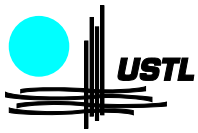
Optimisation Combinatoire Multi-Objectif : Apport des méthodes coopératives et contribution à l'extraction de connaissances

C. Dhaenens



Laboratoire d'Informatique Fondamentale de Lille

Université des Sciences et Technologies de Lille
U.F.R. d'I.E.E.A.

Numéro d'Ordre : xxxx

Année : 2005

THÈSE

pour obtenir le grade de

Habilitation à Diriger des Recherches de l'U.S.T.L.
Discipline : Informatique

présentée et soutenue publiquement,
le 05 Octobre 2005, par

Clarisse Dhaenens-Flipo

Titre :

Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances

Jury :

Président :	Sophie Tison	Professeur, Université de Lille 1
Rapporteurs :	Jacek Blazewicz	Professeur, Université de Poznan (Pologne)
	Jin-Kao Hao	Professeur, Université de Angers
	Jacques Teghem	Professeur, Université de Mons (Belgique)
Examinateurs :	Arnaud Fréville	Professeur, Université de Valenciennes
	Michel Gourgand	Professeur, Université de Clermont II
Directeur :	El-Ghazali Talbi	Professeur, Université de Lille 1

Remerciements

Le travail présenté dans le manuscrit a été réalisé dans l'équipe OPAC du Laboratoire d'Informatique Fondamentale de Lille, dont je tiens à remercier le directeur, M. Jean-Marc Geib pour son accueil.

Je tiens également à remercier M. El-Ghazali Talbi pour la confiance qu'il m'a accordée lors de mon intégration au sein de son équipe Optimisation PArallèle et Coopérative (OPAC).

Mes remerciements vont ensuite aux membres du jury pour leurs encouragements et pour les différentes discussions que nous avons pu avoir avant, et pendant la rédaction de ce mémoire. Je les remercie vivement d'avoir accepté de passer du temps à étudier mon travail.

Je remercie aussi les membres de l'équipe OPAC qui ont permi que ces 6 années à Lille se passent agréablement. Je remercie en particulier les différents étudiants qui ont participé de près à ces travaux de recherche : Laetitia, Julien, Matthieu, Mohammed. Je tiens d'ailleurs à remercier spécialement Laetitia qui en tant que première doctorante m'a permis dans de bonnes conditions, de confimer mon intérêt pour l'encadrement d'étudiants.

Mes remerciements sont aussi pour mes collègues de Polytech'Lille, en particulier Bernard, Claire, Franck, Nathalie, Stéphane. L'ambiance qui règne à Polytech'Lille permet de réaliser un travail de qualité dans de bonnes conditions.

Enfin mes remerciements vont à ma famille : tout d'abord à mes parents et beaux-parents, pouir leur soutien. Et enfin à Jean-Etienne mon mari, et à mes enfants, Odyssée et Ulysse que je remercie pour leur patience. A tous je dédie ce mémoire.

Table des matières

Liste des tableaux

Table des figures

Parcours depuis le doctorat

Soutenue en 1998 au laboratoire LEIBNIZ-IMAG, ma thèse de doctorat en Recherche Opérationnelle traitait de l'optimisation d'un réseau de production et de distribution. Cette problématique d'optimisation combinatoire, issue d'une problématique réelle rencontrée chez Pechiney, cherchait à optimiser l'ensemble d'un réseau de production composé de plusieurs sites de fabrication répartis sur un territoire. Des aspects liés à la distribution (logistique) étaient à prendre en compte.

Arrivée en tant que Maître de Conférences en 1999, dans l'équipe OPAC (Optimisation PArallèle et Coopérative) du LIFL (Laboratoire d'Informatique Fondamentale de Lille), il me fallait déterminer mon thème de recherche. L'équipe était alors composée d'un seul permanent, E-G. Talbi, et s'intéressait à l'optimisation combinatoire par métaheuristiques hybrides.

À cette époque, un nouveau domaine commençait à intéresser les personnes du monde de la Recherche Opérationnelle. Il s'agissait de l'extraction de connaissances. Ce domaine, bien connu du monde des statistiques, de l'apprentissage ou encore des bases de données, restait encore méconnu du monde de la Recherche Opérationnelle. Pourtant l'extraction de connaissances semblait offrir de nouveaux challenges... Ainsi, j'ai choisi d'étudier des problématiques d'extraction de connaissances de façon à voir ce que pouvait apporter la Recherche Opérationnelle à ce domaine.

Ce choix a été conforté par la mise en place de la génopole de Lille à la même époque. En effet, dans ce cadre, des laboratoires des sciences du vivant recherchaient des collaborations avec des laboratoires des sciences de l'information afin de mener des recherches conjointes. Certaines problématiques d'extraction de connaissances ont alors été identifiées.
Ainsi nous avons collaboré avec l'Institut de Biologie de Lille, et en particulier le Laboratoire des Maladies Multifactorielles, afin d'étudier les facteurs de prédisposition à certaines maladies telles que le diabète de type II et l'obésité. Ceci a servi de cadre d'application à la thèse de Laetitia Jourdan. Cette thèse, que j'ai co-encadrée, avait pour objectif de modéliser certains problèmes d'extraction de connaissances rencontrés en génomique en des problèmes d'optimisation combinatoire. Une approche de résolution par métaheuristiques a ensuite été mise en œuvre.
Puis de nouvelles collaborations autour de la génomique (ou bioinformatique de façon plus générale) ont vu le jour, notamment avec l'Institut Pasteur de Lille et la société IT-Omics (Lille). Nous avons ainsi été confrontés à différentes problématiques. Nous avons alors choisi

de nous concentrer sur l'étude de données issues d'expérimentations sur puces à ADN. Cette technologie à haut débit permet de mesurer les niveaux d'expression de milliers de gènes simultanément, sous différentes conditions expérimentales, et génère ainsi d'importantes masses de données. La problématique d'extraction de connaissances retenue fût la recherche de règles d'association pour laquelle un modèle d'optimisation combinatoire multi-objectif a été proposé. Ceci fait l'objet de la thèse de Mohammed Khabzaoui, que je co-encadre.

Ainsi, depuis 5 ans, un travail a été mené sur les apports de la recherche opérationnelle pour l'extraction de connaissances. Ce travail était relativement précurseur, car si nous n'étions pas les seuls à travailler sur ce thème, nous n'étions pas très nombreux... Actuellement, cette thématique est très en vogue et il m'a d'ailleurs été demandé de faire un exposé de synthèse sur *Extraction de connaissances et Recherche Opérationnelle* lors du dernier congrès de la société Française de Recherche Opérationnelle (ROADEF). Un article de synthèse a également été rédigé.

Les problèmes d'extraction de connaissances nous ayant conduit à l'optimisation combinatoire multi-objectif, j'ai donc décidé de m'y intéresser de plus près. De plus, au sein de l'équipe cette problématique devenait de plus en plus présente de part l'augmentation des contrats et collaborations autour d'étude de problèmes réels qui sont très souvent de nature multi-objectif.

M'intéressant donc à cette problématique, j'ai activement participé au groupe PM2O (Programmation Mathématique Multi-Objectif). Actuellement, je suis animatrice de ce groupe qui fait partie des groupes de travail de la ROADEF ainsi que du GDR-I^3. Dans ce cadre, des réunions sont régulièrement organisées.
Ainsi, une grande part de mes travaux actuels concernent l'optimisation combinatoire multi-objectif, notamment avec la recherche de méthodes exactes et la coopération de méthodes. C'est sur cette thématique que je co-encadre la thèse de Julien Lemesre.
C'est d'ailleurs aussi sur ce thème central que nous avons proposé le projet DOLPHIN (*Discrete multi-objective Optimization for Large scale Problems with Hybrid dIstributed techNiques*) qui est récemment passé équipe INRIA de l'entité Futurs.

Enfin, c'est parce que l'optimisation combinatoire multi-objectif est un large domaine plein de challenges que j'ai choisi de rédiger le manuscript autour de ce thème.

Introduction

L'optimisation combinatoire regroupe une large classe de problèmes ayant des applications dans de nombreux domaines applicatifs.
Un problème d'optimisation combinatoire est défini par un ensemble fini de solutions discrètes $\mathcal{D}$ et une fonction objectif f associant à chaque solution une valeur (la plupart du temps, une valeur réelle). Ainsi, un problème d'optimisation combinatoire consiste en l'optimisation (minimisation ou maximisation) d'un certain critère sous différentes contraintes permettant de délimiter l'ensemble des solutions réalisables (ou solutions admissibles).

La variété des problèmes d'optimisation combinatoire est en particulier dûe au large spectre de ses applications. En effet, que l'on s'intéresse à l'optimisation d'un système de production, au design de réseaux de télécommunication, à la bio-informatique ou encore à l'extraction de connaissances, nous pouvons être confrontés à des problèmes d'optimisation combinatoire.

Résolution d'un problème d'optimisation combinatoire

Résoudre un problème d'optimisation combinatoire nécessite l'étude de trois points particuliers :
– la définition de l'ensemble des solutions réalisables,
– l'expression de l'objectif à optimiser,
– le choix de la méthode d'optimisation à utiliser.
Les deux premiers points relèvent de la modélisation du problème, le troisième de sa résolution.

Afin de définir l'ensemble des solutions réalisables, il est nécessaire d'exprimer l'ensemble des contraintes du problème. Ceci ne peut être fait qu'avec une bonne connaissance du problème sous étude et de son domaine d'application. La programmation linéaire peut être utilisée à cet effet.

Le choix de l'objectif à optimiser requiert également une bonne connaissance du problème. La définition de la fonction objectif mérite toute l'attention de l'analyste car rien ne sert de développer de bonnes méthodes d'optimisation si l'objectif à optimiser n'est pas bien défini. Comme nous le verrons par la suite, il peut être très difficile de trouver un objectif unique d'optimisation. Nous pouvons donc être amené à proposer une modélisation multi-objectif.

Enfin, le choix de la méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. En effet, suivant sa complexité, le problème pourra ou non être résolu de façon optimale. Dans le cas de problèmes classés dans la classe $\mathcal{P}$, un algorithme polynomial a été mis en évidence. Il suffit donc de l'utiliser. Dans le cas de problèmes $\mathcal{NP}$-difficiles, deux possibilités sont offertes. Si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé (procédure de séparation et évaluation (Branch & Bound), programmation dynamique...). Malheureusement, ces algorithmes par nature énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles (même si en pratique la taille n'est pas le seul critère limitant). Dans ce cas, il est nécessaire de faire appel à des heuristiques permettant de trouver de bonnes solutions approchées. Parmi ces heuristiques, on trouve les métaheuristiques qui fournissent des schémas de résolution généraux permettant de les appliquer potentiellement à tous les problèmes.

Nous voyons donc ici que la phase de modélisation du problème est très importante puisque c'est elle qui permettra par exemple de reconnaître un problème de la classe $\mathcal{P}$ d'un problème $\mathcal{NP}$-difficile. En particulier, la définition de l'objectif est cruciale mais peut être difficile à réaliser, surtout lors de l'étude de problèmes réels.

Optimisation combinatoire multi-objectif

Les problèmes d'optimisation issus de problématiques réelles sont la plupart du temps de nature multi-objectif car plusieurs critères sont à considérer simultanément. Optimiser un tel problème relève donc de l'optimisation combinatoire multi-objectif.

Les premières études concernant l'optimisation combinatoire multi-objectif transformaient les problèmes multi-objectifs en une succession de problèmes mono-objectifs. Pour cela, un ordre d'importance sur les objectifs pouvait être donné, et l'optimisation consistait à optimiser un objectif sans dégrader les valeurs déjà obtenues pour les objectifs plus prioritaires. Une autre approche consistait en l'optimisation d'une agrégation linéaire des objectifs, chacun pouvant avoir un poids représentant son importance.

Lorsque l'on se trouve dans un réel contexte multi-objectif, il n'est pas toujours possible de trouver un ordre d'importance sur les critères. Il est alors nécessaire de rechercher les solutions de meilleur compromis entre les objectifs. Si cette notion de compromis sera définie plus précisément dans le chapitre 1, il est facile de voir que dans ce contexte la solution recherchée n'est pas une unique solution mais un ensemble de solutions représentant les différents compromis possibles.
Ainsi l'optimisation multi-objectif s'intéresse aux particularités liées à l'existence de ces différentes solutions optimales. En particulier, les méthodes de résolution devront être dédiées à ce type de problèmes qui sont la plupart du temps $\mathcal{NP}$-difficiles. De même, la comparaison de solutions produites par différents algorithmes n'est plus chose facile en multi-objectif car

il faut alors comparer différents ensembles de solutions.

La résolution de problèmes multi-objectifs relève de deux disciplines assez différentes (même si des efforts sont faits pour essayer de diminuer l'écart existant entre ces disciplines). En effet, résoudre un problème multi-objectif peut être divisé en deux phases :

1. **la recherche des solutions de meilleur compromis.** Se pose alors la question de savoir si toutes les solutions doivent être produites (elles peuvent être nombreuses) ou seulement un sous-ensemble représentatif. C'est la phase d'optimisation multi-objectif.
2. **le choix de la solution à retenir.** C'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi-objectif et cela fait appel à la théorie de la décision.

Dans le cadre de ce manuscrit nous ne parlerons que de la première phase qui consiste en la recherche des solutions de meilleurs compromis.

Plan du manuscrit

Ce manuscrit est décomposé en un chapitre introductif et deux parties principales.

Le chapitre 1, pose le contexte du travail. Pour cela les principales définitions liées à l'optimisation combinatoire multi-objectif sont présentées. Puis, les problématiques spécifiques à ce domaine sont exposées et étudiées. Parmi ces problématiques nous parlerons en particulier de la structure de l'ensemble des solutions de compromis (solutions Pareto), du choix des méthodes de résolution et de l'analyse de performances en multi-objectif. Ce chapitre permettra de cerner ce qui est étudié dans le manuscrit et ce qui ne l'est pas.

La première partie (chapitres 2 et 3) traite de la coopération de méthodes en vue d'améliorer les résultats des méthodes d'optimisation combinatoire multi-objectif. Au cours de cette partie, un problème d'ordonnancement - problème de flowshop de permutation bi-objectif - est utilisé à titre d'exemple. Nous commençons donc le chapitre 2 par la présentation de ce problème. Puis, le chapitre s'attarde sur les méthodes exactes pour l'optimisation multi-objectif. Ces méthodes n'étant pas nombreuses, une revue de la littérature est réalisée. Inspirée de cette étude, un nouveau schéma de méthode exacte - **PPM** - est proposé.
Le chapitre 3 s'intéresse à la coopération entre méthodes. Pour cela, la première partie du chapitre concerne l'utilisation des métaheuristiques en multi-objectif. Ainsi, après avoir présenté nos travaux concernant le développement d'un algorithme génétique pour le flowshop bi-objectif, une présentation rapide des méthodes les plus connues est réalisée. Puis, la deuxième partie discute des possibilités de coopération entre les différentes méthodes et présente les résultats obtenus.

La deuxième partie (chapitres 4, 5 et 6) s'intéresse à un tout autre domaine d'application. Il s'agit de l'extraction de connaissances. En effet, de nombreux problèmes d'extraction de

connaissances peuvent être modélisés, entièrement ou en partie, en des problèmes d'optimisation combinatoire. C'est ce que nous présentons dans le chapitre 4. Dans ce chapitre, une partie est consacrée à l'apport du multi-objectif pour ce type de problèmes. Puis, une étude plus approfondie porte sur la problématique de recherche de règles d'association. Pour ce problème, nous exposons la modélisation multi-objectif proposée ainsi que les méthodes de résolution développées.
Le chapitre 5 présente deux approches coopératives : une approche coopérative parallèle mettant en jeux plusieurs métaheuristiques et une coopération avec une méthode exacte.
Finalement, le chapitre 6 donne des indications sur le contexte applicatif utilisé, à savoir l'étude de données issues de la bio-informatique.

Au cours du mémoire, chaque chapitre se termine par un certain nombre de perspectives. Le dernier chapitre, conclusions et perspectives, fait une synthèse des principaux apports des travaux présentés dans le manuscrit ainsi que des perspectives annoncées.

Le reste du manuscrit est composé de trois annexes :
– un Curriculum Vitae qui décrit les différentes activités menées,
– une liste des différentes publications réalisées,
– un ensemble de publications qui se veut être le plus représentatif des travaux de recherches effectués.

Chapitre 1

Optimisation combinatoire multi-objectif : problématique et cadre de travail

Les problèmes d'optimisation combinatoire issus des problématiques réelles sont la plupart du temps de nature multi-objectif car plusieurs critères d'évaluation souvent contradictoires sont à considérer simultanément. Optimiser un tel problème relève donc de l'optimisation combinatoire multi-objectif.

L'optimisation multi-objectif possède ses racines dans les travaux en économie de Edgeworth [21] et Pareto [67]. Elle a ainsi été initialement utilisée en économie et dans les sciences du management, puis graduellement dans les sciences pour l'ingénieur.
Pourtant, malgré l'intérêt indéniable de la modélisation et de la résolution multi-objectifs des problèmes rencontrés en industrie, dans les télecommunications, etc. peu de travaux ont été réalisés en optimisation combinatoire multi-objectif avant les années 80-90. Mais depuis, un fort intérêt a été montré pour l'aide à la décision multi-objectif qui consiste pour un problème comportant plusieurs objectifs, à déterminer, parmi les solutions de meilleurs compromis entre les objectifs, la solution la plus intéressante pour le problème en question. Ainsi, une phase importante concerne l'optimisation multi-objectif qui recherche les solutions de compromis.

Ce chapitre a pour objectif de présenter le contexte de l'optimisation multi-objectif, ses principales définitions et surtout les problématiques liées à ce domaine.

1.1 Définitions

L'optimisation combinatoire multi-objectif fait partie du domaine de l'optimisation combinatoire. Ainsi un certain nombre de définitions s'inspirent de l'optimisation combinatoire, mais différents concepts, spécifiques au multi-objectif, sont également introduits. En effet, la

spécificité principale du multi-objectif étant l'existence de plusieurs fonctions à optimiser, il est en particulier nécessaire de revisiter la notion d'optimalité des solutions.

1.1.1 Problème d'optimisation combinatoire multi-objectif

Un problème d'optimisation combinatoire multi-objectif (PMO) (multi-objective combinatorial optimization problem) peut être défini par :

$$(PMO) \begin{cases} \text{Optimiser} & F(x) = (f_1(x), f_2(x), ..., f_n(x)) \\ \text{sous} & x \in \mathcal{D} \end{cases}$$

où n est le nombre d'objectifs ($n \geq 2$), $x = (x_1, x_2, \ldots, x_k)$ est le vecteur représentant les variables de décision, $\mathcal{D}$ représente l'ensemble des solutions réalisables et chacune des fonctions $f_i(x)$ est à optimiser, c'est-à-dire à minimiser ou à maximiser. Sans perte de généralité nous supposerons par la suite que nous considérons des problèmes de minimisation.

Contrairement à l'optimisation mono-objectif, la solution d'un problème multi-objectif n'est pas unique, mais est un ensemble de solutions non dominées, connu comme l'ensemble des solutions Pareto Optimales (PO).

