covering tour problem*. In both versions, one objective is the minimization of the total length of the tour, while another one is the maximization of the access to the tour for non-visited nodes. To tackle these problems, the authors used a kind of lexicographic method, where a hierarchy is defined between objective functions. Additionally, Doerner et al. [18] recently formulated a three-objective optimization problem of tour planning for mobile health care facilities, closely related to the *Single Vehicle Routing-Allocation Problem* introduced in [19]. A mobile facility has to visit a subset of nodes. Non-visited nodes are then assigned to their closest tour stop or are regarded as unable to reach a tour stop (within a predefined maximum distance). The considered objectives are (i) the minimization of the ratio between medical working time and total working time, (ii) the minimization of the average distance to the nearest tour stops and (iii) the maximization of a coverage criterion. To do so, a Pareto ant colony optimization algorithm as well as two genetic algorithms (namely VEGA [20] and MOGA [21]) were designed to solve real-world instances.

2.4. Real-world applications and industrial concerns

The RSP has a wide range of industrial interests, especially in telecommunications and vehicle routing. Of course, practical real-world applications may contain additional constraints. For instance, as noticed by Labbé et al. [1,2], a ring-based network is designed to interconnect a set of hubs in the case of digital data services [22]. Some concentrators are installed on a subset of locations and are interconnected on a ring network (the Internet) while the remaining locations are assigned to this concentrators (the Intranet). Closely related problems arise in rapid transit systems planning [1] or while designing optical networks [23]. In addition, other kinds of applications appear in the postal collection or delivery routes design, where the distance between a customer and a collection point has to be reasonable. For instance, post-box location while taking both the collection cost and the user inconvenience into account has been studied in [24]. Besides, other applications closely related to the RSP are the location of circular shaped transportation infrastructure (such as metro lines or motorways), the location of recyclable garbage collection bins, and school bus routing. Finally, the routing of essential health care services, already investigated in [18,25] among other authors, consists of a mobile clinic servicing an area without being able to visit every population nodes. Then, unvisited ones have to reach the nearest tour stop by their own to be medically treated.

3. Metaheuristics for the Bi-objective Ring Star Problem

In this section, four population-based metaheuristics are proposed to solve the bi-objective RSP. First is an iterative local search method called IBMOLS [5]. Next, three evolutionary algorithms are designed. They consist of slight variations of IBEA [6] and NSGA-II [7] and of a general-purpose search method called SEEA and introduced in this paper for the first time. IBMOLS and IBEA are both recent indicator-based metaheuristics, whereas NSGA-II can be considered as a state-of-the-art multi-objective resolution approach. A presentation of these metaheuristics is given below, and is followed by a detailed description of problem-specific components.

3.1. A multi-objective local search

Since they are easily adaptable to the multi-objective context, many search methods proposed to tackle MOPs are evolutionary algorithms. However, local search algorithms are known to be effective metaheuristics for solving real-world applications [26,27]. Several multi-objective neighborhood search methods have been proposed in the literature, see [10,28] for a survey. Most of them are based on a set of aggregations of the objective functions. Dominance-based local search algorithms are more rare. However, an indicator-based multi-objective local search (IBMOLS for short) has recently been proposed in [5]. IBMOLS can be seen as a local search variant of IBEA [6], as the same strategy is used for fitness assignment. IBMOLS is a generic population-based multi-objective local search dealing with a fixed population size. This allows to obtain a set of efficient solutions in a single simulation run without specifying any mechanism to control the number of solutions during the search process. Moreover, IBMOLS presents an alternative to aggregation- and dominance-based multi-objective local search algorithms. Indeed, as proposed in [6], it is assumed that the optimization goal is given in terms of a binary quality indicator I [29] that can be regarded as an extension of the Pareto dominance relation. A value $I(A,B)$ quantifies the difference in quality between two approximated efficient sets A and B. So, if R denotes a reference set (that can be the Pareto-optimal set X_E or any other set), the overall optimization goal can be formulated as

$$\arg\min_{A \in \Omega} I(A,R), \tag{4}$$

where Ω denotes the space of all efficient set approximations. As noted in [6], R does not have to be known in advance, it is just required in the formalization of the optimization goal. Since R is fixed, I actually represents a unary function that assigns a real number reflecting the quality of any approximation set according to the optimization goal. If I is dominance preserving [6], $I(A,R)$ is minimum for $A=R$. One of the main advantages of indicator-based optimization is that no additional diversity preservation mechanism is generally required, according to the indicator being used. Indeed, diversity information is usually included into the indicator, since it may quantify the quality of a set both in terms of convergence and diversity.

Principle: The IBMOLS algorithm maintains a population P of size N. Then, it generates the neighborhood of a solution contained in P until an improving solution is found (i.e. one that is better than at least one solution of P in terms of the indicator being used), or until all its neighbors have been explored. If an improving solution is found, it replaces the worst solution of P.

This corresponds to a '*first improving*' strategy. By iterating this simple principle to every solution of P, we obtain a local search step. The whole local search procedure stops when the archive of potentially efficient solutions has not received any new item during a complete local search step. Hence, the IBMOLS method provides a natural stopping criterion.

Quality indicator: Several quality indicators can be used in the frame of IBMOLS. Some examples can be found in [29]. One of them is the binary additive ε- indicator ($I_{\varepsilon+}$) [6,29], inspired by the concept of ε- dominance [30]. This indicator appears to be particularly well-suited to indicator-based search. And its efficiency has been experimentally proven to approximate the efficient set of different kinds of problems, see for instance [5,6]. $I_{\varepsilon+}$ computes the minimum value by which a solution $x \in X$ has to or can be translated in the objective space to weakly dominate another solution $x' \in X$. For a minimization problem, it is defined as follows:

$$I_{\varepsilon+}(x, x') = \max_{i \in \{1,\ldots,n\}} (f_i(x) - f_i(x')), \tag{5}$$

where n stands for the number of objective functions. Furthermore, to evaluate the quality of a solution $x \in X$ according to a whole population P, and then to compute the fitness value of x, different approaches exist. As proposed in [6], we will here consider an additive technique that amplifies the influence of dominating solutions over dominated ones. It can be outlined as follows:

$$fitness(x) = \sum_{x^* \in P \setminus \{x\}} - e^{-I(x^*, x)/\kappa}, \tag{6}$$

where $\kappa > 0$ is a fitness scaling factor, see [6] for more details. However, the initial experiments were not satisfactory because the algorithm was not able to find the extreme points of the trade-off surface. As pointed out in [31], this is known to be one of the drawbacks of the ε- dominance relation, apparently due to the high convexity of the front. Indeed, the authors illustrate that a limitation of ε- dominance is that extreme points of the Pareto front are usually lost. To tackle this problem, we add a simple condition preventing the deletion of solutions corresponding to the extreme non-dominated vectors during the replacement step of IBMOLS.

Iterative version: Firstly, let us remind that IBMOLS handles a natural stopping condition. However, it is commonly known that iteratively repeating a local search algorithm can lead to very high performances and can significantly improve the results. In the frame of single-objective optimization, two main iterative strategies are generally distinguished, that is *random restart* and *perturbation techniques* [32]. Random restart is the simplest possibility, and a well-designed perturbation technique generally achieves better results. Furthermore, within IBMOLS, we now have to deal with a population of solutions that needs to be re-initialized between each local search phase. As proposed in [5], we decide to perturb a set of solutions from the archive to obtain the restarting population. The choice of such a strategy has been motivated by preliminary experiments as well as conclusive results obtained in [5] for another problem. Then, the population re-initialization scheme is based on random noise, such as in a basic simulated annealing [26]. This noise consists of multiple mutations applied to N different randomly chosen solutions contained in the elite set. If the size of the archive is less than N, the population is filled with random solutions, just like in random restart.

More details about IBMOLS and I-IBMOLS as well as algorithm pseudo-codes can be found in [5].

3.2. Multi-objective evolutionary algorithms

The first two multi-objective evolutionary algorithms designed to tackle the bi-objective RSP are small variations of two state-of-the-art search methods, namely IBEA [6] and NSGA-II [7]. Some minor modifications have been carried out in order to save the whole set of non-dominated solutions found during the search process. Finally, a general-purpose approach for MOP solving, called SEEA, is proposed in this paper for the first time and is presented in details.

3.2.1. IBEA

Introduced by Zitzler and Künzli [6], the *Indicator-Based Evolutionary Algorithm* (IBEA) is, like IBMOLS, an indicator-based metaheuristic. The fitness assignment scheme of this evolutionary algorithm is based on a pairwise comparison of solutions contained in a population by using a binary quality indicator. The selection scheme for reproduction is a binary tournament between randomly chosen individuals. The replacement is based on an iterative elitist strategy that consists in deleting, one-by-one, the worst individuals, and in updating the fitness values of the remaining solutions each time there is a deletion; this is continued until the required population size is reached. Moreover, an archive stores solutions mapping to potentially non-dominated points in order to prevent their loss during the stochastic search process. However, in our case, and in contrast to the IBEA defined in [6], this archive is updated at each generation since the beginning of the evolutionary algorithm, so that the output size is not necessarily less than or equal to the population size. Just like for the IBMOLS algorithm, the indicator used within IBEA is the additive ε- indicator; and the same mechanism has been used to prevent the loss of the extreme points on the trade-off surface. More information about this algorithm can be found in [6].

3.2.2. NSGA-II

At each generation of NSGA-II (the *Non-dominated Sorting Genetic Algorithm II* introduced by Deb et al. [7]), the solutions contained in the population are ranked into several classes. Individuals mapping to vectors from the first front all belong to the best efficient set; individuals mapping to vectors from the second front all belong to the second best efficient set; and so on. Two values are computed for every solution of the population. The first one corresponds to the *rank* the corresponding solution belongs to, and represents the quality of the solution in terms of convergence. The second one, the *crowding distance*, consists in estimating the density of solutions surrounding a particular point of the objective space, and represents the quality of the solution in terms of diversity. A solution is said to be better than another solution if it has a best rank value, or in case of equality, if it has the best crowding distance. The selection strategy is a deterministic tournament between two random solutions. At the replacement step, only the best individuals survive, with respect to the population size. Similarly, an external population is added to NSGA-II in order to store every potentially efficient solution found during the search. The reader is referred to [7] for more details about NSGA-II.

3.2.3. SEEA

If evaluating a solution in the objective space is not too much time consuming (which is the case for our problem), computing fitness and diversity information is generally the most computationally expensive step of a multi-objective evolutionary algorithm. Based on this observation, we here propose a simple search method for which none of these phases is required. The resulting evolutionary algorithm, called *Simple Elitist Evolutionary Algorithm*

(SEEA for short), is detailed in Algorithm 1. An archive of potentially efficient solutions is updated at each generation, and the individuals contained in the main population are generated by applying variation operators to randomly chosen archive members. The replacement step is a generational one, i.e. the parent population is replaced by the offspring one. Note that the initial population can, for instance, be filled with random solutions. Thus, as proposed in [33] among other authors, the archive is not only used as an external storage, but it is integrated into the optimization process during the selection phase of the evolutionary algorithm; that is called *elitism*. Elitism is an important issue in the field of evolutionary multi-objective optimization [34] and SEEA is in somehow related to other elitist evolutionary algorithms such as SPEA [33], PESA [35] or SEAMO [36]. But, contrary to other approaches, no strategy to preserve diversity or to manage the size of the archive is involved here, as solutions are selected randomly and the archive is unbounded. Note that to be used for solving optimization problems where an exponential number of efficient solutions are involved, an additional mechanism should be designed to bound the archive size, see [37]. The biggest advantage of this evolutionary algorithm is that the population (or the population size if solutions are randomly initialized) is the only problem-independent parameter. If non-dominated solutions are relatively close to each other in the decision space and if the archive is not too small compared to the main population, we believe that SEEA may convergence to a good approximation of the efficient set in a very short runtime.

Algorithm 1. Simple elitist evolutionary algorithm (SEEA).

Input: P Initial population
Output: A Efficient set approximation
Step 1: **Initialization**. $A \leftarrow$ non-dominated solutions of P; $N \leftarrow |P|$; $P' \leftarrow \emptyset$.
Step 2: **Selection**. Repeat until $|P'| = N$: randomly select an individual from A and add it to the offspring population P'.
Step 3: **Variation**. Apply crossover and mutation operators to individuals of the offspring population P'.
Step 4 : **Replacement**. $P \leftarrow P'$; $P' \leftarrow \emptyset$.
Step 5: **Elitism**. $A \leftarrow$ non-dominated solutions of $A \cup P$.
Step 6: **Termination**. If a stopping criteria is satisfied return A, else go to Step 2.

3.3. Application to the Bi-objective Ring Star Problem

This section presents the problem-specific components that are necessary to instantiate the metaheuristics introduced above for the resolution of our bi-objective RSP. Hence, the encoding mechanism, the population initialization as well as the neighborhood, the mutation and the recombination operators, are described in details below.

3.3.1. Solution encoding

The representation of a RSP solution is based on the random keys mechanism proposed by Bean [38]. Such an encoding mechanism has already been successfully applied for solving a single-objective version of the RSP in [15]. A *random key* $k_i \in [0, 1[$ is assigned to every node v_i that belongs to the ring, with $k_1 = 0$. A special value is assigned to unvisited nodes. Thus, the ring route associated to a solution corresponds to the nodes read according to their random keys in the increasing order; i.e. if $k_i < k_j$, then v_j comes after v_i. A possible representation for the cycle $(v_1, v_7, v_4, v_9, v_2, v_6)$ is given in Fig. 2. Nodes v_3, v_5, v_8 and v_{10} are assigned to a visited node in such a way that the associated assignment cost is minimum.

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Random key	0	0.7	-	0.3	-	0.8	0.2	-	0.5	-

Fig. 2. A RSP solution represented by random keys.

3.3.2. Population initialization

For each search method, the initial population has been generated randomly. Each node has a probability $p = 0.5$ to be visited or not, and to each visited vertex v_i we associate a random key k_i uniformly generated in $[0, 1[$.

3.3.3. Neighborhood and mutation operators

As the RSP is both a routing and an assignment problem, different move and mutation operators have to be designed. For this kind of problems, usual operators consists in removing or adding a node to the cycle. Here we also consider an operator specifically dedicated to the ring improvement by means of a 2-opt move. The resulting three operators are the following:

- *remove operator*: selects a visited node $v_i \in V'$ at random and removes it from the ring;
- *insert operator*: selects an unvisited node $v_i \in V \backslash V'$ at random and adds it to the cycle, the position to insert v_i is chosen so that the increment on the ring cost is minimum;
- 2-*opt exchange operator*: selects two visited nodes $v_i \in V'$ and $v_j \in V'$ at random and applies a 2-opt move between v_i and v_j, so that the sequence of visited nodes located between v_i and v_j is reversed.

Note that it is not necessary to re-evaluate a solution each time a neighborhood or a mutation operator is applied. Thus, after a remove operator, we just have to re-assign the unvisited nodes previously assigned to the one that has been removed. After an insert operator, we just have to re-assign unvisited nodes in order to minimize the assignment cost. Finally, after a 2-opt exchange operator, we just have to recompute the ring cost, the assignment cost being unchanged. For the local search method, the neighbors of a given solution are randomly explored, without considering any order between these three operators. And each neighbor is at most visited once during a search step.

3.3.4. Recombination operator

The recombination operator is a one-point crossover closely related to the one proposed in [15]. Two randomly selected solutions x_1 and x_2 are first divided according to a random position. Then, the first part of x_1 is combined with the second part of x_2 to build a first offspring, and the first part of x_2 is combined with the second part of x_1 to build a second offspring. Every node retains its random key so that it enables an easy reconstruction of the new individuals. Due to the random keys encoding mechanism, solutions having different ring sizes can easily be recombined, even if the initial ring structures are generally broken in the offspring solutions. Fig. 3 illustrates a recombination between two solutions $(v_1, v_7, v_4, v_9, v_2, v_6)$ and $(v_1, v_8, v_4, v_3, v_5)$ after vertex v_6, which gives rise to a couple of new solutions $(v_1, v_8, v_4, v_2, v_6)$ and $(v_1, v_7, v_9, v_4, v_3, v_5)$.

Another recombination operator preserving a bigger part of the initial ring structures, has been experimented. But it was tending to reduce the number of nodes belonging to the cycle and then

Parent 1	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	0.7	-	0.3	-	0.8	0.2	-	0.5	-
Parent 2	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	-	0.8	0.7	0.9	-	-	0.2	-	-
Offspring 1	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	0.7	-	0.3	-	0.8	-	0.2	-	-
Offspring 2	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	-	0.8	0.7	0.9	-	0.2	-	0.5	-

Fig. 3. Recombination operator.

was causing a premature convergence with solutions having a small number of visited nodes.

3.4. Experiments

All the metaheuristics presented in this paper have been implemented using the ParadisEO-MOEO software framework[3] [39]. ParadisEO-MOEO is a C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics for multi-objective optimization. All the algorithms share the same base components for a fair comparison between them. Computational runs were performed on an Intel Core 2 Duo 6600 (2×2.40 GHz) machine, with 2 GB RAM.

3.4.1. Experimental protocol

Benchmark test instances: Experiments were conducted on a set of eight benchmark instances taken from the TSPLIB[4] [40]. These instances contain between 51 and 299 nodes. The number at the end of an instance name represents the number of nodes involved for the instance under consideration. Let l_{ij} denote the distance between two nodes v_i and v_j of a TSPLIB file. The ring cost c_{ij} and the assignment cost d_{ij} have both been set to l_{ij} for every pair of nodes v_i and v_j.

Stopping conditions: From our point of view, there does not exist a standard approach to define an unquestionable condition for stopping a multi-objective metaheuristic. Common strategies are quite basic and generally consist of an arbitrary user-given number of iterations or evaluations. However, in the frame of this paper, an evolutionary algorithm iteration has nothing to do with a local search iteration. Moreover, computing the number of evaluations performed from the beginning of the search process does not make any sense here, as mutated or neighbor solutions are not fully, but incrementally evaluated. Therefore, in this study, the search process simply stops after a fixed amount of runtime. Indeed, all methods share the same base components and their respective implementations can then be compared in term of computational time. But, as small instances are supposedly easier to solve than large ones, this stopping criteria has been arbitrary set according to the size of the instance under consideration, as shown in Table 1. Note that the maximum runtime available stands for a single simulation run *per* instance and *per* algorithm.

Performance assessment: In the frame of multi-objective optimization, the performance assessment of a number of algorithms in solving the same problem is a key issue. In this study, a set of 20 runs *per* instance, with different initial populations, has been performed for each search method. In order to evaluate the quality of the non-dominated front approximations for every instance we experimented, we follow the protocol given by Knowles et al. [41]. For a given instance, let Z^{all} denote the union of the outputs we obtained during all our experiments. Note that this set probably contains both dominated and non-dominated points, as a given approximation may contain vectors dominating the ones of another approximation, and vice versa. We first compute a reference set $Z_N^\star$ containing all the non-dominated points of Z^{all}. Second, we define $z^{max} = (z_1^{max}, z_2^{max})$, where z_1^{max} (respectively, z_2^{max}) denotes the upper bound of the first (respectively, second) objective for all the points contained in Z^{all}.

Now, to measure the quality of an output set A in comparison to $Z_N^\star$, we compute the difference between these two sets by using the unary hypervolume metric [33], z^{max} being the reference point. As illustrated in Fig. 4, the hypervolume difference indicator ($\mathrm{I}_{\bar{H}}$) computes the portion of the objective space that is weakly dominated by $Z_N^\star$ and not by A, cf. Definition 1; the closer this measure to 0, the better the approximation A. Furthermore, we also consider the additive ε- indicator proposed in [29]. Contrary to the one proposed in Eq. (5), this indicator is used to compare non-dominated set approximations, and not solutions. The unary additive ε- indicator ($\mathrm{I}^1_{\varepsilon+}$) gives the minimum factor by which an approximation A has to be translated in the objective space to weakly dominate the reference set $Z_N^\star$. $\mathrm{I}^1_{\varepsilon+}$ can then be defined as follows:

$$I^1_{\varepsilon+}(A) = \min_{\varepsilon}\{\forall z \in Z_N^\star, \exists z' \in A : z'_i - \varepsilon \le z_i, \forall 1 \le i \le n\}. \quad (7)$$

As a consequence, for each test instance, we obtain 20 $\mathrm{I}_{\bar{H}}$ measures and 20 $\mathrm{I}_{\varepsilon+}$ measures, corresponding to the 20 runs, *per* algorithm.

Table 1
Stopping criteria: runtime *per* simulation run.

Instance	Runtime
eil 51	20 s
st 70	1 min
kroA 100	2 ft
bier 127	5 ft
kroA 150	10 ft
kroA 200	20 ft
pr 264	30 ft
pr 299	50 ft

[3] ParadisEO is available at http://paradiseo.gforge.inria.fr.
[4] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

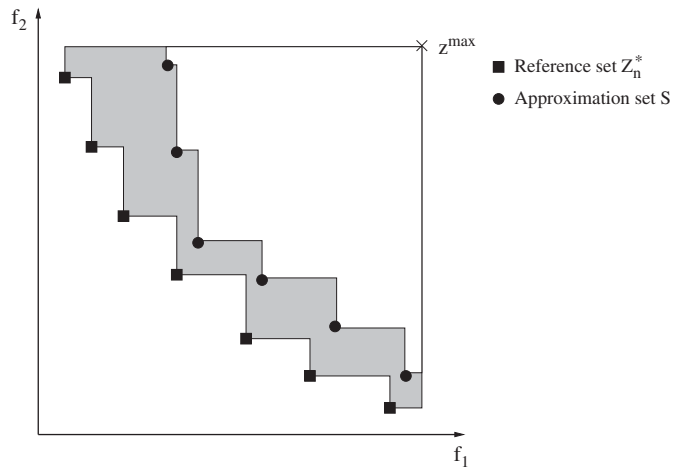


Fig. 4. Illustration of the hypervolume difference (I_H^-) between a reference set Z_N^* and a non-dominated front approximation S (shaded area).

As suggested by Knowles et al. [41], once all these values are computed, we perform a statistical analysis for a pairwise comparison of methods. To this end, we use the Wilcoxon signed rank test. Such a non-parametric statistical test is motivated by the fact that the samples collected here can be considered as matched samples. Indeed, for a given run, both the initial population and the random seed are identical for all algorithms, so that the final indicator values can be taken as pairs. Details for this statistical testing procedure are given in [41]. Hence, for a given test instance, and according to the p-value and to the metric under consideration, this statistical test reveals if the sample of approximation sets obtained by a given search method is significantly better than the ones of another search method, or if there is no significant difference between both. Note that all the performance assessment procedures have been achieved using the performance assessment tool suite provided in PISA[5] [42].

3.4.2. Parameter setting

A preliminary experimental phase has been performed to determine the following parameters, see [43] for more details. Hence, as no general trend has been identified for some algorithms, the population size were fixed according to the instance under consideration, as shown in Table 2. The noise rate for I-IBMOLS has been set to a fixed percentage of the number of nodes involved in the problem at hand. Next, following [6], the scaling factor κ of Eq. (6) has been set to 0.05 for indicator-based metaheuristics, i.e. IBEA and I-IBMOLS. The remainder parameters are shared by all evolutionary algorithms and consist of a crossover probability of 0.25, and of a mutation probability of 1.00, with rates of 0.25, 0.25 and 0.50 for the *remove*, the *insert* and the 2-*opt exchange* operator, respectively.

3.4.3. Computational results

Tables 3 and 4 provide a comparison of SEEA, NSGA-II, IBEA and I-IBMOLS with respect to the I_H^- metric and to the $I_{\varepsilon+}^1$ metric, respectively.

According to the experimental protocol used in the paper, I-IBMOLS is never statistically outperformed by any other algorithm with respect to both metrics. The only exceptions are for the *st* 70 and the *pr* 264 instances, where SEEA obtained better results with respect to the $I_{\varepsilon+}^1$ metric. About NSGA-II, it is outperformed by all the other algorithms on every test instances, except for *pr* 264 where there is no significant difference between its results and the ones of IBEA according to the $I_{\varepsilon+}^1$ metric. At last, with respect to the I_H^- metric, the results of IBEA and SEEA are quite heterogeneous, so that no general trend can be identified. However, according to the $I_{\varepsilon+}^1$ metric, it seems that SEEA is more efficient that IBEA on most test instances.

[5] The package is available at http://www.tik.ee.ethz.ch/pisa/assessment.html.

Table 2
Parameter setting.

Instance	I-IBMOLS		IBEA	NSGA-II	SEEA
	Pop. size	Noise rate (%)	Pop. size	Pop. size	Pop. size
eil 51	20	10	100	100	100
st 70	20	10	50	100	100
kroA 100	30	10	100	200	100
bier 127	30	10	100	200	100
kroA 150	30	10	200	200	100
kroA 200	30	10	200	200	100
pr 264	30	20	50	200	100
pr 299	50	10	50	200	100

Table 3
Algorithms comparison with respect to the I_H^- metric.

			I-IBMOLS	IBEA	NSGA-II	SEEA
eil 51	I-IBMOLS	(4.456)	–	$\succ$	$\succ$	$\equiv$
	IBEA	(6.710)	$\prec$	–	$\succ$	$\prec$
	NSGA-II	(12.573)	$\prec$	$\prec$	–	$\prec$
	SEEA	(4.957)	$\equiv$	$\succ$	$\succ$	–
st 70	I-IBMOLS	(3.143)	–	$\succ$	$\succ$	$\succ$
	IBEA	(4.037)	$\prec$	–	$\succ$	$\equiv$
	NSGA-II	(8.920)	$\prec$	$\prec$	–	$\prec$
	SEEA	(3.718)	$\prec$	$\equiv$	$\succ$	–
kroA 100	I-IBMOLS	(4.251)	–	$\succcurlyeq$	$\succ$	$\succ$
	IBEA	(5.273)	$\preceq$	–	$\succ$	$\equiv$
	NSGA-II	(12.370)	$\prec$	$\prec$	–	$\prec$
	SEEA	(5.015)	$\prec$	$\equiv$	$\succ$	–
bier 127	I-IBMOLS	(3.219)	–	$\succ$	$\succ$	$\succ$
	IBEA	(4.236)	$\prec$	–	$\succ$	$\succ$
	NSGA-II	(9.606)	$\prec$	$\prec$	–	$\prec$
	SEEA	(6.751)	$\prec$	$\prec$	$\succ$	–
kroA 150	I-IBMOLS	(3.959)	–	$\succ$	$\succ$	$\succ$
	IBEA	(4.562)	$\prec$	–	$\succ$	$\equiv$
	NSGA-II	(8.973)	$\prec$	$\prec$	–	$\prec$
	SEEA	(4.747)	$\prec$	$\equiv$	$\succ$	–
kroA 200	I-IBMOLS	(2.875)	–	$\equiv$	$\succ$	$\succ$
	IBEA	(2.980)	$\equiv$	–	$\succ$	$\succ$
	NSGA-II	(8.515)	$\prec$	$\prec$	–	$\prec$
	SEEA	(3.822)	$\prec$	$\prec$	$\succ$	–
pr 264	I-IBMOLS	(1.535)	–	$\succ$	$\succ$	$\equiv$
	IBEA	(1.912)	$\prec$	–	$\succ$	$\prec$
	NSGA-II	(3.663)	$\prec$	$\prec$	–	$\prec$
	SEEA	(1.520)	$\equiv$	$\succ$	$\succ$	–
pr 299	I-IBMOLS	(1.303)	–	$\succ$	$\succ$	$\succ$
	IBEA	(1.964)	$\prec$	–	$\succ$	$\equiv$
	NSGA-II	(4.185)	$\prec$	$\prec$	–	$\prec$
	SEEA	(2.179)	$\prec$	$\equiv$	$\succ$	–

The number in brackets denotes the average metric value for the algorithm and the instance under consideration (multiplied by 10^{-3}). For each instance, either the algorithm located at a specific row significantly dominates the algorithm located at a specific column ($\succ$ or $\succcurlyeq$ for a p-value less or equal to 0.01 or to 0.05, respectively), either it is significantly dominated ($\prec$ or $\preceq$ for a p-value less or equal to 0.01 or to 0.05, respectively), or there is no significant difference between both ($\equiv$).

Table 4
Algorithms comparison with respect to the $I^1_{\varepsilon+}$ metric.

			I-IBMOLS	IBEA	NSGA-II	SEEA
eil 51	I-IBMOLS	(9.307)	–	$\succ$	$\succ$	$\equiv$
	IBEA	(12.094)	$\prec$	–	$\succ$	$\prec$
	NSGA-II	(19.165)	$\prec$	$\prec$	–	$\prec$
	SEEA	(9.613)	$\equiv$	$\succ$	$\succ$	–
st 70	I-IBMOLS	(7.321)	–	$\succ$	$\succ$	$\preceq$
	IBEA	(10.334)	$\prec$	–	$\succ$	$\prec$
	NSGA-II	(13.639)	$\prec$	$\prec$	–	$\prec$
	SEEA	(6.298)	$\succcurlyeq$	$\succ$	$\succ$	–
kroA 100	I-IBMOLS	(9.833)	–	$\succcurlyeq$	$\succ$	$\equiv$
	IBEA	(11.771)	$\preceq$	–	$\succ$	$\preceq$
	NSGA-II	(17.718)	$\prec$	$\prec$	–	$\prec$
	SEEA	(9.606)	$\equiv$	$\succcurlyeq$	$\succ$	–
bier 127	I-IBMOLS	(8.421)	–	$\succ$	$\succ$	$\succ$
	IBEA	(11.993)	$\prec$	–	$\succ$	$\succ$
	NSGA-II	(21.522)	$\prec$	$\prec$	–	$\preceq$
	SEEA	(19.377)	$\prec$	$\prec$	$\succcurlyeq$	–
kroA 150	I-IBMOLS	(7.853)	–	$\succ$	$\succ$	$\succ$
	IBEA	(10.708)	$\prec$	–	$\succ$	$\prec$
	NSGA-II	(13.383)	$\prec$	$\prec$	–	$\prec$
	SEEA	(9.056)	$\prec$	$\succ$	$\succ$	–
kroA 200	I-IBMOLS	(7.829)	–	$\equiv$	$\succ$	$\equiv$
	IBEA	(7.288)	$\equiv$	–	$\succ$	$\succ$
	NSGA-II	(14.473)	$\prec$	$\prec$	–	$\prec$
	SEEA	(8.204)	$\equiv$	$\prec$	$\succ$	–
pr 264	I-IBMOLS	(5.259)	–	$\succ$	$\succ$	$\prec$
	IBEA	(9.055)	$\prec$	–	$\equiv$	$\prec$
	NSGA-II	(8.403)	$\prec$	$\equiv$	–	$\prec$
	SEEA	(4.343)	$\succ$	$\succ$	$\succ$	–
pr 299	I-IBMOLS	(4.023)	–	$\succ$	$\succ$	$\succ$
	IBEA	(8.993)	$\prec$	–	$\succcurlyeq$	$\prec$
	NSGA-II	(10.403)	$\prec$	$\preceq$	–	$\prec$
	SEEA	(5.768)	$\prec$	$\succ$	$\succ$	–

The number in brackets denotes the average metric value for the algorithm and the instance under consideration (multiplied by 10^{-3}). For each instance, either the algorithm located at a specific row significantly dominates the algorithm located at a specific column ($\succ$ or $\succcurlyeq$ for a p-value less or equal to 0.01 or to 0.05, respectively), either it is significantly dominated ($\prec$ or $\preceq$ for a p-value less or equal to 0.01 or to 0.05, respectively), or there is no significant difference between both ($\equiv$).

3.5. Discussion

We can conclude that the iterative version of IBMOLS is globally significantly better than all the evolutionary algorithms we investigated. Nevertheless, the behavior of such a simple search method as SEEA in comparison to state-of-art algorithms like IBEA and NSGA-II is very encouraging with regard to combinatorial problem solving. One of the main characteristics of the problem under consideration seems to be the high number of points located in the trade-off surface. Then, after a couple of iterations, a large part of the population involved in I-IBMOLS, IBEA and NSGA-II might map to non-dominated points. That could explain the low efficiency of NSGA-II. Indeed, since the same fitness value is assigned to the major part of the population, only the crowding distance is used to compare solutions. The indicator-based fitness assignment scheme of I-IBMOLS and IBEA is obviously much more suited to discriminate potentially efficient solutions than the single crowding distance. Note that this is not the case within SEEA because all non-dominated solutions contained in the archive can potentially take part in the evolution engine. However, the good performance of I-IBMOLS might also depend on how close are solutions mapping to non-dominated points in the decision space. If these solutions are close to each other according to the neighborhood operators, a local search method is known to be particularly well-suited to find additional interesting solutions. As a next step, it could then be interesting to design a cooperation scheme between two different search methods in order to benefit of their individual features.

4. Cooperative approaches for the Bi-objective Ring Star Problem

This section presents a general-purpose cooperative approach combining SEEA and the non-iterative version of IBMOLS for multi-objective combinatorial optimization. Two variants are proposed: a periodic one that operates a systematic cooperation and an adaptive one that decides on-line when the cooperation must occur. These hybrid models are next experimented on the bi-objective RSP investigated in this paper.

4.1. Motivations

Designing metaheuristics for solving combinatorial optimization problems is generally a matter of intensification and diversification. This is even more pronounced for MOPs where the goal is to find a well-converged and well-diversified efficient set approximation. However, local search methods are known to be particularly efficient as intensifying methods, whereas evolutionary algorithms are clearly powerful to explore the decision space thanks to their variation operators. Instead of trying to improve one method in term of diversification or the other in term of intensification, a common approach is to hybridize both in order to make them cooperate and then to benefit of their respective behaviors. Thus, hybrid metaheuristics have shown their efficiency to solve different optimization problems [44], including MOPs [45].