1.1.2 Notions de dominances et d'optimalité

Une solution réalisable $x^* \in \mathcal{D}$ est **Pareto optimale** (ou efficace, ou encore non dominée) si et seulement si il n'existe pas de solution $x \in \mathcal{D}$ telle que x domine x^*.
On dit d'une solution $y = (y_1, y_2, \ldots, y_k)$ qu'elle **domine** une solution $z = (z_1, z_2, \ldots, z_k)$, dans le cas d'une minimisation d'objectifs, ssi $\forall i \in [1 \ldots n]$, $f_i(y) \leq f_i(z)$ et $\exists i \in [1 \ldots n]$ tel que $f_i(y) < f_i(z)$.

Ainsi, toute solution de l'ensemble Pareto peut être considérée comme optimale puisque aucune amélioration ne peut être faite sur un objectif sans dégrader la valeur relative à un autre objectif. Ces solutions forment le **front Pareto**.

Dans le cas d'un problème bi-objectif (deux objectifs à minimiser par exemple), les solutions efficaces peuvent être identifiées visuellement dans l'espace objectif, comme étant celles pour lesquelles le rectangle inférieur gauche formé par la solution et le point $(0, 0)$ est vide de solution réalisable (voir Fig. 1.1).

Attachée à la notion de voisinage[1], la notion de **Pareto localement optimale** qualifie une solution qui n'est dominée par aucune solution de son voisinage.

[1]le voisinage d'une solution x est composé des solutions pouvant être atteintes depuis x à l'aide d'une transformation élémentaire, alors appelée opérateur de voisinage.

1.1.3 Points particuliers

En vue d'avoir certains points de références permettant de discuter de l'intérêt des solutions trouvées, des points particuliers ont été définis dans l'espace objectif. Ces points peuvent représenter des solutions réalisables ou non.

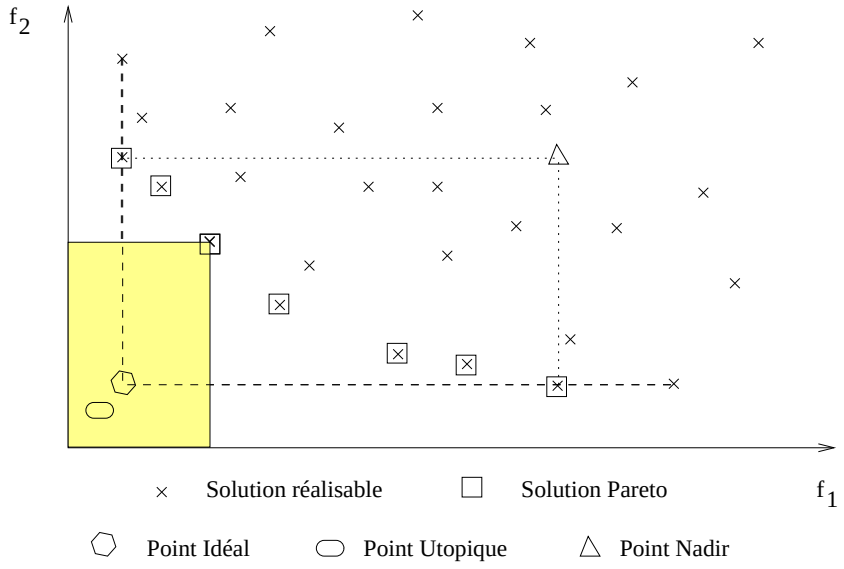


FIG. 1.1 – Illustration des différentes définitions.

1. Tout d'abord, le **point idéal** z^I est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré.

$$z^I \text{tel que } \forall i \in [1...n], f_i(z^I) = opt_{x \in \mathcal{D}} f_i(x)$$

   Ce point ne correspond pas à une solution réalisable car si c'était le cas, cela sous-entendrait que les objectifs ne sont pas contradictoires et qu'une solution optimisant un objectif, optimise simultanément tous les autres, ce qui ramenerait le problème à un problème ayant une seule solution Pareto optimale.

2. De ce point idéal peut être défini le **point utopique** z^U de la façon suivante :

$$z^U = z^I - \epsilon\, U$$

   où $\epsilon > 0$ et U est le vecteur unitaire ($U = (1, \ldots, 1) \in \Re^n$). Il est clair, de par sa définition, que ce point n'est pas réalisable.

3. Enfin le **point Nadir** qui est défini en bi-objectif par :

$$z^N \text{tel que } \forall i \in [1...2], f_i(z^N) = opt_{x \in \mathcal{D}/f_j(x) = f_j(z^I)} f_i(x) \text{ avec } j \neq i$$

Cela revient donc à affecter pour chaque objectif du point Nadir la meilleure valeur possible parmi les solutions optimisant l'autre objectif.

Une visualisation de l'ensemble de ces définitions est donnée sur la figure 1.1.

1.2 Choix de la méthode d'aide à la décision

La résolution d'un problème multi-objectif menant à la détermination d'un ensemble de solutions Pareto, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder.
Ainsi, avant de se lancer dans la résolution d'un problème multi-objectif, il faut se poser la question du type de méthode d'optimisation à utiliser. En effet, on peut répartir les méthodes de résolution de problèmes multi-objectifs en trois familles, en fonction du moment où intervient le décideur. Ainsi nous pouvons trouver les familles suivantes :

- Les méthodes d'optimisation **a priori** : dans ce cas, le compromis que l'on désire faire entre les objectifs a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis.

- Les méthodes d'optimisation **progressives** : ici, le décideur intervient dans le processus de recherche de solutions en répondant à différentes questions afin d'orienter la recherche. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

- Les méthodes d'optimisation **a posteriori** : dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonne solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contre-partie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

Nous nous placerons dans le cadre de cette troisième famille de méthodes où la modélisation des préférences n'est pas requise et où le procédé d'optimisation doit être puissant afin de fournir l'ensemble de solutions Pareto optimales ou à défaut une très bonne approximation de la frontière Pareto.

Dans ce type de méthode, deux phases importantes sont à considérer : la phase de recherche de l'ensemble des solutions Pareto optimales, que nous appelerons de façon abusive, **résolution du problème d'optimisation** et la phase de choix parmi ces solutions, qui relève de l'**aide à la décision**. Cette deuxième phase ne sera pas traitée ici.

1.3 Structure du front Pareto

L'objectif est donc de fournir aux décideurs un ensemble (le plus complet possible) de solutions Pareto, afin qu'ils puissent ensuite choisir les solutions qui les intéressent le plus.
Une question se pose donc sur la nature de ces solutions Pareto et la nécessité de les obtenir toutes. Une étude de la frontière Pareto doit donc être réalisée.

1.3.1 Front minimal / front maximal complet

La définition de front se réfère à l'espace des objectifs. Une solution appartient au front si elle n'est dominée par aucune autre solution réalisable.
Lorsque deux solutions ont exactement les mêmes valeurs pour l'ensemble des objectifs, elles sont équivalentes dans l'espace objectif, mais peuvent correspondre à deux solutions différentes dans l'espace décisionnel. Une question importante est de savoir s'il est intéressant de garder ces deux différentes solutions.
La réponse peut dépendre du contexte (type de problème étudié) en plus de la volonté des décideurs :

- Lors de la résolution d'un problème comportant énormément de solutions Pareto, il est peut être préférable de privilégier une bonne approximation de l'ensemble de la frontière et donc favoriser la diversité (du côté objectif) des solutions retenues.
- Au contraire, lorsque la frontière Pareto comporte peu de solutions, afin d'avoir une bonne représentation de l'ensemble des solutions non dominées, il sera intéressant de rechercher les solutions de même valeur.

Nous parlerons alors de recherche du **front minimal**, dans le premier cas, et du **front maximal complet**, dans le second.

1.3.2 Solutions supportées / non supportées

Sur le front Pareto, deux types de solutions peuvent être différenciées : les solutions supportées et les solutions non supportées. Les premières sont celles situées sur l'enveloppe convexe de l'ensemble des solutions (voir Fig. 1.2) et peuvent donc être trouvées à l'aide d'une agrégation linéaire des objectifs [30]. Elles sont donc plus simples à obtenir que les solutions non supportées. D'ailleurs, les premiers travaux en optimisation combinatoire multi-objectif se sont pour la plupart focalisés sur la recherche de ces solutions supportées en optimisant des combinaisons linéaires des objectifs utilisant différents vecteurs de poids.
Alors pourquoi ne pas se satisfaire des solutions supportées ? Tout d'abord parce que ces solutions peuvent ne représenter qu'un petit sous-ensemble des solutions efficaces. De plus, ces solutions supportées ne sont pas forcément bien réparties le long du front et ne représentent pas toujours un bon compromis. La figure 1.3 nous montre l'exemple d'un problème de flow-shop bi-objectif où les deux seules solutions supportées sont des solutions extrèmes. Donc, si l'on veut obtenir des solutions de bon compromis entre les objectifs, il est nécessaire de considérer les solutions Pareto non supportées.

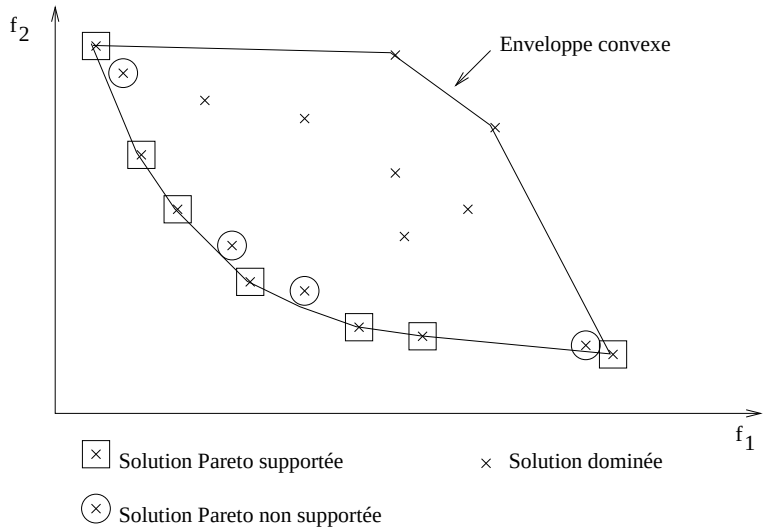


FIG. 1.2 – Représentation des différents types de solutions en bi-objectif.

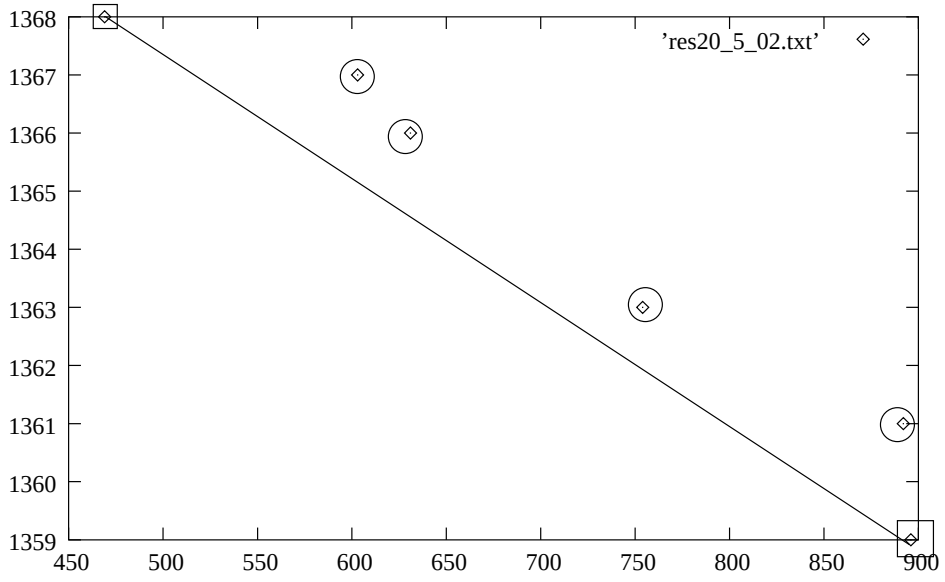


FIG. 1.3 – Exemple de l'importance des solutions non supportées.

1.3.3 Stratégie à adopter

Ainsi, en fonction du problème sous étude, et de la connaissance de la structure de sa frontière Pareto, différentes stratégies peuvent être adoptées.

- *La structure du problème est bien connue et sa frontière a été montrée convexe :* dans ce cas, toutes les solutions Pareto sont supportées. Il est donc possible de rechercher ces solutions en utilisant des agrégations linéaires d'objectifs.

- *La structure du problème est bien connue et sans être convexe, il est montré que les solutions supportées sont bien réparties :* si l'objectif est d'obtenir un ensemble de solutions représentatif du front Pareto (sans nécessairement obtenir toutes les solutions), alors il est encore possible de ne rechercher que les solutions supportées, à l'aide d'agrégations d'objectifs.

- *La structure du problème est pas ou mal connue, ou les solutions supportées sont mal réparties sur le front :* afin de ne prendre aucune hypothèse sur la répartition des solutions sur le front, il est nécessaire de rechercher à la fois les solutions supportées et non supportées. De plus, dans un environnement non stable, la connaissance de plusieurs solutions Pareto optimales permet de mieux appréhender les aléas en permettant parfois de changer le choix de la solution en fonction d'un changement de condition sans avoir à réaliser une nouvelle recherche de solution.

Les deux premiers cas cités sont bien évidemment les plus favorables. Malheureusement ils nécessitent une bonne connaissance du problème sous étude et en particulier une connaissance de la caractérisation de sa frontière Pareto. Cette caractérisation est très difficile à réaliser car il ne faut pas se focaliser sur quelques instances étudiées, mais bien sur des propriétés générales vérifiées par l'ensemble des instances d'un problème.
Ainsi, pour éviter de nous restreindre à quelques types de problèmes, nous nous placerons dans le cas où la structure du problème est mal connue ou non favorable. Notre objectif sera alors de rechercher l'ensemble des solutions supportées et non supportées.

1.4 Approches classiques de résolution

Nous traitons ici des problèmes d'optimisation combinatoire multi-objectifs qui sont pour la plupart $\mathcal{NP}$-difficiles. Pour leur résolution, des méthodes exactes, ainsi que des heuristiques ont été proposées.
Sans avoir la prétention de présenter ici toutes les méthodes dédiées à l'optimisation combinatoire multi-objectif, en voici quelques-unes. Pour avoir une vue plus globale, le lecteur peut se référer à [22].

1.4.1 Méthodes exactes pour l'optimisation multi-objectif

Concernant les **méthodes exactes**, plusieurs approches basées sur des procédures de séparation et évaluation (branch and bound) [82], sur l'algorithme A^* [77] et la programmation dy-

namique [10] ont été proposées pour résoudre de petits problèmes à deux objectifs (problèmes bi-objectifs).
Une approche particulière pour l'optimisation multi-objectif est le "goal programming" (programmation par buts) [72]. Dans ce type d'approche, le décideur indique une valeur cible (but) et l'objectif est de minimiser l'écart avec cette cible. Souvent la programmation par buts est vue comme une discipline en elle-même, différente de l'optimisation multi-objectif[2].
Une approche intéressante a été proposée par B. Ulungu et J. Teghem pour la recherche du front Pareto de problèmes bi-objectifs [82]. Leur méthode en deux phases consiste dans un premier temps à rechercher l'ensemble des solutions Pareto supportées, puis dans un deuxième temps à rechercher de façon indépendante les solutions non-supportées situées entre tous les couples de solutions supportées adjacentes (voir paragraphe 2.3 pour une description précise). Cette approche a été utilisée efficacement sur des problèmes tels que l'affectation ou le sac à dos bi-objectifs. Cette méthode a ensuite été améliorée afin d'obtenir des fronts complets de façon plus efficace [70].

Pourtant, dès que le nombre d'objectifs ou la taille des problèmes augmentent, les méthodes exactes deviennent inefficaces étant donnés la nature $\mathcal{NP}$-difficile des problèmes (déjà en mono-objectif) et l'aspect multi-objectif des problèmes.

Ainsi, il est nécessaire afin de résoudre des problèmes de grande taille et/ou des problèmes avec plus de deux objectifs, de faire appel à des méthodes heuristiques. Les méthodes exactes peuvent néanmoins être utiles lorsque des sous-problèmes peuvent être extraits du problème global. Leur résolution permet en effet de contribuer à la recherche de la solution globale, soit en combinant judicieusement différents sous-problèmes, soit en hybridant résolution exacte de sous-problèmes et résolution heuristique du problème complet. Ceci représente une direction suivie par certains travaux présentés ci-après.

1.4.2 Méthodes heuristiques

Les méthodes heuristiques ne garantissent pas de trouver de manière exacte tout l'ensemble des solutions Pareto, mais une approximation, aussi bonne que possible, de cet ensemble. Les approches heuristiques peuvent être classées en trois catégories :

– **Approches transformant le problème en un ou plusieurs problème(s) mono-objectif(s) :** ces approches transforment le problème initial afin de se ramener à la résolution de un ou plusieurs problèmes mono-objectif. Parmi ces méthodes, on peut citer les méthodes d'agrégation [66, 76], les méthodes ϵ-contrainte [36] ou encore les méthodes utilisant un vecteur cible [12, 60]. En général ces méthodes nécessitent une connaissance du problème et ne fournissent qu'une seule solution. Elles peuvent alors être classées dans la famille des méthodes d'optimisation a priori, présentée précédemment, et ne nous intéressent pas ici.

[2] cf la conférence MOPGP : Multi-Objective Programming AND Goal Programming.

– **Approches Non Pareto :** ces approches transforment le problème d'origine. Elles effectuent leur recherche en traitant indépendamment chacun des objectifs. L'exemple le plus classique est l'algorithme VEGA (*Vector Evaluated Genetic Algorithm* [73]). Ces méthodes ont souvent du mal à trouver les solutions de compromis puisqu'elles se focalisent sur les portions extrèmes du front. Nous pouvons classer dans cette catégorie les méthodes lexicographiques qui donnent un ordre de priorité sur les objectifs à traiter.

– **Approches Pareto :** ces approches utilisent la notion de dominance pour comparer les solutions entre elles. Une seule résolution permet d'approximer l'ensemble de la frontière Pareto. C'est ce type de méthodes que nous allons considérer. L'un des premiers à discuter de l'intérêt de l'utilisation de la notion de dominance pour la recherche de solutions a été Goldberg [31].

Puisque notre objectif est l'optimisation "a posteriori" et donc la génération de l'ensemble des solutions Pareto, les méthodes à base de populations travaillant avec un ensemble de solutions potentielles, tels que les algorithmes évolutionnaires, sont bien adaptées à ce type de problème [14, 16]. Pour en être convaincu, il suffit de regarder la liste de références sur "Evolutionary Multi-objective Optimization" maintenue par Carlos A. Coello Coello[3] qui contient plus de 1850 références et l'engouement pour la conférence EMO (Evolutionary Multi-Criterion Optimization). C.A. Coello Coello, D.A. Van Valdhuisen et G. B. Lamont proposent une classification de ces algorithmes évolutionnaires selon deux générations de méthodes. Ce qui différencie la deuxième génération est la présence d'une population secondaire (archive) et de méthodes de recherche avancées. De nombreuses méthodes ont alors été proposées avec ces différents méchanismes.
Il est à remarquer que deux aspects importants sont à prendre en considération. Tout d'abord, l'heuristique doit converger le plus possible vers la frontière, de façon à s'en approcher au maximum mais elle doit également proposer des solutions diversifiées sur le front afin d'avoir un bon échantillon représentatif et ne pas se concentrer sur une zone de l'espace objectif.