In the previous section, we saw that SEEA and I-IBMOLS were the overall more efficient method to approximate the efficient set for the problem under consideration. Moreover, these two methods are quite different to each other and do not explore the search space in the same way. Indeed, SEEA has been conceived in order to find a rough approximation of the Pareto set in a very short amount of time, whereas the non-iterative version of IBMOLS is able to improve an approximated set in a very efficient way. It could then be interesting to design a cooperation scheme between these two algorithms. The resulting hybrid metaheuristic could be particularly efficient for solving large size problems. Furthermore, both methods maintain a secondary population (the archive) in parallel of the main population to store non-dominated solutions. This archive is not only used as an external storage, but also takes part in the evolution engine as it serves to build new solutions to explore. Thus, each method can manage its own population and therefore use the archive as a single shared memory.

4.2. Cooperative schemes

The general idea of our hybridization scheme is to run SEEA and to launch IBMOLS regularly by using a subset of archive items as an initial population. Since the non-iterative version of IBMOLS naturally stops when its own archive does not receive any new efficient solution anymore, we can restart the SEEA process until the next step of the hybrid algorithm. A step can, for instance, be defined by a certain amount of time or by a certain number of generations. Besides, as SEEA uses the non-dominated solutions found by IBMOLS to create new ones and vice versa, the global archive is the only memory shared by the two search agents to exchange information. Resulting from this, we can imagine two versions of the hybrid algorithm: (i) a *periodic version*, in which

IBMOLS is launched at each step, and (ii) an *adaptive version*, in which IBMOLS is launched at a specific step only if a condition is verified. These two approaches will be denoted by PCS (for *periodic cooperative search*) and ACS (for *adaptive cooperative search*) in the remainder of the paper, and are, respectively, illustrated in Figs. 5 and 6.

The ACS method decides by itself, and on-line, if it is interesting to launch IBMOLS at a given step of the search process. The condition outlined here is that the archive of potentially efficient solutions does not improve enough with regards to the optimization scenario. A possibility is to measure the quality of the current archive A^t in comparison to the one of the previous step A^{t-1}. Different metrics exist to evaluate the convergence properties of an approximated efficient set in comparison to another. For instance, the hypervolume metric could have been envisaged, but its computation has the drawback of being time consuming. Thus, let us introduce the *contribution* metric C proposed by Meunier et al. [46]. This metric gives an idea of the quality of an approximated efficient set in comparison to another one in term of convergence, and can be computed in a reasonable runtime. In our case, at each step t of the ACS, we compute the contribution of the current archive A^t on the archive of the previous step A^{t-1}. Thus, as non-dominated solutions are not lost between two steps, A^t is at least as good as A^{t+1}, so that we know that $C(A^t, A^{t-1}) \in [0.5, 1]$ due to the contribution metric properties; see [46]. Assuming that the archive does not improve enough if the contribution of A^t on A^{t-1} is less than a user-given threshold $\delta \in [0.5, 1]$, we choose to launch IBMOLS only if $C(A^t, A^{t-1}) \leq \delta$. Let us remark that an ACS with a $\delta = 0.5$ is equivalent to SEEA, and that an ACS with a $\delta = 1.0$ is equivalent to PCS, the time spent in calculating the different contribution values in less.

Thus, thanks to the definition of a step t and to the δ parameter, it is possible to intentionally bias the balance between evolutionary search and local search within both PCS and ACS. This issue was pointed out in [47] for hybrid multi-objective metaheuristics. For instance, the higher the δ value, the more often the local search will be launched in the frame of the ACS method. Hence, the proposed hybrid algorithm can directly handle different specifications of the evolutionary-local search balance, i.e. from almost pure SEEA ($\delta = 0.5$) to almost pure IBMOLS ($\delta = 1.0$).

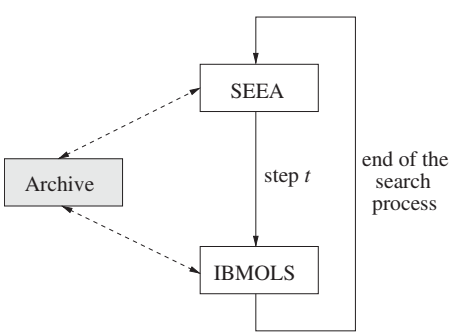


Fig. 5. Illustration of the periodic cooperative search (PCS).

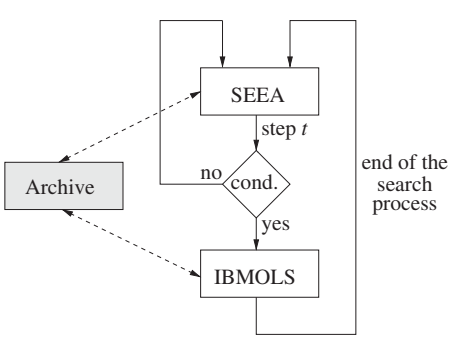


Fig. 6. Illustration of the adaptive cooperative search (ACS).

4.3. Related works

Different schemes exist on how two search methods can be combined. According to the taxonomy proposed in [44], the hybrid metaheuristic investigated in this paper can be classified on the *high-level teamwork hybrid* (HTH) class, and can then be denoted by HTH(SEEA+IBMOLS). In their survey on hybrid metaheuristics to solve combinatorial MOPs [45], Ehrgott and Gandibleux identify three categories of methods hybridizing an evolutionary algorithm with a neighborhood search algorithm: (i) an hybridization to make a method more aggressive, (ii) an hybridization to drive a method and (iii) an hybridization for exploiting complementary strengths. The last one consists of alternating between both search methods, which is the case within our hybridization. But, most existing approaches occur in a pipeline way: first the evolutionary algorithm, then the local search. The few teamwork hybridization techniques from the literature are most often compound of the same metaheuristic, like in the island model. Moreover, this class of hybrid methods often uses a neighborhood search method combining a set of scalarizing functions. The originality of the approach proposed in this paper is that search agents are based on different types of metaheuristics, are hybridized in a teamwork mode, and do not use any scalar approach to convert the multi-objective vector function into a single-objective one. Furthermore, one of the variants we propose, the adaptive one, automatically detects when to start the local search according to the optimization scenario.

4.4. Experiments

In order to experiment the efficiency of our two cooperative approaches, we compare them to SEEA and to the iterative version of IBMOLS by using the same experimental protocol as the one defined in Section 3.4.1. The parameter setting for each hybrid method is followed by some computational results and by a discussion on the contribution of the hybridization for the resolution of the RSP as a bi-objective problem. Finally, we give an idea of the behavior of our approaches in comparison to the state-of-art single-objective approach proposed in [1] for the problem under study.

4.4.1. Parameter setting

For both cooperative search methods, the population size managed by SEEA is set to 100, and the population size managed by IBMOLS is set according to the instance under consideration. These sizes have been set on the same way that for the stand-alone iterative version of IBMOLS on the previous section, see Table 5. But, for large-size instances, initial experiments were not satisfying since the hybrid algorithms were generally not able to launch IBMOLS more than once during the search process, as it was too much time consuming. For this reason, we bounded the IBMOLS population size to 30. Then, a IBMOLS population of 20 individuals has been set for instances with less than 100 nodes, and a IBMOLS population of 30 individuals has been set for instances with 100 nodes and more. Moreover, note that in addition to the benchmark test instances we investigated previously, we experimented two larger TSPLIB problems, namely *pr* 439 and *pr* 1002. Following the general trend identified in the previous section, we set the instance-specific parameters for

Table 5
Stopping criteria for additional test instances.

Instance	Runtime
pr 439	50 ft
pr 1002	50 min

For other instances, see Table 1.

I-IBMOLS as follows: a population of 70 and 100 individuals, respectively, and a noise rate of 10%. The other parameters were set in the same way than for other instances. The step t of the hybrid algorithms has been set to 0.5% of the maximum runtime available for the instance under consideration. In [43], we investigated different δ- values for the ACS method: 0.6, 0.7, 0.8 and 0.9. As pointed out above, an ACS with a δ- value of 0.5 and 1.0 is similar to PCS and SEEA, respectively. This has lead to an effective δ- value of 0.8 for the stopping criterion we adopted.

4.4.2. Computational results

Tables 6 and 7 give a comparison between the results obtained by SEEA, I-IBMOLS, PCS and ACS according to the I_H^- and $I_{\varepsilon+}^1$ metric, respectively. Firstly, except for the *eil* 51 instance, PCS and ACS both always statistically outperform SEEA with respect to, at least, one metric. In comparison to I-IBMOLS, PCS and ACS often obtain better results, in particular for instances with 150 nodes and more. However, in any case, both hybrid metaheuristics are never statistically outperformed by a stand-alone search method. Thus, the benefit of the cooperation scheme appears relevant in most large-size instances, but the addition of an adaptive mechanism does not seem to have a big influence on the results. Indeed, for almost every test instance, the outcome measures did not return a statistically significant difference between PCS and ACS. The few exceptions to notice are for the *bier* 127 and the *pr* 439 instances where PCS obtain better results with regard to one or both metrics, and for the *pr* 1002 instance where ACS outperforms PCS with respect to I_H^-. However, during our experiments, we observed that the difference between the average number of times that IBMOLS is launched during the search process of PCS and ACS is relatively thin. This can be explained by the fact that (i) IBMOLS takes more time to find non-dominated solutions by starting with a population of poorer quality, that is the case in PCS comparing to ACS (at least for the first launch), and (ii) a part of the runtime allocated to the algorithm is used to compute a contribution value at every step of ACS, whereas PCS devotes all of its runtime to the search process. These two aspects lead to the fact that the number of times that IBMOLS is launched is in the end, more or less balanced between both cooperative methods.

Comparison with exact single-objective results: As a last step, we provide a comparison between the results found for the bi-objective RSP investigated in this paper and the ones of the single-objective RSP investigated in [1], where both costs are summed up.[6] In order to provide optimal solutions visiting approximately 25%, 50%, 75% and 100% of the total number of nodes, the authors set the ring cost c_{ij} and the assignment cost d_{ij} between two nodes v_i and v_j in the following way: $c_{ij} = \lceil \alpha l_{ij} \rceil$ and $d_{ij} = \lceil (10-\alpha) l_{ij} \rceil$ with $\alpha \in \{3, 5, 7, 9\}$, where l_{ij} denotes the distance between v_i and v_j given in the TSPLIB files. In order to give a rough idea of the results we obtained, we compare the best found scalar value as detailed above to the optimum[7] found in [1]. Additionally, we compare the best solution visiting every node we have found with the optimal TSP solution available on the TSPLIB website.[8] Table 8 gives the error ratio between the best known value and the best one we have found for every identified (single) objective and every

Table 6
Algorithms comparison with respect to the I_H^- metric.

			I-IBMOLS	SEEA	PCS	ACS
eil 51	PCS	(4.751)	$\equiv$	$\equiv$	–	$\equiv$
	ACS	(4.482)	$\equiv$	$\equiv$	$\equiv$	–
st 70	PCS	(2.691)	$\succcurlyeq$	$\succ$	–	$\equiv$
	ACS	(2.865)	$\equiv$	$\succ$	$\equiv$	–
kroA 100	PCS	(3.738)	$\equiv$	$\succ$	–	$\equiv$
	ACS	(3.326)	$\succcurlyeq$	$\succ$	$\equiv$	–
bier 127	PCS	(3.071)	$\equiv$	$\succ$	–	$\succ$
	ACS	(3.693)	$\equiv$	$\succ$	$\prec$	–
kroA 150	PCS	(2.792)	$\succ$	$\succ$	–	$\equiv$
	ACS	(2.624)	$\succ$	$\succ$	$\equiv$	–
kroA 150	PCS	(2.247)	$\succ$	$\succ$	–	$\equiv$
	ACS	(2.260)	$\succ$	$\succ$	$\equiv$	–
pr 264	PCS	(1.342)	$\succ$	$\succ$	–	$\equiv$
	ACS	(1.404)	$\succcurlyeq$	$\succ$	$\equiv$	–
pr 299	PCS	(1.277)	$\equiv$	$\succ$	–	$\equiv$
	ACS	(1.293)	$\equiv$	$\succ$	$\equiv$	–
pr 439	PCS	(0.348)	$\succ$	$\succ$	–	$\succ$
	ACS	(0.733)	$\succ$	$\equiv$	$\prec$	–
pr 1002	PCS	(2.449)	$\succ$	$\succ$	–	$\prec$
	ACS	(0.707)	$\succ$	$\succ$	$\succ$	–

The number in brackets denotes the average metric value for the algorithm and the instance under consideration (multiplied by 10^{-3}). For each instance, either the algorithm located at a specific row significantly dominates the algorithm located at a specific column ($\succ$ or $\succcurlyeq$ for a p-value less or equal to 0.01 or to 0.05, respectively), either it is significantly dominated ($\prec$ or $\preceq$ for a p-value less or equal to 0.01 or to 0.05, respectively), or there is no significant difference between both ($\equiv$).

Table 7
Algorithms comparison with respect to the $I_{\varepsilon+}^1$ metric.

			I-IBMOLS	SEEA	PCS	ACS
eil 51	PCS	(9.561)	$\equiv$	$\equiv$	–	$\equiv$
	ACS	(9.363)	$\equiv$	$\equiv$	$\equiv$	–
st 70	PCS	(6.328)	$\succcurlyeq$	$\equiv$	–	$\equiv$
	ACS	(7.064)	$\equiv$	$\equiv$	$\equiv$	–
kroA 100	PCS	(8.963)	$\equiv$	$\equiv$	–	$\equiv$
	ACS	(7.533)	$\succ$	$\succ$	$\equiv$	–
bier 127	PCS	(8.114)	$\equiv$	$\succ$	–	$\succ$
	ACS	(9.818)	$\equiv$	$\succ$	$\prec$	–
kroA 150	PCS	(5.450)	$\succ$	$\succ$	–	$\equiv$
	ACS	(5.587)	$\succ$	$\succ$	$\equiv$	–
kroA 150	PCS	(5.057)	$\succ$	$\succ$	–	$\equiv$
	ACS	(5.702)	$\succ$	$\succ$	$\equiv$	–
pr 264	PCS	(4.242)	$\succ$	$\equiv$	–	$\equiv$
	ACS	(4.317)	$\succ$	$\equiv$	$\equiv$	–
pr 299	PCS	(4.501)	$\equiv$	$\succ$	–	$\equiv$
	ACS	(4.048)	$\equiv$	$\succ$	$\equiv$	–
pr 439	PCS	(2.760)	$\succ$	$\succ$	–	$\succ$
	ACS	(5.553)	$\succcurlyeq$	$\succ$	$\prec$	–
pr 1002	PCS	(5.391)	$\succ$	$\succ$	–	$\equiv$
	ACS	(4.309)	$\succ$	$\succ$	$\equiv$	–

The number in brackets denotes the average metric value for the algorithm and the instance under consideration (multiplied by 10^{-3}). For each instance, either the algorithm located at a specific row significantly dominates the algorithm located at a specific column ($\succ$ or $\succcurlyeq$ for a p-value less or equal to 0.01 or to 0.05, respectively), either it is significantly dominated ($\prec$ or $\preceq$ for a p-value less or equal to 0.01 or to 0.05, respectively), or there is no significant difference between both ($\equiv$).

[6] Note that it was not possible to compare our results to the ones of the other formulation of a single-objective RSP investigated in [2,14,15], where the assignment cost is subject to a constraint, due to the way the bound has been fixed.

[7] In fact, the authors imposed a time limit for their experiments. They report the best solution found so far for the instances exceeding this time limit, which is the case for the *kroA* 200 instance with an $\alpha = 3$, 5 and 9 in Table 8.

[8] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

Table 8
Error ratio between the cost value of the best known TSP solution and the best single-objective ring cost value found in [1] in comparison to the best value found during our experiments.

Instance	Optimal TSP solution (%)	Optimal RSP solution [1] (%)			
		$\alpha=3$	$\alpha=5$	$\alpha=7$	$\alpha=9$
eil 51	0.67	0.67	0.75	0.37	0.69
st 70	0.31	0.31	0.49	0.56	0.42
kroA 100	0.08	0.08	0.05	0.12	**0.00**
bier 127	0.64	0.64	0.34	0.52	0.01
kroA 150	1.38	1.01	1.04	0.17	**0.00**
kroA 150	1.22	**−4.82**	1.05	0.69	**−1.55**
pr 264	2.26	–	–	–	–
pr 299	1.72	–	–	–	–
pr 439	4.63	–	–	–	–
pr 1002	14.51	–	–	–	–

benchmark test instance. In comparison to [1], this ratio is always under 1.5%, and is mostly below 1% for every instance. The optimum is even found for the *kroA* 100 and the *kroA* 150 instances with an $\alpha=9$, and a better solution is found for the *kroA* 200 instances with an $\alpha=3$ and an $\alpha=9$. As regards to optimal TSP solutions, our results are quite close for instances with 200 nodes or less. For larger instances, they are not quite as good, especially for the *pr* 1002 instance, where the error ratio is close to 15%. To summarize, in comparison to single-objective optimal or near optimal results, the search methods we proposed in this paper to solve the B-RSP are quite promising with regard to the relatively small computational time available and to the size of the problem instances to be solved. Nevertheless, let us remind that the comparison is here done with the best results we obtained during the whole set of experiments we performed, and may not be as good for a single simulation run.

4.5. Discussion

Two general-purpose cooperative schemes combining SEEA and the non-iterative version of IBMOLS have been designed in this section for multi-objective combinatorial optimization: a periodic one (PCS), and an adaptive one (ACS). Indeed, ACS evolves adaptively according to the search scenario and decides by itself, and on-line, when the cooperation must occur. Another benefit of ACS is the possibility of deliberately orienting the balance between evolutionary search and local search. In comparison to stand-alone metaheuristics, these two hybrid search methods statistically improve the results on a large number of RSP instances, and particularly on large-size ones. However, the efficiency difference between PCS and ACS is almost negligible. This can be explained by the fact that ACS spends, at each step, a significant time to compute whether the cooperation should occur or not. This is not the case in the frame of PCS, so that the latter devotes all of its computational time to the search process. Moreover, in comparison to single-objective approaches from the literature, the algorithms proposed in this paper for the bi-objective RSP seem to provide good-quality solutions.

5. Further comments

A new multi-objective routing problem, the bi-objective ring star problem, has been investigated in this paper for the first time. It aims at locating a cycle through a subset of nodes of a graph while minimizing a ring cost, related to the length of the cycle, and an assignment cost, from non-visited nodes to visited ones. In spite of its clear bi-objective nature, this problem has always been addressed in a single-objective way, either where both costs are combined [1], or where one cost is regarded as a constraint [2,15]. As a first step, we proposed a set of four population-based metaheuristics to approximate the efficient set for the problem under consideration, namely I-IBMOLS [5], IBEA [6], NSGA-II [7], and SEEA, proposed here for the first time. We concluded that I-IBMOLS was the overall most competitive method, and that SEEA was a solid competitor. As a second step, we designed two general-purpose cooperative schemes between SEEA and the non-iterative version of IBMOLS for multi-objective combinatorial optimization problems. And we illustrated the contribution of these hybridization mechanisms to improve the performance of stand-alone methods.

Although the approaches proposed in this paper are already promising, a few research directions are still open. First is the possibility to improve the population initialization strategy used within every search method. Second, we pointed out that the recombination operator designed for the problem under study has a tendency to break the parent ring structures in the offspring individuals. Then, as proposed in [15], we could employ a traveling salesman problem heuristic to improve the ring cost of newly generated solutions. Third, given the number of mutation operators designed within the evolutionary algorithms, it would be interesting to determine the appropriate rate for each one of them in an adaptive way, such as in [48]. Note that the last two points could largely improve the efficiency of evolutionary algorithms, and then would be beneficial for the hybrid methods as well. Finally, we found out that SEEA was a good alternative to state-of-the-art multi-objective evolutionary algorithms when a given, relatively small, amount of computational time is available. In the future, we plan to tackle other kinds of combinatorial problems within SEEA to verify if our observations are still valid, especially for problems where more than two objectives are involved. The same remark can also be done with regard to the cooperative approaches proposed in this paper.

Acknowledgments

A subpart of this work has already been published in [4]. The authors would like to acknowledge Matthieu Basseur and Edmund K. Burke for their helpful contribution. Furthermore, the authors are grateful for valuable discussions held with Nicolas Jozefowiez and Clarisse Dhaenens.

References

[1] Labbé M, Laporte G, Rodríguez Martín I, Salazar González JJ. The ring star problem: polyhedral analysis and exact algorithm. Networks 2004;43:177–89.
[2] Labbé M, Laporte G, Rodríguez Martín I, Salazar González JJ. Locating median cycles in networks. European Journal of Operational Research 2005;160(2):457–70.
[3] Jozefowiez N, Semet F, Talbi E-G. Multi-objective vehicle routing problems. European Journal of Operational Research 2008;189(2):293–309.
[4] Liefooghe A, Jourdan L, Basseur M, Talbi E-G, Burke EK. Metaheuristics for the bi-objective ring star problem. In: Eighth European conference on evolutionary computation in combinatorial optimisation (EvoCOP 2008). Lecture notes in computer science, vol. 4472. Napoli, Italy: Springer; 2008. p. 206–17.
[5] Basseur M, Burke EK. Indicator-based multi-objective local search. In: IEEE congress on evolutionary computation (CEC 2007). Singapore: IEEE Press; 2007. p. 3100–7.
[6] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Eighth international conference on parallel problem solving from nature (PPSN VIII). Lecture notes in computer science, vol. 3242. Birmingham, UK: Springer; 2004. p. 832–42.
[7] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002;6(2):182–97.

[8] Coello Coello CA, Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. Boston, MA, USA: Kluwer Academic Publishers; 2002.
[9] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley; 2001.
[10] Ehrgott M, Gandibleux X. Approximative solution methods for multiobjective combinatorial optimization. Top 2004;12(1):1–89.
[11] Labbé M, Laporte G, Rodríguez Martín I. Path, tree and cycle location. In: Crainic T, Laporte G, editors. Fleet management and logistics. Boston, MA, USA: Kluwer Academic Publishers; 1998. p. 187–204.
[12] Labbé M, Laporte G, Rodríguez Martín I, Salazar González JJ. The median cycle problem. Working paper, CRT-99-29, Université de Montréal, 1999.
[13] Miettinen K. Nonlinear multiobjective optimization, vol. 12. Boston, MA, USA: Kluwer Academic Publishers; 1999.
[14] Moreno Pérez JA, Moreno-Vega JM, Rodríguez Martín I. Variable neighborhood tabu search and its application to the median cycle problem. European Journal of Operations Research 2003;151(2):365–78.
[15] Renaud J, Boctor FF, Laporte G. Efficient heuristics for median cycle problems. Journal of the Operational Research Society 2004;55(2):179–86.
[16] Dias TCS, de Sousa Filho GF, Macambira EM, Cabral LAF, Fampa MHC. An efficient heuristic for the ring star problem. In: Fifth international workshop on experimental algorithms (WEA 2006), Lecture notes in computer science, vol. 4007. Menorca, Spain: Springer; 2006. p. 24–35.
[17] Current JR, Schilling DA. The median tour and maximal covering tour problems: formulations and heuristics. European Journal of Operational Research 1994;73:114–26.
[18] Doerner K, Focke A, Gutjahr WJ. Multicriteria tour planning for mobile healthcare facilities in a developing country. European Journal of Operational Research 2007;179(3):1078–96.
[19] Beasley JE, Nascimento EM. The vehicle routing-allocation problem: a unifying framework. Top 1996;4(1):65–86.
[20] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: First international conference on genetic algorithms (ICGA 1985), Pittsburgh, PA, USA: Lawrence Erlbaum Associates; 1985. p. 93–100.
[21] Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Fifth international conference on genetic algorithms (ICGA 1993). Urbana-Champaign, IL, USA: Morgan Kaufmann; 1993. p. 416–23.
[22] Xu J, Chiu SY, Glover F. Optimizing a ring-based private line telecommunication network using tabu search. Management Science 1999;45(3):330–45.
[23] Baldacci R, Dell'Amico M, Salazar Gonzàlez JJ. The capacitated m-ring star problem. Operations Research 2007;55(6):1147–62.
[24] Labbé M, Laporte G. Maximizing user convenience and postal service efficiency in post box location. Belgian Journal of Operations Research Statistics and Computer Science 1986;26:21–35.
[25] Akinc U, Srikanth K. Optimal routing and process scheduling for a mobile service facility. Networks 1992;22(2):163–83.
[26] Burke EK, Kendall G., editors. Search methodologies: introductory tutorials in optimization and decision support techniques. Berlin, Germany: Springer; 2005.
[27] Glover FW, Kochenberger GA, editors. Handbook of metaheuristics of international series in operations research management science, vol. 57. Boston, MA, USA: Kluwer Academic Publishers; 2003.
[28] Paquete L, Stützle T. Stochastic local search algorithms for multiobjective combinatorial optimization: a review. In: Gonzalez TF, editor. Handbook of approximation algorithms and metaheuristics of computer & information science series, vol. 13. London: Chapman & Hall, CRC; 2007.
[29] Zitzler E, Thiele L, Laumanns M, Foneseca CM, Grunert da Fonseca V. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 2003;7(2):117–132.
[30] Helbig S, Pateva D. On several concepts for ε- efficiency. OR Spektrum 1994;16(3):179–86.
[31] Hernández-Díaz AG, Santana-Quintero LV, Coello CAC, Molina J. Pareto-adaptive epsilon-dominance. Evolutionary Computation 2007;15(4):493–517.
[32] Lourenco HR, Martin O, Stützle T. Iterated local search. In: Glover F, Kochenberger G, editors. Handbook of metaheuristics of international series in operations research & management science, vol. 57. Norwell, MA, USA: Kluwer Academic Publishers; 2002. p. 321–53 [Chapter 11].
[33] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 1999;3(4):257–71.
[34] Laumanns M, Zitzler E, Thiele L. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In: First international conference on evolutionary multi-criterion optimization (EMO 2001), Lecture notes in computer science, vol. 1993. Zurich, Switzerland: Springer; 2001. p. 181–96.
[35] Corne DW, Knowles JD, Oates MJ. The pareto envelope-based selection algorithm for multiobjective optimization. In: Sixth international conference on parallel problem solving from nature (PPSN VI), Lecture notes in computer science, vol. 1917. Paris, France: Springer; 2000. p. 839–48.
[36] Valenzuela CL. A simple evolutionary algorithm for multi-objective optimization (SEAMO). In: IEEE congress on evolutionary computation (CEC 2002). Piscataway, NJ, USA: IEEE Press; 2002. p. 717–22.
[37] Knowles J, Corne D. Bounded Pareto archiving: theory and practice. In: Gandibleux X, Sevaux M, Sörensen K, T'Kindt V, editors. Metaheuristics for multiobjective optimisation of lecture notes in economics and mathematical systems, vol. 535. Berlin: Springer; 2004. p. 39–64 [Chapter 2].
[38] Bean J. Genetic algorithms and random keys for sequencing and optimization. ORSA Journal on Computing 1994;6(2):154–60.
[39] Liefooghe A, Jourdan L, Talbi E-G. A unified model for evolutionary multiobjective optimization and its implementation in a general purpose software framework: ParadisEO-MOEO. Research Report RR-6906, Institut National de Recherche en Informatique et Automatique (INRIA); 2009.
[40] Reinelt G. TSPLIB—a traveling salesman problem library. ORSA Journal on Computing 1991;3(4):376–84.
[41] Knowles J, Thiele L, Zitzler E. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland; 2006 [revised version].
[42] Bleuler S, Laumanns M, Thiele L, Zitzler E. PISA—a platform and programming language independent interface for search algorithms. In: Second international conference on evolutionary multi-criterion optimization (EMO 2003), Lecture notes in computer science, vol. 2632. Faro, Portugal: Springer; 2003. p. 494–508.
[43] Liefooghe A, Jourdan L, Talbi E-G. Metaheuristics and their hybridization to solve the bi-objective ring star problem: a comparative study. Research Report RR-6515, Institut National de Recherche en Informatique et Automatique (INRIA); 2008.
[44] Talbi E-G. A taxonomy of hybrid metaheuristics. Journal of Heuristics 2002;8(2):541–64.
[45] Ehrgott M, Gandibleux X. Hybrid metaheuristics for multi-objective combinatorial optimization. In: Blum C, Blesa MJ, Roli A, Sampels M, editors. Hybrid metaheuristics—an emergent approach for optimization of studies in computational intelligence, vol. 114. Berlin, Germany: Springer; 2008. p. 221–59 [Chapter 9].
[46] Meunier H, Talbi E-G, Reininger P. A multiobjective genetic algorithm for radio network optimization. In: IEEE congress on evolutionary computation (CEC 2000). San Diego, USA: IEEE Press; 2000. p. 317–24.
[47] Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation 2003;7(2):204–23.
[48] Basseur M, Seynhaeve F, Talbi E-G. Design of multi-objective evolutionary algorithms: application to the flow shop scheduling problem. In: IEEE congress on evolutionary computation. Piscataway, NJ, USA: IEEE Press; 2002. p. 1151–6.

J Math Model Algor (2010) 9:195–208
DOI 10.1007/s10852-010-9128-y

Single- and Multi-Objective Cooperation for the Flexible Docking Problem

Jean-Charles Boisson · Laetitia Jourdan · El-Ghazali Talbi · Dragos Horvath

Received: 1 March 2009 / Accepted: 1 December 2009 / Published online: 19 February 2010
© Springer Science+Business Media B.V. 2010

Abstract In this article, the impact of single-objective methods as intensification factors in a multi-objective approach is presented for the flexible docking problem. Based on a novel tri-objective model, a parallel multi-objective genetic algorithm has been designed. However, due to the high variability of the energy objective, intensification methods focused on this objective have been also included in order to improve the convergence speed of the genetic algorithm and the quality of the results. The corresponding approach, combining single- and multi-objective methods, has been proved efficient according to the tested instances and the quality criterion used.

Keywords Multi-objective optimization · Molecular docking · Genetic algorithm

1 Introduction

The molecular docking problem consists in finding how two molecules, generally a very small called ligand, and a bigger one, called receptor, are going to make together a stable complex. This problem is essential in drug design because the action of a (or more) ligand(s) may activate, inhibit or modify the behavior of the receptor. This difficult problem, as many problems in bioinformatics, has a huge associated search space. It is first due to the combination of all the possible location of the molecules from each other. Furthermore, the molecules generally modify their 3D shape during the docking process. Consequently the size of the search space explodes. Each 3D shape of a molecule is called conformation. It corresponds to the specific location of all the atoms that compound the molecule. According to the flexibility of the molecules, three versions of the docking problem exist: rigid, semi-flexible and (full)

J.-C. Boisson (✉) · L. Jourdan · E.-G. Talbi · D. Horvath
INRIA Dolphin project team (Lille), "Infochimie" laboratory, Strasbourg, France
e-mail: jean-charles.boisson@univ-reims.fr

flexible. The docking is called rigid if none of the molecules has its conformation modified during the docking process. In the semi-flexible version of the docking, one molecule, generally the ligand, is allowed to have its conformation modified. The last version of docking, the most resource and time consuming version, is the flexible one. In this version, the both molecules may have conformation modifications during the process.

However, this problem is not a new one. Since the 1990s a lot of approaches have tried to tackle this problem. Bursulaya et al. [21] proposed an interesting study concerning the different methods used to solve this problem. The most known algorithms are the ones included in the following tool suites: Autodock [18], Flex [20], DOCK [19], ... The most recent algorithms are always designed for the flexible version of the docking in order to better simulate the real biological process.

Based on a collaboration with chemists, a new multi-objective model has been designed for the flexible docking problem. It mixes energy criterion and geometric criterion. A multi-objective algorithm has been designed with this model. The aim of our work is to show the impact of single-objective intensification processes in this multi-objective approach. To present our study, the paper is divided into different sections. Firstly, our tri-objective model is detailed. Then the method implementing this model is presented step by step. Due to the cost of the evaluation criteria, a parallelized version of our approach is also proposed. After the validation of the different hypothesis linked to our model, the results of our approach according to the different configurations of our algorithm are given. Lastly, conclusions and perspectives about this work are exposed.

2 Model

In this section, our tri-objective model, mixing energy and geometric criteria, is presented.

2.1 Energy Criterion and Associated Force Field

The lower the associated energy is, the more stable a molecule is. Based on this fact, all the docking approaches (single- and multi-objective) have at least one criterion based on the energy of the ligand/receptor complex. The computation of this energy needs an associated force field. A lot of force fields exist and they are not all dedicated to work with the same types of molecule. It is very expansive to design a force field that exactly simulates all the forces and parameters of the real docking process. Due to this cost, empirical approaches for force field design, as the one presented in this section, are generally preferred. Empirical approaches are often considered as the best compromise between a realistic and an expansive simulation.

In our case, a home made force field inspired of the **C**onsistent **V**alence **F**orce **F**ield has been used. The energy criterion associated to this force field exploits the standard energy contributions and can be divided into two main terms: the bonded atom energy and the non bonded atom one.

The bonded atom energy corresponds to all the energy interactions that occurs between two atoms that are linked together (bond contribution), linked to the same atom (angle contribution) or belonging to the same torsion (torsion contribution).

A torsion is a set of four atoms linked together thanks to three bonds. All these contributions are detailed in the following equation:

$$E_{\text{bonded_atoms}} = \sum_{\text{bonds}} K_b (b - b_0)^2 + \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2 + \sum_{\text{torsions}} K_\phi (1 - \cos n(\phi - \phi_0)) \tag{1}$$

K_b, K_θ and K_ϕ are force constants linked to each contribution (bond, angle and torsion). b_0, θ_0 and ϕ_0 are empirical values for the bond size, the angle and the phase difference. b, θ and ϕ are the real value for the same characteristics. For the torsion contribution, n is the period linked to the type of the central bond of the torsion (double or triple). The bonded atom energy fluctuates during the modification of the conformation of a molecule.