1.5 Analyse de performances en multi-objectif

L'existence de plusieurs solutions optimales (formant la frontière Pareto) et l'absence d'ordre total entre les solutions rendent la mesure de qualité d'un front difficile. En effet, si la notion de dominance au sens de Pareto peut être utilisée pour comparer deux solutions, bien que ces deux solutions puissent être incomparables, la comparaison d'un ensemble de solutions est encore plus délicate. Pourtant lorsque l'on cherche à évaluer un algorithme en terme de qualité des solutions obtenues, il est nécessaire soit de pouvoir qualifier un front soit de le comparer de façon quantitative avec les fronts produits par d'autres algorithmes.

Malheureusement cette tâche est délicate puisque la notion de qualité d'un front est elle-même multi-objectif. En effet, un front intéressant est un front qui montre une intéressante

[3](http ://www.lania.mx/∼ccoello/EMOO/EMOObib.html)

convergence vers le front optimal ET une bonne distribution des solutions. Certains fronts peuvent alors être très bons au regard de l'un des objectifs sans être intéressants pour le second.

De nombreux indicateurs de performances ont été proposés dans la litérature. Des articles de synthèse ont été présentés ([35, 53]) et en particulier, une étude récente analyse ces indicateurs et expose leurs limitations [92]. Voici certaines de ces mesures classées en fonction de leur objectif : mesurer la qualité d'un front isolé, comparer deux fronts...

1.5.1 Indicateurs de qualité s'appliquant à un seul front

L'objectif de ces mesures est de fournir une valeur numérique donnant des indications sur la diversité et/ou la distribution des solutions composant le front. Ces mesures sont très utilisées dans la littérature car elles permettent de qualifier un front indépendemment d'autres fronts. Pourtant elles sont souvent à utiliser avec précaution car ne permettent pas, en général, d'utiliser les valeurs obtenues pour comparer différents fronts. En voici quelques-unes.

ONVG - Overall Non-dominated Vector Generation
Cette mesure comptabilise le nombre de solutions non dominées générées par l'algorithme [84]. Cette mesure indépendante, facile à calculer, doit être manipulée avec précaution si elle est utilisée pour comparer des fronts.

Schott's spacing metric
Cette métrique, basée sur un calcul de distance entre les solutions, a pour objectif de mesurer la distribution des solutions le long du front [53]. Utilisée avec d'autres mesures, elle donne une indication intéressante.

Entropie
L'entropie utilise la notion de niche pour évaluer la distribution des solutions sur le front [6]. Plus proche de 1 est la valeur obtenue, meilleure est la distribution.

1.5.2 Mesures utilisant une référence

Ce type de mesures utilise une référence, qui peut être un point ou l'ensemble Pareto optimal (lorsqu'on a la chance de le connaître), pour évaluer la qualité d'un front. Sans vouloir les citer toutes, en voici des exemples.

Métrique S
Proposée par Zitzler [90], cette mesure calcule l'hypervolume de la région multi-dimensionelle comprise entre le front et un point de référence. L'inconvénient de cette métrique est que le résultat dépend du point de référence choisi. Ainsi, la difficulté réside dans le choix de ce point qui doit en particulier être dominé par toutes les solutions des fronts.

Ratio d'erreur
Ce ratio compare le front à qualifier avec le front optimal [84]. Il dénombre les solutions n'appartenant pas au front optimal. Plus le ratio est faible, meilleur est le front.

Distance par rapport au front optimal
Plusieurs auteurs ont proposé de mesurer la distance entre le front à étudier et le front optimal [35, 53, 84]. En fonction des auteurs, ils préconisent d'utiliser la distance minimale, maximale, moyenne...

1.5.3 Mesures comparant deux fronts Pareto

La comparaison de deux fronts permet de comparer deux méthodes différentes. Lorsque le front optimal est connu cela permet également d'avoir une performance absolue de la méthode sous étude.

Mesure de contribution
La mesure de contribution entre deux fronts ($Cont(F1, F2)$) permet d'évaluer la proportion de solutions Pareto apportée par chacun des fronts [6]. Lorsqu'un front est totalement dominé sa contribution est nulle et $Cont(F1, F2) + Cont(F2, F1) = 1$. Ainsi, une contribution supérieure à 0,5 indique une amélioration du front.

Métrique C
Cette mesure $C(F1, F2)$ indique le ratio de solutions du front $F2$ faiblement dominées par les solutions du front $F1$ [53]. Lorsque $C(F1, F2) = 1$, le front $F2$ est totalement dominé par $F1$.

1.5.4 Conclusion sur l'analyse de performances

Comme nous l'avons vu un ensemble de mesures ont été proposées pour analyser les performances des algorithmes multi-objectifs. Pour une bonne analyse, différentes mesures doivent être utilisées afin de pouvoir analyser à la fois convergence et diversité. L'analyse de performances en multi-objectif est en elle seule un domaine d'étude encore très ouvert puisqu'il n'existe pas de mesure universellement utilisée. Ceci est expliqué par la nature multi-objectif du problème puisque l'on cherche à obtenir des fronts qui approximent le mieux le front optimal et ce suivant différents critères tels que la qualité et la diversification.

1.6 Conclusion sur l'optimisation multi-objectif

Cette partie avait pour objectif de présenter dans un premier temps les principales définitions nécessaires à la présentation des problèmes d'optimisation combinatoire multi-objectifs. Puis différentes problématiques liées aux spécificitées du multi-objectif, comme l'intervention du décideur dans le processus de décision, les choix des méthodes d'optimisation à utiliser ou encore l'analyse de performances ont été évoquées afin de montrer l'étendue du spectre des recherches dans le domaine.
Nous nous sommes également astreints à cerner le cadre des études qui seront présentées ci-après, à savoir l'optimisation "A posteriori" (cherchant à générer l'ensemble du front Pareto) à l'aide de méthodes hybridant algorithmes évolutionnaires et méthodes exactes.

Première partie

Coopération entre méthodes exactes et métaheuristiques pour l'optimisation multi-objectif

Chapitre 2

Méthodes exactes pour l'optimisation multi-objectif

2.1 Motivations et contexte d'étude

Comme nous l'avons dit précédemment, il existe très peu de méthodes exactes dédiées à la recherche de l'ensemble des solutions Pareto. Aussi, il nous a semblé intéressant de voir ce qui pouvait être fait dans ce sens. Ceci est l'objectif du travail de thèse de Julien Lemesre.

Après une étude des méthodes existantes, deux méthodes utilisées en bi-objectif ont tout de même retenu notre attention. Il s'agit d'une méthode basée sur l'utilisation de la méthode ϵ-contrainte et de la méthode deux-phases que nous expliquons ci-après. Ces deux méthodes nous ont conduit à proposer une nouvelle méthode pour les problèmes bi-objectifs : la méthode de décomposition parallèle (PPM : *Parallel Partitionning Method*).

Afin de pouvoir discuter de façon plus précise des apports et limites des approches nous illustrerons notre discours par un problème d'ordonnancement particulier : un problème de flowshop bi-objectif que nous présentons ci-après. Par ailleurs, rappelons que toutes les illustrations sont faites pour des problèmes de minimisation.

Ainsi, après une présentation du problème utilisé comme illustration, le reste du chapitre présente tout d'abord une revue rapide des différentes méthodes exactes multi-objectifs. Puis nous nous attardons sur les deux méthodes ayant retenu notre attention, avant de présenter les améliorations proposées pour la méthode deux-phases. Enfin, nous présentons PPM, la nouvelle méthode par partitions.

2.1.1 Présentation du problème illustratif

Le flowshop est un problème classique d'ordonnancement et a beaucoup d'applications en industrie. Les méthodes de résolution varient entre les méthodes exactes (Branch-and-bound : procédure de séparation et évaluation), les recherches heuristiques et les métaheuristiques.

Or, la majorité des travaux traitent le problème dans sa forme mono-objectif avec pour objectif principal la minimisation du temps de fin des tâches (makespan). Pourtant l'intérêt de considérer plusieurs objectifs n'est plus à démontrer et pour avoir une vue générale des quelques travaux existants en ordonnancement multi-objectif, le lecteur peut se référencer au livre de T'Kindt et Billaut [81]. Ici nous nous intéressons à une version bi-objectif du problème.

2.1.2 Description du problème et notations

Un problème de flowshop consiste en N tâches à exécuter sur M machines. Les machines sont des ressources critiques, ainsi une machine ne peut être affectée à plusieurs tâches simultanément. Chaque tâche t_i est décomposée en M opérations devant être exécutées consécutivement $t_i = (t_{i1}, t_{i2}, \ldots, t_{iM})$, où t_{ij} représente la $j^{ème}$ opération de la tâche t_i nécessitant la machine m_j. A chaque tâche t_i est associée une date de disponibilité r_i, une date de fin souhaitée d_i et à chaque opération t_{ij} est associée une durée d'exécution p_{ij}. Notons C_i la date de fin de réalisation effective de la tâche t_i.
Dans le cadre de ce travail, nous nous intéressons à la recherche de l'ordonnancement de permutation (les tâches sont dans le même ordre sur toutes les machines) minimisant deux objectifs, qui sont la plus grande date de fin des tâches (makespan) et la somme des retards. Ainsi les deux objectifs peuvent être définis par :

$$f_1 = Cmax : minimiser \quad (\max_i C_i)$$

$$f_2 = T : minimiser \quad (\sum_i max(0,\ C_i - d_i))$$

et le problème se note donc $F/perm,\ d_i/(Cmax,\ T)$.
Il s'agit bien ici d'une résolution multi-objectif du problème, où aucun des objectifs n'est plus important que l'autre et où le but est la recherche de l'ensemble des solutions Pareto.

2.2 Revue rapide des méthodes exactes existantes

Nous présentons ici les principales méthodes exactes développées de façon à obtenir l'ensemble ou une partie du front Pareto.

2.2.1 L'agrégation linéaire

Cette méthode populaire transforme le problème multi-objectif en un problème mono-objectif en combinant linéairement les différents objectifs. Ainsi, le nouveau problème obtenu, car il s'agit alors d'un problème différent, consiste à optimiser $\sum_i \lambda_i f_i$. Le théorème de Geoffrion [30] indique qu'en utilisant différentes valeurs pour le vecteur λ, il est possible d'obtenir toutes les solutions supportées du problème multi-objectif initial. Par contre, aucune solution non supportée peut être trouvée par cette méthode.

La méthode d'agrégation linéaire a donc ses limites. Toutefois, elle est intéressante pour des problèmes ayant de nombreux objectifs et/ou un grand nombre de solutions supportées bien réparties. Dans ce contexte, il peut être suffisant de générer les solutions supportées.

2.2.2 La recherche dichotomique

La recherche dichotomique offre un schéma d'application de l'agrégation linéaire permettant d'obtenir les solutions non supportées [19]. Cette méthode consiste à explorer de façon dichotomique des intervalles de front de plus en plus petits.
Tout d'abord les solutions extrèmes sont recherchées. Puis une recherche est menée entre ces solutions r et s suivant une direction perpendiculaire à la droite (r, s). En interdisant de ré-obtenir les solutions r et s et en éliminant les solutions dominées par ces solutions, cette recherche trouve la meilleure solution Pareto relativement à cette direction de recherche, solution qui peut alors être non supportée. Cette nouvelle solution crée deux nouveaux intervalles qu'il faut explorer de la même façon.
Cette méthode, dédiée au bi-objectif, est intéressante mais nécessite de l'ordre de 2^n recherches, si n est le nombre de solutions du front Pareto.

2.2.3 Méthode ϵ-contrainte

Le principe de la méthode ϵ-contrainte qui consiste, dans le cas bi-objectif, à borner l'un des objectifs (en général le plus difficile à résoudre) et à optimiser l'autre objectif (optimisation mono-objectif) en tenant compte de cette borne [33], est intéressant lorsque l'on cherche à énumérer toutes les solutions d'un front Pareto. En effet, en utilisant cette méthode itérativement, en repartant à chaque fois de la solution trouvée pour définir la borne suivante, il est possible en utilisant une méthode exacte mono-objectif de générer, pour des problèmes combinatoires, l'ensemble des solutions Pareto.

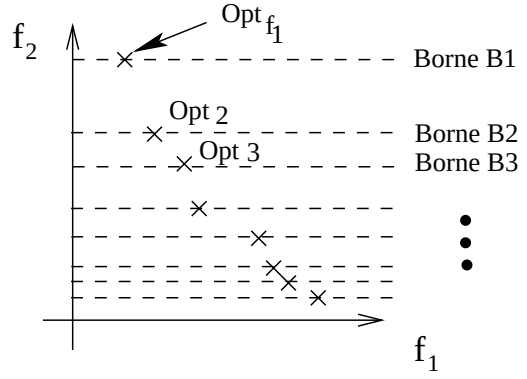


FIG. 2.1 – Illustration de la méthode ϵ-contrainte.

La figure 2.1 illustre un exemple pour lequel, la solution efficace optimale pour l'objectif f_1 est d'abord recherchée (solution Opt_{f_1}). Cette solution détermine la borne $B1$ sur l'objectif

f_2 en dessous de laquelle l'objectif f_2 va devoir être optimisé. Cela nous donne la solution Opt_2 qui elle-même détermine la borne $B2$, etc...

L'inconvénient principal de cette méthode est qu'elle nécessite une résolution mono-objectif pour chacune des solutions du front. Lorsque ce nombre est élevé, cela peut être vu comme une limite, d'autant plus lorsque la méthode de résolution mono-objectif est coûteuse. De plus, lorsqu'il n'existe pas de méthode mono-objectif efficace, rechercher une solution particulière (respectant une borne, par exemple) est souvent synonyme d'énumération de nombreuses autres solutions dont certaines peuvent être Pareto optimales. Ainsi certaines solutions seront énumérées plusieurs fois sans que la méthode les repère.

2.3 Méthode deux-phases

La méthode deux-phases a initialement été proposée par Ulungu et Teghem pour la résolution d'un problème d'affectation bi-objectif [82]. Comme son nom l'indique, cette méthode est décomposée en deux étapes : la première consiste à trouver toutes les solutions supportées du front Pareto, puis la deuxième phase cherche entre ces solutions les solutions Pareto non supportées. Cette méthode travaille donc essentiellement dans l'espace objectif.

2.3.1 Première phase

L'objectif de la première phase est d'obtenir l'ensemble des solutions Pareto supportées. Comme nous l'avons vu précédemment, ces solutions ont l'avantage d'être relativement faciles à trouver puisqu'elles optimisent une certaine combinaison linéaire des objectifs.
Ainsi, durant la première phase de la méthode, les deux solutions extrêmes (solutions optimisant chacune un des deux objectifs) sont recherchées (voir figure 2.2.a). Puis, de façon récursive, dès que deux solutions supportées r et s sont trouvées, la méthode recherche d'éventuelles autres solutions supportées entre r et s, à l'aide de combinaisons linéaires bien choisies des objectifs (voir figure 2.2.b et 2.2.c). A la fin de la première phase l'ensemble des solutions supportées est donc trouvé (voir figure 2.2.d).
Cette première phase rappelle la méthode dichotomique, mais ici seules les solutions supportées sont recherchées. Pour cela, lors de l'exploration entre deux solutions, on s'autorise à retrouver l'une de ces deux solutions, lorsqu'il n'existe pas d'autre solutions supportées dans l'intervalle.

2.3.2 Deuxième phase

La deuxième phase consiste alors en la recherche des solutions non supportées appartenant au front Pareto. Ces solutions ne peuvent être obtenues par combinaisons d'objectifs. Ulungu et Teghem proposent alors d'utiliser les solutions supportées trouvées pour réduire l'espace de recherche en argumentant que les solutions Pareto non supportées restantes sont forcément dans les triangles rectangles basés sur deux solutions supportées consécutives (voir figure

2.2.e). Ainsi, une recherche de type deuxième phase est exécutée entre chaque couple de solutions supportées adjacentes (voir figure 2.2.f et 2.2.g). La méthode de recherche au sein de ces triangles dépend du problème étudié. A la fin de la deuxième phase, toutes les solutions Pareto sont trouvées. Notons, qu'il aura été nécessaire au préalable de préciser si l'on recherche le front minimal ou maximal complet.

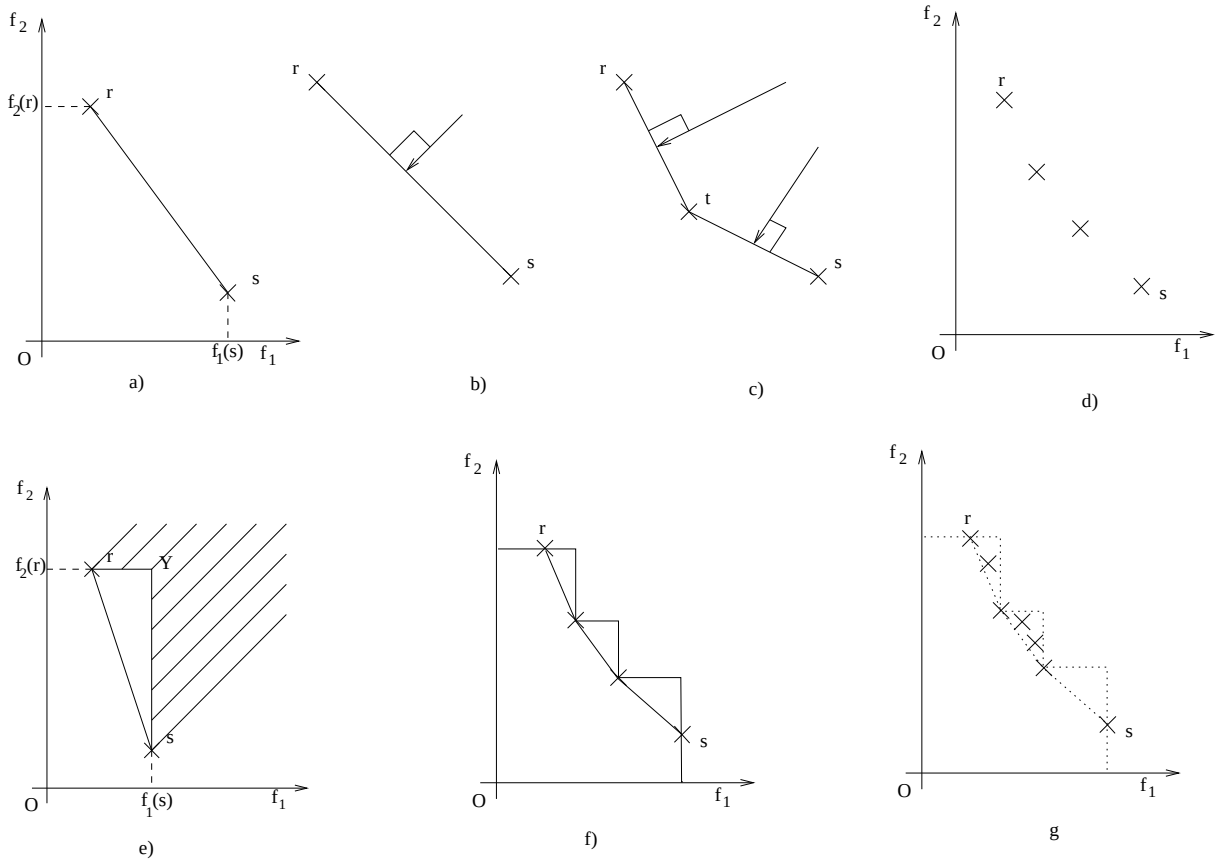


FIG. 2.2 – Illustration des différentes étapes de la méthode deux phases.

2.3.3 Discussion

La méthode deux-phases présente un schéma de résolution exacte très intéressant car très général et qui ne dépend pas du problème. Son intérêt réside dans une décomposition de l'espace de recherche et l'utilisation de méthodes mono-objectifs pour les différentes résolutions successives (recherche des extrêmes, résolution des agrégations...). Appliquer la méthode deux-phases pour la résolution d'un problème bi-objectif nécessite donc d'avoir une méthode mono-objectif efficace (si possible polynomiale). C'est le cas pour les problèmes traités par Ulungu et Teghem, et c'est ce qui rend leur méthode performante pour ces problèmes là.
Cependant, lorsque la restriction du problème à un seul objectif génère un problème déjà $\mathcal{NP}$-difficile, la non existence de méthode exacte efficace pouvant optimiser chaque objectif

séparément peut compromettre l'intérêt de la méthode et notamment rendre la première phase très coûteuse. Il est alors important d'adapter le schéma général. C'est ce que nous avons fait pour le problème du flow-shop bi-objectif présenté précédemment.

2.4 Adaptations et améliorations proposées de la méthode deux-phases

Ayant décidé d'utiliser la méthode deux-phases pour le problème du flowshop bi-objectif, nous l'avons adaptée. De plus certaines spécificités du problème nous ont conduits à proposer des améliorations pour l'utilisation de cette méthode sur des problèmes d'ordonnancement. Ces améliorations ont été réalisées dans le cadre de la thèse de Julien Lemesre.

2.4.1 Adaptations au problème du flowshop bi-objectif

Utiliser la méthode deux-phases pour la résolution d'un problème nécessite une méthode d'optimisation mono-objectif pour chacune des fonctions objectifs. Dans le problème $F/perm, d_i/(Cmax, T)$, chacun des objectifs pris seul est $\mathcal{NP}$-difficile (au sens fort pour le $Cmax$ et au sens faible pour T). Puisqu'il n'existe pas de méthode efficace pour ces deux objectifs, nous avons décidé de développer une procédure de séparation et évaluation (PSE). Bien qu'il existe une méthode pseudo-polynomiale permettant de résoudre le problème de retard total (min T), les expérimentations ont montré que l'utilisation de cette méthode était trop coûteuse en temps dans la PSE développée. Aussi, nous avons proposé, pour l'objectif du retard, une nouvelle borne inférieure [58].

2.4.2 Améliorations pour les problèmes d'ordonnancement

Les problèmes d'ordonnancement (mono ou multi-objectifs) ont la particularité d'avoir des solutions très proches les unes des autres en terme de valeur de la (les) fonction(s) objectif(s). Il n'est par rare non plus de voir plusieurs solutions différentes avoir exactement le même coût. L'une des conséquences de cette remarque est que la recherche des solutions extrêmes peut être très longue. Aussi, nous proposons de rechercher ces extrêmes en utilisant un ordre lexicographique sur les objectifs. Par exemple, pour rechercher la solution extrême du front Pareto pour un objectif f_1, cet objectif est tout d'abord optimisé (sans tenir compte de l'objectif f_2), puis en gardant la valeur obtenue pour f_1, le deuxième objectif est optimisé.

Une autre observation de ce type de problèmes nous montre que les solutions supportées ne sont pas forcément bien réparties sur le front (cf figure 1.3 du chapitre précédent). Rappelons que l'intérêt de rechercher les solutions supportées réside dans la réduction de l'espace de recherche pour la deuxième phase. La question du compromis entre le temps passé à la recherche des solutions supportées et le temps gagné par un plus petit espace de recherche se pose. Lorsqu'il n'est pas coûteux de rechercher les solutions supportées (existence d'une méthode polynomiale pour résoudre l'agrégation), ce n'est pas un problème. Par contre,

lorsque la méthode de résolution de l'agrégation est de type exponentiel, il convient d'affiner la stratégie. Ce que nous proposons dans ce cas est de ne pas exécuter de première phase entre deux solutions supportées trop proches l'une de l'autre. Ceci permet de gagner du temps que nous pouvons utiliser pour la deuxième phase. Remarquons qu'il est possible de ne pas exécuter toutes les recherches de type première phase, puisque la seconde phase est en mesure de trouver les solutions supportées restantes. En effet, la recherche de type deuxième phase entre deux solutions r et s n'est pas réellement restreinte au triangle (r, s, Y) mais au rectangle ($f_2(r)$, O, $f_1(s)$, Y) complet (voir figure 2.2.e). Ainsi, s'il existe une solution supportée entre r et s non trouvée par la première phase (parce que les recherches auraient été arrêtées avant d'avoir terminé la première phase), alors elle sera trouvée par la recherche de type deuxième phase entre les solutions r et s.

2.4.3 Parallélisation

La méthode deux-phases est une méthode naturellement parallèle. En effet :
- Lors de la première phase, lorsqu'une nouvelle solution est trouvée, cela engendre deux nouvelles recherches indépendantes qui peuvent donc être réalisées en parallèle.
- Lors de la deuxième phase, les recherches au sein de chaque triangle sont également indépendantes.

Nous proposons donc d'adopter un modèle de parallélisme de type maître-esclave. Un processeur sert de maître et distribue aux autres processeurs les sous-problèmes (recherche de type première ou deuxième phase) à résoudre.
De plus, afin d'utiliser au mieux les ressources disponibles (P processeurs), nous proposons de lancer en parallèle de la recherche des deux extrêmes (ce qui occupe 2 processeurs), $P-2-1$ recherches uniformément réparties (voir fig 2.3), ce qui permet de trouver, si elles existent, des solutions supportées situées dans ces axes de recherche. Ainsi, dès le début, différentes recherches de solutions supportées pourront être faites en parallèle. Cela permet d'optimiser l'occupation des processeurs.

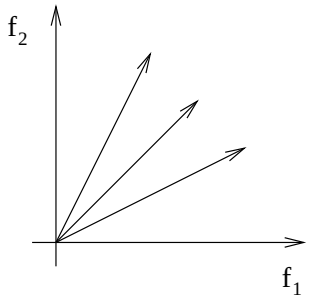


FIG. 2.3 – Optimisation du début de la méthode à l'aide de recherches aléatoires.

Enfin, nous pouvons remarquer que les deux phases sont relativement indépendantes, puisque dès qu'une première phase est terminée entre deux solutions supportées, une deuxième phase peut être lancée entre ces deux solutions, sans que toutes les premières phases soient finies.

Le maître gère donc une liste de priorités des tâches (sous-problèmes) en donnant une plus forte priorité aux tâches correspondant aux recherches de type première phase (puisqu'elles peuvent générer de nouvelles recherches).

2.4.4 Application au problème de Flowshop bi-objectif

Les améliorations proposées ainsi que le modèle de parallélisation ont fait leurs preuves sur le problème de flowshop bi-objectif (qui nous sert de problème test). Cela a donné lieu à une présentation à la conférence MOPGP'04 (Multi-Objective Programming and Goal Programming) [57] et à un article dans la revue EJOR [58].

Instances	Temps		
	Méthode originale	Avec améliorations	Avec parallélisation (4 processeurs)
20×5 (1)	6.5 sec	3.5 sec	inutile
20×5 (2)	3 mn 10 sec	2 mn 50 sec	inutile
20×10 (1)	27 h	12 h	8 h 15 mn
20×10 (2)	18 h 4 mn	7 h 23 mn	4 h 30 mn
20×20 (1)	Non résolu	Non résolu	7 jours

TAB. 2.1 – Temps d'exécution des différentes méthodes : Benchmarks de Taillard.

Le tableau 2.1 présente pour des problèmes de Taillard étendus au cas bi-objectif[1], les différents temps de résolution nécessaires en fonction de la méthode utilisée. Le problème 20×5 (x) indique qu'il s'agit du $x^{ème}$ problème de Taillard à 20 jobs et 5 machines. De même, le tableau 2.2 présente les mêmes résultats pour les Benchmarks de Reeves pour lesquels deux types de problèmes ont été générés. Les problèmes sont dits faciles ou difficiles en fonction de la difficulté d'optimiser l'objectif de la somme des retards dans le cas mono-objectif.
Sont comparées ici :

- la méthode deux-phases initiale telle que proposée par Ulungu et Teghem,
- la méthode deux-phases avec les améliorations proposées (recherche lexicographique des extrêmes et non exécution de recherches de type première phase entre des solutions trop proches),
- la version parallèle.

Ces tableaux illustrent bien l'apport des différentes améliorations en terme de gain de temps d'exécution. Notons d'ailleurs que la parallélisation permet la résolution de problèmes de Taillard de taille 20×20, ce qui n'était pas le cas avec la méthode initiale.

[1] Voir http ://www.lifl.fr/∼lemesre pour les détails sur l'extension.

Instances	Temps		
	Méthode originale	Avec améliorations	Avec parallélisation (4 processeurs)
	Instances Faciles		
20×5 (1)	53 sec	20 sec	inutile
20×10 (7)	3 mn 01 sec	35 sec	inutile
20×15 (13)	5 h 10 mn	2 h 09 mn	1 h 20 mn
20×15 (17)	10 h 41 mn	6 h 01 mn	4 h
	Instances Difficiles		
20×5 (1)	54 sec	39 sec	inutile
20×10 (7)	1 h 22 mn	32 mn 41 sec	15 mn 50 sec

TAB. 2.2 – Temps d'exécution des différentes méthodes : Benchmarks de Reeves.

2.5 PPM : Parallel Partitionning Method

Nous avons vu précédemment deux méthodes (l'utilisation de la méthode ϵ-contrainte et la méthode deux-phases) permettant de générer les fronts Pareto exacts pour des problèmes bi-objectifs. Nous avons indiqué quelques limites à ces méthodes, notamment dans les cas où les méthodes mono-objectifs, permettant d'optimiser l'un des deux objectifs ou bien une agrégation d'objectifs, ne sont pas efficaces. Dans ce cas, il peut être nécessaire de faire appel à des méthodes de type séparation et évaluation et il devient très important de limiter le nombre d'appels à ces méthodes.
Pour palier à ces contre-performances lors de l'étude de problèmes non structurés, nous proposons la méthode de partitionnement parallèle. Cette méthode s'inspire de l'idée de découpage de l'espace de recherche de la méthode deux-phases et s'inspire de la méthode ϵ-contrainte pour réaliser ce découpage. Cette nouvelle méthode tient compte des différentes limites énoncées plus haut et essaie d'y répondre. Ceci constitue également le travail de la thèse de Julien Lemesre.

2.5.1 Fonctionnement de la méthode

Cette méthode est décomposée en trois phases :

1. **Recherche des extrêmes et partitionnement de l'espace de recherche :** les deux solutions extrêmes sont recherchées (cf figure 2.4.a). Ces deux solutions indiquent donc les valeurs minimales et maximales du front Pareto pour chacun des objectifs. L'espace contenu entre ces deux solutions est ensuite découpé de façon uniforme suivant l'objectif le plus difficile à résoudre (supposons un découpage suivant f_2 - cf figure 2.4.b).
2. **Recherche d'une solution par partition :** à la manière de la méthode ϵ-contrainte, la solution optimisant le second objectif (f_1) est recherchée pour chacune des partitions

(cf figure 2.4.c). A la fin de cette deuxième phase, des solutions, que l'on aura voulu les mieux réparties possible, sont trouvées. Remarquons que les solutions obtenues sont toutes Pareto optimales mais pas nécessairement supportées.

3. **Recherche des solutions Pareto dans les sous-espace :** cette dernière phase consiste à rechercher dans chacun des sous-espaces délimités par deux solutions adjacentes obtenues lors de la phase précédente (cf figure 2.4.d), toutes les solutions Pareto (supportées et non supportées) existantes (cf figure 2.4.e).

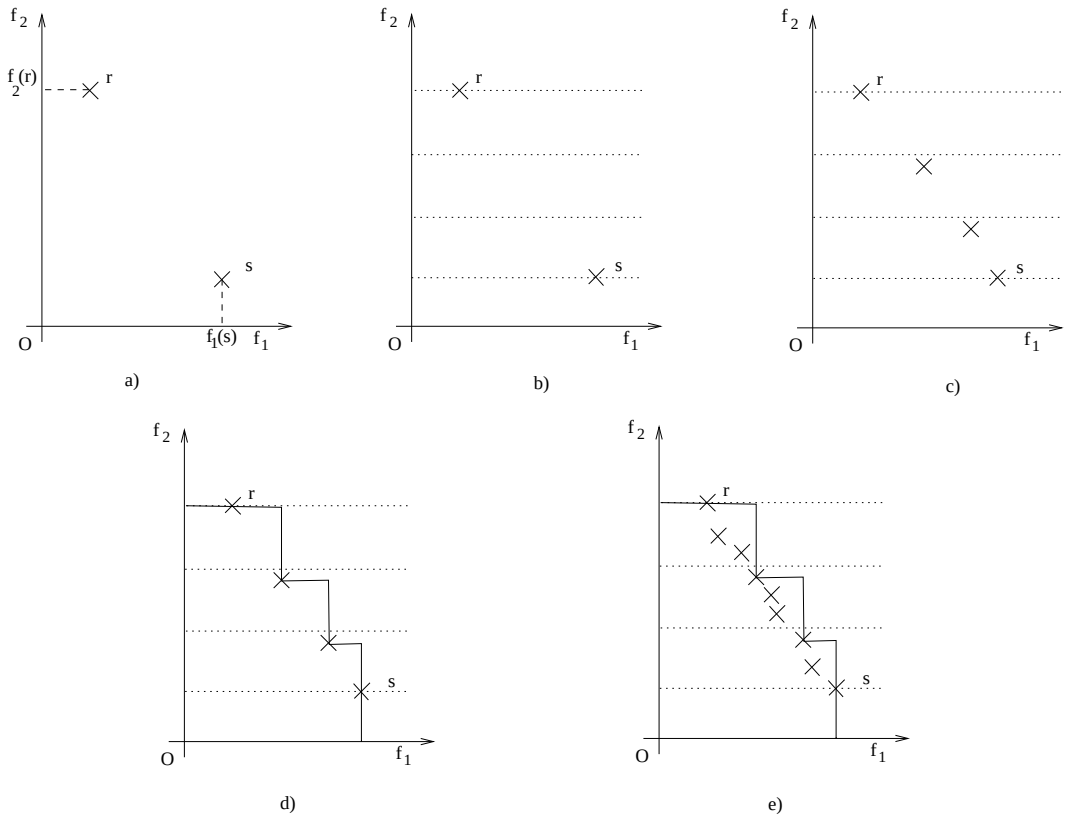


FIG. 2.4 – Illustration de la méthode par partitions (PPM).

Cette méthode a donc l'avantage de découper l'espace de recherche tout en restant plus indépendante de la structure du front Pareto. En effet, aucune hypothèse n'est prise concernant la répartition des solutions supportées le long du front, puisque la deuxième phase recherche des solutions qui ne sont pas forcément supportées (mais qui sont Pareto optimales). Ce faisant, ceci nous permet d'obtenir pour la troisième phase des sous-espaces de même ordre de grandeur, ce qui permet d'équilibrer la recherche parallèle au sein de chacun des sous-espaces. Le nombre de partitions reste un paramètre sur lequel il est possible d'intervenir ce qui permet d'être flexible et de s'adapter à l'environnement parallèle à disposition. Remarquons aussi, que suivant le type de méthode utilisée pour la troisième phase, il peut être possible de trouver toutes les solutions d'une même partition en une seule exécution.

2.5.2 Exemple d'application

Ainsi, cette nouvelle méthode proposée répond aux différents problèmes soulevés lors de l'étude des méthodes existantes, étudions maintenant ses performances sur le problème illustratif que nous avons choisi ($F/perm$, $d_i/(Cmax, T)$). Les mêmes instances que pour la comparaison de la méthode deux-phases et de ses améliorations ont été choisies.

Efficacité de la méthode

Le tableau 2.3 compare PPM avec la version améliorée de TPM.
Dans ce tableau nous pouvons remarquer que PPM permet un gain de temps important même en comparaison avec la version améliorée de TPM qui tient compte des adaptations que nous avions proposées pour le problème de flowshop bi-objectif. Le gain est plus ou moins important en fonction de la structure du front.

Instances	Temps		
	TPM amélioré	PPM	Recherche extrêmes
	Taillard		
20×5 (1)	3.5 sec	3.3 sec	2.6 sec
20×5 (2)	2 mn 50 sec	2 mn 50 sec	2 mn 20 sec
20×10 (1)	12 h	8 h 58 mn	1 h 02 mn
20×10 (2)	7 h 23 mn	5 h 01 mn	1 h 05 mn
	Reeves Faciles		
20×5 (1)	20 sec	18 sec	6 sec
20×10 (7)	35 sec	31 sec	18 sec
20×15 (13)	2 h 09 mn	1 h 55 mn	1 h 13 sec
20×15 (17)	6 h 01 mn	3 h 42 mn	1 h 21 mn
	Reeves Difficiles		
20×5 (1)	39 sec	32 sec	2 sec
20×10 (7)	32 mn 41 sec	22 mn 19 sec	2 mn 01 sec

TAB. 2.3 – Efficacité de la méthode par partitions.

En effet, lorsque les solutions supportées sont relativement bien réparties sur le front, la méthode deux phases se comporte de façon similaire à PPM, surtout dans sa version améliorée où les recherches de type première phase ne sont pas exécutées entre solutions supportées trop proches. Par contre, lorsque les solutions supportées sont moins bien réparties, alors PPM devient plus intéressante.

Influence de la recherche des extrêmes

Le tableau 2.3 montre aussi le temps nécessaire pour obtenir les deux solutions extrêmes. Nous remarquons que ce temps est loin d'être négligeable, puisqu'il peut atteindre plus de

80% du temps de recherche global. Ceci est expliqué par la difficulté de l'optimisation de l'objectif du retard total. Cette phase étant commune aux deux méthodes, elles en patissent toutes les deux.

Front minimal / Front maximal complet

Une autre étude intéressante consiste à comparer l'efficacité de la méthode lorsque l'on s'intéresse plus seulement au front minimal, mais au frontmaximal complet. C'est à dire lorsque l'on veut obtenir, dans l'espace de décision, toutes les solutions ayant les mêmes valeurs pour les objectifs et appartenant au front. Le tableau 2.4 indique pour PPM le temps nécessaire à l'obtention des fronts minimal et maximal complet. Le tableau indique également le nombre de solutions trouvées. Ce tableau montre que même si le nombre de solutions peut être beaucoup plus grand dans le front maximal complet, les temps nécessaires à la résolution ne sont pas forcément beaucoup plus importants qu'avec le front minimal.