The non bonded atom energy describes the energy contributions between atoms that are not near according to the topology of the molecule (the bonds, angles or torsions) but near according to the location of the atoms. The atoms concerned by the non bonded energy terms can belong to the same molecule or not. All these terms are described in the following equation:

$$E_{\text{non_bonded_atoms}} = \sum_{\text{Van der Waals}} \frac{K_{ij}^a}{d_{ij}^{12}} - \frac{K_{ij}^b}{d_{ij}^6} + \sum_{\text{Coulomb}} \frac{q_i q_j}{4\pi\epsilon d_{ij}} + \sum_{\text{desolvation}} \frac{K q_i^2 V_j + q_j^2 V_i}{d_{ij}^4} \tag{2}$$

In this equation, q_i is the charge of the atom i; d_{ij} is the atomic distance between the atoms i and j; V_i is a volumetric measure for the atom i; K and K_{ij}^x are constant forces linked to each contribution. The non bonded energy is the one giving the best description of the interactions between the ligand and the receptor. These interactions combine attractive and repulsive forces including the action of a solvent. Generally it is the water that is considered.

According to these two equations, our first criterion is the global energy:

$$E_{\text{total}} = E_{\text{bonded_atoms}} + E_{\text{non_bonded_atoms}} \tag{3}$$

The associated force field, originally designed for protein structure prediction algorithms, has been tuned on several types of proteins and it is constantly improved.

2.2 Surface Criterion

The surface is a geometric criterion based on the hypothesis that the surface area describing both the receptor and the ligand is going to decrease because of the penetration of the ligand into the receptor. There are several ways to draw molecular surfaces. An atom is represented as a sphere with its Van der Waals radius. By summing all the spheres of a molecule, the Van der Waals surface is obtained.

However, this surface does not model the solvent action. The solvent accessible surface and the Connoly surface include the simulation of the solvent action, they are been described in [8] and [9] respectively. These surfaces model the path that follows the center of a sphere that represents the solvent. Generally it is a sphere of 1.4 Å[1] because the standard solvent is the water. The sphere center rolls on the atom spheres. The raw area draws by the sphere is the solvent accessible surface, the smoother version of the drawing is the Connoly surface.

In our model, for a good cost/quality compromise, the solvent accessible surface has been chosen as surface criterion. The algorithm used has been presented in [10] and approximate the Shrake and Rupley method [12]. A recent work using this algorithm has been presented in [11].

2.3 Stability Criterion: A Hypothesis to Validate

This criterion is based on an entropy calculus on a ligand/receptor complex. It describes the resistance of the complex against location modifications of the ligand into the receptor (rotation/translation). This calculus needs a sampling of neighbor complexes. According to the number of neighbors nbConf and their associated energy, the stability criterion is given by the Eq. 4.

$$\text{Stability} = -\frac{1}{\beta} * ln \sum_{i=1}^{\text{nbConf}} -\beta * \text{Energy}[i] \tag{4}$$

The β value is $\frac{1}{kT}$ with k being the Boltzmann constant and T being the temperature. The aim is to estimate if a complex is more probable than an other. The hypothesis contained in Eq. 4 is that a good ligand/receptor complex is rather in a valley of equivalent energies than in a narrow well surrounded by energy peaks.

3 Methods

In this section, the design of our multi-objective algorithm is detailed.

3.1 The Algorithm Choice

Three reasons have motivated the choice of genetic algorithms to solve our problem with the proposed multi-objective model:

1. A population-based method allows to propose naturally a set of solutions as a result.
2. The genetic algorithms have a good power of exploration that can be combined with intensification mutation operators. Consequently, they are able to tackle problems with a huge search space.
3. For genetic algorithms, multi-objective schemes proved to be efficient and robust exist.

[1] 1 angstrom (Å) is equivalent to 0.1 nanometer

3.2 Multi-Objective Genetic Algorithm (MOGA)

The design of genetic algorithms (GAs) is mainly based on the choice of the solution representation and evaluation. From these choices, the operators and all the other mechanisms of a GA can be defined. In our case, additional mechanisms are needed since our GA is a multi-objective one.

3.2.1 Solution Representation

In order to code a ligand/receptor complex, several representations are proposed in the literature. The most used is the torsion based representation. All the torsions that describes a molecule are coded as a vector of angle values. Knowing the angle of each torsion is sufficient to determine the conformation of the molecule. This representation is the smallest in memory but may generate additional calculi for the solution evaluation, the GA operators and the molecules drawing. Another representation is the Cartesian coordinates of each atom. This second representation is bigger than the torsion based one, but is very easy to apply on solution evaluation. With the atomic coordinates of each atom, the conformation of the molecule is directly known. That is why the ligand/receptor complexes are represented as a couple of atomic coordinate lists: one for the ligand and one for the receptor.

3.2.2 Multi-Objective Scheme

In order to design MOGAs, robust schemes already exist. Lot of them compare solutions using the dominance notion. A solution dominates another one if at least one of its criterion value is strictly better and the others criteria are better or equal to the other solution. Maybe the scheme the most known and used is the Non Dominated Sorting Genetic Algorithm II NSGA-II [13]. Also based on the notion of dominance, MOGA (Multi-Objective Genetic Algorithm) [15] and SPEA (Strength Pareto Evolutionary Algorithm) [16, 17] can be also cited. IBEA [14] (Indicator Based Evolutionary Algorithm) is a bit different. This algorithm is based on binary quality indicators.

Two of theses schemes have been tested thanks to the ParadisEO platform [23][2] and more precisely MOEO for Multi-Objective Evolving Object [22] : NSGA-II and IBEA. Statistical results have proved that for our approach and our instances, IBEA is better than NSGA-II [1].

3.2.3 Final Quality Evaluation

According to our home made force field, the only criterion to evaluate the quality of our approach and allowing to be comparable with other ones is the **R**oot **M**ean **S**quare **D**eviation (RMSD). According to [7], the RMSD is generally defined as follows:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{n}(dx_i^2 + dy_i^2 + dz_i^2)}{n}} \tag{5}$$

[2] http://paradiseo.gforge.inria.fr

In the Eq. 5, n is the total number of heavy atoms (no hydrogen), dx_i, dy_i and dz_i are the deviations of the Cartesian coordinates of the atoms between the same molecules at different locations. The unity of the RMSD is the *angström* noted Å. 1 Å is equivalent to 0.1 nanometer. For our docking approach, the RMSD computation is made between the ligand in its optimal location and the locations found at the end of the algorithm. According to other approaches, a docking is considered as a good one with solution of RMSD in [1, 2] Å and as a very good one with solutions of RMSD < 1 Å.

Although the RMSD allows to easily compare different docking approach, this criterion is not a robust quality indicator. Due to its definition, several parameters may have an impact of the result of the RMSD computation: the size of the considered molecule or symmetric parts in the molecule. A small deviation in a small molecule generates a great perturbation of the RMSD but not in a bigger molecule. Symmetric parts in a conformation can generate a bad RMSD despite a good global location of the molecule.

Furthermore, using the RMSD as a quality indicator assumes that the optimal location of the ligand is known. In real studies, this location is the goal to reach and is unknown.

3.2.4 Operators

Specific recombination and mutation operators have been designed.

3.2.4.1 *Recombination:* according to the solution representation, a ligand swap between two complexes has been chosen. This type of mechanism is simple and does not need a verification step because of the criteria used in our model. If a ligand swap generates atom collisions or too narrow positions for the atoms, the associated energy will greatly increase, mainly due to the Van der Waals force. This type of recombination allows to rapidly share the good properties of the best individuals.

3.2.4.2 *Mutation:* two main types of mutations have been designed: standard ones and specific ones dedicated to our approach.

Standard mutations: during a flexible docking process, let's introduce the well known three standard modifications:

1. Conformation rotation: one member of the ligand/receptor complex (generally the smaller, so the ligand) has its conformation rotated. The center of the rotation is a virtual point in the middle of the chosen molecule and the rotation angle is randomly determined according to the X, Y and Z axis. In our case, the rotation always concerns the ligand.
2. Conformation translation: one member of the complex follows a translation according to a 3D vector randomly generated. As for the first mutation, only the ligand will be translated.
3. Torsion rotation: this modification of conformation is only feasible in the flexible case. One torsion angle is modified by rotation and consequently the conformation of the molecule changes. The impact of the rotation on the energy criterion depends of the place of the torsion in the backbone of the concerned molecule.

Advanced mutations: these mutations are divided in two families: the specific ones and the intensive ones. The first family contains mutations that follow or exploit the specific characteristics of our problem although the second one incorporates intensification mechanism dedicated to the speed-up of the stable complexe search.

– *Specific mutations:* each mutation has its own goal:

1. Mutation reverse: according to the standard definition of the RMSD, molecules with symmetric part may produce bad RMSD value despite a good position in the receptor. The reverse mutation allows not only to avoid this type of situation but also allows to restart exploration of the search when the ligand is trapped in a local optimum. This mutation makes a big rotation of the ligand, typically 180°.
2. Big and small rotations: this mutation is an extension of the standard rotation mutation. The aim is to be able to produce rotation according to two ranges of values: one with small values and the other one with high values. A priori, during the docking process, small rotations, respectively big rotations, may be effective during several steps but not during the entire search.
3. Several Mutations in One mutation (SMO): this is simply the application of the other mutations several times without evaluation. This type of mutation is dedicated to tackle the limit linked to each mutation. This mutation can be viewed as a variable neighborhood search mechanism. It allows to exit from local optimum and access to other areas of the search space. However,

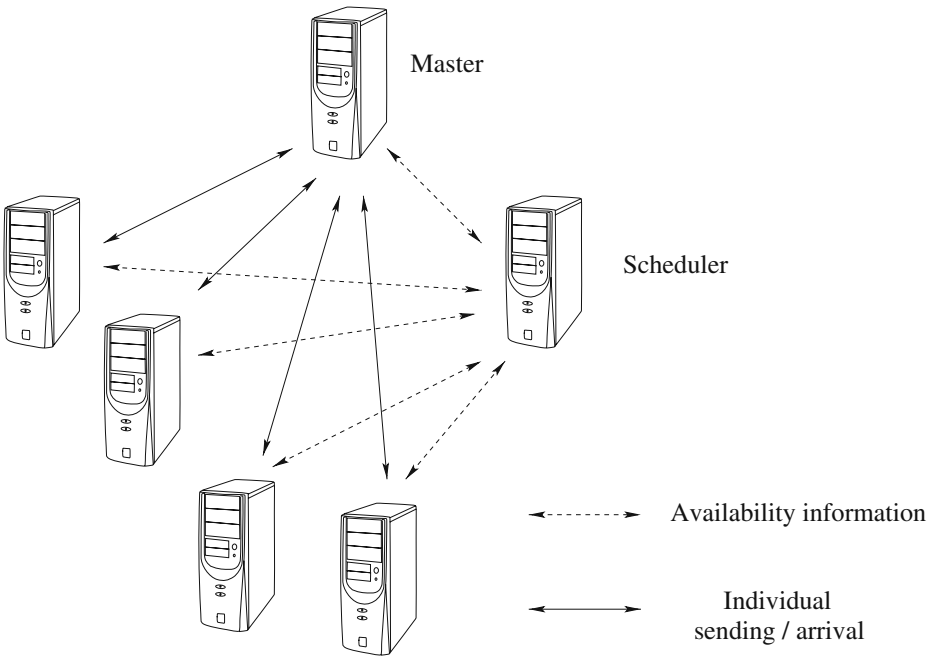


Fig. 1 Master/slave paradigm under ParadisEO-PEO. Example for four slaves (six processors are needed: four slaves, one master and one scheduler)

Table 1 Execution time in seconds (ExecTime), speed-up (S_N) and efficiency (Eff) on a cluster according to the number of slaves (Nb slaves)

Nb slaves	ExecTime	S_N	Eff
1	347	1	1
2	234	1.48	0.74
4	135	2.57	0.64
8	84	4.13	0.52
16	66	5.26	0.33
32	67	5.18	0.16
48	72	4.82	0.1
54	69	5.03	0.09

this type of mutation can be dangerous has to be applied carefully. Indeed, it can produce very deprecated solutions.

– *Intensive mutations:* due to the high variability of the energy criterion, two molecules may have a similar conformation but very different associated energies. Based on this remark, two types of mutation have been designed. These mutations extends standard mutations to decrease rapidly the energy to gain more stable complexes. Each mutation corresponds to a hill climbing algorithm based on a specific partial neighborhood. These hill climbing algorithms have been designed thanks to the Moving Object part (MO) of ParadisEO. More information about ParadisEO-MO may be found in [5]. These mutations are defined as follows:

  - ligand rotation based hill climbing: in a range of given values, new complexes are generated by ligand rotation. The search space associated to this mechanism is directly linked to the size of this interval of values (minimum and maximum angle value and step between two values in the three directions corresponding to the three dimensions). Then only a partial neighborhood is generated around an individual. The number of neighbours generated and also the cost of the evaluation function reduces the interest to make a too large neighbourhood around an individual.
  - torsion rotation based hill climbing: as the other hill climbing approach, the aim is to find a more stable conformation around an individual. For this mutation, the neighbourhood is based on the rotation of one particular torsion to find the angle that minimizes the global energy. In this case also, only a partial neighbourhood is explored to not spend too much time in applying this mutation.

Table 2 Full description and PDB identifier of the ligand/receptor complexes used

PDB means Protein Data Bank and allows to identify the instances

Ligand/receptor complexes	PDB
Ribonuclease A / Uridine-2',3'-Vanadate	6rsa
HIV-1 Protease / G26	1mbi
Thymidilate / CB3	2tsc
HIV-1 Protease / G26	1htf
Glucoamylase-471 / Alpha-d-mannose	1dog
IGA-kappa MCPC603 FAB / Phosphocholine	2mcp

Table 3 Molecular details concerning the instances used

PDB	NbAtom		DegFree	
	Ligand	Receptor	Ligand	Receptor
6rsa	31	1610	3	19
1mbi	9	2539	0	22
2tsc	56	2914	11	0
1htf	79	3127	15	4
1dog	24	3362	5	6
2mcp	25	1225	8	5

PDB means Protein Data Bank and allows to identify the instances, NbAtom corresponds to the number of atoms that compounds the instance and DegFree gives the freedom degree of the corresponding molecule. For the receptor, the DegFree value concerns the part considered as flexible (not the full molecule)

4 Parallel MOGA

The MOGA has been design under the master/slave paradigm of ParadisEO-PEO (the parallel part of ParadisEO) in order to be runnable on several processors. This paradigm, according the ParadisEO platform, is described in the Fig. 1. The main genetic algorithm runs on the master, the slaves wait for work and the scheduler knows which worker is available or not.

In our MOGA, not only the population evaluation is parallelized but also all the GA operators. An evaluation of the speed-up obtained by the parallel version of the MOGA is given in the Table 1. The speed-up and the efficiency are given by the following formulae:

$$S_N = \frac{St}{Pt} \qquad\qquad Eff = \frac{S_n}{N}$$

St is the sequential execution time and *Pt* the execution time with *N* slaves.

According to this table, it is logical that using several processors reduce the execution time of the algorithm. But a high number of processors is not necessary to keep a good efficiency. 32, 48 or 54 slaves leads to comparable results.

5 Experiments

In this section, the data preparation, the validation of the hypothesis concerning the stability criterion and the configuration of the parallel MOGA are detailed before presenting the results.

Table 4 RMSD between the seed ligand and the ligand in its optimal position for each instance

Instance	RMSD seed VS optimal (Å)
6rsa	7.15
1mbi	7.93
2tsc	13.48
1htf	14.45
1dog	10.68
2mcp	7.07

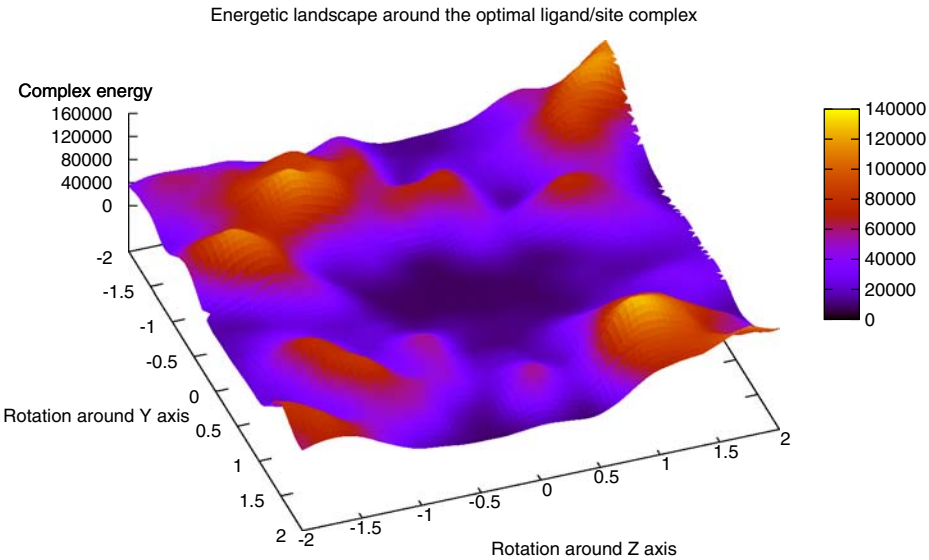


Fig. 2 Energy landscape around the crystallographic complex for the *6rsa* instance. This landscape is based of rotational modification of the ligand location according to the Y and Z axis

5.1 Data

Our simulation data are based on crystal structures taken from the RCSB Protein Data Bank (PDB). These data and the associated details are summarized in Tables 2 and 3. These complexes are taken from the clean list of the CCDC/Astex dataset. The work presented in this article is a part of global study about the multi-objectivization of the docking problem. Then, a reduce number of instances has been used. In [6], a lot of models are tested to complete the algorithm configuration.

Table 5 The eleven configurations (config) tested for the genetic algorithm: six for rigid docking (CX_R) and six for flexible docking (CX)

Config	SR	TorsRot	Reverse	SBRot	SMO	LRHC	TRHC
C1_R	X						
C1	X	X					
C2_R	X		X				
C2	X	X	X				
C3_R	X			X			
C3	X	X		X			
C4_R	X				X		
C4	X	X			X		
C5_R	X					X	
C5	X	X				X	
C6	X	X					X

SR standard rotation and translation of the ligand, *TorsRot* rotation of one torsion, *SBRot* rotation with small and big values for the rotation angle, *SMO* single mutations in one, *LRHC* ligand rotation driven by a hill climbing approach, *TRHC* torsion rotation driven by a hill climbing approach

Table 6 Statistic results for each instance according to the configuration of the algorithm

	C1	C1_R	C2	C2_R	C3	C3_R	C4	C4_R	C5	C5_R	C6
Minimum	2	2.1	2.4	2.3	2	2.1	*1.9*	2	2.2	*1.9*	2.4
First quartile	*3.2*	3.4	3.3	3.425	3.4	3.225	3.3	3.5	3.825	3.6	3.25
Median	*4.05*	4	4.5	5.75	*4.1*	*4*	4.35	*4.1*	*4.1*	4.05	4.6
Mean	*4.343*	4.4	4.729	5.293	5.126	4.686	5.169	5.129	4.733	4.964	4.776
Third quartile	5.075	4.875	6	6.75	6.3	5.875	6.325	6.15	*4.775*	7	6
Maximum	10.5	*7.7*	10.8	10	11	11	12.6	12	14	11.8	8.5
Standard deviation	1.651	*1.556*	1.937	2.078	2.524	2.183	2.865	2.576	2.29	2.42	1.794

From these complexes, two tool suites have been used in order to prepare our instances: the Vega ZZ tool[3] [4] and the UCSF Chimera tool suite[4] from the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco funded by the NIH P41 RR-01081 [2, 3].

In Table 3, the degree of freedom of the receptor corresponds to the part considered as an actor in the flexible docking i.e. the torsions considered as flexible in the process. Only lateral chains of the amino acids that are nearer than 3 Å of one amino acid of the ligand in its optimal configuration are considered. Of course, this specific limitation of the freedom degree of the receptor is applied to reduce the size of the associated search space. It is only possible with instances where the best configuration (the crystallographic one) is already known.

In order be used in our approach, the instances coming from the PDB have been prepared as follows:

- For each instance, the receptor and the ligand are separated into two different files in a specific file format: Tripos mol2. A complete description of this format can be found in the document located at this address http://tripos.com/tripos_resources/fileroot/pdfs/mol2_format2.pdf. If necessary missing hydrogen atoms are added.
- The initial position of the ligand is saved as the optimal position (to reach). Then the ligand is manually extracted from this location, thanks to the Chimera tool suite, but its position is kept in front of the receptor pocket. This new position is considered as the ligand seed for the initial population of the genetic algorithm. According to the instance, the distance, in term of RMSD, between the seed position and the optimal position is given by the Table 4. This distance is useful to compare the other methods that start with a ligand nearer of its optimal location.

5.2 Stability Criterion Validation

In order to validate the hypothesis that ligand/receptor complexes are robust to modifications of the location of the ligand into the receptor, the energy landscape around each instance has been drawn according to rotational and translational modification of the ligand position. Figure 2 shows one of these landscapes.

[3]Drug Design Lab : http://www.ddl.unimi.it

[4]http://www.cgl.ucsf.edu/chimera/

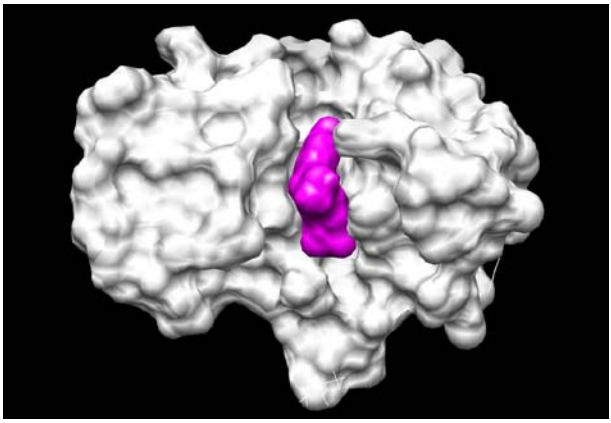


Fig. 3 Optimal complex for the 6rsa instance

According to all the obtained landscapes,[5] the optimal complex always belongs in valleys of equivalent energies for small modifications of the ligand location.[6]

5.3 Algorithm Configuration

The aim of our approach is to evaluate the impact of advanced mutations in our multi-objective model. In order to compare the results, several configurations of the algorithm have been tested. All these configurations are summarized in the Table 5. It can be noticed that the C6_R configuration does not exist. Torsion rotation based mutation can not be used in the rigid docking. The impact of each mutation has been independently evaluated with the standard mutations always activated.

Concerning the parameters of the MOGA, the population of solutions has been set to 100, the maximum number of generations was 10,000, but the algorithm may stop earlier if 500 consecutive generations without any new non-dominated solution discovered are observed. The probability of recombination and mutation has been set to 0.9 and 0.5 respectively.

5.4 Results

The Table 6 summarizes the results obtained for the all the instances according to each configuration of our genetic algorithm. All the configurations globally allows to find good docking solutions (with RMDS < 2 Å).

According to this table, the use of advanced mutations operators allows to get better results than with only standard mutations. It can be remarked that also the configuration C6 using a mutation based on a torsion rotation driven by a hill climbing does not give as good results as the C5 approach. This mutation is to restrictive in the area of the search space that is analyzed. It is difficult to find a good time/quality compromise for this specific mutation. In the C5 configuration,

[5]The graphics are available on demand

[6]The valleys are larger for translational modification based landscape

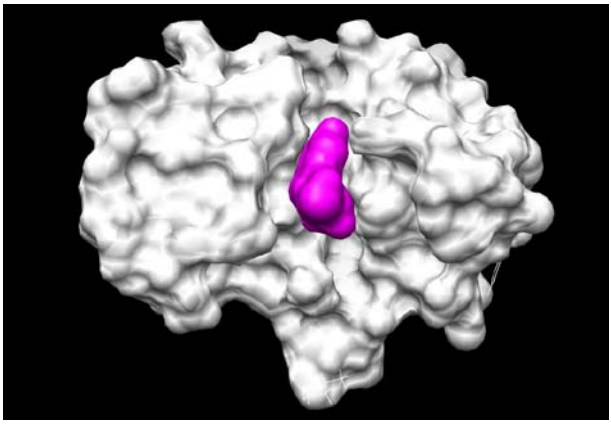


Fig. 4 Best complex found for the 6rsa instance

the hill climbing using ligand rotation explores enough search space to improve the quality of the results. Finally, the C4 configuration based on a mutation that acts like a variable neighborhood approach gives the best result. This configuration is able to obtain efficient solutions more rapidly than other mutations. This process may be made cleverer in order to improve the results.

The Figs. 3 and 4 show respectively the ligand/receptor complex with the optimal location for the ligand and best found location. These figures has been obtained thanks to the Chimera tool suite.

Comparing to the optimal location, the ligand is well enter into the receptor. However, the conformation has not been completely modified to respect the optimal one.

6 Conclusion

In this paper, a parallel multi-objective genetic algorithm has been designed for the flexible docking problem. Standard and advanced mutation operators including single-objective intensification processes based on the energy evaluation has been proposed. According to the results, the designed model is valid and the advanced mutations allows to get better results. Currently, improvement of the methods are tested on new instances by combining the best advanced mutations. Furthermore, GPU versions of our algorithm are currently designed. A multi-objectivization study of the molecular docking problem is also in progress.

Acknowledgements The model design of the docking problem has been supported by the French National Research Agency through the “Dock” project.

References

1. Boisson, J.-C., Jourdan, L., Talbi, E.-G., Horvath, D.: Parallel multi-objective algorithms for the molecular docking problem. In: IEEE 2008 Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Sun Valley Resort, Idaho, USA, 15–17 September 2008

2. Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E.: UCSF Chimera - a visualization system for exploratory research and analysis. J. Comput. Chem. **25**(13), 1605–1612 (2004)
3. Sanner, M.F., Olson, A.J., Spehner, J.C.: Reduced surface: an efficient way to compute molecular surfaces. Biopolymers **38**(3), 305–320 (1996)
4. Pedretti, A.A., Villa, L., Vistoli, G.: VEGA - an open platform to develop chemo-bio-informatics applications, using plug-in architecture and script programming. J.C.A.M.D. **18**, 167–173 (2004)
5. Boisson, J-C., Jourdan, L., Talbi, E-G.: ParadisEO-MO: A Framework to Design Single Solution Metaheuristics. INRIA Technical report, INRIA Lille Nord Europe (2008)
6. Boisson, J-C.: Modélisation et résolution par métaheuristiques coopératives : de l'atome à la séquence protéique PhD Thesis, Université des Sciences et Technologie de Lille (2008)
7. Thomsen, R.: Protein-Ligand Docking with Evolutionary Algorithms Chapter in the book entitled Computational Intelligence in Bioinformatics (2008)
8. Lee, B., Richard, F.M.: The interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. **55**, 379–400 (1971)
9. Richard, F.M.: Areas, volumes, packing and protein structure. Annu. Rev. Biophys. Bioeng. **6**, 151–176 (1977)
10. Le Grand, S.M., Merz, K.M., Jr.: Rapid approximation to molecular surface area via the use of boolean logic and look-up tables. J. Comput. Chem. **14**, 349–352 (1993)
11. Leaver-Fay, A., Butterfoss, G.L., Snoeyink, J., Kuhlman, B.: Maintaining solvent accessible surface area under rotamer substitution for protein design. J. Comput. Chem. **28**(8), 1336–1341 (2007)
12. Shrake, A., Rupley, J.A.: Environment and exposure to solvent of protein atoms. Lysozyme and insulin. J. Mol. Biol. **79**(2), 351–364 (1973)
13. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 181–197 (2002)
14. Zitzler, E., Kunzli, S.: Indicator-based selection in multiobjective search. Parallel Problem Solving from Nature, PPSN VIII, vol. 3242, pp. 382–842. Springer, Germany (2004)
15. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423. San Mateo, California, USA, (1993)
16. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
17. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, In: Proceedings of the Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), pp. 95–100 (2001)
18. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Har, W.E., Belew, R.K., Olson, A.J.: Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J. Comput. Chem. **19**, 1639–1662 (1998)
19. Ewing, T.J.A., Kuntz, I.D.: Critical evaluation of search algorithms for automated molecular docking and database screening. J. Comput. Chem. **18**(9), 1175–1189 (1997)
20. Rarey, M., Kramer, B., Lengauer, T., Klebe, G.: A fast flexible docking method using an incremental construction algorithm. J. Mol. Biol. **261**(3), 470–489 (1994)
21. Bursulaya, B.D., Totrov, M., Abagyan, R., Brooks, C.L.: Comparative study of several algorithms for flexible ligand docking. J. Comput.-Aided Mol. Des. **17**(11), 755–763 (2003)
22. Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.-G.: ParadisEO-MOEO: a framework for multiobjective optimization. In: The Proceedings of EMO'2007, pp. 457–471 (2007)
23. Cahon, S., Melab, N., Talbi, E.-G.: ParadisEO: a framework for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics **10**(3), 357–380 (2004)

European Journal of Operational Research 199 (2009) 620–629

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Hybridizing exact methods and metaheuristics: A taxonomy

L. Jourdan *, M. Basseur, E.-G. Talbi

LIFL/INRIA/CNRS, Bat M3, Cité Scientifique, 59655 Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 22 September 2006
Accepted 10 July 2007
Available online 13 April 2008

Keywords:
Taxonomy
Combinatorial optimisation
Metaheuristics
Exact methods

ABSTRACT

The interest about hybrid optimization methods has grown for the last few years. Indeed, more and more papers about cooperation between heuristics and exact techniques are published. In this paper, we propose to extend an existing taxonomy for hybrid methods involving heuristic approaches in order to consider cooperative schemes between exact methods and metaheuristics. First, we propose some natural approaches for the different schemes of cooperation encountered, and we analyse, for each model, some examples taken from the literature. Then we recall and complement the proposed grammar and provide an annotated bibliography.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

NP-hard problems are difficult to solve and no polynomial time algorithm are known for solving them. Unfortunately, most combinatorial optimization problems, such as the Travelling Salesman, N-Queens, Bin Packing, 0/1 Knapsack, Graph Partitioning, are NP-hard. Two approaches can be considered to solve this kind of problems depending on their size.

For small instances, researchers usually use exact methods. Exact methods find the optimal solution and assess its optimality. There exist numerous exact methods such as the family of Branch and X (Branch and Bound algorithm [58], Branch and Cut algorithm [42], Branch and Price algorithm [12]), Linear Programming, Dynamic Programing, etc. A branch and X algorithm uses a divide and conquer strategy to partition the solution space into subproblems and then optimizes individually each subproblem. Exact methods are known to be time expensive, so they can not be applied to large NP-hard problems or difficult ones.

When instances become too large for exact methods, heuristics and in particular metaheuristics are often used. There are two main categories of metaheuristics: single solution algorithms and population based algorithms. The first category gathers local search (LS) [54], greedy heuristic (GH) [70], simulated annealing (SA) [50], tabu search (TS) [40], Iterated Local Search (ILS) [56] etc. The second category, which is more and more studied, regroups evolutionary algorithms such as genetic algorithms [44], evolution strategies [74], genetic programming [52], and also ant colonies (AC) [31], scatter search (SS) [39], immune systems [48] etc. However, in general, metaheuristics are not able to solve the problems to optimality and some convergence problems can be encountered.

During the last years, many works have been realized on cooperative (or hybrid) optimization approaches. In many cases, best results are obtained with this kind of approaches, especially on real-life problems. At the beginning, cooperations were mainly realized between several metaheuristics. But nowadays, more and more cooperation schemes between metaheuristics and exact approaches are proposed. These strategies usually give good results because they are able to exploit simultaneously the advantages of both types of methods. For example, it may allow to give quality guarantees to the identified solutions.

In this article, we propose to survey the different cooperation between these two types of method. The fact that more and more papers deal with this kind of approaches (see Fig. 1) clearly indicates that it is an important issue for the operational research community. So it seems interesting to classify these works. A state of the art of this type of cooperation has been proposed recently by Stützle and Dumitrescu [34]. They distinguish five classes of approaches for cooperation between exact and local search methods; they also provide an example for each type. The five classes proposed are:

- Use exact algorithms to explore large neighborhoods in local search algorithms.
- Perform several runs of a local search and exploit information in high quality solutions to define smaller problems that are amenable for solution with exact algorithms.
- Exploit bounds in constructive heuristics.
- Use information from relaxations of integer programing problems to guide local search or constructive algorithms.
- Use exact algorithms for specific procedures in hybrid metaheuristics.

* Corresponding author.
E-mail addresses: laetitia.jourdan@inria.fr (L. Jourdan), basseur@lifl.fr (M. Basseur), talbi@lifl.fr (E.-G. Talbi).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.07.035

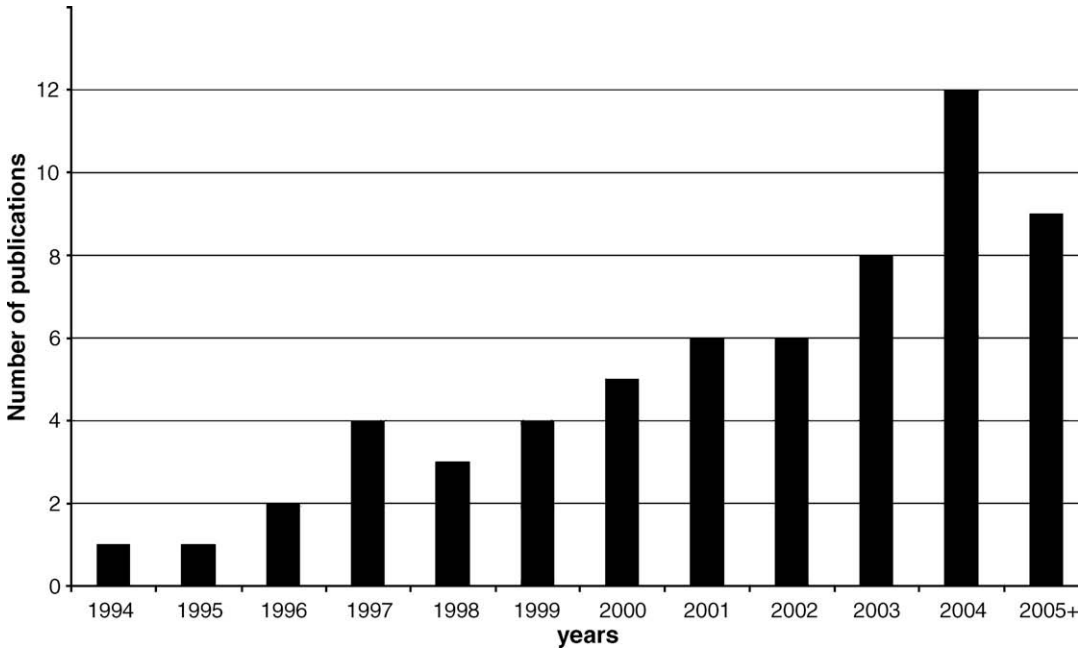


Fig. 1. The evolution of the publication activity on hybridizaton between exact methods and metaheuristics.

The survey proposed by Stützle and Dumitrescu presents several cooperative approaches to explain the different classes but would have more interest if it were generalized to every optimization methods. Their paper also excludes some combination such as preprocessing.

In [68], Puchinger and Raidl, propose a survey of the state-of-the-art approaches that combine exact methods and metaheuristics. Their survey provides a classification of methods thanks to different classes. The first one deals with collaborative combinations where no algorithm is contained in any other. This class is divided into subclasses:

- Sequential execution.
- Parallel and intertwined execution.

The second class regroups integrative combinations and is subdivided into two subclasses:

- Incorporating exact algorithms in metaheuristics.
- Incorporating metaheuristics in exact algorithms.

Puchinger and Raidl illustrate each subclass with examples issued from the literature.

In this article, we propose also to classify different articles issued from the literature but we will also propose a taxonomy of methods that combines exact and heuristic approaches. Important contributions of this article are the formal grammar proposed to classify the methods, integration of conceptual and hierarchical aspects. A separation between design and implementation is also taken into account. Our survey is far from providing an exhaustive list but it will constitute a good way for authors of new cooperation papers to classify their approach or for developers to find ideas on how to combine methods efficiently. In this article, cooperation and hybridization will be used in the same way. These terms will indicate algorithms which combine different optimization methods. The present paper uses the taxonomy proposed by Talbi [80] as we observe that it is valuable for cooperative methods between metaheuristics and exact approaches.

The remainder of the article is organized as follows. In Sections 2–4 the taxonomy used in [80] is recalled and illustrated with examples of cooperation between exact methods and metaheuristics, as in [80] the author only considers cooperation between metaheuristics. The taxonomy is divided into three general aspects: cooperation method design (Section 2), approach design (Section 3), and implementation issues (Section 4). In Section 5, the grammar for hybrid metaheuristics is extended, and an annotated review of different references is presented according to the taxonomy. Conclusions are drawn in Section 6.

2. Cooperation method design

Cooperation involves two main components: the design and the implementation. The former category concerns the cooperative algorithm itself, involving issues such as the functionality and the architecture. The implementation takes into account the hardware platform, programming model and the environment.

In this section, we will focus on the design of the cooperative mecanisms, i.e. how the methods will cooperate. For each type of classification, the derived classes are presented and some examples from the literature are described.

To facilitate the reading of our survey, the terms used in [80] are recalled. The design of metaheuristics can be classified in two types of design classification:

- Low-level/High-level
  - *Low-level*: The functional composition of a single optimization method. A given function of a metaheuristic is replaced by another method.
  - *High-level*: The different algorithms are self-contained.

- Relay/Teamwork
  - *Relay*: A set of methods is applied one after another, each using the output of the previous as its inputs, acting in a pipeline fashion.
  - Teamwork represents cooperative optimization models.

Four classes are derived from this hierarchical taxonomy (see Fig. 2).

2.1. LRH (Low-level Relay Hybrid)

This class corresponds to the algorithms in which a given method is embedded into another method; the embedded method has to be executed sequentially, i.e. the global method execution is dependant to the results obtained by the embedded method. This type of cooperation is commom when a heuristic approach is used to improve an exact approach. In the context of cooperation between metaheuristics, the most proposed approaches is to run an evolutionary algorithm then launch a local search in order to intensify the search on the best solutions. If we consider the cooperation between exact and heuristics methods, the most natural approach is to design an heuristic to improve the search strategy of the exact method (Fig. 3). For this scheme of cooperation, the heuristics will work on a problem which is from a different nature (node selection, column generation) that the considered optimization problem.

An example of this type of cooperation has been proposed by Augerat et al. [2]. In this study, a branch and cut algorithm is proposed to solve a capacitated vehicle routing problem (CVRP). The cutting plane generation is a crucial part of branch and cut algorithms. Indeed, it greatly determines their efficiency. The authors remark that the linear inequality resulting from the constraint capacities are those which provide the best cutting planes. So they propose different heuristic approaches (constructive heuristics, greedy algorithms, and Tabu search algorithms) to extract a set of violated capacity constraints of the relaxed problem.

This class of cooperation is not widely used for cooperative models between exact algorithms and heuristic approaches. Indeed, in many studies, the authors use simple (node exploration) or specific heuristics (column generation) to optimize the exact search strategy.

2.2. LTH (Low-level Teamwork Hybrid)

In this class, an element of a given method is replaced by another method. This kind of cooperation can drastically improve a metaheuristic. Oppositely to LRH cooperations, LTH consist in an embedded method which can be executed in parallel with the global method.

In the context of cooperation between metaheuristics, a well known LRH cooperative class of algorithms are memetic algorithm, i.e. genetic algorithms with a local search replacing a transformation operator, which is in many cases the mutation operator. Memetic algorithms are classified as LTH cooperation since the GA can be executed while a local search is running, applied on previously selected individuals from the GA population. Concerning the meta/exact cooperations, let us consider two main types of approaches:

- Exact search LTH cooperation (Fig. 4): the exact approach build partial solutions, which are used to define a search space for the heuristic approach. Then, the results obtained by the heuristic are analized in order to refine bounds, or column to generate in a branch & cut algorithm.
- Heuristic search LTH cooperation (Fig. 5): the heuristic search works like memetic algorithms, but in this case, the genetic operator is replaced by an exact search within a subspace of the global search space.

Cotta et al., propose a framework which lays on the cooperation between genetic algorithms and a Branch and Bound (B&B)

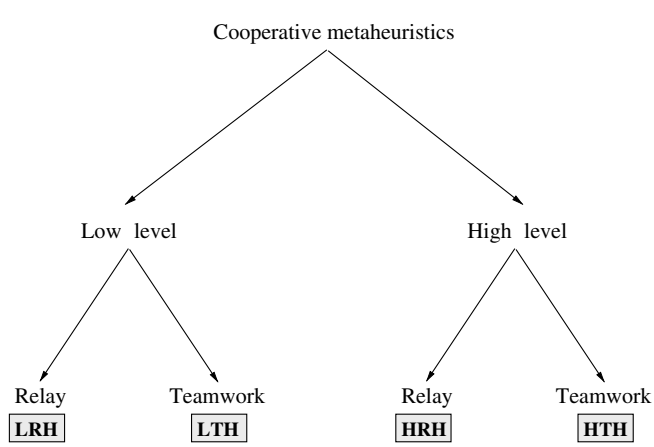


Fig. 2. The four classes derived from the cooperation method design classification: LRH, LTH, HRH, and HTH.

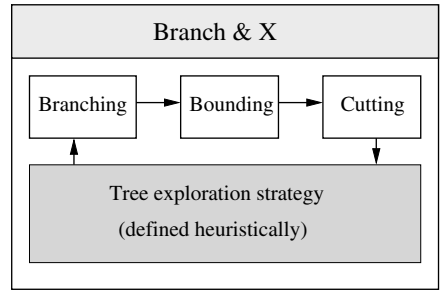


Fig. 3. LRH cooperation (exact search): Heuristic designed to define the exact search strategy (selection of the node to explore, column generation selection...). The heuristic is usually designed to help to define the exploration search strategy. (in gray).

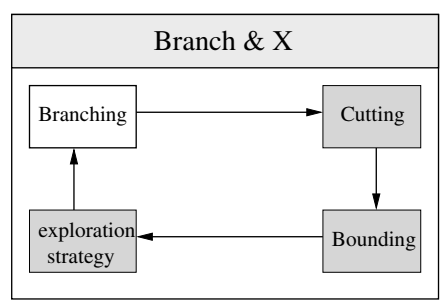


Fig. 4. LTH cooperation (exact search): Heuristic designed to explore the search space associated with a partial solution, in order to define bounds for cutting, or exploration strategy.

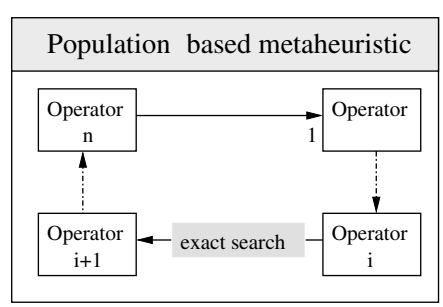


Fig. 5. LTH cooperation (heuristic search): Exact search is realized on solutions in order to intensify the search within an evolution search, with proof of the local optimality of the new solutions. The exact search could replace a genetic operator, like in memetic searches.

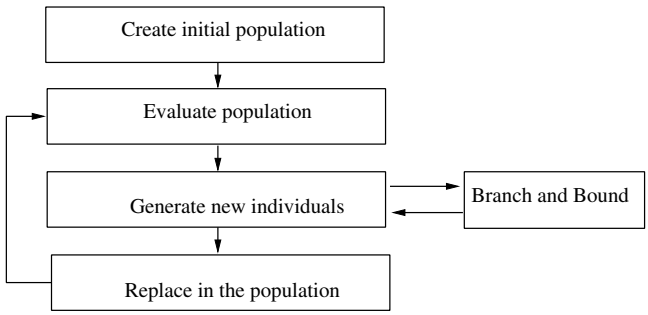


Fig. 6. Outline of the hybrid genetic algorithm of Cotta et al.: An example of Low-level TeamWork Hybrid (LTH).

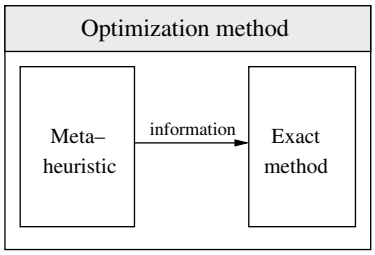


Fig. 7. HRH cooperation: some information is provided by the heuristic to the exact method. Two cases: (1) In exact search, some initial bounds/solutions are given. (2) In heuristics search, initial solutions are given to the exach algorithm which works on a subspace (partitioning, large neighborhood search).

algorithm which is used as an operator [26]. The resulting hybrid operator cleverly explores the dynastic potential (possible children) of the solutions being recombined, providing the best combination of formae (see Fig. 6).

In [49], Kostikas and Fragakis present the application of Genetic Programming (GP) in B&B based on Mixed Integer linear Programming (MIP). The hybrid architecture employs the GP as a node selection expression generator: a GP run, embedded into the B&B process, exploits the characteristics of the particular MIP problem being solved. The evolved method replaces the default one for the rest of the B&B.

Jahuira et al., propose different hybridizations between genetic algorithms and exact methods applied to the Travelling Salesman Problem (TSP) [45,46]. The cooperation is introduced in the genetic functions as the authors replace the genetic crossover by a branch and bound algorithm and a Minimal Spanning Tree solution algorithm. The initial solutions are also generated by means of the minimum spanning tree construction algorithm.

Large neighborhood search algorithms are typically *LTH* cooperations. These algorithms can be viewed as local search algorithms which use a large neighborhood to improve the efficiency of the search. The exploration of this large neighborhood can be either heuristic or exact. A survey of these methods can be found in [4]. Several studies propose exact methods to explore these large neighborhoods to find the best solution in a subspace of the global search space of the optimized problem. These types of approaches have been proposed by Bent and Van Hentenryck to solve the asymmetric TSP problem [16], or Shaw for VRP problem [78].

Globally, the frontier between LRH and LTH is thin. The difference mainly depends on the possibility, or not, to propose a parallel version of the proposed algorithm without applying drastical changes to the initial algorithm.

2.3. HRH (High-level Relay Hybrid)

In this class, the different methods are self-contained and are executed in sequence. This cooperation scheme is the most represented for general hybridization.

Of course, like in the other cooperation schemes, different types of resolution could be considered. However, in general, the most natural approach is to design a sequential execution of a metaheuristic which is launched before an exact approach (Fig. 7). The metaheuristic is designed in order to give information to the exact algorithm. If we consider an exact search, the information given could be initial bounds, for example, which helps the exact algorithm to speed up the search. If we consider a heuristic search, the metaheuristic gives initial solution(s) to the exact search, which helps to define a reduced search space to launch the exact search. For example, the search space could be reduced by defining partitions of the proposed solutions, or by defining large neighborhoods around the proposed solutions.

Klepeis et al., propose a cooperation between the alpha Branch and Bound algorithm and a conformational space annealing (CSA) algorithm for protein structure prediction [53]. The alpha branch and bound algorithm is a global optimization algorithm based on a branch and bound algorithm. It is applicable to a large class of nonlinear optimization problems that have twice differentiable functions [5]. The CSA is a stochastic method that employs elements of both simulated annealing and genetic algorithms [57].

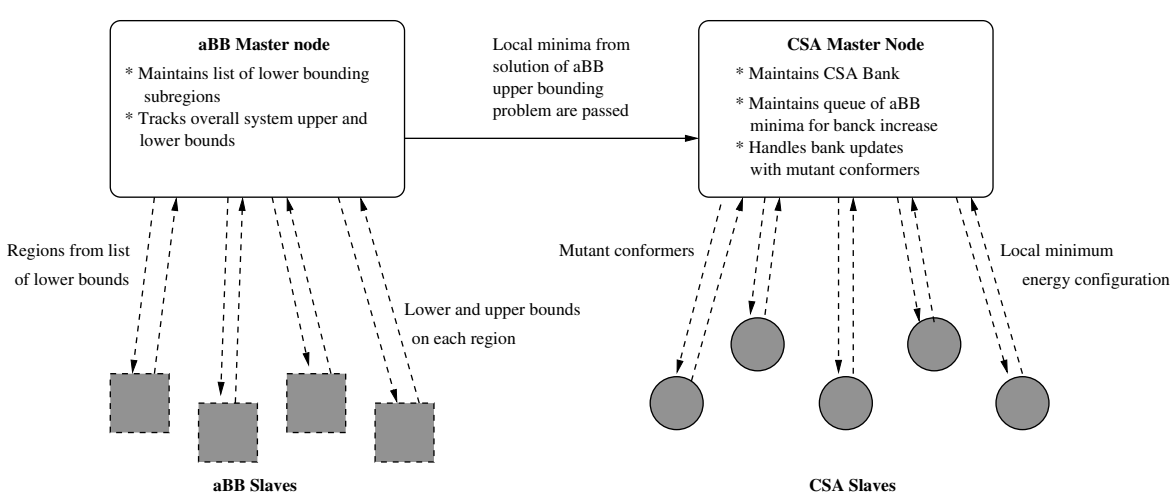


Fig. 8. An example of High-level Relay Hybrid for the protein structure prediction.

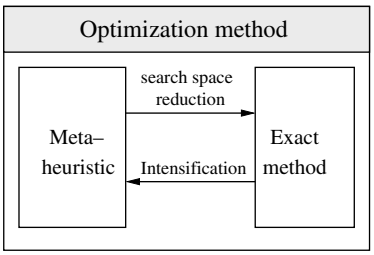


Fig. 9. HTH cooperation: two algorithms are launched in parallel, and exchange information (the metaheuristic provides information for search space reduction and the exact method provides improved solution to intensify the search).

In this algorithm, the authors alternate several runs of B& B and CSA. They parallelize their algorithm in a master–slave model (see Fig. 8).

In [16], Bent and Van Hentenryck propose a two-stage hybrid algorithm for the vehicle routing problem with time windows. First, the algorithm minimizes the number of vehicles by means of a simulated annealing algorithm. Then it minimizes the travel cost using a large neighborhood search technique which may relocate a large number of customers.

In [14], Basseur et al., investigate several cooperative approaches for a biobjective permutation flow-shop scheduling problem. These schemes are designed around a biobjective hybrid metaheuristic (Adaptive Genetic/Memetic Algorithm) and the two-phase method [84], a biobjective exact method. The main approach consists in fixing a part of initial solutions and optimizing exactly between two points of each individual. Then the process is iterated for each part of the best initial solutions proposed by the metaheuristic. Pareto fronts obtained by the Adaptive Genetic/Memetic Algorithm are strictly improved by this hybridization.

2.4. HTH (High-level Teamwork Hybrid)

This class contains algorithms where self-contained methods are performing a search in a parallel and cooperative manner.

Considering cooperation between metaheuristics, this cooperation involve principally island parallel models. It is almost the same with exact methods, with two different types of islands, those which are dedicated to exact search, and those dedicated to heuristic search. During the execution, the different algorithms exchange information, which is dependant of the type of the island (Fig. 9). The major difficulty is to set parameters (when and how the exchange is realized for example).

For example, in [62], Simulated Annealing and Branch and Bound are hybridized such that the two optimisation methods work on the same problem. The model exchanges information when conditions are satisfied. The model is parallel and is detailed in part 4.2.

In some cases, the two different cooperative approaches are not dedicated to solve instances of the same size. So, to obtain a *HTH* cooperation, the two approaches have often to solve different parts of the same problem in an independent manner. In [20], Chabrier et al. propose a cooperation between a local search and a column generation algorithm to solve VRP. The scheme of this *HTH* cooperation is shown in Fig. 10. The problem is divided into subproblems (partial model).

3. Approach design

Three criteria have been selected for the flat classification of cooperation between exact and heuristic methods: the complete resolution (exact or approximated), the resolution space (global or partial), and the nature of the cooperation (general or specialist).

3.1. Exact/approximated resolution

The type of the whole cooperative method could be either exact or heuristic. The exact cooperative approaches take useful information from heuristics to speed up the enumeration of the whole search space by upgrading bounds, finding initial solutions, defining useful cutting planes and so on.

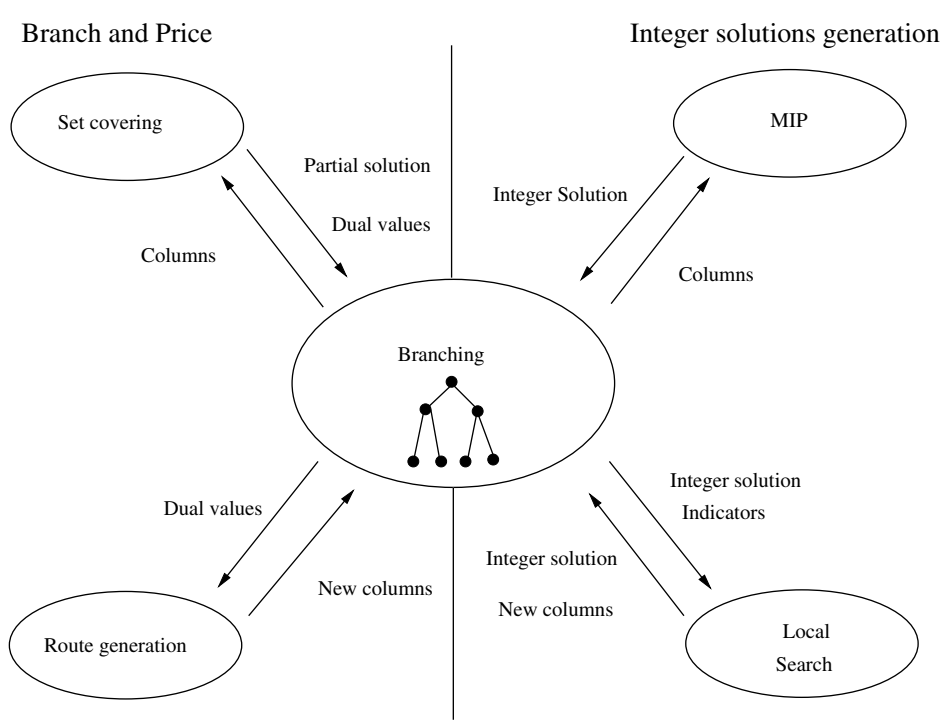


Fig. 10. An example of High-level Teamwork Hybrid scheme for the vehicle routing problem.

In the article of Chabrier et al. [20], a local search is used to generate new columns for a branch and cut algorithm. So the whole approach is exact, despite the use of a metaheuristic in the cooperative algorithm. Another way for this type of cooperation is the approach proposed by Cotta et al. [17]. The metaheuristic works on the same problem than the exact method and gives bounds to speed up the exact resolution.

Burke et al., describe a cooperative heuristic [8]. In this study, the exact approach is included in a local search mechanism to explore exactly interesting subspaces.

3.2. Global/partial cooperation

The components of the cooperative approach may work on the whole search space or only on a part. On one hand, in the global cooperation all the algorithms explore the same search space and on the other hand, in the partial cooperation, the problem is decomposed into sub-problems, each having its own search space. It may be noted that when the cooperation is a global one, the purpose is generally to solve the problem optimally, while partial cooperations are usually associated to approached solutions.

3.2.1. Partial hybrids

The majority of metaheuristic/exact cooperation approaches are often partial since the search space is generally too large for the exact method. Indeed, there are many examples of partial cooperation.

An example is the Mimausa method designed by Mautor and Michelon [60] for the quadratic assignment problem. The method builds at each iteration a subproblem and solves it by a branch and bound algorithm. Palpant et al., propose an heuristic for solving the Project scheduling with resource constraint. The heuristic integrates an exact method into a local search heuristic [65]. At each iteration, the process generates a main subproblem whose exact resolution provides a starting point for solving the remaining subproblem. Other approaches propose to solve the subproblems with an exact method and to integrate it in a metaheuristic: Budenbender for the Direct Flight Network Design Problem [11], the forget-and-extend algorithm [23] etc.

In [25], Chelouah and Siarry propose an hybrid method combining a tabu search algorithm and a simplex algorithm to deal with the global minimization of functions depending on continuous variables. The simplex algorithm is used to accelerate the convergence toward a minimum value.

In [14], branch and bound iterations are made on the potentially optimal Pareto solutions found by an hybrid evolutionary algorithm for a biobjective flowshop problem. However, only a small part of the search space is explored by the exact approach, because a large part of each individual is fixed before.

3.2.2. Global cooperation

In global cooperation, two optimization methods have to work on the same search space, usually the whole search space.

First, we consider as global cooperation exact algorithms which use a heuristic to improve bounds and then speed up an enumerative approach. In [72], Portmann et al., propose an exact approach to solve a hybrid flowshop problem. A specific heuristic computes solutions to provide an initial bound for a branch and bound algorithm. The two algorithms explore the same search space making this cooperation global.

3.3. Specialist/general

In a general hybrid model, all the algorithms solve the same optimization problem whereas the components of a specialist cooperation solve different problems. Many cooperations proposed in the literature are general, as for example the study of Cotta et al. [17].

The study of Chabrier et al. [20] is a good example of a specialist cooperation on the VRPTW (vehicle routing problem with time windows). One method solves a set covering problem and the other one a routing problem. In [83], T'kindt et al., investigate a biobjective flowshop problem with two machines. The objective are treated in a lexicographic way. The first objective, the *Makespan* is exactly solved by an exact approach (Johnson algorithm), then an ant system is applied to optimize the second objective. In many cases, specialist cooperations are used to solve problems with specific features, which can be solved exactly. Another possibility is a *LRH* approach, which optimizes heuristically (resp. exactly) a specific mechanism of an exact (resp. heuristic) method.

4. Implementation issues

The implementation choices of a cooperative algorithm can be a good way to improve the scalability of a method. Hence, an interesting point is to classify algorithms through the kind of implementation. In an optimization point of view, it is interesting to investigate the use of parallelism.

4.1. Sequential implementation

The majority of the proposed implementations are sequential. In many papers, it is suggested to parallelize the proposed cooperation in order to design better cooperation schemes and to speed up the execution. However, in these papers, the proposed approaches are not implemented, but only proposed as research perspectives. It is mainly due to the number of areas which are involved (parallelism, exact approaches, heuristics approaches...), and by the fact that the parallel models are not really natural (not like island models for genetic algorithms, for example).

4.2. Parallel implementation

Only a few studies propose parallel implementation for cooperation between exact and heuristics approaches. One very important point to determine is what kind of information is exchanged and when. In parallel implementations several architectures can be encountered: Multiprocessor, cluster or grid architecture. In metaheuristic and exact method cooperation the majority of the work use small cluster implementation. In [17], Cotta et al., propose a parallel implementation of their hybrid GA/B&B approach. GAs are executed in a parallel way, and a processor is dedicated to launch B&B resolutions on restricted problems. This processor allows to give new individuals to the GAs.

In [62], Simulated Annealing and Branch and Bound are hybridized in a parallel manner. The model uses two processors. On the first one, the simulated annealing is running and on the second one the Branch and Bound. Bounds and variable decisions are exchanged. Upper bounds obtained from SA are immediately passed onto the B&B code. If these bounds are better than B&B incumbent bounds, the B&B execution updates its current best bound value. Moreover, any integer bound obtained by the B&B execution is passed on to the SA code and used as an alternative reheated solution. Secondly, variable choice information is exchanged, once both SA and B&B have calculated their respective variable ranking and pseudocosts information, a single variable choice list is constructed by averaging the ranking positions from both strategies.

For future direction, it is important to notice that more and more frameworks allow to facilitate the parallelization of cooperation. To this purpose, we present several frameworks that allow to realize cooperation between metaheuristics and exact methods.

4.3. Frameworks

4.3.1. A-Teams

An asynchronous team (A-Team) is a strongly cyclic computational network [81]. The number of agents can be arbitrarily large and the agents may be distributed over an arbitrarily wide area. Agents cooperate by working on one another's results. Each agent is completely autonomous (it decides which results it is going to work on and when).

4.3.2. TECHS

TECHS (TEams for Cooperative Heterogeneous Search) allow the use of very different search agents within a search team while still achieving synergetic effects [32]. An agent could be either a metaheuristic or an exact method.

4.3.3. MALLBA and TRACER

n MALLBA (university of MAlaga, university of La Laguna, university of BArcelona), each optimization method is encapsulated in a skeleton [1]. Different exact methods are proposed (Branch and Bound, Divide and Conquer, Dynamic Programming) and also heuristic methods. MALLBA is one of the few frameworks that proposes the possibility to directly develop parallel hybrid methods between exact and heuristic methods. The parallelism is available for LAN and WAN computer platforms. TRACER is the following project of MALLBA.

Since many cooperation scheme could be designed to cooperate exact methods with metaheuristics, frameworks are useful in order to reuse mechanisms proposed by the literature. However, there exists only first experiments in this area since this research area is very recent.

5. Global overview

As we have extended the taxonomy [80] to the case of cooperation between metaheuristics and exact methods, we recall the grammar for cooperation schemes. In our case, we consider only heterogeneous models, since metaheuristics and exact methods are definitely different!

For the heuristics methods, we use these following abbreviations:

- SA: *Simulated Annealing*
- GA: *Genetic Algorithm*
- MA: *Memetic Algorithm*
- ES: *Evolution Strategy*
- GP: *Genetic Programming*
- NN: *Neural Network*
- DW: *Descent Walk*
- LS: *Local Search*
- ILS: *Iterated Local Search*
- TS: *Tabu Search*

<*cooperative-algorithm* >	⟶	<*design-issues*><*implementation-issue*>
<*design-issues*>	⟶	<*hierarchical*><*flat*>
<*hierarchical*>	⟶	<*LRH*> | <*LTH*> |
		<*HRH*> | <*HTH*>
<*LRH*>	⟶	LRH (<*metaheuristic*>(<*exact*>))|
		(<*exact*>(<*metaheuristic*>))
<*HRH*>	⟶	HRH (<*metaheuristic*>+ <*exact*>|
		<*exact*>+ <*metaheuristic*>)
<*LTH*>	⟶	LTH (<*metaheuristic*> (<*exact*>))|
		(<*exact*>(<*metaheuristic*>))
<*HTH*>	⟶	HTH (<*metaheuristic*>+ <*exact*>)|
		<*exact*>+ <*metaheuristic*>)
<*flat*>	⟶	(<*resolution*>, <*optimization*>, <*function*>)
<*resolution*>	⟶	Exact | Approximate
<*optimization*>	⟶	global | partial
<*function*>	⟶	general | specialist
<*implementation-issue*>	⟶	sequential | parallel | <*scheduling*>
<*scheduling*>	⟶	static | dynamic | adaptive
<*metaheuristic*>	⟶	SA | GA | MA | ES | GP | NN | DW | LS |
		ILS | TS | GH | AF | SS | CSA | SH | ... |
		<*cooperative-algorithm*>
<*exact*>	⟶	B&B | *alpha*B&B | B&C | B&P | ... |
		B&C&P | LP | MLP | DP | Splx | SG | SM |
		<*cooperative-algorithm*>

Fig. 11. The proposed grammar for cooperation between metaheuristics and exact methods.

- GH: *Greedy Heuristic*
- AC: *Ant Colonies*
- SS: *Scatter Search*
- CSA: *Conformational Space Annealing*
- SH: Specific Heuristic

For the exact methods, the abbreviations used are:

- B&B: Branch and Bound
- *alpha*B&B: the alpha Branch and Bound
- B&C: Branch and Cut
- B&C&P: Branch and Cut and Price
- BS: Beam Search (Breadth first search process without backtracking)
- LP: Linear Programming (as CPlex, etc.)
- DP: Dynamic Programming
- CP: Constraint Programming
- Splx: Simplex Search
- SG: Search Goal
- SES: Specific exact search

Table 1
Annotated bibliography

Refs.	Design	Optimization problem
[3]	LTH (DP(ILS)) (approximate, partial, general) sequential	TSP
[6]	HRH(B&P + SH) (approximate, partial, general) sequential	Large-scale set partitioning
[14]	HRH(HTH(GA + MA),B&B) (approximate, partial, general) sequential	Biobjective M-machines Flow-shop problem
[13]	HRH(HRH(GH + LS) + B&C) (exact, global, general) sequential	VRP with time windows
[10]	HRH(TS + LP) (approximate, partial, general) sequential	Irregular stock cutting
[16]	HRH(SA + LTH (LS(B&B))) (approximate, partial, general) sequential	VRP with time windows
[11]	HRH(GH + LTH (TS(B&B))) (approximate, partial, general) sequential	Direct flight network design
[8]	LRH(LS + DP) (approximate, partial, general) sequential	Asymmetric TSP
[17]	LTH(GA(B&B)) (approximate, partial, general) parallel static	TSP
[24]	LTH(LS(DP)) (approximate, partial, general) sequential	One machine Flow-shop (weighted sum of tardiness)
[19]	HRH(RS + LP) (approximate, partial, general) sequential	Fiber-optic cable manufacturing (scheduling)
[20]	HTH(B&P(HRH(MLP+LS))) (exact, global, specialist) sequential	VRP
[25]	HRH(TS + Splx) (approximate, partial, general) sequential	Multiminima continuous functions
[26]	LTH(GA(B&B)) (approximate, partial, general) sequential	Generalized Schwefel function, rule base learning
[28]	LTH(TS(B&B)) (approximate, partial, general) sequential	VRP
[33]	HTH(AS(HRH(B&B + AC))) (approximate, partial, specialist) sequential	Local access network design
[35]	HRH(LP + GA) (approximate, partial, general) sequential	Generalized assignment
[36]	LRH(B&B + GA) (exact, global, general) sequential	Max-SAT
[45]	LRH(MST + LTH (GA(B&B—ACM))) (approximate, global, general) sequential	TSP
[46]	LRH(MST + LTH (GA(ACM))) (approximate, global, general) sequential	TSP
[51]	HRH(LP + MA) (approximate, partial, general) sequential	Prize-collecting steiner tree
[53]	HRH(αB&B + CSA) (approximate, partial, specialist) parallel static	Protein structure prediction
[47]	LTH (GA(Splx)) (approximate, partial, general) sequential	Gene regulatory network models
[49]	LTH (MLP(GP)) (exact, global, specialist) sequential	MIPLIB3 benchmarks
[59]	LRH(AS + LP) (approximate, global, specialist) sequential	Quadratic assignment
[60]	HRH(B&B + TS) (approximate,partial,general) sequential	Quadratic assignment
[65]	HTH(LS(CP)) (approximate, partial,general) sequential	Project scheduling with resource constraints
[69]	HRH(DW + B&B) (exact, global,general) sequential	Quadratic assignment
[66]	HTH(LS(CP)) (approximate, partial,general) sequential	TSP
[72]	HRH(SH + (HTH(B&B(GA)))) (exact, global, general) sequential	Hybrid flow-shop
[76]	HRH(SH + B&B) (approximate, partial, general) sequential	*p-median* problem
[78]	LTH (LS(B&B)) (approximate, partial, general) sequential	VRP
[79]	HRH(B&B + NN) (approximate, partial, general) sequential	Biobjective broadcast scheduling problem
[83]	HRH(SES + GA) (approximate, partial, specialist) sequential	Biobjective 2 machines flow-shop problem
[85]	LRH(ILS + Splx) (approximate, global, specialist) sequential	One-dimensional cutting stock variant (pattern restricted problem)
[87]	HRH(AS + RT) (approximate, partial, general) sequential	0–1 multi-dimensional knapsack
[75]	LTH(LP(GA)) (exact, partial, general) sequential	graph coloring
[55]	HRH(GA + MIP) (exact, partial, general) sequential	Markov Decision Processes
[61]	LTH(GA(SES)) (approximate, partial, specialist) sequential	graph colouring
[63]	HRH(B&B + GA) (approximate, partial, specialist) sequential	Flowshop
[71]	HRH(IP + SS) (approximate, partial, specialist) sequential	0–1 programming
[67]	LTH(B&C&P(GA)) (approximate, partial, general) sequential	2D bin packing
[82]	HRH(GA + IP) (exact, partial, general) sequential	Jobshop
[37]	HTH(B&B + EA) approximate, partial, general sequential	Multidimensional Knapsack
[29]	HRH(IPL + AC) (approximate, partial, specialist) sequential	Project portfolio selection
[41]	LTH(SA(LP)) (approximate, partial, general) sequential	Irregular strip packing
[38]	LTH(SS(Splx)) (approximate, partial, general) sequential	Bicriteria 0,1-knapsack
[27]	HTH(GA + LP) (approximate, partial, general) sequential	Scheduling in power systems
[73]	HTH (B&B + GA) (exact, global, general) sequential	Hybrid flowshops
[62]	HTH(SA + B&B) (approximate, global, general) parallel	Zero-one LP
[77]	HRH(Cplex + GA) (approximate, partial, general) sequential	Single line scheduling
[18]	HRH(LR + TS) (approximate, global, general) sequential	Single source capacitated location problem
[7]	HRH(BS + TS) (approximate, partial, general) sequential	One-dimensional bin packing problem
[9]	LTH(GRASP(OCTANE)) (approximate, partial, specialist) sequential	Multiconstraint knapsack problem
[21]	LTH(TS(LP)) (approximate, partial, specialist) sequential	Capacitated network design
[15]	LTH(AC(BS)) (approximate, global, general) sequential	open shop scheduling
[22]	LTH(LS(SES)) (approximate, partial, general) sequential	Routing
[30]	LTH(BS(LS)) (approximate, partial, general) sequential	2-machine flow shop, uncapacitated p-median location problem
[64]	HRH(LP + GRASP) (approximate, partial, specialist) sequential	Weighted maximal planar graph problem
[86]	LTH (LR + TS) (approximate, partial, general) sequential	Workshift and rest assignment of nursing personnel
[43]	LTH (LS(B&B)) (approximate, partial, general) sequential	Flow shop problem

The grammar is proposed in Fig. 11 and is particularly adapted to the design of cooperative schemes where it is interesting to know which method uses the other one. So the order in the description of a method has an importance. For example [17] is a Low Level Co-Evolutionary Hybrid scheme denoted by *LTH* (GA (B&B)) for the Design and this means that the B&B is used as an operator by the genetic algorithm. So for *LTH* scheme the first method uses the second in its design.