Instances	Front minimal		Front maximal complet	
	Temps	Nb Solutions	Temps	Nb Solutions
	Taillard			
20×5 (1)	3.3 sec	4	4 sec	18
20×5 (2)	2 mn 50 sec	6	3 mn 15 sec	21
20×10 (1)	8 h 58 mn	42	9 h 07 mn	43
20×10 (2)	5 h 01 mn	31	5 h 15 mn	31
	Reeves Faciles			
20×5 (1)	18 sec	24	19 sec	50
20×10 (7)	31 sec	6	40 sec	13
20×15 (13)	1 h 55 mn	11	1 h 59 sec	12
20×15 (17)	3 h 42 mn	19	3 h 55 mn	28
	Reeves Difficiles			
20×5 (1)	32 sec	42	33 sec	71
20×10 (7)	22 mn 19 sec	40	25 mn 24 sec	368

TAB. 2.4 – Influence de la recherche du front maximal complet.

2.5.3 Conclusions et perspectives

Les expérimentations sur PPM montre l'intérêt de l'utiliser pour des problèmes pour lesquels l'optimisation de chacun des objectifs est difficile puisque PPM limite le nombre de recherches. De plus, PPM est plus robuste car moins sensible à la structure du front Pareto.

La méthode PPM ayant donné de bons résultats pour les problèmes bi-objectifs, une perspective naturelle concerne l'extension de cette méthode pour k objectifs. Les travaux actuellement en cours dans la thèse de Julien Lemesre, montrent que théoriquement l'extension

est possible. La limite de ce type de méthode - méthode exacte pour k objectifs - est la taille des problèmes pouvant être résolus. En effet, lors du passage du mono-objectif au bi-objectif, dans le cas du problème du flowshop, la taille maximale des problèmes pouvant être résolus optimalement passe de 50×20 à 20×20. Aussi l'ajout d'un objectif supplémentaire fera encore diminuer cette taille. Il est cependant à remarquer que la diminution de la taille maximale des problèmes pouvant être résolus optimalement en passant de k à $k+1$ objectifs dépend fortement de la difficulté intrasèque de l'objectif ajouté.

Une autre perspective intéressante concerne une meilleure exploitation du parallélisme. En effet, même si PPM optimise le parallélisme de haut niveau en rendant indépendantes chaque recherche dans chacune des phases, certaines recherches restent *trop* longues. Il serait donc intéressant de paralléliser chaque méthode de recherche (recherche des extrêmes, des solutions représentantes de chaque partition, des solutions Pareto au sein des triangles) de façon à les accélérer. Ceci est tout à fait réalisable surtout lorsque les problèmes associés sont difficiles et que les seules méthodes possibles sont basées sur des méthodes énumératives. La parallélisation de bas niveau de ce type de méthodes fait d'ailleurs partie de la thèse de Mohand Mezmaz qu'il réalise au sein de l'équipe.

Chapitre 3

Coopération entre méthodes exactes et métaheuristiques pour l'optimisation multi-objectif

L'utilisation de méthodes exactes étant limitée à des problèmes de petites tailles, il convient, notamment pour pouvoir traiter des problèmes réels, de développer des méthodes heuristiques. Comme nous l'avons déjà signalé, les métaheuristiques et en particulier les algorithmes évolutionnaires, travaillant sur des populations de solutions, sont bien adaptés à l'optimisation multi-objectif. Nous commençons donc ce chapitre par montrer l'apport des métaheuristiques pour l'optimisation multi-objectif. Puis, nous présentons des méthodes coopératives.

3.1 Optimisation multi-objectif et métaheuristiques

Afin d'étudier les possibilités offertes par les métaheuristiques pour l'optimisation multi-objectif, nous avons utilisé comme exemple d'application le même problème de flowshop bi-objectif. Ce travail a été initié en 2000 dans le cadre d'une collaboration avec l'université USTHB d'Alger et en particulier avec Mohamed Hakim Mabed et Malek Rahoual. Il a donné lieu à une présentation à MOSIM'01 [59] et à EMO'01 [78]. Ce travail a ensuite été poursuivi par Matthieu Basseur dans le cadre de sa thèse qu'il a réalisée au sein de l'équipe [4].

La difficulté de l'optimisation multi-objectif réside dans l'absence d'une relation d'ordre total qui lie l'ensemble des solutions du problème. Sur le plan des algorithmes évolutionnaires, ce manque apparaît dans la difficulté de concevoir un opérateur de sélection qui affecte à chaque individu une probabilité de sélection proportionnelle à la performance de cet individu. Un autre inconvénient est lié à la perte prématurée de la diversité. D'où la nécessité de concevoir des techniques de maintien de la diversité au sein de la population.
Nous exposons ici ce premier travail préliminaire réalisé en 2000, puis donnerons les principales évolutions ayant eu lieu depuis autour des métaheuristiques pour l'optimisation multi-

objectif.

3.1.1 Algorithme génétique pour le flowshop bi-objectif

Codage : L'application des AGs à un problème donné nécessite une représentation chromosomique d'une solution (dans notre cas, un ordonnancement des jobs). La séquence de passage des jobs sur les machines étant identique (Flow-Shop de permutation), il suffit de coder le séquencement des jobs sur une seule machine (permutation).
L'évaluation d'une séquence donnée, nécessite le calcul des dates de début et de fin des tâches. Les objectifs à optimiser étant réguliers, ce calcul est réalisé par la construction de l'ordonnancement au plus tôt des tâches. Le calcul se fait simplement de manière récursive, à commencer par les tâches planifiées en premier.

Opérateurs de recombinaison : De même, le choix d'opérateurs génétiques qui serviront à faire évoluer la recherche est à réaliser. Nous nous sommes inspirés des opérateurs définis par Murata et Ishibuchi [66]. L'opérateur de mutation consiste en un choix aléatoire de deux points dans le chromosome. Une rotation est alors effectuée. L'opérateur de croisement, aussi nommé "croisement deux points", consiste également en un choix aléatoire de deux points de croisement. Un individu fils est alors généré en conservant les extrémités du chromosome du parent1 et en complétant avec les jobs non déjà insérés en suivant leur ordre d'apparition dans le parent2.

Opérateur de sélection : L'absence d'une relation d'ordre totale entre les différentes solutions possibles d'un problème multi-objectif, nécessite de redéfinir la notion d'optimalité. En effet, la dominance au sens de Pareto représente une relation d'ordre partiel sur l'ensemble des points de l'espace de recherche.
Lors de la phase de sélection, l'aspect multi-objectif apparaît dans la façon dont les individus sont triés suivant leurs performances en vu d'être sélectionnés. On distingue alors :

1. La sélection lexicographique.
2. La sélection par Ranking (NSGA [76], NDS [25], WAR [8]).
3. La sélection parallèle [73].
4. La sélection par somme pondérée des objectifs à poids variable [66].

Une comparaison de six opérateurs de sélection implémentés montre une importante amélioration de la recherche avec l'introduction de l'élitisme dans la phase de sélection [78]. Cette comparaison montre également que les statégies non Pareto (somme pondérée, sélection parallèle) ne semblent pas adaptées à ce type de problème. Les stratégies Pareto (NSGA, NDS, WAR) ont des performances similaires.

Maintien de la diversité : Les AGs classiques sont réputés pour être très sensibles quant au choix de la population initiale ainsi qu'aux mauvais échantillonnages lors de la sélection. Cette fragilité est observable sur le plan de la perte de diversité ou ce qu'on appelle aussi la

dérive génétique. Pour pallier à cet inconvénient plusieurs approches visant à maintenir la diversité dans la population ont été proposées dans la littérature :

- Introduction de nouveaux individus (random immigrant).
- Stochastic Universal Sampling (SUS) [3].
- Maintien d'une distance minimale [61].
- Crowding [37] : Holland suggère qu'après la génération d'un individu, celui-ci remplace dans la population l'individu qui lui est le plus semblable.
- Restriction de voisinage [29] : l'idée est de ne permettre la reproduction entre deux individus que s'ils sont similaires, ou au contraire, empêcher la reproduction entre individus similaires pour éviter l'inceste.
- Niches écologiques (sharing) : le principe du Sharing [32] est la dégradation de l'adéquation des individus appartenant à des espaces de recherche de forte concentration de solutions.

Les résultats des expérimentations menées montrent que le sharing phénotypique (espace des objectifs) donne de meilleurs résultats que le sharing génotypique (espace des solutions), même si ce dernier apporte plus de diversité. Le sharing combiné apporte ces deux aspects : performances et diversification.

Hybridation avec une recherche locale : Notre intérêt s'est ensuite porté sur l'utilisation d'une recherche locale (RL) comme moyen d'accélération et de raffinement de la recherche. L'idée dans ce cas est de lancer l'AG en premier lieu afin d'approcher la frontière Pareto, après quoi la recherche locale s'occupera du raffinement des solutions trouvées par l'AG afin de mieux approcher l'ensemble des solutions Pareto. Dans ce contexte l'AG est utilisé pour son pouvoir d'exploration et doit fournir une approximation de l'ensemble du front Pareto. Puis, la recherche locale utilise les solutions trouvées par l'AG comme point de départ, en vue de les améliorer. Ceci est illustré par la figure 3.1.
L'utilisation de la recherche locale nécessite premièrement le choix de la manière de générer le voisinage d'une solution donnée. Nous nous sommes inspirés pour cela de l'opérateur de mutation. Le procédé d'hybridation consiste à générer pour chaque individu de la population Pareto trouvé, l'ensemble de son voisinage. Les voisins non dominés dans la population Pareto sont insérés dans celle-ci, et les solutions nouvellement dominées sont supprimées. Ce procédé est réïtéré jusqu'à ce qu'aucun voisin d'aucune solution Pareto ne soit inséré dans la population Pareto. Les mesures de performances de l'AG hybride montrent que la recherche locale ne présente aucun intérêt pour les problèmes de petites tailles pour lesquels de bonnes solutions sont trouvées par l'AG seul. Cependant l'apport de la coopération se fait sentir dès que la taille du problème augmente.
Les résultats montrent la capacité de la coopération à trouver de bonnes solutions dispersées ce qui nous offre un bon échantillonnage du front Pareto.

Conclusion et perspectives

La construction de ce premier AG multiobjectif, à une époque où les travaux dans ce domaine étaient encore très jeunes, s'est faite par introduction progressive de concepts tels que la sélection, le maintien de la diversité et la coopération. A chaque étape nous avions illustré

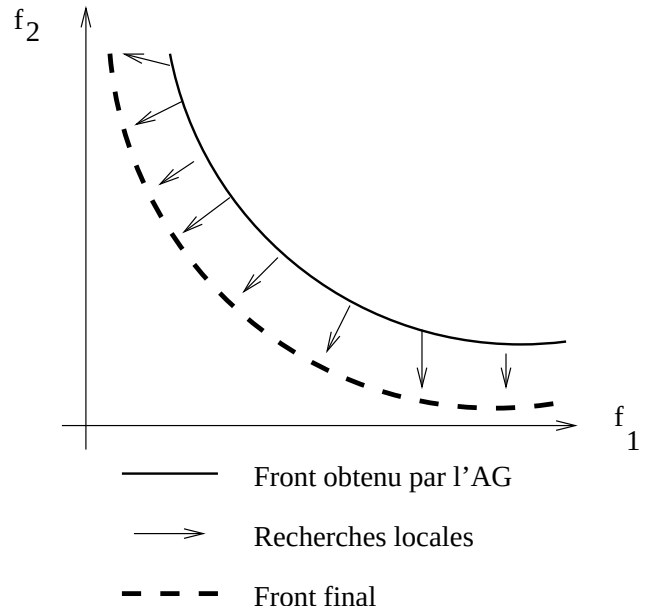


FIG. 3.1 – Hybridation avec une recherche locale.

l'apport du mécanisme introduit, ce qui nous a permis de formuler les conclusions suivantes. Les stratégies de sélection Pareto (NSGA, NDS, WAR) sont mieux adaptées au cas multi-objectif. L'efficacité de telles méthodes est améliorée avec l'introduction de l'élitisme lors de la phase de sélection. Cependant, les risques de dérive génétique et d'instabilité de la recherche restent présents. Les stratégies de diversification permettent de prévenir de tels problèmes. Ensuite, la recherche locale a été utilisée comme moyen de raffinement et d'accélération de la recherche. L'apport de la coopération apparaît surtout pour des problèmes de grande taille.

Continuité des travaux

Suite à ce travail préliminaire datant de 2000, Matthieu Basseur a proposé dans le cadre de sa thèse différents schémas d'algorithmes évolutionnaires basés sur différents agents aux propriétés complémentaires. Actuellement ses méthodes parviennent à trouver le front exact pour des problèmes de Taillard de petites et moyennes tailles (jusque $20jobs \times 20machines$) en des temps raisonnables (moins de 20 minutes) et fournissent pour les problèmes de grandes tailles de bonnes approximations. Pourtant pour ces plus grands problèmes, il est légitime de chercher à mieux faire et c'est ce qui motive la recherche de coopération entre méthodes exactes et métaheuristiques. Mais avant de nous intéresser à la coopération, regardons l'état des recherches autour des métaheuristiques pour l'optimisation multi-objectif.

3.1.2 Petit état de l'art sur métaheuristiques en optimisation multi-objectif

Les métaheuristiques ont été largement utilisées pour la résolution de problèmes multi-objectifs. Ceci est d'autant plus vrai pour les algorithmes évolutionnaires travaillant sur des populations de solutions et étant bien adaptés pour générer des fronts de solutions. Ainsi Deb publiait dès 2001 un livre sur l'utilisation des algorithmes évolutionnaires pour l'optimisation multi-objectif [16].
Récemment un livre publié par Coello Coello et Lamont montre le spectre des applications multi-objectifs abordées par les algorithmes évolutionnaires [13]. Ces applications vont du design d'appareils électromagnétiques, au design de circuits logiques, à l'optimisation de problèmes de tournées en passant par l'analyse de promoteurs dans le domaine de la bio-informatique. Chaque application apportant de nouveaux challenges, de nouveaux opérateurs et de nouvelles stratégies ou combinaisons de méthodes sont proposés.

Principaux Challenges : Comme nous l'avons dit plus haut, deux aspects liés au multi-objectif sont à prendre particulièrement en compte. Il s'agit :
- de la non existence d'ordre total entre les solutions. Ainsi une question est : comment sélectionner les individus afin que ceux qui sont non dominés soient préférés aux autres ?
- de la recherche d'un ensemble de solutions et non d'une solution unique. Ainsi, comment maintenir la diversité de façon à garder dans la population suffisament de solutions pour pouvoir approximer au mieux l'ensemble Pareto ?

En ce qui concerne la sélection, nous avons vu dans le paragraphe précédent différentes possiblités. Il s'avère qu'actuellement, la majorité des algorithmes évolutionnaires multi-objectifs utilisent un "ranking" faisant intervenir la notion d'optimalité de Pareto. Cette notion de ranking, initialement proposée par Goldberg [31], propose de trier la population d'un algorithme évolutionnaire de telle façon que toutes les solutions non dominées soient de meilleur rang. Cette notion de ranking peut ensuite être implémentée de différentes façons.
Concernant le maintien de la diversité, également beaucoup d'études ont été proposées. Les différentes approches comprennent le "sharing" ou "niching" dans l'espace objectif, le clustering, des notions de répartition géographique, l'utilisation de l'entropie... De même différents auteurs ont également choisi de limiter la reproduction entre individus différents, permettant ainsi de générer des enfants variés. Récemment a émergé la notion de relaxation de la notion de dominance dans le but d'encourager plus d'exploration et ainsi d'apporter de la diversité. En particulier la notion de ϵ-dominance devient de plus en plus populaire [55].

Evolution des algorithmes évolutionnaires multi-objectifs (Multi-Objective Evolutionnary Algorithms - MOEAs) : Suite au premier travail recensé dans le domaine, consistant en la proposition de l'algorithme VEGA (Vector Evaluated Genetic Algorithm) par Shaffer en 1984-85 [73], la proposition de Goldberg consistant à utiliser la notion d'optimalité de Pareto dans la sélection a été suivie [31]. Ceci a donné naissance à différents algorithmes évolutionnaires multi-objectifs. Nous pouvons citer MOGA [25], NSGA [76] ou encore NPGA [39]. Dans ces algorithmes, la qualité d'une solution est évaluée en fonction

de sa dominance au sein de la population et la diversité maintenue à l'aide de statégie de "niching". Puis, dans le but d'assurer la convergence vers le front Pareto, la question de préservation de l'élite est devenue fondamentale. Ainsi de nouveaux algorithmes, pouvant faire intervenir ou non des archives de solutions non dominées ont été proposés. Nous pouvons citer parmi ces algorithmes élitistes SPEA [91], PAES [52] ou encore NSGA-II [17]. Ainsi une classification des Algorithmes Evolutionnaires multi-objectifs est souvent utilisée. Cette classification distingue les algorithmes non élitistes, n'ayant aucun opérateur de préservation de l'élite, des algorithmes élitistes prévoyant un opérateur préservant l'élite des solutions.

Autre classification : Il existe différentes manières de classer les algorithmes évolutionnaires pour l'optimisation multi-objectif (MOEAs). Ils peuvent l'être en fonction des opérateurs mis en œuvre (c'est ce que nous avons exposé juste avant) ou bien en fonction de la fonction objectif utilisée et du mécanisme de sélection associé. C'est la classification qui était proposée dans le chapitre introductif. Dans cette classification, communément utilisée, trois catégories peuvent être identifiées :
– Les algorithmes travaillant sur des agrégations de critères,
– les algorithmes non Pareto,
– les algorithmes utilisant la notion de dominance Pareto.
Ainsi, les algorithmes évolutionnaires dédiés à l'optimisation combinatoire multi-objectif ont bien évolué ces dernières années. Cependant ces méthodes ne restent que des approches heuristiques et si sur des applications réelles elles semblent être performantes, l'objectif est toujours de chercher à améliorer les résultats. Dans cet objectif, une approche prometteuse concerne la coopération de différentes méthodes et en particulier la coopération entre méthodes exactes et méthodes heuristiques.

3.2 Coopération de méthodes

Ainsi, nous avons vu et illustré avec l'exemple du flowshop bi-objectif que les méthodes exactes permettaient de résoudre des petits problèmes tandis que les (méta)heuristiques sont capables d'appréhender de grands problèmes sans pouvoir donner la solution optimale (ou prouver que la solution fournie est optimale). Notre objectif ici est de combiner ces différents types de méthodes afin d'obtenir toujours de meilleurs résultats. Ce travail a été réalisé par une collaboration entre deux doctorants de l'équipe : Matthieu Basseur (partie métaheuristique) et Julien Lemesre (partie méthode exacte).

3.2.1 Etat de l'art sur la coopération de méthodes

L'idée de faire coopérer différents types de méthodes n'est pas nouvelle. Très vite, il est apparu que toutes les méthodes n'avaient pas les même propriétés et on a cherché à profiter des avantages des différentes méthodes. Pourtant l'essentiel des études de coopération ont jusqu'alors porté sur la collaboration entre méthodes heuristiques. Dans son état de l'art sur la coopération des métaheuristiques réalisé en 2002, E-G. Talbi propose une taxonomie des

méthodes coopératives, en fonction du schéma de coopération utilisé.

Nous ne nous intéressons pas ici aux méthodes hybrides basées exclusivement sur des métaheuristiques, mais nous nous focalisons sur deux aspects : 1/ les méthodes faisant coopérer métaheuristiques et méthodes exactes et 2/ les méthodes de coopération pour l'optimisation multi-objectif. Ceci a fait l'objet du mémoire de stage de DEA de Vanessa Chantreau [11]. Nous ne présentons pas ici un état de l'art exhaustif, mais cherchons à ressortir les principales tendances.