For the relay model, the order used indicates the order of the different methods used by the authors. For example in [14], the authors use in one of their models a *LRH*(GA + exact) which means that they apply an exact method on the solutions of a genetic algorithm in order to have the exact Pareto front in a multi-objective flowshop problem.

In Table 1, we propose a annoted survey of the literature, with for each reference the corresponding classification and grammar and the optimization problem solved.

6. Conclusion

The interest for hybrid metaheuristics is still growing as we can observe that several workshops and conferences are dealing with it: Hybrid Metaheuristics (HM 2005, 2006) workshop of ECAI, Conference on Hybrid Intelligent Systems (HIS 2005, 2006), Application of Hybrid Evolutionary Algorithms to NP-complete problems workshop of GECCO, etc.

In this paper, we have presented an extension of the work of Talbi [80] to cooperation between exact methods and metaheuristics. The objective was to offer a short review of the literature on the subject but also the paper had the purpose to show how the methods can be combined and how to describe such cooperations in order to favorize discussion in future papers on cooperation.

With this classification, we remark that several ways of cooperation can be explored. Moreover, only a few parallel implementation are realized for cooperative approaches between exact and heuristic approaches. Parallelism could be a good way to improve algorithms efficiency, as in many cases for hybrid metaheuristics [80]. A second important point is that most of the found cooperative algorithms involving on exact scheme are for mono-objective problems. The major difficulty for applying such cooperation to multi-objective problems is that there are few exact methods that can treat the specificities of multi-objective optimization [14].

In order to solve these difficulties, the use of frameworks seems to be a promizing way, if the user is not expert in all the topics involved in these cooperations. For example, some heuristics could be designed by an expert of this area, which also uses a framework in order to solve subproblems. The frameworks which could be involved concerns exact search, heuristics search, parallelism, or multi-objective optimization. It will be also very interesting to propose cooperative frameworks, which try to unify the different types of existing framework, in order to allow the user to define complex cooperative models in a few effort.

References

[1] E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Dfaz, I. Dorta, J. Abarr, J. Gonzlez, C. Len, L. Moreno, J. Petit, J. Roda, A. Rojas, F. Xhafa, Mallba: A library of skeletons for combinatorial optimisation, in: Proceedings of the Euro-Par, volume LNCS 2004, 2002, pp. 927–932.

[2] P. Augerat, J.M. Belenguer, E. Benavent, A. Corber'n, D. Naddef, Separating capacity constraints in the CVRP using tabu search, European Journal of Operational Research 106 (2–3) (1998) 546–557.

[3] D. Applegate, R. Bixby, V. Chvátal, W. Cook, Finding tours in the TSP. Technical Report 99885, Forschungsinstitut fnr Diskrete Mathematik, Universität Bonn, Germany, 1999.

[4] R.K. Ahuja, O. Ergun, J.B. Orlin, A.P. Punnen, A survey of very large-scale neighborhood search techniques, Discrete Applied Mathematics 123 (1-3) (2002) 75–102.

[5] I. Androulakis, C. Maranas, C. Floudas, ffbb: A global optimization method for general constrained nonconvex problems, Journal of Global Optimization 7 (1995) 337–363.

[6] A. Atamturk, G.L. Nemhauser, M.W.P. Savelsbergh, A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems, Journal of Heuristics 1 (1996) 247–259.

[7] A.C.F. Alvim, C. Ribeiro, F. Glover, D.J. Aloise, A hybrid improvement heuristic for the one-dimensional bin packing problem, Journal of Heuristics 10 (2) (2004) 205–229.

[8] E.K. Burke, P.I. Cowling, R. Keuthen, Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem, in: Applications of Evolutionary Computing. EvoWorkshops2001, Lecture Note in Computer Science, vol. 2037, Springer-Verlag, Como, Italy, 2001, pp. 203–212.

[9] M. Barake, P. Chardaire, G.P. McKeown, Metaheuristics: Computer decision-making, The Probe Metaheuristic and its Application to the Multiconstraint Knapsack Problem, Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[10] J.A. Bennell, K.A. Dowsland, Hybridising tabu search with optimisation techniques for irregular stock cutting, Management Science 47 (8) (2001) 1160–1172.

[11] K. Büdenbender, T. Grünert, H-J. Sebastian, A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem, Transportation Science 34 (4) (2000) 364–380.

[12] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: Column generation for solving huge integer programs, Operations Research (1998) 316–329.

[13] J.F. Bard, G. Kontoravdis, G. Yu, A branch-and-cut procedure for the vehicle routing problem with time windows, Transportation Science 36 (2002) 250–269.

[14] M. Basseur, J. Lemesre, C. Dhaenens, E-G. Talbi, Cooperation between branch and bound and evolutionary approaches to solve a biobjective flow shop problem, Workshop on Evolutionary Algorithms (WEA'04), vol. 3059, Springer-Verlag, 2004, pp. 72–86.

[15] C. Blum, Beam-aco – hybridizing ant colony optimization with beam search: An application to open shop scheduling, Computers and OR 32 (2005) 1565–1591.

[16] R. Bent, 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.

[17] C. Cotta, J.F. Aldana, A.J. Nebro, J.M. Troya, Hybridizing genetic algorithms with branch and bound techniques for the resolution of the tsp, in: D.W. Pearson, N.C. Steele, R.F. Albrecht (Eds.), Artificial Neural Nets and Genetic Algorithms 2, Springer-Verlag, 1995, pp. 277–280.

[18] M.J. Cortinhal, M.E. Captivo, Upper and lower bounds for the single source capacitated location problem, EJOR 151 (2) (2003) 333–351.

[19] D. Clements, J. Crawford, D. Joslin, G. Nemhauser, M. Puttlitz, M. Savelsbergh, Heuristic optimization: a hybrid AI/OR approach, in: Workshop on Industrial Constraint-Directed Scheduling, Schloss Hagenberg, Austria, 1997.

[20] A. Chabrier, E. Danna, C. Le Pape, Coopération entre génération de colonnes sans cycle et recherche locale appliquée au routage de véhicules, in: JNPC, 2002.

[21] T.G. Crainic, M. Gendreau, J.M. Farvolden, A simplex-based tabu search method for capacitated network design, INFORMS Journal on Computing 12 (3) (2000) 223–236.

[22] P.I. Cowling, Ralf Keuthen, Embedded local search approaches for routing optimization, Computers and OR 32 (2005) 465–490.

[23] Y. Caseau, F. Laburthe, Effective forget-and-extend heuristics for scheduling problems, in: CP-AI-OR'02 – Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 1999, Ferrara (Italy).

[24] R.K. Congram, C.N. Potts, S.L. van de Velde, An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem, INFORMS Journal on Computing 14 (1) (2002) 52–67.

[25] R. Chelouah, P. Siarry, A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multiminima functions, European Journal of Operational Research 161 (3) (2005) 636–654.

[26] C. Cotta, J.M. Troya, Embedding branch and bound within evolutionary algorithms, Applied Intelligence 18 (2003) 137–153.

[27] K.P. Dahal, C.J. Aldridge, S.J. Galloway, Evolutionary hybrid approaches for generation scheduling in power systems, European Journal of Operational Research (2006) [Available online, 25 January 2006].

[28] B. De Backer, V. Furnon, P. Kilby, P. Prosser, P. Shaw, Solving vehicle routing problems using constraint programming and metaheuristics, Journal of Heuristics 6 (4) (2000) 501–524.

[29] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection, European Journal of Operational Research 171 (3) (2006) 830–841.

[30] F. Della Croce, M. Ghirardi, R. Tadei, Recovering beam search: Enhancing the beam search approach for combinatorial optimization problems, Journal of Heuristics 10 (1) (2004) 89–104.

[31] M. Dorigo, V. Maniezzo, A. Colorni, Positive feedback as a search strategy. Technical Report No. 91016, Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie, 1991.

[32] J. Denzinger, T. Offermann, On cooperation between evolutionary algorithms and other search paradigms, in: CEC-99, 1999, pp. 2317–2324.

[33] C. Duhamel, A. Quilliot, Coupling a metaheuristic with an exact method for the local access network design problem, In Optimization 2001, 2001.
[34] I. Dumitrescu, T. Stützle, Combinations of local search and exact algorithms, in: EvoWorkshops 2003, 2003, pp. 211–223.
[35] H. Feltl, G.R. Raidl, An improved hybrid genetic algorithm for the generalized assignment problem, Symposium on Applied Computing, SAC'04, ACM Press, 2004, pp. 990–995.
[36] A.P. French, A.C. Robinson, J.M. Wilson, Using a hybrid genetic-algorithm/branch and bound approach to solve feasibility and optimization integer programming problems, Journal of Heuristics 7 (6) (2001) 551–564.
[37] J.E. Gallardo, C. Cotta, A.J. Fernández, Solving the multidimensional knapsack problem using an evolutionary algorithm hybridized with branch and bound, in: J. Mira, J.R. Álvarez (Eds.), IWINAC (2), Lecture Notes in Computer Science, vol. 3562, Springer, 2005, pp. 21–30.
[38] C. Gomes da Silva, J. Figueira, J. Climaco, Integrating partial optimization with scatter search for solving bi-criteria 0,1-knapsack problems, European Journal of Operational Research (2006) (Available online, 17 February 2006).
[39] F. Glover, Heuristics for integer programming using surrogate constraints, Decision Sciences 8 (1) (1977) 56–166.
[40] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 13 (1986) 533–549.
[41] A.M. Gomes, J.F. Oliveira, Solving irregular strip packing problems by hybridising simulated annealing and linear programming, European Journal of Operational Research 171 (3) (2006) 811–829.
[42] R. Gomory, Outline of an algorithm for integer solutions to linear programs, Bulletin of the American Mathematical Society 64 (1958) 275.
[43] M. Haouari, T. Ladhari, A branch-and-bound-based local search method for the flow shop problem, Journal of the Operational Research Society 54 (10) (2003) 1076–1084.
[44] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[45] C.A.R. Jahuira, Hybrid genetic algorithm with exact techniques applied to tsp, in: Second International Workshop on Intelligent Systems Design and Application, Dynamic Publishers, 2002, pp. 119–124.
[46] C.A.R. Jahuira, E. Cuadros-Vargas. Solving the tsp by mixing gas with minimal spanning tree, in: Ernesto Cuadros-Vargas, Eduardo Tejada-Gamero, Adenilso da Silva Simao, (Eds.), First International Conference of the Peruvian Computer Society, Lima – Perù, Peruvian Computer Society, January 2003, pp. 123–123.
[47] P. Koduru, S. Das, S. Welch, J.L. Roe, A multi-objective ga-simplex hybrid approach for gene regulatory network models, in: Congress on Evolutionary Computation (CEC'04), vol. 2, IEEE Service Center, Portland, Oregon, USA, June 2004, pp. 2084–2091.
[48] J.O. Kephart, A biologically inspired immune system for computers, in: Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Cambridge, MA, US, MIT Press, 1994, pp. 130–139.
[49] K. Kostikas, C. Fragakis, Genetic programming applied to mixed integer programming, in: EuroGP 2004, Lecture Notes in Computer Science, vol. 3003, 2004, pp. 113–124.
[50] S. Kirkpatrick, D.C. Gelatt, M.P. Vechhi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[51] G.W. Klau, I. Ljubic, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, R. Weiskircher, Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem, in: K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, P. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, A. Tyrrell, (Eds.), Lecture Note in Computer Science, Proceedings of the 2004 Genetic and Evolutionary Computation Conference, vol. 3102, Seattle, Washington, USA, June 2004, pp. 1304–1315.
[52] J.R. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge, MA, 1992.
[53] J.L. Klepeis, M.J. Pieja, C.A. Floudas, Hybrid global optimization algorithms for protein structure prediction: Alternating hybrids, Biophysical Journal 4 (84) (2003) 869–882.
[54] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Reinhart, and Winston, New York, 1976.
[55] Z-Z. Lin, J.C. Bean, C.C. White III, A hybrid genetic/optimization algorithm for finite-horizon, partially observed markov decision processes, INFORMS Journal on Computing 16 (1) (2004) 27–38.
[56] H.R. Lourenço, O. Martin, T. Stützle, Handbook of Metaheuristics, Iterated Local Search, Kluwer Academic Publishers, Norwell, MA, 2002.
[57] J. Lee, H. Scheraga, S. Rackovsky, New optimization method for conformational energy calculations on polypeptides: Conformational space annealing, Journal of Computational Chemistry 18 (1997) 1222–1232.
[58] E.L. Lawler, D.E. Wood, Branch and bound methods: A survey, Operations Research 14 (1966) 699–719.
[59] V. Maniezzo, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, INFORM Journal on Computing 11 (4) (1999) 358–369.
[60] T. Mautor, P. Michelon, Mimausa: A new hybrid method combining exact solution and local search, in: Second International Conference on Meta-Heuristics (MIC'97), 1997.
[61] A. Marino, A. Prugel-Bennett, C. Glass, Improving graph colouring with linear programming and genetic algorithms, in: K. Miettinen, M.M. Makela, J. Toivanen (Eds.), EUROGEN99, Jyvaskyla, Finland, 1999, pp. 113–118.
[62] V. Nwana, K. Darby-Dowman, G. Mitra, A co-operative parallel heuristic for mixed zero-one linear programming: Combining simulated annealing with branch and bound, European Journal of Operational Research 164 (2005) 12–23.
[63] A. Nagar, S.S. Heragu, J. Haddock, A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem, Annals of Operations Research 63 (1996) 397–414.
[64] I.H. Osman, M. Hasan, A. Abdullah, Lp-based meta-heuristics for the weighted maximal planar graph problem, Journal of Operational Research Society 53 (10) (2002) 1142–1149.
[65] M. Palpant, C. Artigues, P. Michelon, LSSPER: Solving the resource-constrained project scheduling problem with large neighbourhood search, Annals of Operations Research 31 (1) (2004) 237–257.
[66] G. Pesant, P. Gendreau, A constraint programming framework for local search methods, Journal of Heuristics 5 (3) (1999) 255–279.
[67] J. Puchinger, G.R. Raidl, An evolutionary algorithm for column generation in integer programming: An effective approach for 2d bin packing, In PPSN VIII, 2004, pp. 642–651.
[68] J. Puchinger, G.R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification, in: IWINAC, pp. 41–53, 2005.
[69] P.M. Pardalos, K.G. Ramakrishnan, M.G.C. Resende, 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 (1997) 280–294.
[70] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, 1982.
[71] A. Plateau, D. Tachat, P. Tolla, A hybrid search combining interior point methods and metaheuristics for 0–1 programming, International Transactions in Operational Research 9 (6) (2002) 731–746.
[72] M.C. Portmann, A. Vignier, D. Dardihac, D. Dezalay, Branch and bound crossed with ga to solve hybrid flowshops, European Journal of Operational Research 107 (2) (1998) 389–400.
[73] M.C. Portmann, A. Vignier, D. Dardilhac, D. Dezalay, Branch and bound crossed with ga to solve hybrid flowshops, European Journal of Operational Research 107 (1998) 389–400.
[74] I. Rechenberg, Evolutions Strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973.
[75] G. Ribeiro Filho, L.A.N. Lorena, Constructive genetic algorithm and column generation: An application to graph coloring, in: APORS 2000 – The Fifth Conference of the Association of Asian-Pacific Operations Research Societies within IFORS, 2000.
[76] C. Rosing, C. ReVelle, Heuristic concentration: Two stage solution construction, European Journal of Operational Research 97 (1997) 75–86.
[77] R. Spina, L.M. Galantucci, M. Dassisti, A hybrid approach to the single line scheduling problem with multiple products and sequence-dependent time, Computers and Industrial Engineering 45 (4) (2003) 573–583.
[78] 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, Lecture Notes in Computer Science, vol. 1520, Springer-Verlag, Pisa, Italy, 1998, pp. 417–431.
[79] H. Shi, 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, IOS Press, 2003, pp. 42–49.
[80] E-G. Talbi, A taxonomy of hybrid metaheuristics, Journal of Heuristics 8 (2) (2002) 541–564.
[81] S. Talukdar, L. Baerentzen, A. Gove, P.S. de Souza, Asynchronous teams: Cooperation schemes for autonomous agents, Journal of Heuristics 4 (4) (1998) 295–321.
[82] H. Tamura, A. Hirahara, I. Hatono, M. Umano, An approximate solution method for combinatorial optimization – a hybrid approach of genetic algorithm and Lagrange relaxation method, Transactions of the Society of Instrument and Control Engineers (Japan) 30 (3) (1994) 329–336.
[83] V. T'kindt, N. Monmarché, F. Tercinet, D. Laügt, An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem, European Journal of Operational Research 142 (2002) 250–257.
[84] E.L. Ulungu, J. Teghem, The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems, Foundations of Computing and Decision Sciences 20 (2) (1995) 49–165.
[85] S. Umetani, M. Yagiura, 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, Valencia Spain, August 2004, pp. 101–114.
[86] C. Valouxis, E. Housos, Hybrid optimization techniques for the workshift and rest assignment of nursing personnel, Artificial Intelligence in Medicine 20 (2) (2000) 55–175.
[87] M. Vasquez, J.-K. Hao, A hybrid approach for the 0/1 multidimensional knapsack problem, in: IJCAI, 2001, pp. 328–333.

Using Datamining Techniques to Help Metaheuristics: A Short Survey

Laetitia Jourdan[1], Clarisse Dhaenens[2], and El-Ghazali Talbi[1,2]

[1] INRIA Futurs, Bât M3, Cité Scientifique 59655 Villeneuve d'Ascq France
[2] LIFL, CNRS, Université de Lille I, 59655 Villeneuve d'Ascq France
{jourdan, dhaenens, talbi}@lifl.fr

Abstract. Hybridizing metaheuristic approaches becomes a common way to improve the efficiency of optimization methods. Many hybridizations deal with the combination of several optimization methods. In this paper we are interested in another type of hybridization, where datamining approaches are combined within an optimization process. Hence, we propose to study the interest of combining metaheuristics and datamining through a short survey that enumerates the different opportunities of such combinations based on literature examples.

1 Introduction

Hybrid metaheuristics are more and more studied and a first taxonomy has been proposed in [44]. Many works propose to combine two or more metaheuristics, but other works present also hybridizations between exact optimization methods and metaheuristics. Another promising approach to hybridization is to use datamining techniques to improve metaheuristics. Datamining (DM), also known as Knowledge-Discovery in Databases (KDD), is the process of automatically exploring large volumes of data e.g., instances described according to several attributes, to discover patterns. In order to achieve this, datamining uses computational techniques from statistics, machine learning, pattern recognition or combinatorial optimization.

Datamining tasks can be organized into a taxonomy, based on the desired outcome of the algorithm. Usually a distinction is made between supervised and unsupervised learning. Classical tasks of supervised learning are:

- Classification: examining the attributes of a given instance to assign it to a predefined category or class.
- Classification rule learners: discovering a set of rules in the database which forms an accurate classifier.

The most common tasks of unsupervised learning are:

- Clustering: partitioning a data set into subsets (clusters), so that data in each subset share some common aspects. Partitioning is often indicated by a proximity evaluated using a distance measure.

F. Almeida et al. (Eds.): HM 2006, LNCS 4030, pp. 57–69, 2006.
© Springer-Verlag Berlin Heidelberg 2006

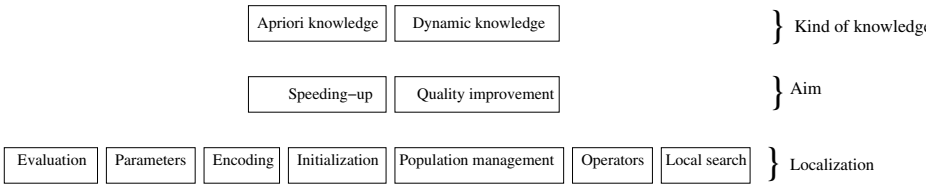


Fig. 1. The proposed taxonomy

– Association rule learners: discovering elements that occur in common within a given data set.

Using metaheuristics for knowledge extraction has become common while the other way which consists in using knowledge discovery to improve metaheuristic is less studied. This research way can be refereed as knowledge incorporation in metaheuristics and may be performed by using informed operators, approximation of fitness, etc.

To illustrate the different ways to integrate knowledge into metaheuristics, a small taxonomy that summarizes compositions found in several articles, is proposed in Figure 1.

– Two kinds of knowledge can be distinguished: a previously acquired knowledge which is called *Apriori Knowledge* and a dynamically acquired knowledge which is extracted or discovered during the search.
– Another useful information to classify algorithms is to distinguish the aim of the cooperation. Either the cooperation is used to reduce the computational time *i.e.*, speed up techniques, by simplification of the fitness *i.e.*, fitness approximation, or by significantly reducing the search space e.g., leading the metaheuristic in promising area; or the cooperation is used to improve the quality of the search by introducing knowledge in operators or in other parts of the metaheuristic. In fact, the insertion of datamining techniques often leads to both speeding up the metaheuristic and improving the quality.
– The last point used to distinguish the hybridizations is to determine which part of the metaheuristic is concerned by the knowledge incorporation. Hybridization can occur in each part of the metaheuristic: parameters, encoding, evaluation, initialization, operators, etc.

This paper aims to provide a quick comprehensive picture of the interest of combining datamining techniques and metaheuristics. We do not consider here the vast topic of incorporating knowledge but the use of knowledge algorithms also called datamining algorithms. In order to present this literature review, we have chosen to classify references with respect to the localization of the knowledge integration.

The remainder of this paper is set out as follows. Section 2 highlights the potential of datamining to speed-up metaheuristics by using datamining during

the evaluation. Section 3 discusses how datamining can help to set the parameters of the metaheuristic. Section 4 presents the use of datamining techniques for the initialization of metaheuristics. Section 5 is devoted to population management. Section 6 details the benefit of datamining in crossover operators. Section 7 shows the local search datamining applications in evolutionary computation. Section 8 exhibits that some metaheuristics are based on datamining incorporation. Finally, conclusions and perspectives are drawn in the last section.

2 Using Datamining During the Evaluation

In some real cases, the fitness function can be very expensive to compute. Thus decreasing the number of complete evaluations would be beneficial. To achieve this, some approximations of the fitness functions could be used, and datamining techniques may be interesting to obtain good approximations. A very complete and comprehensive survey on fitness approximation has been proposed in [15]. It shows that fitness approximation can be used either for expensive fitness functions or multi-modal fitness functions and may be realized by several approaches exposed below.

2.1 Replace the Evaluation Function by a Datamining Algorithm

Datamining techniques can be used to build approximate models of the fitness function. In this context, previously calculated fitnesses are learned by a datamining algorithm to approximate the fitness of new individuals. Many works use neural networks (both multi-layer perceptrons and radial-basis-function networks) to realize an approximation of the function to optimize. For example in [6], the authors use an artificial neural network (ANN) with a multiple objective genetic algorithm. They evaluate a large part of the population with an ANN and a small part is still simultaneously evaluated with the original function. Rasheed et al. propose to cluster data and to construct separate approximation models for the different clusters [30,32,29]. The approximation model can be used each time or alternatively with the real objective function.

2.2 Using Datamining Techniques to Avoid Evaluations

When the fitness function can be approximated, some authors use the approximation only within operators, such as initialization, mutation, crossover and selection. This approach avoids the calculation of time consuming fitness for individuals that may be of very bad quality and that will not be kept in the population. For example, in [31], the authors use approximations to improve operators. They generate several possible new individuals and then choose the best according to a reduced model. To compute the model, they maintain a large sample of points encountered during the course of the optimization and divide it into dynamic clusters. To compute the approximate model of an individual, they use the weighted k-nearest-neighbor approach which is a classification technique.

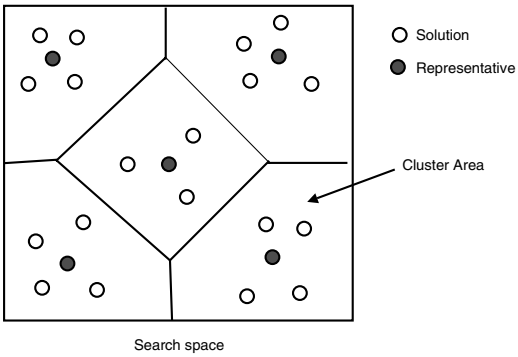


Fig. 2. Using representatives for the evaluation process through clustering

2.3 Using a Datamining Algorithm to Estimate a Representative

Approximating the fitness could be not satisfying because of the quality of the approximation for example. Another way to speed up the metaheuristic is to reduce the number of calls to the fitness function. This may be realized using fitness imitation. In this kind of approach, a set of individuals is considered as similar to another one and their fitness is fixed as equal to the reference individual which is called the representative. To determine the different sets, clustering techniques are often used.

Hence, in [16,18,47], the author proposes to maintain a large population size by using clustering algorithms. For example, in [47] Yoo et al. propose to use a fuzzy clustering approach to divide the population and to elect a representative of each cluster. Only the representatives are evaluated which allows the reduction of the evaluation costs. The fitness value of an individual of a cluster is estimated in respect with its associated representative (Figure 2).

3 How Datamining May Help to Set Parameters

A very difficult part in designing metaheuristics deals with the setting of the parameters of such methods. How can we fix in advance parameters such as the probability of application of a given operator, the size of the population or the number of iterations, for example? Two approaches may be used in this context:

- A first approach which is empirical consists in both executing several times the method with different parameter values and trying to select the best values. If the number of executions or the number of parameters are high, determining the best set of parameters may require statistical analyses. This may be seen as a datamining help.
- To set the probability of application of an operator, another approach may be used. It consists in analyzing the performance of the operators during the algorithm execution. In particular, this approach may be used when several

operators are available for the same operation (crossover or mutation, for example). In [12], the author proposes to compute the rate of appliance of a mutation operator by calculating the progress of the last applications of this operator. Hence, it becomes possible to determine the probabilities of appliance of a given operator in an adaptively way where the more efficient an operator is, the more it will be used. Another approach could be to analyse in details the new individuals generated by operators (in term of quality, diversity) using clustering algorithms for example. This would give valuable information that can help to set the new application probabilities.

These two approaches give examples on the way the datamining techniques may help to set parameters.

4 Using Datamining for Initialization

Generally, metaheuristics generate their initial solution(s) randomly. In continuous optimization, this generation may also be done using a grid initialization. It could be also interesting to cleverly generate the initial population in order, for example, to reduce the search space by leading the metaheuristic to promising area.

For example, in [28], Ramsey et al. propose to initialize a genetic algorithm with case-based reasoning in a tracker/target simulation with a periodically changing environment. Case-based initialization allows the system to automatically bias the search of the genetic algorithm toward relevant areas of the search space.

5 Datamining and Population Management

Datamining techniques are often used to manage the population. Several works deal with introducing new individuals in the population. Some common methods try to inject new individuals into the population. This could be realized to diversify the population like in the random immigrant strategy. In order to lead the search to promising search spaces it could be also interesting to regularly introduce individuals that are built based on information of the past encountered solutions.

In [20,21], Louis presents CIGAR (Case Injected Genetic AlgoRithm). The aim of CIGAR is to provide periodically to the genetic algorithm solutions that suit to similar problems. CIGAR has been successfully applied to several problems such as jobshop, circuit modelling, etc.

In [4,39], the authors propose to hybridize a genetic algorithm and the Apriori algorithm (Apriori is a classical algorithm to generate association rules [1]) to discover interesting subroutines for the oil collecting vehicle routing problem. They insert the found subroutines into the new individuals.

In the work of Ribeiro et al. [37,38,40], the authors present a GRASP hybridized with several frequent item set mining algorithms: the Direct Count and

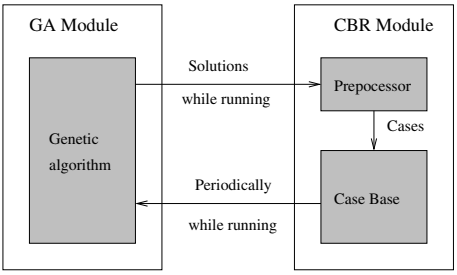


Fig. 3. Case Injected Genetic AlgoRithm (CIGAR)

the Intersect algorithms in [38] and the FPMax* in [40], which are Apriori-like approaches. These algorithms are used to extract patterns that are promising only on elite solutions. The hybridization is realized after a fixed number of seconds or iterations and new starting solutions for the GRASP are computed thanks to the found patterns. The authors apply their approach to the Set Packing Problem and the Maximum Diversity Problem. The method allows for the speed up of the convergence of the algorithm and for the improvement of the robustness of the GRASP.

Another example of the use of datamining techniques in the population management is the use of clustering algorithms in Multi-objective population metaheuristics where the result to produce is a set of solutions of best compromise (Pareto solutions). An archive is often used to store these solutions and the clustering is used to avoid a bias towards a certain region of the search space. Such a bias would lead to an unbalanced distribution of the Pareto solutions. Authors often use the average linkage method as this clustering algorithm performs well for Pareto optimization [48].

6 Using Datamining Within Operators

Incorporating knowledge in operators could be useful if, for example, it allows to cleverly exploit the search space. In the following section, some examples using machine learning approaches in crossover to explore the search space are presented.

Handa et al. propose a co-evolutionary genetic algorithm, which uses an hybridization between a GA and C4.5 (a classification algorithm) in order to discover the schemata to use in the crossover [10,9]. In their early work [8], Handa et al. have proposed a co-evolutionary algorithm in order to discover the good schemata to use that have not been discovered by the GA. The method works well but was just presented for bit representation.

LEM [22,23] integrates a symbolic learning component to evolutionary computation; it seeks out rules explaining the differences between the better and worse performers in the population, and generates new individuals based on the templates specified in these rules. An example of behaviour of LEM in the search

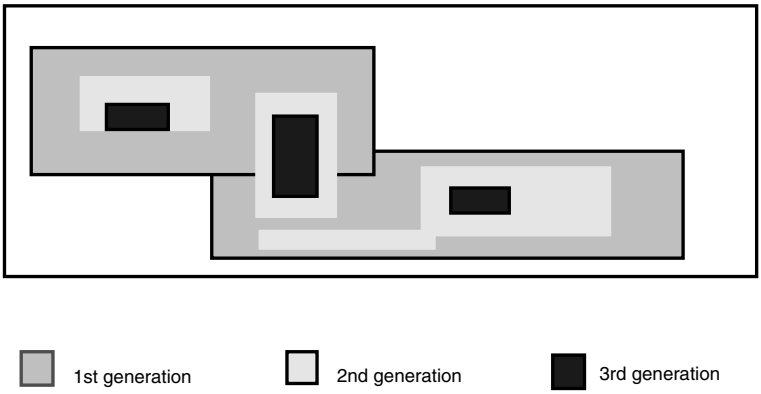


Fig. 4. LEM: Example of search region reductions defined by description of the 1st, 2nd and 3rd generations

space is shown on Figure 4 where the search regions associated to each generation are illustrated. This figure shows how the search space is reduced. The LEM methodology has proved able to improve the efficiency of the evolutionary process. LEM uses AQ learning algorithm (a general decision rules learning algorithm) in order to produce rules. Let us remark that, existing implementations, such as AQ11 [24], AQ15 [25] handle noise with pre and post-processing techniques. The basic AQ algorithm however, heavily depends on specific training examples during the search (the algorithm actually employs a beam search). This approach has been used with the C4.5 algorithm for mono-objective jobshop problems in the work of Huyet [14,13].

In the work of Jourdan et al. [17,45], the authors propose to extend LEM to LEMMO for the multi-objective case in order to seek for rules that explain why some individuals dominate others in a multi-objective point of view and why some individuals are dominated by other. They generate new individuals thanks to the rules by creating solutions that match to positive rules and do not match to negative rules. This approach has shown good results on a water system application both in speeding up the multi-objective algorithm and in improving the quality of the solutions.

We can remark that usually the authors use classification methods (C4.5, AQ, etc) to identify the genes that induce the good quality of the individuals. Some authors propose to also determine the genes that characterize bad quality solutions and to use them to repair the constructed solutions [17,45].

7 Datamining in Local Search

Metaheuristics are often hybridized with local search methods to improve the intensification part. Some datamining algorithms are themselves local searches and could be used as part of the metaheuristics.

In [6], the authors propose to use inverse Artificial Neural Networks (ANN) as local search to exploit specific region for candidate solutions. They note that ANNs can be adequate as they construct a smooth mapping. The ANN is trained in a reverse way as the input layer presents the criteria and the output the parameters to be optimised.

Moreover, datamining problems can be often modelized as optimization problems and in this case, the hybridization of the metaheuristic with a machine learning algorithm could be realized such that the machine learning algorithm treats a subproblem. For example, when clustering or grouping problems are solved using a metaheuristic, the metaheuristic searches for the optimal subset of genes that act as initial cluster centers. At the lower level, a local learning method performs local clustering from these initial centers. The objective is to combine the strength of EAs and clustering methods to produce a global efficient clustering algorithm. Kmeans is often used as a local search [5,11,46] for initialization of the solutions or to realize a local search during the search. Another approach, presented in [7], uses fuzzy c-means and hard c-means as an objective function. This article shows the importance of the initialization of the solution(s). In [3], the authors also use Expectation Maximization (EM) as a local search to analyze gene trajectory.

8 Datamining Based Metaheuristics

Some metaheuristics are designed to directly care of dynamic knowledge. We decide to create a specific part for them as they are now considered as new metaheuristic and not as improvement of previous ones. A lot of these algorithms are classified as Non-Darwinian evolutionary computation as they replace Darwinian operators by other operators. As identified in the taxonomy (Figure 1), the cooperation can appear in different localizations but we observe that in the proposed metaheuristics, the integration is often localised in the operator part. For example, the *Population-based Incremental Learning* (PBIL) creates a real-valued probability vector characterizing high fitness solutions [2] (Figure 5). This vector is then used to build new solutions. Generally, PBIL does not use mutation and crossover. PBIL can be considered as both encoding and initialisation localization hybridizations.

Specifically, Muhlenbein and Paass have estimated the probability density functions of binary variables in their chromosomes by the product of the individual probability density functions in the UMDA (Univariate Marginal Distribution Algorithm) [26]. Hence, UMDA is a special class of the PBIL algorithm.

Pelikan and Goldberg developed an algorithm "BOA" (Bayesian Optimization Algorithm) that extends the above ideas by using Bayesian Networks to model the chromosomes of superior fitness [27] (Figure 6). BOA can be classified as localization operator algorithm with dynamic knowledge.

A similar approach has also been proposed by Larranaga and Lozano, who have given the term "EDA" (Estimation of Distribution Algorithms) to the statistical estimation approach to EC [19].

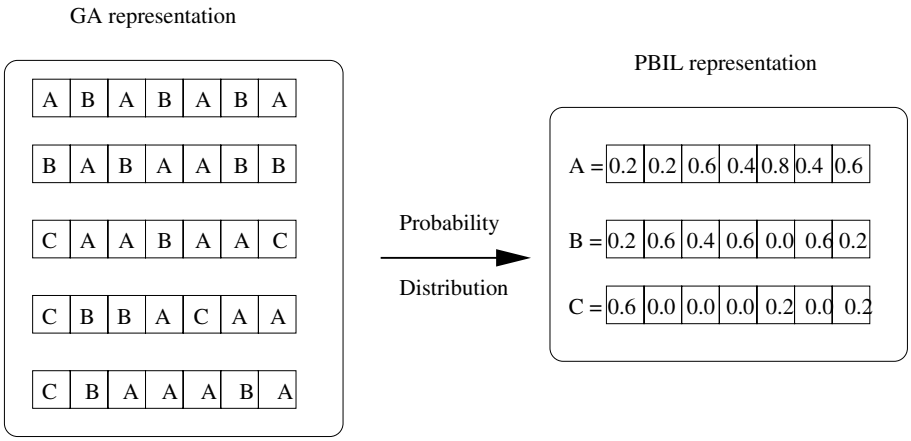


Fig. 5. The PBIL probability vector

```
begin
    t=0;
    Initialise randomly Population POP(0);
    Evaluate(POP(0));
    repeat
        Select a set of promising strings S(t) from POP(t);
        Construct the network B using a given metric and constraints;
        Generate a set of new strings O(t) according to the joint distribution
         encoded by B;
        Create a new population POP(t+1) by replacing some strings from P(t)
         with O(t);
        Evaluate(POP(t));
        t=t+1;
    until (termination condition)
end
```

Fig. 6. Overview of the Bayesian Optimization Algorithm

Similarly, cultural algorithms use high performing individuals to develop beliefs constraining the way in which individuals are modified by genetic operators [35,36] (Figure 7). In cultural algorithm, beliefs are formed based on each entity's individual experiences. The reasoning behind this, as outlined by [35], is that cultural evolution allows populations to learn and adapt at a rate faster than pure biological evolution alone. Importantly, the learning which takes place individually by each entity is passed on to the remainder of the group, allowing learning to take place at a much faster rate. Cultural algorithm can be classified as operator localization algorithm in the taxonomy.

Ravise and Sebag [43,34,41,33,42] worked on civilized genetic algorithms that differ from Darwinian's ones as they keep information of the population in order to avoid doing the same errors. The knowledge is dynamically updated during

```
begin
    t=0;
    Initialise Population POP(0);
    Initialise Belief Network BLF(0);
    Initialise Communication Channel CHL(0);
    Evaluate(POP(0));
    t=1;
    repeat
        Communicate(POP(0), BLF(t));
        Adjust(BLF(t));
        Communicate(BLF(t), POP(t));
        Modulate Fitness (BLF(t), POP(t));
        t=t+1;
        Select POP(t) from POP(t-1);
        Evolve(POP(t));
        Evaluate(POP(t));
    until (termination condition)
end
```

Fig. 7. Overview of the cultural evolution algorithm (Reynolds 1994)

generations. The datamining hybridization accelerates and improves the convergence of the algorithm. But it has been tested only on bit representation. The authors have observed that the GA must be run first with Darwinian operator in order to have diversity in its population. After a fixed number of generations, the civilized operator is used. They keep history of the past results in order to not reproduce the same error (and produce bad individuals). Civilized genetic algorithms can be classified as operator based dynamic knowledge.

9 Discussion and Conclusion

We have seen that there are multiple reasons to integrate datamining methods within metaheuristics. It could be to approximate the fitness function, to improve the convergence of the metaheuristics or to create an operator that is adapted to the problem.

In a research point of view, the actual major interest is to use datamining to extract useful information from the history of the metaheuristic in order to move the search in interesting space areas. Moreover, the hybridization between metaheuristics and datamining techniques have not been studied a lot in multi-objective optimization.

The major drawback of hybridization is the setting of parameters. When applying the datamining method, how many solutions have to be stored in dynamic knowledge, etc ? Many articles realize experimentally the parameter settings and many authors remark that clearly the performances are correlated with the parameters. A very promising search investigation is to automatically determine during the search all these parameters for designing adaptive efficient metaheuristics.

References

1. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proceeding 20th International Conference Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.
2. S. Baluja. Population based incremental learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994. http://www.cs.cmu.edu/afs/cs/user/baluja/www/techreps.html.
3. Z.S.H. Chan and N. Kasabov. Gene trajectory clustering with a hybrid genetic algorithm and expectation maximization method. In *IEEE International Joint Conference on Neural Networks*, pages 1669–1674, 2004.
4. F.L. Dalboni, L.S. Ochi, and L.M.A. Drummond. On improving evolutionary algorithms by using data mining for the oil collector vehicle routing problem. International Network Optimization Conference, 2003.
5. Emanuel Falkenauer. A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, 2(2):123–144, 1994.
6. A. Gaspar-Cunha and A.S. Vieira. A hybrid multi-objective evolutionary algorithm using an inverse neural network. In *Hybrid Metaheuristic*, pages 25–30, 2004.
7. L. O. Hall, I. B. Özyurt, and J. C. Bezdek. Clustering with a genetically optimized approach. *IEEE Trans. on Evolutionary Computation*, 3(2):103–112, 1999.
8. H. Handa, N. Baba, O. Katai, and T. Sawaragi. Coevolutionary genetic algorithm with effective exploration and exploitation of useful schemata. In *Proceedings of the International Conference on Neural Information Systems*, volume 1, pages 424–427, 1997.
9. 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*, 2002.
10. 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 Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, page 764, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.
11. J. Handl and J. Knowles. Improvements to the scalability of multiobjective clustering. In IEEE, editor, *IEEE Congress on Evolutionary Computation*, pages 438–445, 2005.
12. T.P. Hong, H. Wang, and W. Chen. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of heuristics*, 6:439–455, 2000.
13. A.-L. Huyet. *Extraction de connaissances pertinentes sur le comportement des systemes de production : une approche conjointe par optimisation evolutionniste via simulation et apprentissage.* PhD thesis, Université Blaise Pascal Clermont II, October 2004.
14. A.-L Huyet and J.-L. Paris. Configuration and analysis of a multiproduct kanban system using evolutionary optimisation coupled to machine learning. In *Proceedings of CESA 2003, the IMACS Multiconference Computational Engineering in Systems Applications*, July 2003. ISBN 2-9512309-5-8, CDROM.
15. Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 9(1):3–12, 2005.

16. 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.
17. L. Jourdan, D. Corne, D.A. Savic, and G A. Walters. Preliminary investigation of the learnable evolution model for faster/better multiobjective water systems design. In LNCS 3410, editor, *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO'05)*, pages 841–855, 2005.
18. 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.
19. P. Larranaga and J.A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.
20. S. J. Louis. Genetic learning from experience. In IEEE, editor, *Congress on Evolutionary Computation (CEC'03)*, pages 2118 – 2125, Australia, Dec 2003. IEEE.
21. S. J. Louis. Learning for evolutionary design. In *Proceedings of the 2003 Nasa/DoD Conference on Evolutionary Hardware*, pages 17–23, July 2003.
22. R.S. Michalski. Learnable evolution model: Evolutionary processes guided by machine learning. *Machine Learning*, 38(1–2):9–40, 2000.
23. R.S. Michalski, G. Cervon, and K.A. Kaufman. Speeding up evolution through learning: Lem. In *Intelligent Information Systems 2000*, pages 243–256, 2000.
24. R.S. Michalski and J.B. Larson. Selection of most representative training examples and incremental generation of vl1 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.
25. R.S. 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.
26. H. Muhlenbein and G. Paass. From recombination of genes to the estimation of distributions: I. binary parameters. *Lecture Notes in Computer Science*, 1141:178–187, 1996.
27. M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 13-17 1999. Morgan Kaufmann Publishers, San Fransisco, CA.
28. C. Ramsey and J. Grefenstette. Case-based initialization of genetic algorithms. In *Fifth International Conference on Genetic Algorithms*, pages 84–91, 1993.
29. K. Rasheed. An incremental-approximate-clustering approach for developing dynamic reduced models for design optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC'00)*, pages 986–993, California, USA, 6-9 2000. IEEE Press.
30. K. Rasheed and H. Hirsh. Using case based learning to improve genetic algorithm based design optimization. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, San Francisco, CA, 1997. Morgan Kaufmann.
31. K. Rasheed and H. Hirsh. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. In L. Darrell Whitley, David E. Goldberg, Erick Cantú-Paz, Lee Spector, Ian C. Parmee, and Hans-Georg Beyer, editors, *GECCO*, pages 628–635. Morgan Kaufmann, 2000.

32. K. Rasheed, S. Vattam, and X. Ni. Comparison of methods for developing dynamic reduced models for design optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, pages 390–395, 2002.
33. C. Ravise and M. Sebag. An advanced evolution should not repeat its past errors. In *International Conference on Machine Learning*, pages 400–408, 1996.
34. C. Ravise, M. Sebag, and M. Schoenauer. A genetic algorithm led by induction. url: citeseer.ist.psu.edu/126354.html.
35. R. G. Reynolds, Z. Michalewicz, and Michael J. Cavaretta. Using cultural algorithms for constraint handling in genocop. In *Evolutionary Programming*, pages 289–305, 1995.
36. R. G. Reynolds and B. Peng. Cultural algorithms: computational modeling of how cultures learn to solve problems: an engineering example. *Cybernetics and Systems*, 36(8):753–771, 2005.
37. M. Ribeiro, A. Plastino, and S. Martins. Hybridization of grasp metaheuristic with data mining techniques. *Special Issue on Hybrid Metaheuristic of the Journal of Mathematical Modelling and Algorithms*, 5(1):23–41, April 2006.
38. M. Ribeiro, V. Trindade, A. Lastino, and S. Martins. Hybridization of GRASP metaheuristic with data mining techniques. In *Workshop on Hybrid Metaheuristics 16th European Conference on Artificial Intelligence (ECAI)*, pages 69–78, 2004.
39. H.G. Santos, L.S. Ochi, E.H. Marinho, and L.M.A. Drummond. Combining an evolutionary algorithm with data mining to solve a vehicle routing problem. *NEUROCOMPUTING*, 2006. (to appear).
40. L. Santos, M. Ribeiro, A. Plastino, and S. Martins. A hybrid GRASP with data mining for the maximum diversity problem. In LNCS 3636, editor, *Hybrid Metaheuristic*, pages 116–128, 2005.
41. M. Sebag, C. Ravise, and M. Schoenauer. Controlling evolution by means of machine learning. In *Evolutionary Programming*, pages 57–66, 1996.
42. M. Sebag and M. Schoenauer. Controlling crossover through inductive learning. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature – PPSN III*, pages 209–218, Berlin, 1994. Springer.
43. M. Sebag, M. Schoenauer, and C. Ravise. Toward civilized evolution: Developing inhibitions. In Thomas Bäck, editor, *Proceeding of the Seventh Int. Conf. on Genetic Algorithms*, pages 291–298, San Francisco, CA, 1997. Morgan Kaufmann.
44. E-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of heuristics*, 8(2):541–564, Sept. 2002.
45. L. Vermeulen-Jourdan, D. Corne, D.A. Savic, and G.A. Walters. Hybridising rule induction and multi-objective evolutionary search for optimising water distribution systems. In *Proceeding of Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, pages 434–439, 2004.
46. L. Vermeulen-Jourdan, C. Dhaenens, and E-G. Talbi. Clustering nominal and numerical data: A new distance concept for a hybrid genetic algorithm. In Jens Gottlieb and Günther 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.
47. S-H. Yoo and S-B. Cho. Partially evaluated genetic algorithm based on fuzzy c-means algorithm. In LNCS 3242, editor, *Parallel Problem Solving From Nature (PPSN)*, pages 440–449, 2004.
48. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Information Processing Letters 109 (2009) 887–896

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis

J. García-Nieto [a,1], E. Alba [a,*,1], L. Jourdan [b,2], E. Talbi [b,2]

[a] Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Málaga - 29071, Spain
[b] LIFL/INRIA Futurs Parc scientifique de la Haute-Borne, Bât. A, 40 Avenue Halley, 59650 Villeneuve d'Ascq Cedex, France

ARTICLE INFO

Article history:
Available online 15 April 2009
Communicated by A.A. Bertossi

Keywords:
Algorithms
Analysis of algorithms
Combinatorial problems
Databases
Design of algorithms
Performance evaluation
Sensitivity and specificity analysis
Multiobjective genetic algorithm
Microarray gene selection

ABSTRACT

The study of the sensitivity and the specificity of a classification test constitute a powerful kind of analysis since it provides specialists with very detailed information useful for cancer diagnosis. In this work, we propose the use of a multiobjective genetic algorithm for gene selection of Microarray datasets. This algorithm performs gene selection from the point of view of the sensitivity and the specificity, both used as quality indicators of the classification test applied to the previously selected genes. In this algorithm, the classification task is accomplished by Support Vector Machines; in addition a 10-Fold Cross-Validation is applied to the resulting subsets. The emerging behavior of all these techniques used together is noticeable, since this approach is able to offer, in an original and easy way, a wide range of accurate solutions to professionals in this area. The effectiveness of this approach is proved on public cancer datasets by working out new and promising results. A comparative analysis of our approach using two and three objectives, and with other existing algorithms, suggest that our proposal is highly appropriate for solving this problem.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Microarray technology [1] (DNA Microarray) allows biologists to simultaneously analyze thousands of genes and can thus provide important insights into cell functioning, since changes in the physiology of an organism are generally associated with changes in gene expression patterns. The vast amount of data involved in a typical Microarray experiment naturally leads to using statistical analysis, in addition to classifying the dataset into correct classes. The key issue in this classification procedure is to identify significant and representative gene subsets that may be used to predict class membership for new external samples of genes. The main difficulty in the Microarray classification problem is the availability of a relatively small number of samples in comparison with the large number of genes in each sample. In addition, expression data are highly redundant and noisy, and most genes are believed to be uninformative with respect to classes studied, as only a fraction of genes may present distinct profiles for different classes of samples.

In this context, feature selection is considered a necessary preprocess step to analyze large datasets, as this method can reduce the dimensionality of the datasets and often leads to better analyses [2]. Therefore, in feature selection, the objective is to select subsets, which are as small as possible, of informative features from the initial dataset, in order to obtain high classification accuracy.

Nevertheless, optimal feature selection is a complex problem proved to be NP-hard [3], and hence, only meta-

* Corresponding author.
E-mail addresses: eat@lcc.uma.es (J. García-Nieto), jnieto@lcc.uma.es (E. Alba), laetitia.jourdan@lifl.fr (L. Jourdan), el-ghazali.talbi@lifl.fr (E. Talbi).

[1] Authors acknowledge funds from the Spanish Ministry of Sciences and Innovation FEDER under contract TIN2008-06491-C04-01 (M* project, http://mstar.lcc.uma.es) and CICE, Junta de Andalucía under contract P07-TIC-03044 (DIRICOM project, http://diricom.lcc.uma.es).

[2] Authors acknowledge funds from the INRIA-PERFOM 3 + 3 Méditérrané.

0020-0190/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2009.03.029

heuristic approaches are capable of solving it accurately and efficiently. Evolutionary Algorithms (EAs) have been successfully used in the past to tackle the gene selection of Microarray datasets [4–6]. Specifically, Multiobjective Evolutionary Algorithms (MOEAs) [7–9] are actually quite popular for feature selection, since they allow a given solution to be evaluated (selected subset) in a more suitable and straightforward way. From this point of view, two objectives are clearly involved in the evaluation of solutions: minimizing the *number of genes* and maximizing the classification *accuracy*. However, using this approach, the range of good solutions (i.e., $\leqslant$10 genes and $\geqslant$90% accuracy) is normally limited, which is a problem for the subsequent expert decision making process. In this sense, the analysis of the sensitivity and the specificity (ROC analysis) [10] of a diagnostic test constitutes a powerful analysis in supervised classification since it provides the specialists with more detailed information about the validity of a solution. Moreover, the use of the sensitivity and the specificity as objectives in the evaluation task, instead of the accuracy rate, offers a mathematically equivalent method since these two factors are proportionally (weighted to the *prevalence*) included in the overall accuracy calculation. Therefore, despite the intuitive appeal of using only the overall accuracy as a single measure of test validity, its dependence on the *prevalence* renders it inferior to a careful and balanced consideration of sensitivity and specificity [11].

Recent studies have used multiobjective algorithms in order to optimize the sensitivity and the specificity for a given classifier [12,13]. However, the main goal of these approaches consisted of looking for a favorable trade-off between sensitivity and specificity, considering neither subset selection nor dataset reduction purposes. In this paper, we extend these works by using a MultiObjective Genetic Algorithm (MOGA) for gene selection and classification of Microarray datasets, which optimizes three objectives: maximize the sensitivity, maximize the specificity, and minimize the number of genes. Initially, this algorithm selects the subsets of genes encoded in the tentative solutions manipulated by the algorithm. After that, each solution is evaluated using the Support Vector Machines (SVMs) classifier, and 10-fold cross-validation is then applied to assess the percentages of sensitivity and specificity. In addition, each generation, specialized crossover and mutation operators, both adapted to feature selection, are applied to the population. Our contribution is noticeable since this approach is able to offer a number of accurate solutions to professionals in this area. As we will show in the experiments, the effectiveness of this approach is evaluated on three well-known datasets, and new and promising results are obtained.

The remaining of this paper is organized as follows. In Section 2, we provide the reader with basic concepts about the feature selection problem, the sensitivity and specificity analysis, the Support Vector Machines classifier and the Microarrays technology. Section 3 gives the details of our specialized MOGA algorithm for gene selection and classification. Experimental results and comparisons are presented in Section 4. Conclusions and further work are finally given in Section 5.

2. Basic concepts

In this section, preliminary concepts of the feature selection problem, the sensitivity and specificity analysis, the SVMs classifier and the Microarrays technology are briefly introduced.

2.1. Feature selection

When applied to Biology, feature selection is also called *gene selection*, targeted to distinguish influential genes from irrelevant ones based on DNA Microarray datasets. This technique is normally coupled with learning algorithms that use the reduced subset of features in order to proceed as a supervised classifier. The formal definition of the feature selection problem is given as follows:

Definition. Let $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ be a set of features; find a subset $F' \subseteq F$ that maximizes a scoring function $\Theta : \Gamma \to G$ such that $F' = \text{argmax}_{G \subset \Gamma}\{\Theta(G)\}$, where Γ is the space of all possible feature subsets of F and G a subset of Γ [14].

In feature selection, two different models may be used depending on whether the learning algorithm is coupled with the selection method or not, respectively *wrapper* model and *filter* model. On the one hand, the *filter model* carries out the selection and classification regardless of the learning algorithm used. Filter methods are based on a performance evaluation metric calculated directly from the data.

On the other hand, the *wrapper model*, which performs feature subset selection and classification in the same single process, internally uses a learning algorithm to measure the accuracy.

2.2. Sensitivity and specificity

The sensitivity is a statistical value measuring how well a binary classification test correctly identifies a condition. The sensitivity is the proportion of *true positives* of all diseased cases in the entire population. For example, for a medical test to determine if a person has a certain disease, the sensitivity to the disease is the probability that if the person has the disease, the test will be positive. However, the sensitivity alone does not tell us how well the test predicts other classes.

The specificity is a statistical measure of how well a binary classification test correctly identifies the negative cases, or those cases that do not meet the condition being studied. For example, given a medical test that determines if a person has a certain disease, the specificity of the test to the disease is the probability that the test indicates "negative" if the person does not have the disease. The specificity is thus the proportion of *true negatives* of all negative cases in the population.[3]

[3] The sensitivity depends on the number of true positives (#*TP*) and true negatives (#*TN*). The specificity depends on the number of true negatives (#*TN*) and false positives (#*FP*). The number of false negatives (#*FN*) is included in the calculation of a third value: accuracy.

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#TN}, \tag{1}$$

$$\text{Specificity} = \frac{\#TN}{\#TN + \#FP}. \tag{2}$$

Eqs. (1) and (2) calculate the sensitivity and specificity factors of a prediction test, respectively.

A combination of these two measures is the most widely used method to quantify the diagnostic ability of a test, since these measures constitute basic factors included in all well-known statistical analysis such as the *receiver operating characteristic* (ROC curve) and the *F-measure*. A useful measure, the accuracy value of a test, may be determined by combining sensitivity and specificity, with another measure called Prevalence,[4] using the following equation:

$$\text{Accuracy} = \text{Sensitivity} \times \text{Prevalence} + \text{Specificity} \times (1 - \text{Prevalence}) \tag{3}$$

being

$$\text{Prevalence} = \frac{\#\text{classes (2 in this work)}}{\#\text{genes (in the dataset)}}. \tag{4}$$

In addition, the accuracy can also be expressed in terms of the true/false positive/negative factors as formulated in Eq. (5):

$$\text{Accuracy} = \frac{\#TP + \#TN}{\#TP + \#FP + \#FN + \#TN}. \tag{5}$$

2.3. Support Vector Machines

Support Vector Machines, derived from statistical learning theory, are used to classify points by assigning them to one of two disjoint half spaces. The objective is to provide a model which predicts, as efficiently as possible, the class of a given data instance in the testing set where only the values of the features are known and there is no information about the classes. Vapnik and Cortes [15] defined the SVMs method as follows:

Definition. Given a training set of instance-label pairs (x_i, y_i) with $i = 1, 2, \ldots, l$ where $x_i \in R^n$ (training vectors) and $y_i \in \{1, -1\}^l$ (classes), the Support Vector Machines require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{6}$$

being

$$y_i(w^T \phi(x_i) + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0. \tag{7}$$

Here, $C > 0$ is the penalty parameter of the error term. For linearly separable data, SVMs obtains the vector which maximizes the distance between the training samples and the class boundary. For non-linearly separable data, samples (x_i) are mapped to a high-dimensional space by means of the function ϕ, where a separating hyperplane can be found. The assignment is carried out by means of the equation $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ called the *kernel* function.

In the classification of informative genes embedded in a given dataset of gene expression levels, SVMs uses the kernel function to find an orthogonal hyperplane to a specific gene dimension. Many works in the literature have successfully used SVMs for gene selection and classification [4, 5,16]. In this work, a linear kernel function ($K(x_i, x_j) \equiv x_i^T x_j$) is used by the SVMs classifier.

2.4. Microarrays and gene expression

Microarrays or gene arrays/chip [1] consist of a thin glass substrate containing specific DNA gene samples spotted in an array by a mechanical procedure. These DNA samples are spread with fluorescently labeled m-RNA from an experimental condition. This m-RNA hybridizes strongly with some DNA gene samples and weakly with others, depending on the inherent double helical characteristics. The array is then scanned and the resulting image is processed in order to detect fluorescence levels (using red and green dyes), indicating the strength with which the sample expresses each gene.

The logarithmic ratio between the two intensities of each dye is used as the gene expression data. The relative abundance of the DNA sequences spotted in a pair of DNA or RNA samples is assessed by evaluating the differential hybridization of the two samples compared to the sequences on the Microarray. The gene expression levels can be determined for samples taken either at multiple time instants of a given biological process or under various conditions, such as for tumor samples with different histopathological diagnosis. Each sample corresponds to a high-dimensional row vector of its gene expression profile.

3. Gene selection and classification by MOGA

As explained in the introduction, MOEAs are already being used for solving the gene selection and classification of gene expression datasets [7–9]. In most of these works, the optimized function is computed using two main objectives: the classification accuracy and the number of genes. Our approach here is to use a Multiobjective Genetic Algorithm, in which the evaluation of solutions involves the sensitivity and the specificity of the classification test, as well as the number of genes, but without using the classification accuracy. In this section, we present the main elements of our MOGA focusing on the fitness evaluation, solution encoding and adapted operators. Multiobjective specific aspects such as the preservation of the diversity and the selection of solutions are also described.

3.1. The multiobjective approach

Formally, each gene subset belonging to two classes, $s_i = \{X_i, y_i\}$ where $\{X_i\}$ represents the n training samples and $y_i \in \{-1, +1\}$ their class labels, is associated with

[4] In Biology, the Prevalence of a disease is the proportion of total cases (classes) of the disease in the studied population (set) divided by the number of individuals (genes) in this population (studied subset).

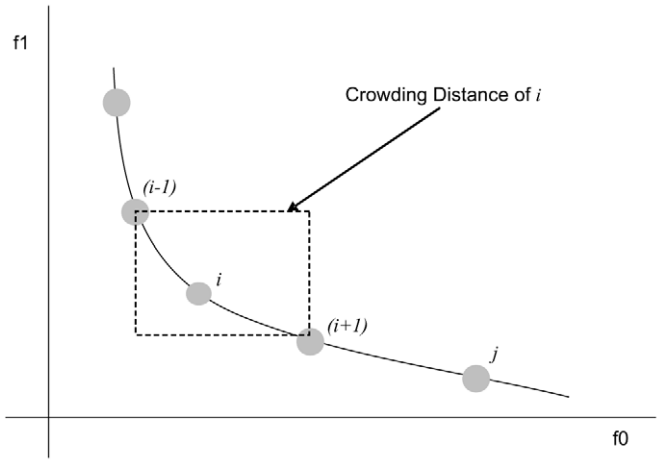


Fig. 1. Crowding distance operation.

a vector evaluation function $F(s_i) = \langle F_1(s_i), F_2(s_i), F_3(s_i)\rangle$, each function corresponding to:

$$F_1(s_i) = \frac{\#(\text{genes in } s_i)}{\#(\text{total genes})} \quad (\text{minimize \#genes}), \tag{8}$$

$$F_2(s_i) = \frac{\#TP}{\#TP + \#TN} \quad (\text{maximize sensitivity}), \tag{9}$$

$$F_3(s_i) = \frac{\#TN}{\#TN + \#FP} \quad (\text{maximize specificity}). \tag{10}$$

Our MOGA evolves by optimizing simultaneously these three fitness functions. In this scenario, a selected gene subset can be more informative (better than), less informative (worse than), equal, but also indifferent to another gene subset with respect to the objective values. Here, one subset s_i is said to "dominate" or be "more informative" than another subset s_j when:

$$\forall k \in \{1, 2, 3\}: \; F_k(s_i) \succeq F_k(s_j) \tag{11}$$

and

$$\exists k: \; F_k(s_i) \succ F_k(s_j). \tag{12}$$

In these equations, given two subsets of genes s_a and s_b, $F_k(s_a) \succeq F_k(s_b)$ means that s_a is not "less informative" than s_b, and $F_k(s_a) \succ F_k(s_b)$ means that the subset s_a is "more informative" than s_b. The "Pareto front" is then defined as the set of nondominated gene subsets which constitutes the best solutions. The MOGA guides the search towards the Pareto front, keeping the nondominated solutions as diverse as possible, and preventing nondominated solutions from being rejected, in order to delay premature convergence.

In our approach, the reinforcement of the diversity in the front is carried out by means of the "Crowding" distance operator (typical in NSGA-II [17]). The Crowding distance operator (Fig. 1) assigns the highest value to the boundary solutions, and the average distance of two solutions $(i-1)$th and $(i+1)$th on either side of the solution i in each of the objectives. The complete front (with new and old nondominated solutions) is sorted and the Crowding distance operator is performed to get a new front with spread solutions. The selection task is then accomplished by a tournament Crowding strategy, in which, given two solutions i and j, the solution i is selected if it dominates the solution j. If neither solution dominates the other, then the one less densely allocated in the search space (i.e., with less Crowding distance) is selected.

3.2. Solution encoding

In our MOGA, each individual encodes a selected subset of genes by using a binary vector where each bit represents a gene in the dataset. If a bit is '1', it means that this gene is selected for the reduced subset and '0' indicates that the gene is not selected. Therefore, the length of the individuals is equal to the number of genes in the initial Microarray dataset.

3.3. Adapted crossover and mutation operators

Specific crossover and mutation operators, adapted to feature selection, are used in our MOGA as reproduction methods. The first one, *Subset Size-Oriented Common Feature* crossover (SSOCF) [18,19], is one of the most commonly used when facing the feature selection. As proved in [18], the SSOCF keeps useful informative patterns and produces offspring which have the same number of features (genes) as the parents. Here (see Fig. 2), the common features (bits '1') are kept by offspring and each non-shared feature is inherited from the ith parent (feature) with a probability $(n_i - n_c/n_u)$. Where n_i is the number of selected features of the ith parent, n_c is the number of commonly selected features from the mating parents, and n_u is the number of non-shared selected features.

The second operator, consisting of a *weighted* mutation, is applied with a probability of $p_{mut} = 0.1$ to the population. When an individual is mutated, its bits are flipped with different probabilities (p_{flip}) in order to adjust the number of flips from '1' to '0', and vice versa. We carried out tuning experiments with three options of flipping a simple bit as indicated in Table 1.

In this table, if we choose *one reduction* bit-flip mutation, if a given bit is '1' then it is flipped to '0' with

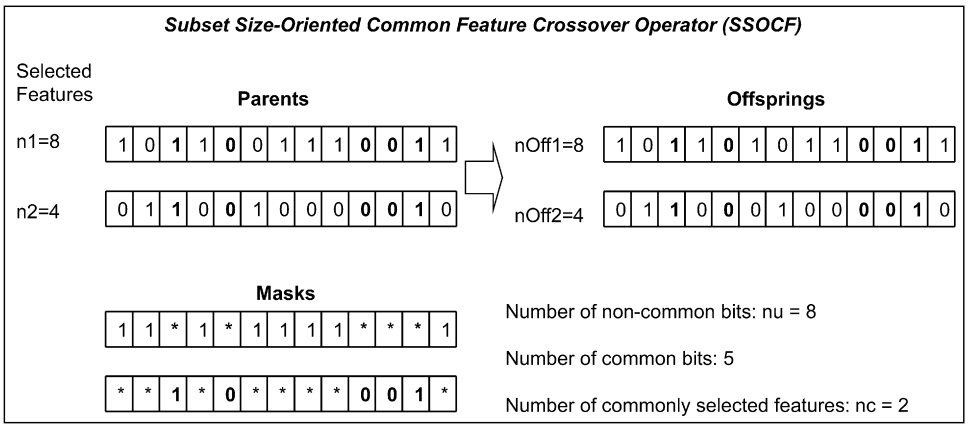


Fig. 2. Operation scheme of the SSOCF crossover. The mask promotes common shared features from parents to offspring.

Table 1
Three different kinds of bit-flip mutations. The values indicate the probability of bit-flip (p_{flip}).