Coopération entre méthodes exactes et métaheuristiques

Les algorithmes évolutionnaires et les méthodes de type séparation et évaluation (B & B) ont souvent été couplés. Différentes études mono-objectifs ont utilisé ce type de coopération. Par exemple, Cotta et al. ont proposé en 1995 différents schémas de coopération entre des méthodes de type B & B et un algorithme génétique pour des problèmes de type Voyageur de Commerce (TSP) [15]. Pour le même problème, Jahuira et al. ont proposé d'utiliser différentes méthodes exactes (B & B, arbre couvrant, ...) en tant qu'opérateur de croisement d'un Algorithme génétique [40, 41].
Cependant les schémas de coopération entre méthodes exactes et méthodes heuristiques sont rarement très originaux et de belles perspectives de recherches portent sur ce domaine. De plus, ces schémas ont été majoritairement utilisés en optimisation mono-objectif.

Coopération en optimisation multi-objectif

En optimisation multi-objectif, la coopération est plus récente et concerne essentiellement la coopération entre méthodes heuristiques.
Ainsi, Ishibushi et Yoshida proposent pour le flowshop de faire coopérer des algorithmes évolutionnaires multi-objectifs (SPEA, NSGA II) en utilisant de la recherche locale en tant qu'opérateur de mutation.
Il existe cependant des études en multi-objectif faisant coopérer méthodes exactes et métaheuristiques. Ainsi, dans sa thèse, Nicolas Jozefowiez propose une méthode coopérative pour la résolution d'un problème de tournées (le problème de la tournée couvrante) bi-objectif [47]. Pour cela un algorithme génétique propose une approximation puis, un algorithme de séparation et coupes (Branch and Cut) est utilisé pour résoudre optimalement des sous-problèmes suivant l'un des objectifs.

3.2.2 Proposition de schémas coopératifs pour le Flowshop bi-objectif

Dans notre étude, nous avons cherché à faire coopérer une méthode exacte bi-objectif (la méthode deux-phases - TPM) avec une métaheuristique développée pour le problème de

flow-shop bi-objectif. Cette métaheuristique met en œuvre des opérateurs adaptatifs lui permettant d'alterner la recherche entre l'utilisation d'un algorithme génétique et l'utilisation d'un algorithme mimétique [7].

Différents schémas de coopération ont été proposés. Ces différents schémas ont permis soit de confirmer l'optimalité de certains fronts obtenus par la métaheuristique, soit d'améliorer ces fronts. Ce travail fait partie de la thèse de Matthieu Basseur et a donné lieu à une publication à WEA'04 (Workshop on Efficient and Experimental Algorithms) [5].

Recherche de fronts exacts

Une première idée de coopération consistait à utiliser les solutions obtenues par la métaheuristique comme solutions initiales pour la méthode deux phases. Dans ce cas, la métaheuristique est d'abord exécutée. Puis la méthode deux-phases est lancée et chaque fois qu'une recherche de type Branch-and-bound est lancée (pour rechercher les extêmes, les solutions supportées,...) les solutions obtenues par la métaheuristique sont utilisées en tant que borne. Ainsi, l'objectif est de pouvoir couper rapidemment un certain nombre de nœuds et pouvoir exécuter la méthode deux phases sur de plus grandes instances. Cette approche a permi de prouver l'optimalité de certains fronts obtenus par la métaheuristique.
Malheureusement cette approche, bien que permettant de réduire de façon importante le temps nécessaire à la résolution de certains problèmes, ne permet pas de résoudre des problèmes de plus grande taille. En effet, même en donnant à TPM le front optimal, le temps nécessaire à la vérification de l'optimalité du front reste important. Ceci est expliqué par la difficulté des problèmes mono-objectifs associés au problème du flow-shop de permutation utilisé en exemple, et la simple recherche des solutions extrêmes (comme nous l'avons vu avant) peut demander un temps important.

Utilisation d'une méthode exacte pour approximer le front

Puisque la méthode exacte TPM ne peut être utilisée sur l'ensemble du problème dès que celui-ci atteint une taille moyenne, la deuxième approche consiste à utiliser TPM sur des sous-problèmes qui peuvent être déterminés au fur et à mesure de l'exécution de la métaheuristique. Pour cela deux approches ont été proposées. La première utilise TPM comme un opérateur de voisinage large, la deuxième comme une optimisation locale.

TPM - Un algorithme de recherche par voisinage large : Explorer le voisinage d'une solution peut permettre de trouver des solutions proches (en terme de caractéristiques des solutions) de meilleure qualité. Au plus le voisinage est large, au plus la chance d'améliorer la qualité d'une solution augmente. Ainsi il serait tentant d'explorer des voisinages de taille exponentielle. Il faut cependant que l'exploration ne demande qu'un temps raisonable et on préfèrera s'intéresser aux voisinages larges. Pour définir le voisinage, il est nécessaire de caractériser l'opérateur de voisinage (comment se définit un voisin direct, un voisin d'ordre 2...). Dans le cas du Flow-shop, un opérateur classique de voisinage est l'opérateur d'insertion.

C'est donc celui qui a été utilisé pour définir le voisinage d'une solution. Ainsi, la distance entre deux solutions dépendra du nombre minimum d'applications de l'opérateur d'insertion pour passer d'une solution à une autre.
L'idée ici est donc d'appliquer TPM à partir d'une solution de référence (faisant partie du front obtenu par la métaheuristique) en réduisant l'espace de recherche puisque seules les solutions appartenant au voisinage de la solution référence seront énumérées.
En appliquant cette procédure à partir de chaque solution de front obtenu par la métaheuristique, une amélioration du front est obtenue. Bien sûr, étant donné que TPM n'est pas exécutée dans son ensemble, seules des approximations de front sont trouvées.

TPM - Une méthode d'optimisation locale : L'idée principale consiste à n'explorer à l'aide de TPM qu'une petite partie de l'espace décisionnel. Pour cela, à partir d'une solution de départ trouvée à l'aide de la métaheuristique, TPM va fixer certaines caractéristiques de la solution et explorer les possiblités offertes par ce qui n'a pas été fixé. Les solutions non dominées produites par cette recherche sont sauvegardées.
Ainsi, dans le cas de l'application au flowshop, TPM va fixer la position de certains jobs dans l'ordonnancement et explorer l'espace de recherche associé aux différentes possibilités de placements des jobs non fixés.
Ceci est expliqué dans la figure 3.2 sur laquelle la solution initiale $a{-}b{-}c{-}d{-}e{-}f{-}g{-}h{-}i{-}j$ est optimisée. Pour cela deux points de coupure sont définis, permettant de délimiter la partition qui va être étudiée. Les jobs situés à l'extérieur de cette partition garde leur place. Pour les autres toutes les possibilités sont énumérées (énumération non exhaustive à l'aide de TPM).

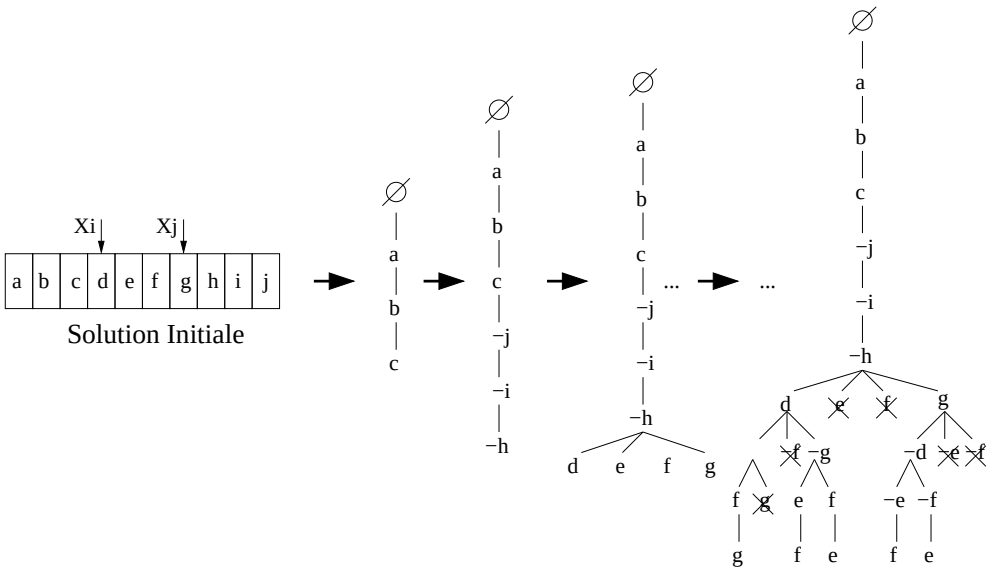


FIG. 3.2 – Exploration d'une partition.

Appliquer cette optimisation demande de s'intéresser à deux paramètres fondamentaux qui

sont :
– la taille des partitions : des partitions trop grandes rendent leur exploration trop coûteuse en temps. Des partitions trop petites ne permettent pas d'améliorer de façon intéressante les solutions. Pour le problème du flowshop des partitions contenant 12 (problèmes à 20 machines) ou 15 (problèmes à 10 machines) jobs sont utilisées.
– le nombre de partitions : de même avoir trop de partitions ralentit le calcul. Pourtant il faut tout de même recouvrir l'ensemble de la solution par différentes partitions de façon à permettre à un job qui aurait été ordonnancé au début de pouvoir être reculé si cela va dans le sens de l'optimisation.

Résultats

Comme nous l'avons dit précédemment, la coopération en vue d'obtenir un front exact n'a pas permis d'augmenter la taille des problèmes résolus optimalement. Nous allons donc nous intéresser aux améliorations de fronts apportées dans le cadre de recherches d'approximations de fronts à l'aide de la recherche par voisinage large et la recherche par optimisation locale. Il apparaît que ces deux méthodes donnent des résultats similaires sur les problèmes à 50 machines. Malheureusement le temps d'exécution de la méthode par voisinage large, telle que proposée ici est exponentiel et ne permet pas de traiter de très grands problèmes avec cette approche. Aussi, nous nous concentrons ici sur les résultats obtenus par la dernière approche, c'est à dire en utilisant TPM comme une méthode de recherche locale.

Au chapitre 1, nous énoncions les problèmes liés à la comparaison de fronts de solutions obtenus par différentes méthodes. Les indicateurs choisis ici pour la comparaison sont *la contribution* et *la S-métrique*.

Instances	Contribution		S-métrique	
	Moyenne	Déviation standard	Moyenne	Déviation standard
50×10 (1)	0.594	0.026	0.185%	0.122%
50×20 (1)	0.525	0.015	0.093%	0.095%
100×10 (1)	0.986	0.015	1.199%	0.387%
100×20 (1)	0.876	0.062	0.970%	0.412%
200×10 (1)	1.000	0.000	13.094%	1.974%

TAB. 3.1 – Apport de l'utilisation de TPM en temps que recherche locale.

Ce tableau montre que pour les problèmes de taille moyenne, l'amélioration obtenue n'est pas très importante. Ceci est expliqué par le fait que la métaheuristique utilisée pour trouver le front initial parvient pour ces tailles de problèmes à trouver de très bons fronts. Par contre, pour les problèmes de grandes tailles l'approche coopérative parvient à améliorer les fronts obtenus par l'éxécution de la métaheuristique seule. Ceci peut être visualisé sur les figures

3.3 et 3.4 sur lesquelles $PAGMA$ représente la métaheuristique initiale et $SSPOBB$ son hybridation avec TPM pour la recherche locale.

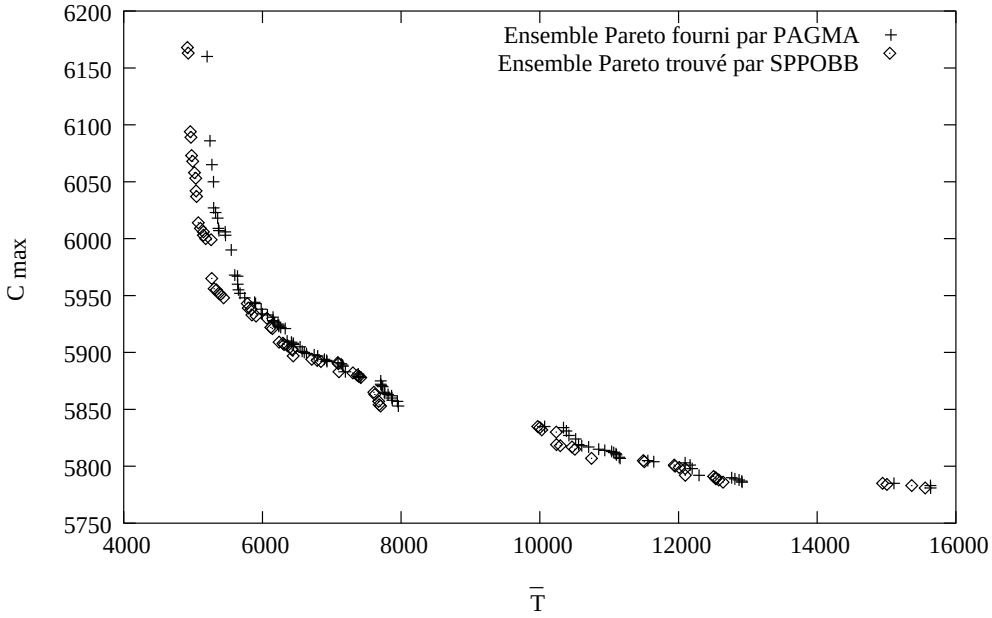


FIG. 3.3 – Résultat de la coopération - Instance 100×10 (1).

3.3 Conclusions et perspectives

Comme pour tout problème d'optimisation combinatoire (mono-objectif), la résolution d'un problème d'optimisation combinatoire multi-objectif peut passer par différents types d'approches.

Parmi ces approches, les méthodes exactes sont toujours intéressantes car elles permettent d'obtenir la solution optimale au problème. Le problème est que ces méthodes sont très souvent gourmandes en temps et elles ne sont pas utilisables pour des problèmes difficiles de grande taille. Ceci est d'autant plus vrai en multi-objectif où la présence de plusieurs objectifs augmente en général la difficulté du problème. Ainsi nous avons vu au chapitre précédent les limites des méthodes exactes lorsque : 1/ les problèmes mono-objectifs sous-jacents sont eux-mêmes difficiles, 2/ lorsque le nombre d'objectifs à optimiser augmente.

Ainsi, les approches heuristiques et en particulier les métaheuristiques à base de populations de solutions sont bien adaptées à la résolution de problèmes multi-objectifs. Leur inconvénient concerne leur performance garantie. En effet, il est très difficile de pouvoir évaluer la qualité d'une méthode, puisque les objectifs de qualité eux-mêmes ne sont pas clairement définis. Quant à pouvoir affirmer l'erreur à l'optimum de telles méthodes, nous en sommes

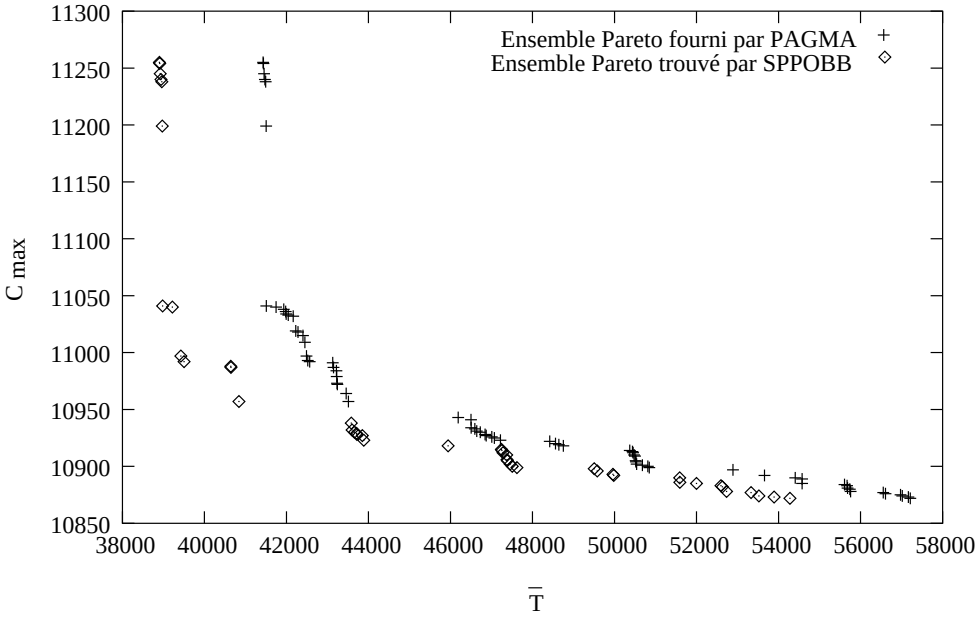


FIG. 3.4 – Résultat de la coopération - Instance 200×10 (1).

loin.

C'est dans ce cadre, que la coopération de méthodes peut s'avérer intéressante. Tout d'abord parce que cela permet en général d'améliorer dans certains cas les fronts obtenus, mais aussi parce que cela permet de garantir l'optimalité de certains fronts.

Tout cela constituait donc les motivations des travaux présentés dans ce chapitre.
Ces travaux ayant porté leurs fruits, des perspectives intéressantes sont maintenant à étudier :

– **Amélioration des méthodes exactes -** Les méthodes exactes ont montré leurs limites. Les perspectives concernent essentiellement trois points : 1/ l'étude de problèmes bi-objectifs spécifiques et l'amélioration d'une méthode en fonction du problème étudié, 2/ la proposition de méthodes multi-objectifs pour plus de deux objectifs et 3/ l'utilisation de façon plus intensive du parallélisme.

– **Proposition de nouveaux schémas de coopération -** Comme il a été dit avant les schémas de coopération entre méthodes au niveau du multi-objectif sont rarement originaux. Dans ce qui existe et ce que nous avons étudié, une première méthode (souvent une métaheuristique) est utilisée pour obtenir une première approximation du front qui est ensuite affinée par une méthode d'optimisation locale (basée ou non sur une méthode exacte). Il serait intéressant d'enrichir ces schémas en faisant coopérer les méthodes pendant leur phase de résolution. Plusieurs voies sont possibles. Par exemple une méthode peut être vue comme un opérateur de l'autre méthode. Ou bien les deux méthodes parti-

cipent conjointement à la construction des solutions. De même la coopération de méthodes ne doit pas forcément se restreindre à l'utilisation de deux méthodes mais peut faire intervenir un plus grand nombre de méthodes, autour d'une archive centralisée, par exemple.

– **Analyse de performances -** Les opérateurs d'analyse de performances proposés à ce jour ne satisfont pas tout le monde. En effet certains privilégient la mesure de diversité, d'autres la mesure de performance... L'analyse de performances d'algorithmes multi-objectifs est donc elle-même multi-objectif. Pour aider à s'y retrouver, un logiciel - GUI-MOO[1] - intégrant une bonne partie des indicateurs classiques et proposant des visualisations de fronts a été développé dans l'équipe. Ceci représente une avancée dans le domaine. Pourtant dans le cadre d'utilisation de méthodes approchées des perspectives persistent autour d'éventuelles analyses de performances garanties de certaines méthodes.

[1]Graphical User Interface for Multi Objective Optimization - www.lifl.fr/OPAC.

Deuxième partie

Méthodes coopératives pour des problèmes d'optimisation multi-objectif en extraction de connaissances

Chapitre 4

Optimisation combinatoire multi-objectif et extraction de connaissances

L'Extraction de connaissances est communément décomposée en quatre phases :
– l'acquisition et le stockage des données (Data warehousing),
– le pré-traitement des données (Pre-processing),
– la fouille de données (Data Mining),
– le post-traitement (Post-processing).
Chacune de ces phases mérite toute l'attention lors de la mise en place d'un processus d'extraction de connaissances. En ce qui nous concerne, nous nous intéressons à la troisième étape, à savoir l'étape de fouille de données qui cherche à décrire le comportement actuel et/ou futur d'un procédé. Pour cela nous modélisons ces problèmes de fouille de données en des problèmes d'optimisation combinatoire.

4.1 Optimisation combinatoire et extraction de connaissances

L'étape de fouille de données consiste donc, de façon très schématique, à décrire des relations liant des informations contenues dans une base de données. Dans la suite du mémoire, nous simplifierons la présentation en faisant l'hypothèse que les données sont représentées sous forme de fichiers plats dans lesquels nous appelerons *instances* les objets décrits dans la base de données (les lignes) et *attributs* les caractéristiques de ces instances (les colonnes).
L'étape de fouille de données peut se décomposer en quatre tâches principales :
– la sélection d'attributs qui consiste à réduire le nombre de facteurs (attributs) décrivant les instances,
– la classification supervisée (discrimination),
– la segmentation (classification non supervisée, clustering) qui consiste à trouver des groupes homogènes au sein d'une population,

– la recherche de règles d'association qui consiste à découvrir des règles mêlant une condition et une prédiction.

Pour chacune de ces tâches, différentes approches issues des statistiques, de l'analyse de données, de l'apprentissage ou encore de l'optimisation combinatoire, ont été proposées. Dans notre étude, nous cherchons à exprimer ces problèmes sous forme de problèmes d'optimisation combinatoire, où un ensemble de combinaisons sont "réalisables" et une ou plusieurs fonctions d'évaluation de ces combinaisons sont à optimiser.

4.1.1 Modélisation en des problèmes d'optimisation combinatoire

Modéliser un problème d'extraction de connaissances en un problème d'optimisation combinatoire nécessite la définition d'un certain nombre d'éléments. En effet, un problème d'optimisation combinatoire est généralement caractérisé par un ensemble fini de solutions admissibles D et une fonction objectif $f : D \rightarrow \mathcal{R}$ qui associe à chaque solution admissible une valeur représentant de façon générale un coût ou un gain. Résoudre le problème d'optimisation consiste alors à déterminer la (ou les) solution(s) admissible(s) optimisant (minimisant ou maximisant) la fonction f. Afin de modéliser des problèmes d'extraction de connaissances en problèmes d'optimisation combinatoire, il est donc nécessaire de définir l'ensemble des solutions admissibles (l'espace de recherche associé au problème) ainsi que la fonction à optimiser. Ces deux éléments sont très étroitement liés.

L'ensemble des solutions admissibles dépend évidemment de la tâche sous étude. Lors d'un problème de segmentation, par exemple, une solution admissible pourra être le regroupement des différentes instances de la base. Ce regroupement pourra être exprimé de différentes façons : détermination d'un centre pour chaque groupe, liste des instances appartenant à un même groupe... Tandis que lors de la recherche de règles d'association, ce sont des relations entre les attributs de la base qui sont recherchées. Ainsi une solution admissible ne sera pas définie de la même façon dans les deux cas.

En ce qui concerne la définition de la fonction à optimiser, ce point est crucial. En effet, à quoi sert de développer de très bons algorithmes d'optimisation, permettant de rechercher de très bonnes solutions par rapport à un critère, si ce critère n'est pas bien défini (ou ne correspond pas tout à fait à ce que l'on cherche). Ainsi, pour la définition de la fonction à optimiser, de bonnes connaissances du domaine d'application sont souvent nécessaires et un dialogue doit s'instaurer entre informaticiens et spécialistes du domaine d'application.

Une fois l'ensemble des solutions admissibles et la fonction d'optimisation définis, il est possible de considérer le problème d'extraction de connaissances en un problème d'optimisation. Malheureusement la plupart des problèmes d'optimisation obtenus sont $\mathcal{NP}$-Difficiles, et il n'existe alors pas de méthode polynomiale pour les résoudre.
Aussi, différentes approches issues de la recherche opérationnelle et de l'optimisation, aussi diverses que des méthodes basées sur des graphes, ou sur de la programmation mathématique, des méthodes énumératives ou des méthodes heuristiques ont été utilisées pour résoudre des

problèmes d'extraction de connaissances. Quelques-unes de ces méthodes ont été référencées dans l'article de synthèse qui accompagnait la session plénière invitée qu'il m'a été demandé de faire à ROADEF 2005 [20]. Nous nous focalisons ici sur les méthodes exactes (énumératives) et les métaheuristiques, puisque ce sont les méthodes que nous utiliserons par la suite.

4.1.2 Méthodes de type énumérative

Les problèmes d'extraction de connaissances concernent, par nature, de grandes bases de données, il est en général très difficile d'énumérer les différentes possibilités. Pourtant une méthode très utilisée peut être classée dans cette catégorie.

En effet, la méthode la plus utilisée pour la recherche de règles d'association de type *IF C THEN P* est l'algorithme *Apriori* qui est basé sur un principe d'énumération [1]. Cet algorithme réalise une énumération efficace de toutes les règles ayant une occurence minimale (support minimal - qui représente la présence conjointe de C et P dans la base) et étant souvent vérifiées (confiance minimale - qui mesure si P est vrai lorsque C est vrai). L'algorithme fonctionne en deux phases. Tout d'abord les ensembles d'attributs fréquents (*frequent itemsets*) sont construits, puis les meilleures règles construites à partir de ces ensembles sont extraites. L'efficacité de l'algorithme se base sur la propriété de monotonie du *support* qui permet de rechercher les ensembles fréquents de taille k qu'à partir des ensembles fréquents de taille $k-1$. Ainsi, cette méthode, bien qu'énumérative, permet de rechercher des règles dans de grandes bases de données.
Un regret concernant cette méthode, est qu'elle n'est pas transposable à d'autres critères que le support, car les autres critères permettant de mesurer la qualité des règles ne sont pas monotones, ce qui a pour conséquence que la valeur du critère pour un ensemble peut être meilleure que pour l'un de ses sous-ensembles, ce qui n'est pas le cas du *support*. De même le développement de méthodes de types séparation et évaluation (*branch and bound*) et de façon générale *branch and X* est compromis par ce manque de monotonie qui ne permet pas de couper les branches de l'arbre d'énumération sans risquer de perdre des solutions de meilleures qualités.

Ainsi, les méthodes énumératives ne pouvant pas exploiter d'intéressantes propriétés de dominance, elles sont souvent mises à défaut devant des problèmes de grandes tailles, en particulier les problèmes ayant un grand nombre d'attributs, ce qui génère de très grands espaces de recherche. Cela explique qu'il est souvent nécessaire de faire appel à la résolution heuristique de ces problèmes afin d'obtenir un compromis entre la qualité des solutions obtenues et le temps de recherche de ces solutions. Les métaheuristiques, en particulier, permettent d'appréhender de tels problèmes.

4.1.3 Résolution par métaheuristiques

Qu'il s'agisse de métaheuristiques à solution unique telles que la méthode de descente, le recuit simulé ou la recherche tabou, ou de métaheuristiques à population de solutions, il existe déjà quelques travaux intéressants concernant leur application aux problèmes d'extraction de connaissances. Dans son mémoire de thèse[1], Laetitia Jourdan fait une revue des différentes métaheuristiques appliquées essentiellement à des problèmes de sélection d'attributs, de segmentation et de règles d'association [42]. De plus, différentes applications issues de la bio-informatique ont été étudiées dans le cadre de cette thèse. En particulier, pour la recherche de facteurs de prédisposition au diabète, une étude sur le déséquilibre de liaison des gènes a été modélisée en un problème de recherche de règles d'association par Jourdan et al. et traité par un algorithme génétique [45]. Différents opérateurs spécifiques pour les règles d'association sont proposés et leurs probabilités d'application sont modifiées au cours de la recherche en fonction de leurs performances (algorithmes adaptatifs).

Ainsi, les algorithmes évolutionnaires, de par leur capacité d'adaptation aux problèmes de différentes natures, ont déjà été bien utilisés pour l'extraction de connaissances.
Dans son état de l'art [28] et son livre [27], Freitas se focalise sur l'utilisation de tels algorithmes pour la recherche de règles d'association (IF-THEN rules). Dans ce contexte, un individu correspond à une règle ou à un ensemble de règles. La fonction objectif permet de mesurer la qualité de la règle et les opérateurs adaptés au problème de recherche de règles permettent de transformer une règle ou un couple de règles en d'autres règles.

Mais l'intérêt de l'utilisation de telles méthodes pour l'extraction de connaissances se voit surtout à travers les différentes applications qui ont été étudiées à l'aide de ces méthodes.
Ainsi, pour la classification automatique des bouchons de liège (détermination de leur catégorie en fonction de leur qualité), Pech-Gourg et al. comparent différentes métaheuristiques (recuit simulé et algorithmes génétique) [68].
Concernant la recherche d'informations sur internet (problématique du Web mining), Picarougne et al. proposent de modéliser le problème comme un problème d'optimisation où la fonction d'évaluation est définie par la requête de l'utilisateur [69]. Ils développent un algorithme génétique dans lequel la population initiale est créée en utilisant les pages référencées par des moteurs standards et la notion de voisinage des solutions est définie par les liens hypertextes entre les pages web.
Dans un contexte similaire, la classification hiérarchique de pages web, un modèle basé sur les fourmis artificielles est proposé par Azzag et al. [2]. Chaque fourmi porte l'information (la page) à classer et les comportements d'auto-assemblage sont comparés entre deux genres de fourmis.
Les algorithmes d'optimisation en essaim ont également été utilisés pour la classification de données par Monmarché et al. [63]. Dans ce cas aussi, chaque insecte représente une donnée et les déplacements des insectes, mouvements complexes déterminés à partir de règles locales simples, visent à créer des groupes de données homogènes.

[1] Thèse réalisée au sein de l'équipe et que j'ai co-encadrée.

Enfin, la programmation génétique a également été utilisée. En effet, une approche pour la classification consiste à construire de nouveaux attributs, à partir des attributs initiaux, afin de définir des combinaisons ayant de forts pouvoirs prédictifs. Pour cela, Muharram et Smith ont développé un algorithme à base de programmation génétique. Ils montrent alors que les *'classifiers'* classiques obtiennent de meilleurs résultats avec la base de données ainsi augmentée [64, 65].
Les métaheuristiques permettent donc d'appréhender de grandes bases de données. Pourtant, certaines études seraient impossibles sans l'utilisation du parallélisme.

4.1.4 Apports du parallélisme

Le parallélisme apporte deux perspectives intéressantes pour le Data Mining : la distribution des données sur différents sites et l'accès à de performantes ressources de calcul. Ces deux aspects ont leurs problématiques propres, présentées dans [48].

Distribution des données

Pour de très grandes bases de données, il peut être nécessaire de les distribuer sur différents sites. Dans ce cas, soit les données sont homogènes sur chacun des sites (représentent les mêmes informations avec le même codage), soit elles sont hétérogènes (informations différentes). L'objectif du Data Mining Distribué (DMD) est d'offrir des techniques pour découvrir de nouveaux modèles à travers ces données distribuées en minimisant les communications. En effet, si la base de données est de grande taille, il n'est pas réalisable de centraliser toutes les données en un site et l'objectif consiste alors à rechercher un modèle global en ne chargeant qu'une petite fraction des données. En général, des modèles partiels sont recherchés en chacun des sites et une agrégation est ensuite réalisée de façon centrale (en utilisant des données partielles des différents sites).

Algorithmes parallèles

Puisque les bases de données sont en général de grande taille, leur analyse peut nécessiter d'importantes ressources de calcul. Ainsi, le Data Mining Parallèle (DMP) cherche à exploiter des environnements d'exécution parallèles de haute performance afin d'augmenter la taille des problèmes étudiés.
Depuis les arbres binaires de classification [75]jusqu'à la parallélisation de l'algorithme *Apriori* [88], en passant par la parallélisation de l'évaluation coûteuse d'une solution [85], les algorithmes pour lesquels une version parallèle a été proposée sont nombreux [89].

Ainsi, le parallélisme et la distribution offrent de bonnes perspectives qui sont multipliées avec la mise en place des grilles (de données et/ou de calculs).

4.1.5 Apport du multi-objectif

Comme il a été exposé dans la partie modélisation, le critère d'optimisation choisi est très important. Or, comme nous l'a montré l'expérience pour un problème de recherche de facteurs de prédisposition à certaines maladies multi-factorielles [46], définir le critère d'optimisation est une phase très délicate car intrinsèquement plusieurs paramètres entrent en jeu. Dans cette première étude, seule une somme pondérée de différents critères, définie de façon expérimentale a pu être mise en œuvre. Cette approche non satisfaisante montre qu'il est important de se questionner sur l'intérêt de définir un unique critère d'évaluation.
De la même façon, dans une étude sur les règles d'association, la question du choix de ce critère (unique ?) s'est posée. Puisqu'il existe plus d'une vingtaine de critères proposés dans la littérature [79], une étude statistique de ces critères a été réalisée afin de rechercher les corrélations éventuelles entre ces critères [50]. Cette étude a conduit à la proposition d'un modèle multi-objectif pour les règles d'association et la proposition d'un algorithme génétique multi-objectif permettant de trouver non pas une règle optimale, mais un ensemble de règles de meilleur compromis entre les critères [51]. C'est le travail que nous exposerons dans la prochaine partie.
Ainsi le multi-objectif permet d'assouplir la phase de sélection de la fonction à optimiser. Et même si en contre-partie les problèmes à résoudre sont (en général) plus difficiles, l'existence d'un ensemble de solutions de meilleur compromis entre les critères peut satisfaire le commanditaire de l'étude qui obtient alors non pas une seule solution, mais un ensemble de solutions intéressantes.

Dans nos études nous nous sommes focalisés sur la recherche de règles d'association pour l'étude de base de données issues de la génomique. Nous exposons dans un premier temps les études réalisées sur les règles d'association multi-objectifs puis exposons ensuite leur application en bio-informatique.

4.2 Règles d'association multi-objectifs

4.2.1 Motivations

Une tâche importante de l'extraction de connaissances est la recherche de règles d'association. Une règle d'association traduit des relations entre certains items (attributs) d'une base de données. Le premier problème traité a été l'étude du panier de la ménagère (étude des tickets de caisse) où des relations entre les différents achats sont recherchés (ex : SI Pain ET Vin ALORS Boursin, comme disait une certaine publicité.) [1].
Plus formellement, une règle d'association est une implication de la forme *SI C Alors P* (*If C then P*) où C et P sont des conjonctions de termes. C représente la condition de la règle et P la prédiction. Un terme peut représenter la présence/absence d'un attribut (attribut binaire) ou associer un attribut à une valeur (lorsque l'on considère des attributs nominaux ou numériques).
Obtenir de bonnes règles permet de décrire des relations et d'anticiper des comportements.

Une question fondamentale est donc : qu'est-ce qu'une bonne règle ? Comment évaluer la qualité d'une règle ? Plusieurs communautés scientifiques se sont intéressées à cette question. Chacune a essayé de proposer différents indicateurs de mesure et l'on peut maintenant en dénombrer plus d'une vingtaine. Nous avons ici étudié différents critères, afin de trouver des relations (corrélation, ...) entre eux pour, in fine, extraire un ensemble restreint de critères complémentaires, indépendants permettant de mettre en évidence toutes les propriétés des règles et ainsi proposer une modélisation multi-objectif du problème de recherche de règles. Dans le cadre de nos études, nous nous sommes focalisés sur les règles ayant une prédiction composée d'un seul terme. Si ce terme était toujours le même (et correspondant à un attribut à prévoir) nous aurions traité des règles de prédiction. Dans nos études, le terme composant P peut être basé sur n'importe quel attribut de la base.

4.2.2 Etude de critères de mesure de qualité

L'objectif de cette étude est de rechercher les éventuelles relations existantes entre les principales mesures de qualité des règles proposées dans la litérature. Ce travail, réalisé dans le cadre des travaux de thèse de Mohammed Khabzaoui, a donné lieu à une collaboration avec Assi N'Guessan, Maître de conférences en statistiques. Il s'est conclu par une présentation lors des Journées Françaises des Statistiques JDS'03 [50].

Considérons la règle d'association R, qui représente une implication de la forme $C \rightarrow P$ (Si C $Alors$ P) où C est la condition et P la prédiction. La qualité d'une règle peut dépendre de plusieurs caractéristiques (sa force de prédiction, son nombre d'occurrences...) en fonction du contexte.

Nous présentons ici les critères étudiés. Lors d'une première étude, nous en avions sélectionnés onze provenant des statistiques, de la théorie de l'information, du datamining. Pour avoir plus de détails sur ces critères, le lecteur peut se référer à [80].

- Support : c'est la mesure classique des règles d'association. Elle permet de mesurer la fréquence de la règle dans la base de données.
- Confiance : elle mesure la validité de la règle, c'est la probabilité conditionnelle de P sachant C.
- Intérêt et Conviction : l'intérêt mesure la dépendance en privilégiant les motifs rares dont le support est faible. L'intérêt a un comportement symétrique. Afin de pallier à ce problème, un nouvel indice a été proposé : la Conviction.
- Surprise : la surprise est utilisée pour mesurer l'affirmation. Elle permet de chercher les règles étonnantes.
- Jaccard : c'est une mesure de similarité utilisée pour calculer la similarité (ou distance) entre deux mots ou textes.
- Phi-coefficient : c'est une mesure de dépendance dérivée du test de χ^2.
- Cosinus : cette mesure est dérivée de la corrélation statistique. Elle est très intéressante dans la région de faible support et de fort intérêt.
- J-mesure : mesure combinant support et confiance utilisée en Optimisation.

TAB. 4.1 – Critères de qualité étudiés.

Critère	**Formulation mathématique**
Support - $Supp.$	$\frac{\lvert CandP\rvert}{N}$
Confiance - $Conf.$	$\frac{\lvert CandP\rvert}{\lvert C\rvert}$
Intérêt - $Int.$	$\frac{N*\lvert CandP\rvert}{\lvert C\rvert*\lvert P\rvert}$
Conviction - $Conv.$	$\frac{\lvert C\rvert*\lvert\overline{P}\rvert}{N*\lvert Cand\overline{P}\rvert}$
Surprise - $Surp.$	$\frac{\lvert CandP\rvert-\lvert Cand\overline{P}\rvert}{\lvert\overline{P}\rvert}$
Jaccard - $Jacc.$	$\frac{\lvert CandP\rvert}{\lvert C\rvert+\lvert P\rvert-\lvert CandP\rvert}$
Phi-coefficient - $\phi.$	$\frac{(\lvert CandP\rvert*\lvert\overline{C}and\overline{P}\rvert-\lvert Cand\overline{P}\rvert*\lvert\overline{C}andP\rvert)^2}{\lvert C\rvert*\lvert P\rvert*\lvert\overline{C}\rvert*\lvert\overline{P}\rvert}$
Cosinus - $Cos.$	$\frac{\lvert CandP\rvert}{\sqrt{\lvert C\rvert*\lvert P\rvert}}$
J-mesure - $JMe.$	$\frac{\lvert P\rvert}{N} * [\frac{\lvert CandP\rvert}{\lvert P\rvert}\log(\frac{N*\lvert CandP\rvert}{\lvert C\rvert*\lvert P\rvert}) +(1-\frac{\lvert CandP\rvert}{\lvert P\rvert})\log(\frac{1-\frac{\lvert CandP\rvert}{\lvert P\rvert}}{1-\frac{\lvert C\rvert}{N}})]$
Piatetsky-Shapiro - $PS.$	$\frac{\lvert CandP\rvert}{N}-\frac{\lvert C\rvert}{N}*\frac{\lvert P\rvert}{N}$
Laplace - $Lapl.$	$\frac{\lvert CandP\rvert+1}{\lvert C\rvert+2}$

– Piatetsky-Shapiro : autre mesure de dépendance utilisée en datamining.
– Laplace : cette mesure est très proche de la confiance par sa définition.