Bit-flip option	1 to 0	0 to 1
uniform	0.3	0.3
zero reduction	0.3	0.6
one reduction	0.6	0.3

a probability of $p_{flip} = 0.6$, and if this bit is '0' then it is flipped to '1' with $p_{flip} = 0.3$. For the tuning experiments, we first used a *uniform* bit-flip (always $p_{flip} = 0.3$). This kind of mutation led the algorithm to obtain solutions with low percentages of sensitivity and specificity (hence accuracy) close to 65% and 60%, respectively. Secondly, when we used the *zero reduction* mutation, the feature selection procedure never obtained subsets with less than 30 genes which provokes the stagnation of the reduction process prematurely. Finally, we incorporated the *one reduction* mutation to our algorithm which obtained subsets always with less than 10 genes and percentages of sensitivity and specificity higher than 85% and 81%, respectively. Therefore, for the subsequent experimentation we used the *one reduction* mutation since it showed the best performance.

3.4. The general MOGA

The MOGA employed here generates an initial population P of individuals randomly (uniformly) initialized. Each individual, codifying a gene subset, is evaluated by means of the SVMs classifier and then 10-fold cross-validation is applied to assess the percentage of sensibility and specificity.

The population is sorted by using the dominance Crowding criteria described in Section 3.1. From this, a new elitist population E is generated selecting 10% of the best individuals. This selection is accomplished using the crowding selection operator of Section 3.1. The adapted SSOCF crossover and mutation are applied to the elitist population (E) to generate the offspring P'. Both, offspring and parent populations are then combined ($P' \cup P$). Finally, the best members replace the worst parents. When it is not possible to accommodate all the members of a particular front, that front is sorted according to the crowding distance. The individuals are selected on the basis of higher crowding distance. This selection is repeated to completely fill the new population with one of the same size the old one. The MOGA evolves for a fixed number of generations.

4. Experiments

We have implemented the proposed MOGA for gene selection in C++ using the ParadisEO [20] framework. As SVMs classifier, we have used a set of object classes provided by the LIBSVM [21] library consisting of training, testing, and validation tools. This classes were coupled with the MOGA algorithm in the evaluation phase. In this section, the experiments are described concerning the datasets, the experimentation setup, the analysis of results, and discussions.

4.1. Datasets

The used instances are classified into three well-known datasets obtained from real-word Microarray experiments. All of them were taken from the public UPITT Cancer Gene Expression Data Set Link Database in URL http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html.

- The *ALL-AML Leukemia* dataset consists of 72 tissue samples with 7129 gene expression levels. Two classes exist in this dataset: *Acute Myeloid Leukemia* (AML) and *Acute Lymphoblastic Leukemia* (ALL). The complete dataset contains 25 AML and 47 ALL samples. The original data are divided into a training set of 38 samples and a test set of 34 samples.
- The *Colon tumor* dataset consists of 62 tissue samples collected from colon-cancer patients with 2000 genes. Among them, 40 tumor biopsies are from *tumors* and 22 biopsies are from healthy parts of the colons of the same patients.
- *Types of Diffuse Large B-cell Lymphoma* dataset consists of 47 tissue samples, 24 of them are from the *germinal centre B-like group* while the remaining 23 are *activated B-like group*. Each sample is described by 4026 genes.

The expression levels were normalized in order to scale the intensities, enabling thus a comparison of the different datasets previously introduced. Each attribute was scaled to $[-1, 1]$ (as LIBSVM recommends) by means of:

$$a'_j(x_i) = 2 \times \frac{a_j(x_i) - min_j}{max_j - min_i} - 1, \tag{13}$$

where max_j and min_j correspond to the maximum and minimum gene expression values for attribute a_j in all samples.

4.2. Experimental setting

As explained in the introduction, the individuals of the population, representing gene subsets, are evaluated by means of the SVMs classifier and 10-fold cross-validation. At each iteration, the dataset is divided into ten subsets, nine of them constitute the training set and the remaining one is the test set. The SVMs is trained using the training set and then the sensitivity and the specificity of the classifier (once trained) are evaluated on the test set. This evaluation is repeated ten times, alternating the test set used each time. This method reinforces the validation process, so that the sensitivity and specificity values are the average of the ten validation folds. Moreover, such a strong validation is necessary when the number of samples is low in relation to the number of features, which is the case in this work.

Therefore, an optimal configuration of the SVMs classifier is crucial since it influences the training effectiveness. In these experiments, the main kernel parameters, γ and C coefficient (explained in Section 2), were systematically optimized in a preprocess phase for each dataset (by means of the Grid Tool of LIBSVM [21]) as follows:

- Leukemia: $C = 8$ and $\gamma = 1.220703125 \times 10^{-4}$.
- Colon: $C = 128$ and $\gamma = 1.220703125 \times 10^{-4}$.
- Lymphoma: $C = 8$ and $\gamma = 3.05175578125 \times 10^{-5}$.

These parameters were set using the SVMs classifier independently of the MOGA, in order to obtain accuracy as close to 100% as possible.

For the MOGA algorithm, the population size was fixed to 100 individuals; and 30 independent runs were performed with 2000 generations each one. The crossover and mutation operators were applied as explained in Section 3.3.

4.3. Results

In this section, we first report the results obtained by MOGA operating with three objectives. Following the standard methodology when comparing classification rates, the average and standard deviation (obtained after 30 runs) of the sensitivity, the specificity and the number of genes are shown in Table 2. The number of solutions that constitute the final Pareto front is reported in the last column. Since the number of final solutions is always higher than one, we report the best solution (in each final front) to discuss internal details like the number of genes, to assess the average and standard deviation.

Table 2
Results obtained by our MOGA in 3 objective optimization. The columns indicate the average and standard deviation of the number of genes (*NG*), the sensitivity rate (*Sen.*), the specificity rate (*Spe.*), and the number of solutions in the final Pareto front (*NSf*).

Dataset	*NG*	*Sen.* (%)	*Spe.* (%)	*NSf*
Leukemia	7.44 ± 2.14	87.63 ± 5.60	81.56 ± 10.34	9.11 ± 2.31
Colon	2.25 ± 0.20	85.93 ± 6.44	83.89 ± 4.08	12.25 ± 3.63
Lymphoma	5.00 ± 1.94	91.55 ± 7.56	86.36 ± 3.79	5.60 ± 3.43

As we can observe in Table 2, our algorithm obtains subsets of between 9 (Leukemia) and 2 (Colon) genes with rates of sensitivity and specificity higher than 85% and 81%, respectively. These results lead us to state that our MOGA performs very efficiently, since considering a *Prevalence* with a constant value, it can obtain over 85% accuracy, as explained in Section 2.2. Moreover, the number of solutions provided (between 2 and 16) reinforces this claimed effectiveness. In this sense, as well as in the number of solutions, we must consider the diversity in the final Pareto front, since this informs how different the final solutions are. Fig. 3 illustrates three representative fronts of solutions obtained in our experiments. We can observe that solutions are sufficiently scattered between 98% and 78% of sensitivity, and between 90% and 70% of specificity, despite the low number of genes in subsets.

4.4. 3-objective versus 2-objective approaches

In order to analyze the performance of our MOGA, an additional experimentation was carried out to compare the effectiveness when operating with 3 and 2 objectives.

The 3-objectives MOGA optimizes the sensitivity, the specificity and the number of genes. However, the 2-objectives MOGA optimizes the accuracy and the number of genes.

In these experiments, the subsets of genes resulting from both approaches after 30 runs were evaluated using the same cross validation method external to the selection process. Initially, the datasets were divided into two subsets, a training set and a test set. The selection algorithm was applied to the training set, and when an optimized subset of genes was obtained (either by the 3-objectives or the 2-objectives MOGA), its accuracy was evaluated on the external test set. This way, we were able to accurately compare the two different approaches.

Table 3 shows the average and standard deviation (in 30 runs) of the results reported by our MOGA in both 3-objective and 2-objective mode. We have carried out a set of statistical tests in order to find significant differences between both approaches. In each case the procedure for generating the statistical information was the following.

First a Kolmogorov–Smirnov test was performed in order to check whether the variables were normal or not and the Levene test to check the homocedasticity of samples (equality of variances). If they were (normal with equal variances), an ANOVA I test was performed, otherwise we performed a Kruskal–Wallis test. A level of significance of

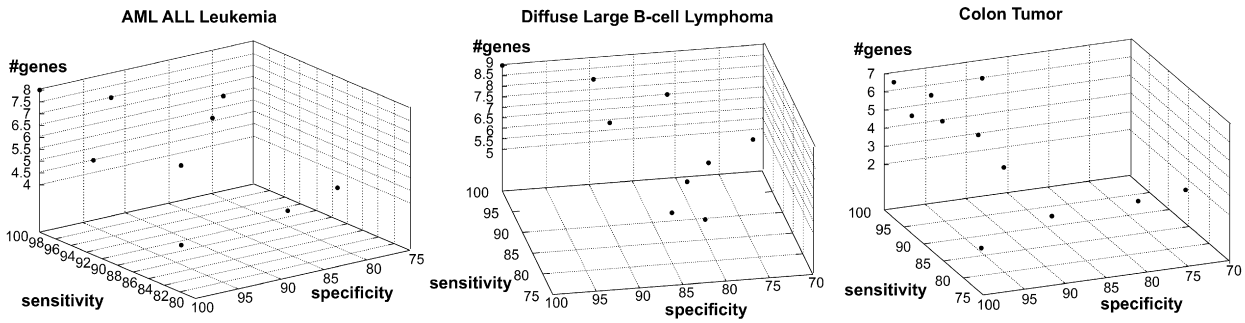


Fig. 3. Pareto fronts obtained by our MOGA in the classification of Leukemia, Lymphoma, and Colon datasets. Three objectives are optimized: the number of genes (#genes), the sensitivity, and the specificity.

Table 3
Comparison of 3 versus 2 objective MOGA. The columns indicate the average and standard deviation of the number of selected genes (*NG*), and the accuracy rate (*Ac.*). T_S indicates the percentage of samples of the external test set.

Dataset	3 obj.			2 obj.			T_S (%)	Statistical test
	NG	*Ac.* (%)	*NSf*	*NG*	*Ac.* (%)	*NSf*		
Leukemia	6 ± 2.64	$\mathbf{98.03 \pm 1.61}$	9.11 ± 2.31	7.33 ± 1.52	95.27 ± 2.21	6.66 ± 0.57	47%	+
Colon	3.33 ± 1.52	$\mathbf{89.58 \pm 1.80}$	12.25 ± 3.63	3.66 ± 0.57	86.45 ± 4.77	5.66 ± 0.57	50%	+
Lymphoma	3.75 ± 1.50	$\mathbf{96.05 \pm 3.04}$	5.60 ± 3.43	3.33 ± 1.52	92.97 ± 3.03	3 ± 1	50%	+

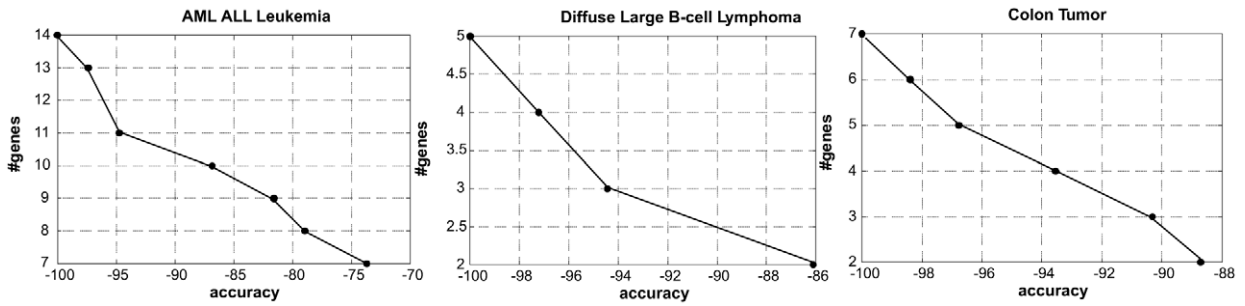


Fig. 4. Pareto fronts obtained by our MOGA in the classification of Leukemia, Lymphoma and Colon datasets. Two objectives are optimized: the number of genes (#genes), and the accuracy rate.

95% ($\alpha = 0.05$) is always applied in order to check if statistically significant differences exist. After that, we did a multiple comparison test whose results are presented in the last column of Table 3 where a plus sign means that the difference is significant (minus sign would mean that it was not).

This way, we can observe that the accuracy rate obtained by the 3-objective approach is better than the one obtained by a 2-objective approach for the three datasets. Furthermore, for all the instances, the results are statistically different (+), which leads us to ensure the final higher performance of the 3-objective approach.

As a secondary observation, the 3-objective approach obtains a larger number of solutions in the final fronts, which is an important issue when facing the decision making process. This property is clearly observable in Fig. 4, where several 2-objective fronts obtained in these experiments are shown in contrast with Fig. 3. Nevertheless, the diversity and quality of solutions shown in Fig. 4 are also suitable for gene selection, leading us recommend our MOGA with 2 objectives for other future scenarios.

4.5. Comparison with other approaches

In this section we first compare the performance of our MOGA (in 3-objective mode) with a base-line method for the gene selection. This method runs a K-Mean procedure for clustering, in which we have used the same number of genes of the final subsets selected by MOGA as representative centroids ($K = NG$). Then, each resulting subset (gene centroids) is used to train the SVM classifier (configured as explained in Section 4.2), and cross-validated with an external test set.

For this purpose, we used the K-Means procedure available in Weka tools for data mining [23]. A number of 30 independent runs were performed with 2000 iterations of each one in order to obtain as accurate a result as possible. Table 4 shows the results obtained by K-Means and MOGA in terms of mean and standard deviation of the accuracy percentage.

We can observe in this table that MOGA clearly outperforms the K-Means procedure in all the datasets. Specifically, the difference regarding the accuracy percentage in the lymphoma dataset (57.89 ± 1.11 of K-Means in con-

Table 4
Comparison with base-line method: K-Means clustering. The columns indicate the average and standard deviation of the number of selected genes (*NG*), and accuracy rate (*Ac.*). T_S indicates the percentage of samples of the external test set.

Dataset	*NG*	K-Means *Ac.* (%)	MOGA 3 obj. *Ac.* (%)	T_S (%)
Leukemia	6 ± 2.64	85.29 ± 1.02	$\mathbf{98.03 \pm 1.61}$	47%
Colon	3.33 ± 1.52	78.12 ± 2.70	$\mathbf{89.58 \pm 1.80}$	50%
Lymphoma	3.75 ± 1.50	57.89 ± 1.11	$\mathbf{96.05 \pm 3.04}$	50%

trast with 96.05 ± 3.04 of MOGA 3 obj.) gives us some insights into the power of our proposal. We would expect these differences in results, since K-Means is a naive method without any information about the problem in its procedure. For this reason, we have carried out further comparisons with two related metaheuristic approaches found in the literature.

The first work, Liu and Iba (2002) [8], consists of a multiobjective evolutionary algorithm which optimizes simultaneously 3-objectives: the misclassification rate, the difference in the error rate among the classes, and the number of selected genes. In the second approach, Hernandez et al. (2007) [22], a genetic algorithm embedded with a pre-filtering criteria is used, which optimizes an aggregative function using the accuracy rate and the number of genes.

Table 5 summarizes our results together with those reported in [8] and [22] on the same three datasets. As shown in bold face, the accuracy rate reported by our MOGA is the best in all the datasets, although it is clear that the number of genes selected by [22] in the Leukemia dataset is smaller. In addition, our results are competitive with respect to those reported in a third work [7]. For this reason, we can claim that our approach shows an efficient and better performance in comparison with existing state of the art algorithms.

4.6. Biological analysis

Finally, in this section we provide a biological analysis of the computed gene subsets. Although this biological study is very important, only a few papers considered it in the past [24,25]. Most articles addressed just the machine learning problem regardless of the actual meaning of the datasets. We will show here the much broader impact of our technique, capable of computing real biological ensembles of genes that have been suggested in the domain only (e.g., Science [26]).

In Fig. 5, a graphical distribution of the most frequently obtained genes in 30 independent executions of the MOGA (in 3-objective mode) are reported. We have used the Leukemia dataset, since it is the one commonly studied by other related works in the literature. In this figure, we highlight in bold face the genes also reported in the list of the 30 most important genes (selected from 7129 in Leukemia) suggested in Golub et al. [26]. In Table 6, we arrange these genes by means of the rank assigned in the Golub et al. list (column 1 in the referenced table).

The genes reported in Table 6 were also selected as the most informative genes in recent specialized works. Specifically, Dramiński et al. [24] used a Monte Carlo method for feature selection and supervised classification on Leukemia and Lymphoma datasets. Wang and Zhu [25] proposed a Nearest Shrunken Centroid (NSC) classifier on the Leukemia dataset. Both works assign the gene M31166_at great importance, which matches with our main results. In addition, in Table 7 we present a list of

Table 5
Comparison with other authors. The columns indicate the average and standard deviation of the number of selected genes (*NG*), the accuracy rate (*Ac.*), and the number of solutions in the final Pareto front (*NSf*). T_S indicates the percentage of samples of the external test set.

Dataset	Liu and Iba (2002) [8]			Hernandez et al. (2007) [22]			MOGA 3 obj.		
	NG	*Ac.* (%)	T_S (%)	*NG*	*Ac.* (%)	T_S (%)	*NG*	*Ac.* (%)	T_S (%)
Leukemia	15.2 ± 4.54	90.00 ± 7.00	30%	3.17 ± 1.16	91.5 ± 5.9	47%	6 ± 2.64	$\mathbf{98.03 \pm 1.61}$	47%
Colon	11.4 ± 4.27	80.00 ± 8.3	30%	7.05 ± 1.07	84.6 ± 6.6	50%	3.33 ± 1.52	$\mathbf{89.58 \pm 1.80}$	50%
Lymphoma	12.9 ± 4.40	90.00 ± 3.4	30%	5.29 ± 1.31	93.3 ± 3.1	50%	3.75 ± 1.50	$\mathbf{96.05 \pm 3.04}$	50%

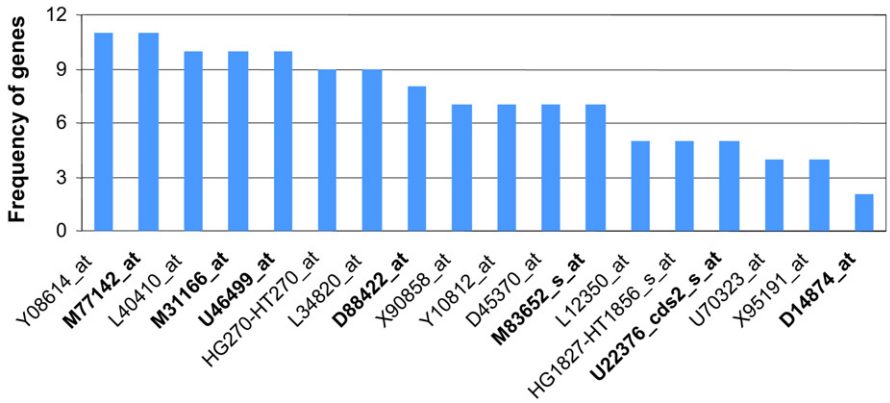


Fig. 5. Distribution of the most frequently obtained genes (in 30 independent executions) by our MOGA in Leukemia dataset.

Table 6
Top 7 genes ranked with MOGA (3-objectives) also reported in the list of the 30 most important genes suggested by Golub et al. on the Leukemia dataset.

Rank	Index	Accession	Gene description
2	1926	M31166_at	"PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta"
5	760	D88422_at	CYSTATIN A
13	2233	M77142_at	NUCLEOLYSIN TIA-1
17	3256	U46499_at	"GLUTATHIONE S-TRANSFERASE, MICROSOMAL"
18	6379	M83652_s_at	"PFC Properdin P factor, complement"
22	249	D14874_at	ADM Adrenomedullin
29	5772	U22376_cds2_s_at	"C-myb gene extracted from Human (c-myb) gene"

Table 7
New genes suggested in this work for the Leukemia dataset.

Index	Accession	Gene description
5975	J03778_s_at	MICROTUBULE-ASSOCIATED PROTEIN TAU
1584	L40410_at	Thyroid receptor interactor (TRIP8) mRNA, 3′ end of cds
865	HG270-HT270_at	Lymphocyte Chemoattractant Factor
1494	L34820_at	NAD+-dependent succinate-semialdehyde dehydrogenase (SSADH) mRNA, 3′ end
4780	X90858_at	Uridine phosphorylase
5021	Y10812_at	GB DEF = Fructose-1,6-bisphosphatase
442	D45370 at	ApM2 mRNA for GS2374 (unknown product specific to adipose tissue)
1295	L12350_at	THBS2 Thrombospondin 2
5731	HG1827-HT1856_s_at	Cytochrome P450, Subfamily Iic, Alt. Splice Form 2
3635	U70323_at	SCA2 Spinocerebellar ataxia 2
4835	X95191_at	GB DEF = Delta-sarcoglycan

new genes that we consider to be important, since they were the ones that overlapped most in the resulting subsets (together with the ones shown in Table 6) in our experiments.

5. Conclusions

In this paper, we propose the use of the sensitivity and the specificity rates, in addition to the number of selected genes, as three main objectives to optimize when facing the gene selection and classification of DNA Microarrays. We have used a NSGA-II based MOGA algorithm which evolves optimizing simultaneously these three objectives. In this algorithm, the classification task is accomplished by SVMs, and 10-fold cross-validation is applied to the resulting subsets to evaluate the solutions. The effectiveness of this approach is proved on public Microarray datasets (Leukemia, Lymphoma, and Colon).

The first statistical analysis confirms that breaking up the accuracy factor among the sensitivity and the specificity factors can increase, in terms of quality and diversity, the number of good solutions. The comparisons presented in Table 2 show the difference in the accuracy percentage and number of solutions of both strategies: 2 and 3 objectives.

The accuracy percentages of 98.03 ± 1.61 obtained by MOGA 3 obj., in contrast with the ones of 95.27 ± 2.21 obtained by MOGA 2 obj. for Leukemia guarantee the usefulness of our proposal. In a second analysis, we compare our approach with a naive method based on K-Means clustering, and with two related multiobjective approaches. Our results suggest that the MOGA 3 obj. is highly appropriate for solving the gene selection, outperforming the compared techniques for all the datasets.

A final biological analysis reports a list of the most representative genes selected by our approach. We can notice that seven of these genes were also reported as the most relevant ones in the original work of Golub et al. concerning the Leukemia dataset. Specifically, the M31166_at gene was also considered very important in related works, which is consistent with our results.

As to future work, we are interested in evaluating our algorithm in new Microarray datasets, and plan to test and compare different MOGA approaches (IBEA, SPEA, etc.) in order to offer fresh points of view to this problem.

References

[1] A. Pease, D. Solas, E. Sullivan, M. Cronin, C.P. Holmes, S. Fodor, Light-generated oligonucleotide arrays for rapid DNA sequence analysis, Proc. Natl. Acad. Sci. USA 96 (1994) 5022–5026.
[2] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1–3) (2002) 389–422.
[3] M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (1977) 917–922.
[4] E. Alba, J. García-Nieto, L. Jourdan, E.-G. Talbi, Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, in: IEEE Congress on Evolutionary Computation CEC-07, Singapore, 2007, pp. 284–290.
[5] E.B. Huerta, B. Duval, J.-K. Hao, A hybrid GASVM approach for gene selection and classification of microarray data, in: F. Rothlauf, et al. (Eds.), EvoWorkshops, in: LNCS, vol. 3907, Springer, 2006, pp. 34–44.
[6] T. Juliusdottir, D. Corne, E. Keedwell, A. Narayanan, Two-phase EA/K-NN for feature selection and classification in cancer microarray datasets, in: CIBCB, 2005, pp. 1–8.
[7] K. Deb, A. Raji, Reliable classification of two-class cancer data using evolutionary algorithms, BioSystems 72 (2003) 111–129.
[8] J. Liu, H. Iba, Selecting informative genes using a multiobjective evolutionary algorithm, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC '02, vol. 1, 2002, pp. 297–302.
[9] M. Banerjee, S. Mitra, H. Banka, Evolutionary rough feature selection in gene expression data, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 37 (4) (2007) 622–632.
[10] C. Metz, Basic principles of ROC analysis, Seminars in Nuclear Medicine 8 (4) (1978) 283–298.
[11] A.J. Alberg, J.W. Park, B.W. Hager, M.V. Brock, M. Diener-West, The use of "overall accuracy" to evaluate the validity of screening or

diagnostic tests, J. of General Internal Medicine 19 (5) (2004) 460–465.
[12] M. Kupinski, M. Anastasio, Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves, IEEE Trans. on Medical Imaging 18 (8) (1999) 675–685.
[13] R.M. Everson, J.E. Fieldsend, Multi-class ROC analysis from a multi-objective optimisation perspective, Pattern Recogn. Lett. 27 (8) (2006) 918–927.
[14] H. Liu, H. Motoda, Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer Academic Publishers, Norwell, MA, USA, 1998.
[15] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[16] T. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machines classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[17] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. on EC 6 (2) (2002) 182–197.
[18] C. Emmanouilidis, A. Hunter, J. MacIntyre, A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator, in: IEEE Congress on Evolutionary Computation, California, USA, 2000, pp. 309–316.
[19] L. Jourdan, C. Dhaenens, E.G. Talbi, S. Gallina, A data mining approach to discover genetic and environmental factors involved in multifactorial diseases, Knowledge-Based Systems 15 (4) (2002) 235–242.
[20] A. Liefooghe, M. Basseur, L. Jourdan, E.-G. Talbi, ParadisEO-MOEO: A framework for evolutionary multi-objective optimization, in: EMO, in: LNCS, 2007, pp. 386–400.
[21] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, Software available at URL http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2002.
[22] J. Hernandez, B. Duval, J.-K. Hao, A genetic embedded approach for gene selection and classification of microarray data, in: E. Marchiori, et al. (Eds.), EvoBIO, in: LNCS, 2007, pp. 90–101.
[23] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, M. Kaufmann, 2005.
[24] M. Dramiński, A. Rada-Iglesias, S. Enroth, C. Wadelius, J. Koronacki, J. Komorowski, Monte Carlo feature selection for supervised classification, Bioinformatics 24 (1) (2008) 110–117.
[25] S. Wang, J. Zhu, Improved centroids estimation for the nearest shrunken centroid classifier, Bioinformatics 32 (2) (2007) 972–979.
[26] R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

POLYMERS FOR ADVANCED TECHNOLOGIES
Polym. Adv. Technol. 2008; **19**: 762–769
Published online 13 February 2008 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/pat.1030

PAT

New analysis of the optimization of electromagnetic shielding properties using conducting polymers and a multi-objective approach

Oliver Schütze[1]*, Laetitia Jourdan[2], Thomas Legrand[2], El-Ghazali Talbi[2] and Jean-Luc Wojkiewicz[3]

[1]Computer Science Department, CINVESTAV-IPN, Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07300, Mexico
[2]INRIA Futurs, LIFL, CNRS Bât M3, Cité Scientifique, 59655 Villeneuve d'Ascq Cedex, France
[3]Ecole des Mines de Douai, 941, rue Charles Bourseul, BP 10838 - 59508 Douai Cedex, France

Received 30 August 2007; Revised 18 October 2007; Accepted 22 October 2007

Recently, an approach was proposed to optimize multi-layer shields of polyaniline–polyurethane (PAni/PU) conducting composites in the microwave band. Though by this method shields for different applications can be obtained which are light-weight and offer a low percolation threshold, the full potential of the design process could not be tapped since the underlying optimization problem includes only one objective. In this work we go one step beyond and re-formulate the design problem as a *multi-objective optimization problem* (MOP). To be more precise, we involve simultaneously the shielding efficiency as well as the weight and the cost of the material—i.e. all the requirements for modern shielding materials—within the optimization process. After having stated the model we present two possible ways to approximate the solution set—the so-called *Pareto set*—and address the related and important decision-making problem. All steps are demonstrated on a particular three-layered composite in order to show the applicability of the novel approach. **Copyright © 2008 John Wiley & Sons, Ltd.**

KEYWORDS: conducting polymers; polyurethane; electromagnetic shielding; multi-objective optimization; decision making

INTRODUCTION

Electromagnetic interferences have become an important problem due to the proliferation of commercial, military and scientific electrical devices, and equipment in high frequencies. Electronic devices must be shielded to be protected against the incoming and potentially disturbing radiation.

Conducting polymer composites (CPCs) like polyaniline–polyurethane (PAni/PU) are very promising for applications in electromagnetic interference shielding.[1–3] These materials are, e.g. characterized by relatively high conductivities and permittivities. Since these properties can easily be tuned via chemical processes in the making of these composites, CPCs are well suited for the demanding optimization in this field. Further, these materials are lighter, more flexible, and offer better environmental stability compared to the classical shielding materials subjected to corrosion which make them an interesting potential alternative. A first step to optimize the materials has been done by Hoang *et al.*[4], where the target objective is to achieve the desired shielding efficiency of the multi-layered compound for a given application based on a particular theoretical model of these materials. As a side effect, the resulting shields are typically characterized by suitable low thicknesses of the layers, resulting in a light-weight compound.

In this work, we are particularly interested in the design of new high-protecting and light-weight materials which are realizable for reasonable prices. In search of these materials, we extend the approach of Hoang *et al.*[4] and propose in the following a new multi-objective optimization model which involves all the desired objectives at once. Further, we address the numerical treatment of these problems, and present possible techniques which are designed to support the decision maker (DM) to find the preferred solution according to the specific problem. The applicability of the new approach—approximation of the solution set and in particular the decision making process—is demonstrated on a three-layered material.

The remainder of this work is organized as follows: in Section ''Experimental'' we state the background required for the understanding of the particular design problem which is proposed afterwords. Section ''Results and discussion'' deals with the approximation of the Pareto sets of the resulting multi-objective optimization problems (MOPs), and we further show how these sets can be visualized according to the preference of the DM. Finally in our conclusion we address topics of future research.

**Correspondence to*: O. Schütze, Computer Science Department, CINVESTAV-IPN, Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07300, Mexico.
E-mail: schuetze@cs.cinvestav.mx

Copyright © 2008 John Wiley & Sons, Ltd.

EXPERIMENTAL

Background

In this section we briefly summarize the background required for this work: we introduce the electromagnetic properties which are interesting in our context, present a theoretical model for these properties which serves as the basis for further considerations, and finally address the concept of multi-objective optimization.

Electromagnetic properties

Since our aim is to design new protecting materials we are particularly interested in what happens when an electromagnetic wave (EM) arrives at the surface of a material. In that case, three physical phenomena can occur: *absorbtion*, *reflection*, and *transmission* of the incidental wave (Fig. 1).

For our purpose it is sufficient just to consider the reflection and the transmission. Naishadham[5] proposed a theoretical model for these two wave interactions which will be used in this work and which will be described in the following. For this, we consider a compound consisting of N layers and assume each layer to be homogeneous and isotropic. The design parameters of the ith layer, $i = 1, \ldots, N$, are the conductivity σ_i, the permittivity ε_i, and the thickness d_i of the material of each layer.

The *characteristic matrix* $\mathbf{M}_i \in \mathbb{C}^{2\times 2}$ of the ith layer is given by:

$$\mathbf{M}_i = \begin{bmatrix} \cos(A_i) & -jZ_i\sin(A_i) \\ -\frac{j}{Z_i}\sin(A_i) & \cos(A_i) \end{bmatrix}$$

where

$$A_i = \omega d_i\sqrt{\mu_0\varepsilon_0[\varepsilon_i - j(\sigma_i/\omega\varepsilon_0)]}, \qquad Z_i = \sqrt{\frac{\mu_0}{\varepsilon_0[\varepsilon_i - j(\sigma_i/\omega\varepsilon_0)]}}$$

with $\omega = 2\pi f$, where f is the frequency of the EM and j denotes the imaginary unit. Z_i is the impedance of the ith layer. Due to their contact to air media, the impedances of the outer layers are set to $Z_0 = Z_{N+1} = 377(\Omega)$.

The characteristic matrix of the entire compound is given by the product of the characteristic matrices for each layer, i.e.

$$\mathbf{M} = \mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \ldots \cdot \mathbf{M}_N = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

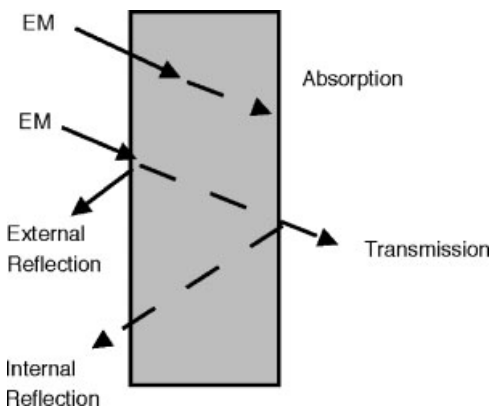


Figure 1. The three kinds of physical wave interaction.

Now we are in the position to state the coefficients for the reflection R and the transmission T:

$$R = \frac{(M_{11}Z_0 - M_{12}) - Z_0(M_{22} - M_{21}Z_0)}{(M_{11}Z_0 - M_{12}) + Z_0(M_{22} - M_{21}Z_0)} \tag{1}$$

and

$$T = \frac{2[M_{22}(M_{11}Z_0 - M_{12}) + M_{12}(M_{22} - M_{21}Z_0)]}{(M_{11}Z_0 - M_{12}) + Z_0(M_{22} - M_{21}Z_0)} \tag{2}$$

Multi-objective optimization

In a variety of applications in industry and finance a problem arises that several objective functions have to be optimized concurrently. One important feature of these problems is that the different objectives typically contradict each other and therefore certainly not have identical optima. As an example related to the current study, a material offering a high-shielding efficiency (say, $\geq$80 dB) may not be the cheapest nor the lightest one, and vice versa. Thus, the question arises how to approximate one or several particular ''optimal compromises'' (e.g. by *interactive methods*[6]) or how to compute *all* optimal compromises of this MOP. For this, for instance a huge variety of mathematical[6,7] and heuristic[8,9] strategies have been proposed during the last years.

Mathematically speaking, an MOP can be stated in its general form as follows:

$$\min_{x\in S}\{\mathbf{F}(x)\}, \quad S = \{x \in \mathbb{R}^n : h(x) = 0,\ g(x) \leq 0\}$$

where $\mathbf{F}$ is defined as the vector of the objectives, i.e.

$$\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^k, \quad \mathbf{F}(x) = (f_1(x), \ldots f_k(x))$$

with $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $m \leq n$, and $g : \mathbb{R}^n \to \mathbb{R}^q$. A vector $\mathbf{v} \in \mathbb{R}^k$ is said to be *dominated* by a vector $\mathbf{w} \in \mathbb{R}^k$ if $\mathbf{w}_i \leq \mathbf{v}_i$ for all $i \in \{1, \ldots, k\}$ and $\mathbf{v} \neq \mathbf{w}$ (i.e. there exists a $j \in \{1, \ldots, k\}$ such that $\mathbf{w}_j < \mathbf{v}_j$). A vector $\mathbf{v}$ is called *nondominated* with respect to a set P, if none of the vectors $\mathbf{p} \in P$ dominate $\mathbf{v}$.

A point $x \in S$ is called optimal or *Pareto optimal*[1], if $\mathbf{F}(x)$ is not dominated by any vector $\mathbf{F}(y), y \in S$. The solution set—the so-called *Pareto set*—consists typically not of finitely many points as for scalar optimization problems, but forms a $(k-1)$-dimensional object. The image of the Pareto set is called the *Pareto front*. Figure 2 displays the two sets of the following MOP:

$$\mathbf{F} : \mathbb{R} \to \mathbb{R}^2$$
$$\mathbf{F}(x) = \begin{pmatrix} (x-1)^2 \\ (x-1)^2 \end{pmatrix} \tag{3}$$

Since two objectives are under consideration, both Pareto set and Pareto front form curves in their spaces. Here, both sets consist of one connected component, which, however, does not have to be the case for a given MOP.

The design problem

In this section we propose a novel multi-objective model for the design of CPCs for shielding electrical devices. We aim in particular at high-shielding *and* light-weight materials

[1]Named after the economist Vilfredo Pareto, 1848–1923.

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

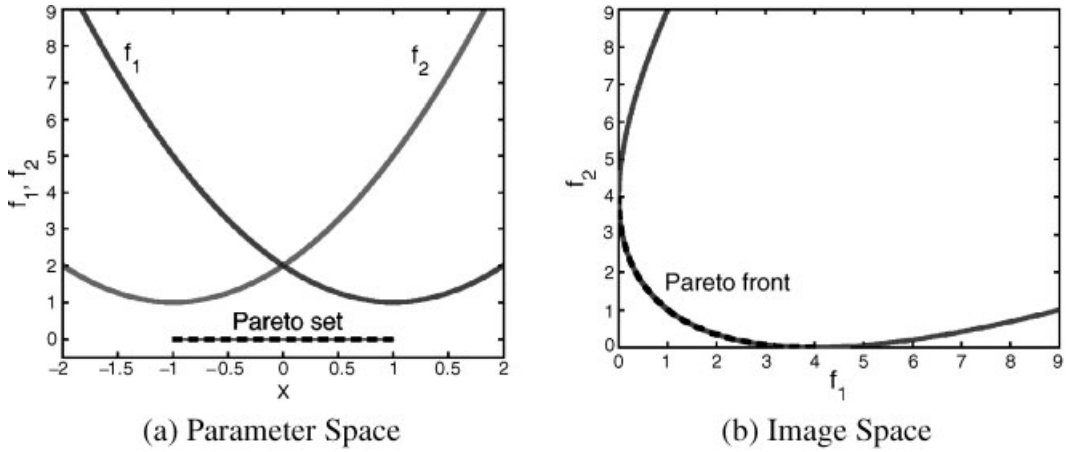


Figure 2. Pareto set (a) and Pareto front (b) of MOP (3) within the domain $Q = [-2, 2]$.

since there seems to be a growing interest in alternatives to classical materials like metals, which are too heavy, e.g. for aeronautic applications.

The electromagnetic shielding of CPCs and, partly, the related optimization have been considered in some works so far.[5,10,11] Albeit this is of course the most important feature of this material, the mono-objective approach reveals some limitations since it does not consider other physical properties of the compound which are getting more and more important for commercial products (as weight and cost).

Now we propose the objectives—without loss of generality formulated as minimization problems—which have to be considered in search for modern CPCs for the shielding of electronic devices.

The first objective is the *electromagnetic shielding*, i.e. the ''classical'' objective, which can be expressed as follows[5]:

$$f_s(x) = 20log(|T|) \tag{4}$$

where T is the transmission coefficient defined in eqn (2).

Alternatively, it can be desirable to aim in particular for a high reflection coefficient [refer to eqn (1)], which leads to the objective

$$f_r = -|R| \tag{5}$$

Since reflection and transmission of an EM are closely related, only one of these two objectives—depending on the preference of the DM—is required for the formulation of the MOP.

Next, we propose to take the *mass percentage* of a particular material (*polyaniline*) inside the polymer compound in each layer into account since this value is highly responsible for the (relatively high) cost of these composites. Thus, to be efficient and realizable, the materials must have a small mass percentage, which leads to the following minimization problem[4]:

$$f_p = log\left(\sum_i^N p_i\right), \quad p_i = \left(\frac{\sigma_i}{\sigma_0}\right)^{\frac{1}{t}} + \text{pc}, \; i = 1, \ldots, N \tag{6}$$

where σ_0 is a reference conductivity, pc the percolation threshold, and t a critical exponent. p_i is the mass percentage of the ith layer.

Finally, we propose to take the thickness of the compound into account since this has a direct influence on the weight and the cost of the resulting material. Thus, the fourth objective reads as follows:

$$f_t = \sum_i^N d_i \tag{7}$$

There exist of course other possible goals as well as other models which could be interesting for particular applications and which cannot all be stated here. However, the objectives presented above seem to be the most generic ones.

RESULTS AND DISCUSSION

Approximation of the Pareto fronts

In this section we introduce the two methods which were used and adapted to compute the Pareto fronts of the MOPs which grew out of the design problem under consideration. Since so far mainly compounds with few layers are being studied, we are faced with low or moderate dimensional models which do not represent a challenge to state of the art evolutionary (multi-objective) algorithms. Consequently, the approximation of the solution sets is not the main contribution in this work; however, this is and will be one important task in multi-objective optimization, and has to be accomplished thoroughly.

Legrand[12] proposed a multi-objective evolutionary algorithm (MOEA) which is designed for the present context. The genetic algorithm used for the optimization of electromagnetic shielding properties allows to obtain diversified and pertinent results. For all the steps of the algorithm, a satisfying diversity of the population was maintained which allowed to present a large number of different solutions to the DM. In the first step of this algorithm, the components of each individual belonging to the initial population are generated at random from subdivided intervals in order to gain homogeneity. The selection step combines two popu-

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

lations, the current one and another one which is stored in a Pareto archive in an elitist manner. In this algorithm, the crossover and mutation operators which are presented by Chelouah and Siarry[13] are used as they have proven their efficiency on continuous optimization problems. Excellent results have been obtained on different benchmarks with genetic algorithm using these operators; their flexible configuration represents an advantage to get varied components. Furthermore, these operators have not increased the time of computation. Also, the obtained individuals had diversified components and the results in the Pareto fronts offered a large palette of solutions to the DMs.

The algorithm has been developed using the platforms EO[14] and its extension ParadiseEO[15]

In order to compare the results obtained by the method descibed above, we have alternatively used and adapted *subdivision techniques* [17,18] for this problem. These techniques have been primarily designed for unrestricted MOPs and work particularly well for moderate dimensions, i.e. when few layers are considered in the compound.

Algorithm 1: MOEA for shielding design problems

1: choose initial population P_0
2: set $A_0 := $ nondominated points of P_0 (archive)
3: $i := 0$
4: **repeat**
5: compute P_{i+1} by the following steps
  – perform NSGA selection from P_i and A_i as proposed in Reference 16
  – perform Crossover and Mutation as described in Reference 13
  – perform the generational replacement
6: $A_{i+1} := $ nondominated points of $A_i \cup P_{i+1}$
7: $i := i+1$
8: **until** (stopping criteria fulfilled).

The algorithms of this type start with a compact subset $Q \subset \mathrm{D}$ of the domain, represented by a collection of n-dimensional boxes (where n is the dimension of D). Each box gets subdivided into smaller sub-boxes and after certain conditions it is decided if a box is promising—i.e. if it could contain a part of the Pareto set—or not. The ''unpromising'' boxes are deleted from the collection while the process—subdivision and selection—is continued successively on the remaining boxes until the desired granularity of the boxes is reached. In our design problem the minimal radii of the boxes are given in a natural way by the manufacturing accuracy of the material (which in turn results in a certain accuracy for the parameters ε and σ).

The approach is of global nature, i.e. in principle capable of detecting the entire Pareto set. However, it is restricted to moderate dimensions and not rigorous. That is, boxes which are deleted once from the collection but contain a part of the Pareto set will not be reconsidered in further iteration steps. Schütze *et al.*[19] described a variant which hybridizes with a MOEA in order to reduce this problem, which allows to attack higher dimensional and more complicated models, and which was used to compute the Pareto sets of the present design problems. It is planned to integrate the MOEA described above into the subdivision techniques in order to unite the strengths of both algorithms.

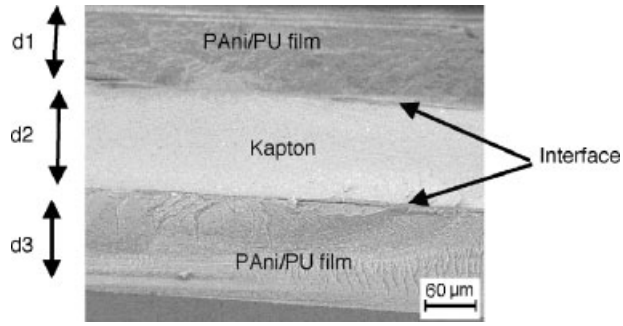


Figure 3. Electron microscope image of the three-layered material under consideration.

Example: a three-layered material

As a general test example which will serve for the remainder of this work we have chosen a three-layered material which is a compound of PAni/PU and Kapton™ (refer to Fig. 3).[2] We have decided to include Kapton into the compound since it is a polymer which offers high chemical resistance and good interaction with the PAni/PU solution in the chemical production process.[20] Using this compound and the model described above this leads to a design problem with four free parameters (refer to Table 1). Due to the low dimensionality of the parameter the corresponding model is easy to handle with our (and other) algorithms, and we are able to include all four objectives proposed above into the design problem. Doing so, this leads to the MOP

$$\min_x F_c : Q \subset \mathbb{R}^4 \to \mathbb{R}^4 \tag{8}$$

where Q is the hyper-rectangle which is given by the box constraints shown in Table 1.

Figure 4 shows two projections of a front as well as a short discussion. We have chosen $f = 50\,\mathrm{MHz}$ for the frequency of the incoming wave.[3] The results are certainly highly satisfying—from the point of view of the developer of the optimization algorithm. However, it is *ad hoc* more than doubtful if and how this huge amount of data can help the DM to find the ''right'' material according to the given problem. Therefore, the next section deals with the problem specific visualization of these solution sets.

Since this multi-objective approach to the shielding problem is novel and since every application has its special environmental peculiarities, the result of this optimization can hardly be compared to existing materials documented in the literature (but is a task for future work). However, the results seem to be promising regarding (a) the large portion of the front where international standards for the shielding efficiency[21,22] are satisfied, and (b) the significant diversity

[2]Kapton is a registered trademark of DuPont.
[3]Since the frequency of the incidental wave may vary, it is desired in this situation to have a high shielding efficiency in the entire frequency range between $5.0 \times 10^6$and 1.0×1.0^9 Hz. Refer discussion below.

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

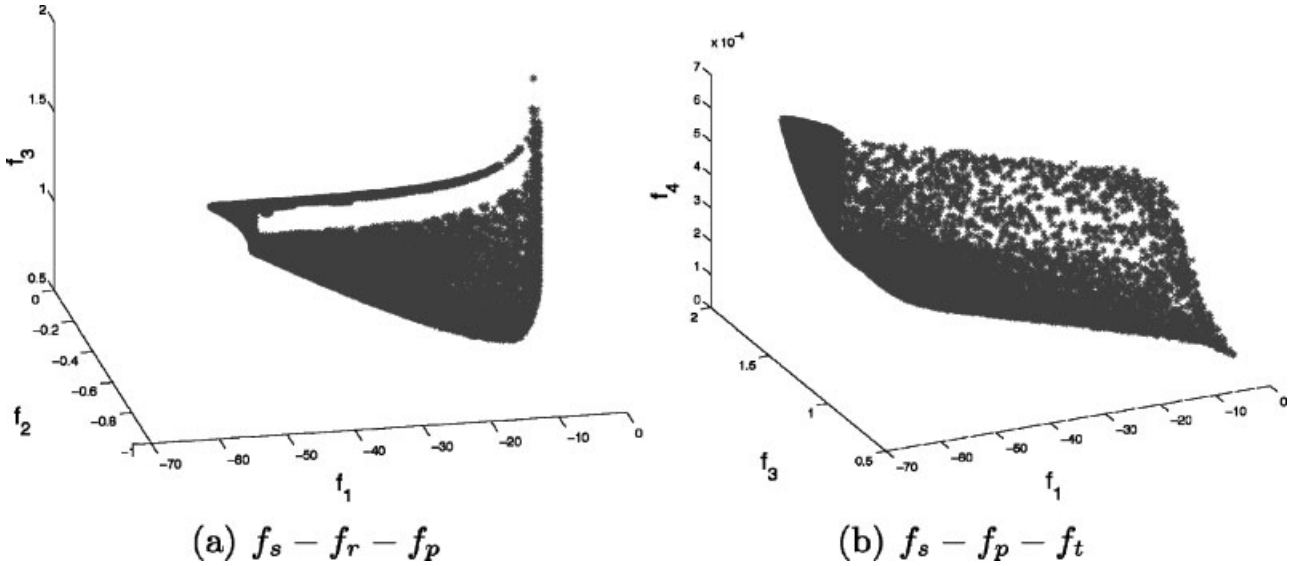


Figure 4. Two projections of a Pareto front of MOP (8): subfigure (a) displays the projection f_s–f_r–f_p and (b) displays the projection f_s–f_p–f_t. In order to obtain this set we have used the algorithm *EA=Subdivision* proposed by Schütze *et al.*[19] which has used a total of 724,600 fitness evaluations. Since the objective values are given in short analytical form the computation took less than one minute on a standard computer. Using the MOEA described above very similar results were obtained.

Table 1. Parameters MOP (8)

No. of layer	Material	ε	σ (S m^{-1})	d (μm)
1	PAni/PU	0.0	From 30 to 10^4	From 0 to 300
2	Kapton	3.1	0	125
3	PAni/PU	0.0	From 30 to 10^4	From 0 to 300

Since the permittivities of the outer layers are fixed and we consider Kapton as the second layer (refer to Hoang *et al.*[11] for a motivation of this choice), merely four free design parameters have to be considered.

with respect to f_p and f_t, which influence cost and weight of the material. An example for the latter can be seen in Table 2. A motivation for the choice of the points—we have chosen $x_i \in B_i, i = 1, 2$, refer to Fig. 6—is given in the next section. The material related to point x_1 reaches the maximal value for the thickness and is further relatively expensive due to the high mass percentage ($p_1 = 31.27$ %m and $p_3 = 31.30$ %m for the 1st and 3rd layer, respectively) but offers in turn a shielding efficiency of 61.20 dB. The material which is given by x_2 is much thinner and less expensive due to a significant lower mass percentage ($p_1 = 22.39$ %m and $p_3 = 2.92$ %m). Practically, this indicates that in fact the third layer is not required in case the function values of x_2 have been selected which would lead to further reduction of the cost due to a simplification of the making process.

Table 2. Two selected solutions of MOP (8) with different properties

Point	f_s (dB)	f_r (dB)	f_p (%m)	f_t (μm)	d_1 (μm)	σ_1 (S m^{-1})	d_3 (μm)	σ_3 (S m^{-1})
x_1	−61.21	−0.9991	1.80	598.4	298.6	9945.5	299.8	9965.7
x_2	−40.86	−0.9909	1.40	127.9	126.5	4587.8	1.3	37.3

Selecting the preferred material

In this section we present two possible ways to present the Pareto fronts of a given design problem in a way that allows the DM to obtain a suitable, problem specific, and maybe subjective overview of the available possibilities, and thus, to help to find the preferred solution. In the following we report on the applicability of an existing visualization tool which offers an unbiased overview on the entire front and propose alternatively a new way of the visualization of two-dimensional (2D) solution sets which can involve preferences of the DM.

The PARETO FRONT VIEWER[4] is based on the interactive decision maps technique[23] and was particularly developed for the exploration of Pareto fronts with more than two objectives. In this approach, a nonnegative cone is added to every point of the approximation of the Pareto front. The combination of these cones approximates the Edgeworth–Pareto Hull (EPH). By displaying various *decision maps*—i.e. collections of 2D slices of the EPH—which are depicted on the value of a third objective, the decision maps help to understand the influence of this objective. The influence of further objectives can be experienced by using sliders, which move the efficiency frontiers described above according to the values of these objectives. Though 2D maps are used for the visualization of the fronts, this approach allows the DM to handle multiple objective values. Refer to Lotov *et al.*[23] for various applications where up to nine objectives are involved.

[4]For further description and a demo version of the software tool see http://www.ccas.ru/mmes/mmeda/mcdm.htm

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

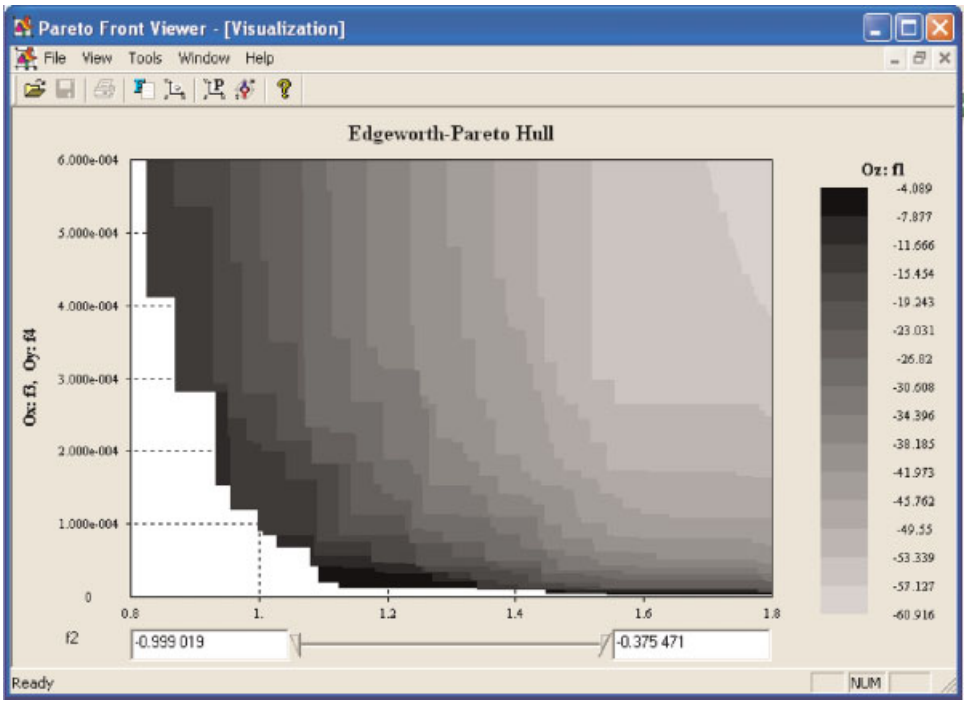


Figure 5. Snapshot of the PARETO FRONT VIEWER which shows a visualization of a Pareto front which is similar to the one displayed in Fig. 4. This figure is available in color online at www.interscience.wiley.com/journal/pat

Figure 5 shows a snapshot of a decision map which displays a Pareto front of MOP (8). The values of f_m and f_p are plotted in the x-axis and y-axis, respectively. The map shows an amount of 15 such efficiency frontiers using different values of the shielding efficiency. Finally, these maps can be moved according to the value of the reflection coefficient. Using this tool, a good understanding of the criterion tradeoff can be obtained.

One important requirement for a proper visualization of a Pareto front is certainly to tame the complexity of the huge

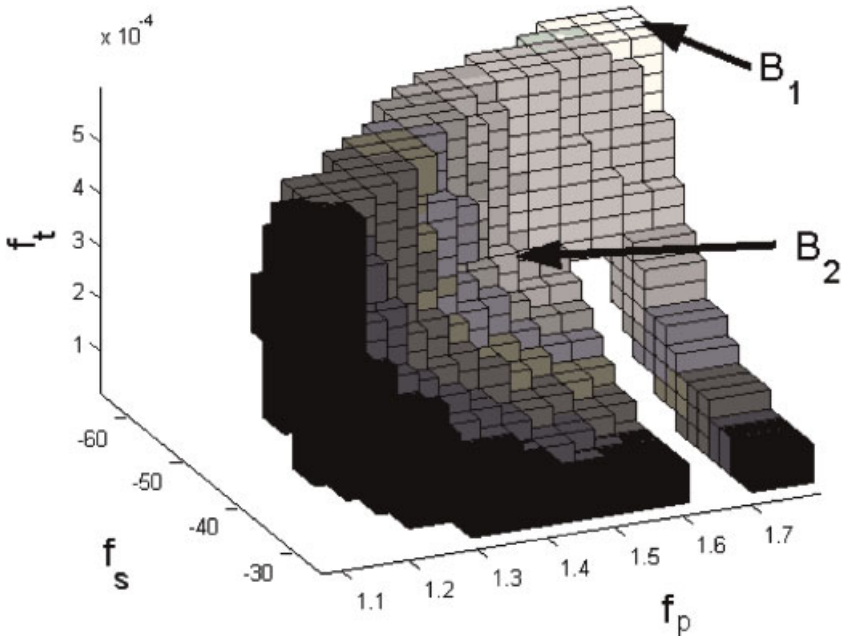


Figure 6. Classification of an obtained Pareto front consisting of 15,987 non-dominated solutions (refer to Fig. 4) into 469 different boxes which are 15 different tones displayed in ranging from black (corresponding to the smallest value of the indicator) to white (highest value). This figure is available in color online at www.interscience.wiley.com/journal/pat

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

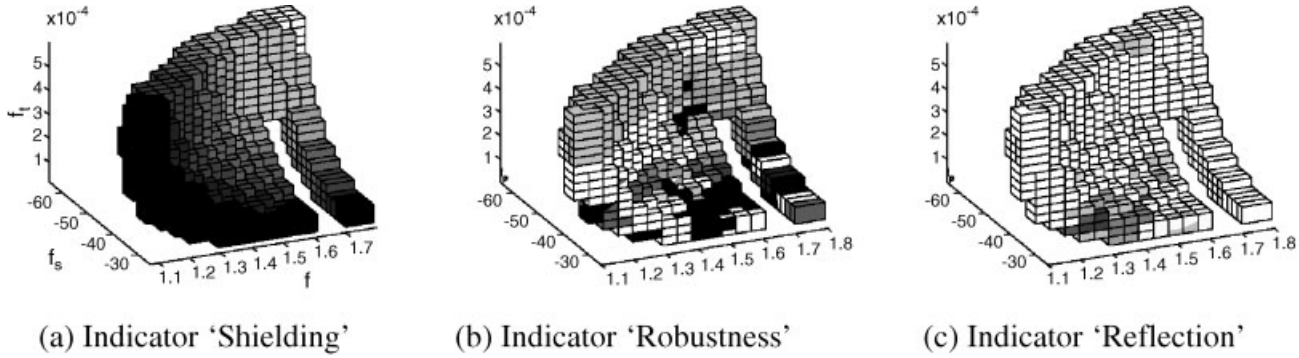


Figure 7. Further examples of possible shading of the Pareto front leading to different results: in (b) the value of $(\partial \mathrm{SE}/\partial d_1) + (\partial \mathrm{SE}/\partial d_3)$ is taken as an indicator for the robustness of the production with respect to a manufacturing error in the thickness of the PAni/PU compounds, while for (c) the value of the reflection coefficient f_r is used. In (a) the shielding indicator discussed above is shown for comparison.

amount of data, in particular when more than two objectives are under consideration. For this, the authors of this work share the opinion that it makes sense for this application—and certainly for others as well—to introduce an additional, problem-specific indicator which can (hopefully) help the DM to identify parts of the Pareto front which are potentially promising for the current situation.

Let us consider one example. The multi-objective model presented above fixes the frequency of the incoming wave. However, since it is desired to have a high-shielding efficiency in an entire frequency range $[f_{\min}, f_{\max}]$, one could, e.g. consider a ''shielding indicator'' as follows:

$$I_s(x) := \int_{f_{\min}}^{f_{\max}} f_s(x, f)\,df \tag{9}$$

where $x = (\varepsilon, \sigma, d) \in \mathbb{R}^{3N}$ and $f_s\ (\cdot,\cdot)$ denotes the shielding efficiency analog to eqn (4). If the value of $I_s(x)$ is high, it does obviously not follow that $f_s(x, f)$ is high for all relevant frequencies, but the underlying heuristic is that in this case the values of f_s must be high in at least one subregion.

Further examples for possible indicators in the current design problem are, e.g. the value of the reflection, the cost or the weight of the material, the robustness of the material against possible errors in the production process, etc. Refer to Figs 6, 7 for examples.

For the visualization of the 2D fronts in combination with a property indicator (or without), we propose to use *boxes* since they bring the required three-dimensional (3D) effect to the appearance of the fronts.[5] Further, they can be used to reduce the complexity of the data since several ''neighboring'' points are collected in one box, depending on the location and the size of the boxes which can be both adjusted according to the problem.

A k-dimensional box B can be represented by a center $c \in \mathbb{R}^k$ and a radius $r \in \mathbb{R}^k$:

$$B = B(c, r) = \{x \in \mathbb{R}^k : |x_i - c_i| \le r_i\ \forall i = 1, \ldots, k\}$$

In order to obtain a clear view of these new designed fronts it is certainly advantageous to build a box collection where the interiors of its boxes are mutually nonintersecting. Algorithm 2 represents one possible way to construct such a collection B given a set of points P, a domain $Q = [a_1, b_1] \times \ldots \times [a_k, b_k]$, and a number sd of subdivision steps. This algorithm does not treat adequately the fact that the same boxes may be constructed several times. For this, we refer to Dellnitz and Hohmann,[24] where the same data structure is used for a different purpose.

Figure 6 shows a box collection where the Pareto front was used which is displayed in Fig. 4. For the shading of the boxes the shielding indicator (9) was used.

Using this example we want to demonstrate on two (hypothetical) settings how this visualization form can be of advantage for the DM. First, we assume that we are aiming at a high-shielding material where the cost is of a minor interest (e.g. in a military application). The image of the Pareto front displays one connected component which is shaded in white (corresponding to the highest value of the indicator). Thus, a point in this section can be chosen—maybe in Box B_1—or the search can be continued in this region, e.g. in an interactive manner. Secondly we assume we want to design a material for a ''standard'' device. Thus, it is sufficient to fulfill the required norm for the shielding efficiency while it is desired to minimize cost and weight of the material. Using the shielding indicator (e.g. by looking at the boxes which correspond to a value of $I_s(x) \ge \mathrm{SE}_{\mathrm{norm}}(f_{\max} - f_{\min})$) this could lead to the conclusion that points inside B_2 have to be examined for possible realization.

The results demonstrate that the two visualization techniques are well suited to screen and filter the available possibilities offered by the multi-objective approach in their own way. Both approaches achieved to reduce the complexity of the incoming data in the required amount for this particular application, which motivates that this can also be possible for other design problems.

[5]For the visualization we have used MATLAB, see http://www.mathworks.com

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat

Algorithm 2: $\mathcal{B} := Build_BC(P = \{p_1, \ldots, p_1\}, Q = [a_1, b_1] \times \ldots \times [a_k, b_k], \mathrm{sd})$

```
1: B := ∅
2: for all i = 1, ..., l do
3:   y := p_i
4:   for all j = 1, ..., k do
5:     l := a_j
6:     r := b_j
7:     c := (l + r)/2
8:     for all s = 1, ..., sd do
9:       if y_j ≤ c then
10:        r := c
11:        c := (l + c)/2
12:    else
13:        l := c
14:        c := (c + r)/2
15:      end if
16:      c_j := c
17:      r_j = (b_j − a_j)/2
18:    end for
19:  end for
20:  B := B ∪ B(c, r)
21: end for
```

Conclusions and future work

We have presented a multi-objective approach to the design of CPCs for the shielding of modern devices which require high-shielding and light-weight materials at reasonable prices. For this, we have proposed a novel model, have shown the applicability of state of the art multi-objective optimization methods (which were tuned for this purpose) to these MOPs, and have addressed the related decision support problem. For the latter we have proposed a particular technique for the problem-specific visualization of these 2D Pareto fronts, which can certainly be used in other applications.

For future work, there are a lot of interesting topics which can be addressed to advance the present work. For instance, one can take the uncertainties coming from the manufacturing process into account. Furthermore, it could make sense to extend the model by including, e.g. the number of layers as well as further properties of the CPCs (e.g., the magnetic permeability) as additional design parameter. This will in turn call for new techniques which can handle the hybrid model efficiently as well as for more sophisticated decision support.

REFERENCES

1. Makela T, Pienimaa S, Taka T, Jussila S, Isotalo H. Thin polyaniline films in EMI shielding. *Synthetic Met.* 1997; **85**: 1335–1336.
2. Lee CY, Song HG, Jang KS, Oh EJ, Epstein AJ, Joo J. Electromagnetic interference shielding efficiency of polyaniline mixtures and multilayer films. *Synthetic Met.* 1999; **102**: 1346–1349.
3. Satheesh Kumar KK, Geetha S, Trivedi DC. Freestanding conducting polyaniline film for the control of electromagnetic radiations. *Curr. Appl. Phys.* 2005; **5**: 603–608.
4. Hoang NH, Wojkiewicz J-L, Miane J-L, Biscarro RS. Lightweight electromagnetic shields using optimised polyaniline composites in the microwave band. *Polym. Adv. Technol.* 2007; **17**: 257–262.
5. Naishadham K. Shielding effectiveness of conductive polymers. *IEEE Trans. Electromagn. Compat.* 1992; **34**(1): 47–50.
6. Miettinen K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers: Boston, 1999.
7. Ehrgott M. *Multicriteria Optimization*. Springer: Berlin, 2005.
8. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley: Chichester, 2001.
9. Coello Coello CA, Van Veldhuizen DA, Lamont GB. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers: New York, 2002.
10. Colaneri NF, Shacklette LW. EMI shielding measurements of conductive polymer blends. *IEEE Trans. Instrum. Meas.* 1992; **4**(2): 291–297.
11. Hoang NH, Wojkiewicz J-L, Miane J-L. Optimization of electromagnetic shielding effectiveness of composites. In *8th International Workshop on Optimization and Inverse Problem in Electromagnetism IOPE 2004*, 2004; pp. 61–62.
12. Legrand T. Algorithmes génetiques et composites conducteurs. Master's thesis, Polytech Lille, France, 2006; http://www2.lifl.fr/OPAC/index.php?page=Publications/publiRec
13. Chelouah R, Siarry P. A continuous genetic algorithm designed for the global optimization of multimodal functions. *J. Heur.* 2000; **6**(2): 191–213.
14. Keijzer M, Merelo JJ, Romero G, Schoenauer M. Evolving objects: a general purpose evolutionary computation library. In *5. Conférence Internationale sur l'Évolution Artificielle (EA'01)*, October 2001.
15. Cahon S, Melab N, Talbi E-G. ParadisEO: a framework for metaheuristics. In *International Workshop on Optimization Frameworks for Industrial Applications (ICOPI 2005)*, October 2005.
16. Srinivas N, Deb K. Multiobjective optimisation using non-dominated sorting in genetics algorithms. In *Evolutionary Computation* **2**(3): 221–248.
17. Schütze O. *Set oriented methods for global optimization*. PhD thesis, University of Paderborn, 2004; http://ubdata.*uni*-paderborn.de/ediss/17/2004/schuetze/
18. Dellnitz M, Schütze O, Hestermeyer T, Covering Pareto sets by multilevel subdivision techniques. *J. Optimization Theory Appl.* 2005; **124**: 113–155.
19. Schütze O, Mostaghim S, Dellnitz M, Teich J. Covering Pareto sets by multilevel evolutionary subdivision techniques. In *Evolutionary Multi-Criterion Optimization*, Fonseca CM, Fleming PJ, Zitzler E, Deb K, Thiele L (eds). Lecture Notes in Computer Science, Springer, Heidelberg, 2003.
20. Hoang NN, Wojkiewicz J-L, Biscaro RS. Genetic algorithm applied to the optimisation of the electromagnetic properties of conducting polymer composites in the microwave band. In *Proceedings of ICMS'05*, 2005.
21. Kaiser B. *Principles of Electromagnetic Compatibility*. Artech House: Norwood, MA, 1987.
22. Duff WG. Fundamentals of electromagnetic compatibility. In *A handbook series of electromagnetic interference and compatibility* (vol. 1), Interference Control Technologies, Inc. Gainesville, VA, 1998.
23. Lotov A, Bushenkov VA, Kamenev G. *Interactive Decision Maps. Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers: Boston, 2004.
24. Dellnitz M, Hohmann A. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Num. Math.* 1997; **75**: 293–317.

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/pat