4.2.3 Analyses statistiques

Afin d'étudier les relations entre critères, nous avons énuméré pour un problème donné (le problème classique *Nursery* de *l'UCI Data Repository*[2] - site sur lequel un grand nombre de bases de données classiques pour l'extraction de connaissances sont à disposition), toutes les règles pouvant exister (énumération exhaustive). Nous avons alors mesuré chacune de ces règles suivant l'ensemble des critères étudiés ci-dessus. Nous avons ainsi généré un tableau de 2002 lignes et de 11 colonnes dans lequel une ligne représente une règle d'association et chaque colonne la qualité de la règle par rapport à l'un des critères. Nous avons soumi ce premier tableau de données à l'analyse en composantes principales (ACP) normée disponible sous le logiciel SPAD 5.5 [18, 56]. Des corrélations fortes entre critères (mesures) revient donc à trouver des corrélations entre les colonnes de la matrice. Ainsi il est possible de mettre en évidence, des critères ayant des comportements similaires pour l'ensemble des règles. L'ensembles de ces comportements est résumé par le tableau 4.2 qui représente la matrice des corrélations linéaires entre les onze critères étudiés.

[2]ftp ://ftp.ics.uci.edu/pub/machine-learning-databases.

	Supp.	Conf.	Int.	Conv.	Surp.	Jacc.	ϕ.	Cos.	JMe.	PS.	Lapl.
Supp.	1,00										
Conf.	0,62	1,00									
Int.	-0,09	0,20	1,00								
Conv.	0,27	0,56	0,47	1,00							
Surp.	0,17	0,48	0,07	0,17	1,00						
Jacc.	**0,87**	0,62	0,32	0,55	0,20	1,00					
ϕ.	0,38	0,50	0,62	**0,81**	0,26	0,76	1,00				
Cos.	**0,86**	0,68	0,34	0,56	0,19	**0,98**	0,76	1,00			
JMe.	0,34	0,50	0,40	**0,84**	0,15	0,64	**0,89**	0,62	1,00		
PS.	0,29	0,49	0,25	0,71	0,15	0,51	0,75	0,51	**0,93**	1,00	
Lapl.	0,63	**0,99**	0,18	0,54	0,53	0,61	0,49	0,67	0,50	0,51	1,00

TAB. 4.2 – Matrices des corrélations linéaires.

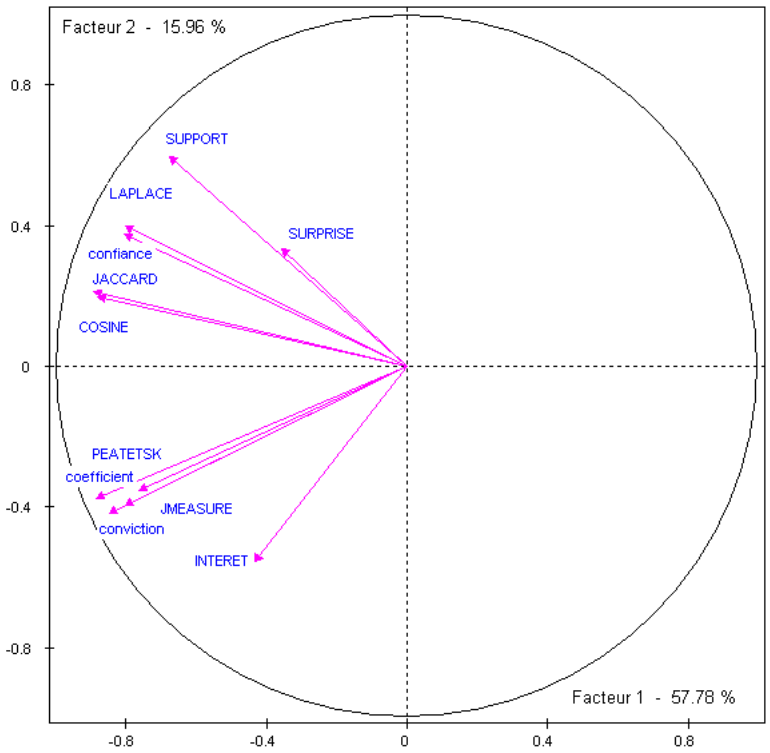


FIG. 4.1 – Cercle des corrélations.

Cette matrice donne une première indication sur la nature des relations linéaires entre ces différents critères par rapport aux 2002 règles. On peut ainsi remarquer de très fortes corrélations entre "Cosinus" et "Jaccard" (0,98), entre "J-mesure" et "Piatestky-Shapiro" (0,93), entre "Confiance" et "Laplace" (0,99). "Support" peut être rattaché au premier groupe via sa corrélation avec "Jaccard". De même "Phi-Coefficient" et "Conviction" peuvent être associés au deuxième groupe. Les critères "Intérêt" et "Surprise" sont dans l'immédiat assez atypiques. Le cercle des corrélations (voir figure 4.1) avec 75,44 % d'inertie, confirme les tendances précitées et met en relief l'importance du premier axe factoriel.

Ce travail a été confirmé dans le cadre d'un projet de statistiques réalisé par des étudiants de $2^{ème}$ année de la filière Génie Informatique et Statistique de Polytech'Lille. En effet, dans le cadre de ce projet, d'autres bases de règles ont été générées sur des problèmes différents. Les études sur ces autres bases montrent que les corrélations entre les critères restent valides.

4.2.4 Conclusions et perspectives de l'étude des critères

Ces analyses nous ont permis de mettre en évidence 5 classes de critères. Une analyse plus complète, étudiant 24 critères a montré un même regroupement en 5 classes (thèse de M. Khabzaoui). Chaque classe rassemble des critères très fortement corrélés, ce qui veut dire des critères qui mesurent les mêmes propriétés. Ces classes sont données dans le tableau 4.3. L'idée est maintenant de choisir un critère représentant chacune des classes afin d'obtenir une modélisation multi-objectif considérant des critères réellement complémentaires.

Tendance 1	Jaccard, Cosinus, Support
Tendance 2	Laplace, Confiance
Tendance 3	Phi-Coefficient, Conviction, J-mesure, Piatetsky-Shapiro
Tendance 4	Intérêt
Tendance 5	Surprise

TAB. 4.3 – Récapitulatif des différentes corrélations.

Ce travail exploratoire doit maintenant être accompagné d'autres études afin de s'assurer que les corrélations identifiées sur l'ensemble de l'espace des solutions sont vérifiées également sur des sous-espaces. En particulier, un aspect important consiste à vérifier la validité de l'analyse sur le front Pareto. En effet, les solutions qui nous intéressent le plus, sont les solutions non dominées (ou les solutions proches) et il est important de vérifier que sur ces solutions les classes de critères restent valides.

Ce travail a permis de proposer une modélisation multi-objectif du problème de recherche de règles. Notons que ce problème peut également être vu comme un problème d'optimisation combinatoire puisque chaque règle consiste en une combinaison de termes.
Ainsi, le problème de recherche de règles d'association ayant été défini comme un problème d'optimisation combinatoire multi-objectif, différentes possibilités s'offrent à nous pour le

résoudre. Nous discutons dans un premier temps d'une résolution par méthode exacte puis nous présentons une approche à l'aide de métaheuristiques.

4.3 Résolution par méthodes exactes

La recherche d'une méthode exacte pour le problème des règles d'association multi-objectif pousse naturellement à chercher à se rapprocher de la méthode *Apriori*, puisqu'elle consiste à énumérer de façon exhaustive toutes les règles satisfaisant certaines conditions sur les critères de support et de confiance. Malheureusement, comme nous l'a montré l'étude menée lors du stage de Frédéric Blondel, l'efficacité de cette méthode est basée sur la propriété de monotonie du support qui permet de n'énumérer qu'un sous-ensemble des ensembles possibles de termes.
Ainsi, dans ce contexte, la seule façon d'obtenir l'ensemble des solutions Pareto est d'utiliser une méthode énumérative qui construit toutes les règles possibles étant donnés nb attributs. Malheureusement, sans l'existence de propriétés intéressantes, cette procédure d'énumération n'est pas très efficace. Lorsqu'il s'agit d'énumérer toutes les règles en autorisant pour chaque attribut d'avoir tour à tour toutes ses valeurs possibles, il est nécessaires de restreindre l'application de la procédure à un petit sous-ensemble d'attributs.
Ainsi, cette procédure ne peut raisonnablement pas être utilisée sur des bases de données de taille intéressante. Nous en gardons tout de même la mémoire et verrons dans le chapitre suivant comment dans le cadre de méthodes coopératives, elle peut être utilisée.

Avant cela, regardons ce que les métaheuristiques peuvent proposer.

4.4 Résolution par métaheuristiques

Afin de pouvoir traiter le problème de recherche de règles dans sa forme multi-objectif nous avons développé un algorithme génétique multi-objectif adapté pour la recherche de règles. Dans le modèle choisi, cinq objectifs (un représentant par groupe) ont été sélectionnés. Il s'agit donc de maximiser le **Support**, la **Confiance**, la **J-mesure**, l'**Intérêt** et la **Surprise**.

Des opérateurs spécifiques pour les règles ainsi que des mécanismes pour le multi-objectif sont implémentés. Cette étude a été initiée, pour le cas mono-objectif, dans la thèse de Laetitia Jourdan [42] et a été ensuite poursuivie dans le cadre de la thèse de Mohammed Khabzaoui, pour la partie multi-objectif. Cet algorithme a été présenté lors de la conférence CEC (Congress on Evolutionary Computation) en 2004 [51]. Les principales caractéristiques sont données ci-dessous. L'algorithme a été développé à l'aide du Framework ParadisEO [3] (*PARallel and DIStributed Evolving Objects*) qui est un extension de la plateforme EO permettant de concevoir des algorithmes évolutionnaires parallèles et distribués. L'usage de

[3]Ce framework est téléchargeable sur www.lifl.fr/OPAC.

telles plateformes permet de développer des algorithmes flexibles pouvant facilement utiliser différents mécanismes fournis par ces plateformes.

4.4.1 Coder les règles

Une règle d'association consiste en une condition et une prédiction qui sont chacune une conjonction de termes. Chaque terme est un attribut associé à une valeur. Dans notre cas, nous nous sommes restreints à des prédictions composées d'un seul terme. Ainsi, le codage proposé consiste en une suite de termes (<attribut, valeur>) et le dernier terme représente la prédiction.
Attention ce n'est pas parce que le terme de prédiction est unique qu'il est toujours le même ! !

4.4.2 Opérateurs pour la recherche de règles

Les opérateurs classiques de croisement et mutation ont été dédiés à la recherche de règles.

Opérateur de croisement

Deux versions sont proposées en fonction des caractéristiques des parents.
- **Croisement par changement de valeur -** Si deux règles X et Y ont un (ou plusieurs) attribut(s) commun(s), la valeur de l'un de ces attributs est échangée entre les parents (voir Figure 4.2).
- **Croisement par insertion -** Inversement, si X et Y n'ont pas d'attribut commun, un terme est choisi aléatoirement dans X et inséré dans Y. L'opération inverse est exécutée pour insérer un terme de Y dans X (voir Figure 4.3).

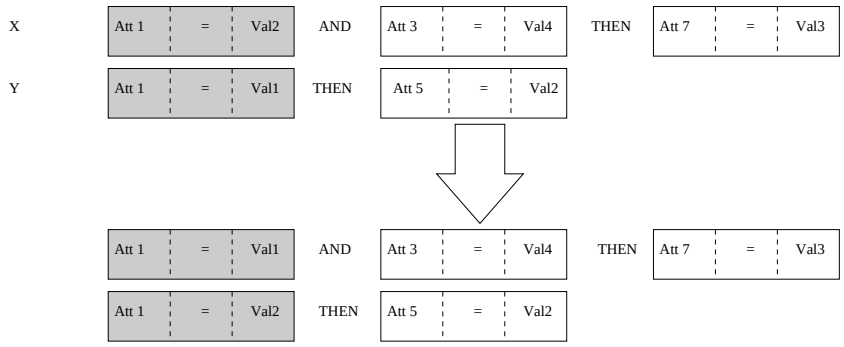


FIG. 4.2 – Croisement par changement de valeur.

Mutation

Quatre opérateurs de mutation ont été mis en œuvre.
- **Mutation par valeur** où un attribut d'une règle voit sa valeur être modifiée.

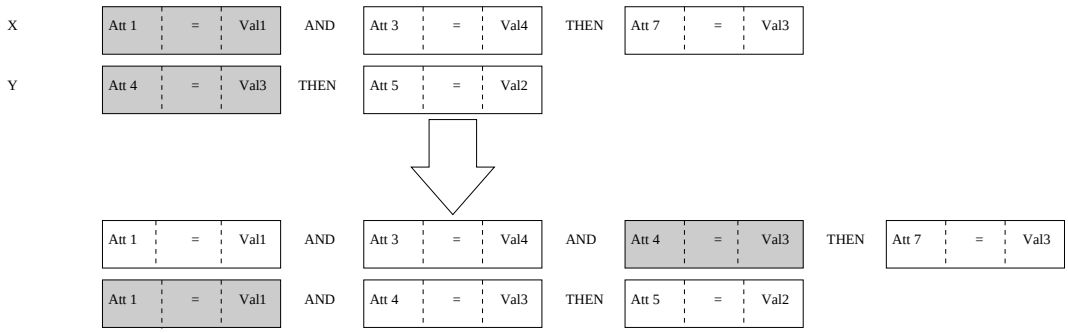


FIG. 4.3 – Croisement par insertion.

- **Mutation par attribut** qui remplace un terme par un autre (dans ce cas, un attribut est sélectionné aléatoirement ainsi que sa valeur).
- **Mutation par insertion** qui ajoute un nouveau terme à une règle (terme généré aléatoirement).
- **Mutation par suppression** qui supprime un terme de la règle (si le nombre de termes est suffisant).

Mutation adaptative - Définir la probabilité d'application des différents opérateurs de mutation n'est pas chose facile. En effet, un opérateur intéressant lors d'une partie de la recherche ne le reste pas forcément tout le temps. Pour cela nous proposons de définir ces probabilités d'application de façon adaptative en tenant compte des performances des opérateurs. Cette stratégie, proposée pour l'optimisation mono-objectif par Hong et al. [38], a été étendue pour le cas multi-objectif. Il a donc été nécessaire de redéfinir le progrès apporté par un opérateur, non plus au regard d'un seul objectif, mais de plusieurs.

4.4.3 Aspects multi-objectifs

Comme nous l'avons vu précédemment, résoudre des problèmes d'optimisation multi-objectifs nécessite d'une part de revoir certains opérateurs de l'algorithme et d'autre part de mettre en œuvre des mécanismes spécifiques permettant de gérer l'existence de plusieurs solutions de meilleur compromis. En particulier, ici la notion de dominance est utilisée pour comparer les solutions entre-elles et la gestion de la population est revue.

Opérateur de sélection - Il consiste ici en une sélection par roulette basée sur la notion de "*Pareto ranking*" définie par Fonseca [26] et qui affecte à chaque solution (chaque règle d'association), un rang correspondant au nombre de solutions qui la dominent.

Opérateur de remplacement - Le remplacement élististe consiste à remplacer les solutions de moins bon rang (*Pareto ranking*) par les nouvelles solutions générées qui les dominent (s'il en existe). La taille de la population est fixe.

Archive Pareto - Les solutions non dominées trouvées au cours de la recherche sont archivées dans une seconde population appelée "Archive Pareto" de façon à ne pas les perdre. Cette archive doit être mise à jour dès qu'une nouvelle solution y est ajoutée, car cette nouvelle solution peut dominer des solutions qui appartenaient à l'archive et donc étaient jusque là non dominées.

Elitisme - Les solutions de l'archive ne sont pas seulement stockées de façon permanente mais participe à la phase de sélection et peuvent donc prendre part à la reproduction.

4.4.4 Premiers résultats

La mise en place d'un tel algorithme a toujours son lot de questions quant aux opérateurs efficaces et aux mécanismes à mettre en œuvre. A travers une première série d'expérimentations nous avons testé, dans le contexte de la recherche des règles, différentes configurations.
Les bases de données utilisées (*BD1* et *YeastBD*) sont relatives à des expérimentations sur puces à ADN. Comme nous l'expliquerons plus en détails dans le chapitre présentant les applications, ces expérimentations permettent de mesurer l'activité de milliers de gènes simultanément et l'objectif consiste à trouver des règles d'association permettant de traduire les relations entre les niveaux d'activité de ces gènes. Dans cette partie, le côté applicatif n'est pas très important, puisque les approches proposées permettent de rechercher des règles d'association dans tout type de bases de données.

Afin de comparer les fronts Pareto obtenus par ces différentes configurations, les mesures de la contribution et de l'entropie, sont utilisées (voir chapitre 2). Rappelons qu'une contribution supérieure ou égale à 0.5 indique une amélioration du front et qu'au plus l'entropie est forte au plus le front est diversifié.

Contribution	Sélection Pareto sans élitisme	Sélection type NSGA sans élitisme
Sélection Pareto avec élitisme	0.64	
Sélection type NSGA avec élitisme		0.75

Tab. 4.4 – Contribution de l'élitisme (sur BD1).

Le tableau 4.4 montre la contribution de l'élitisme pendant la phase de sélection qui permet d'obtenir au final de meilleurs fronts. Notons que ceci est vérifé que ce soit une sélection de type *Pareto Ranking* ou une sélection de type *NSGA* qui est utilisée.
Le tableau 4.5 montre la contribution de la version adaptative (pour deux bases de données d'expression génique). Tout d'abord le nombre de solutions Pareto obtenues est plus important avec la version adaptative qui permet donc une meilleure approximation de l'ensemble du front. De plus, la contribution supérieure à 0.5 indique que les fronts sont meilleurs (puisqu'ils dominent en partie les fronts obtenus sans l'adaptativité). Ainsi, ce processus d'adaptativité, qui nécessite des calculs supplémentaires pour mesurer l'efficacité des opérateurs

	BD1	Yeast BD
Non adaptatif	9 sol.	19 sol.
Adaptatif	12 sol.	31 sol.
Contribution adaptatif/non adaptatif	0.54	0.71

TAB. 4.5 – Comparaison version adaptative vs non adaptative.

tout au long de la recherche, semble indispensable dès que plusieurs opérateurs sont en jeu.

Bien sûr, nous pouvions nous attendre aux résultats présentés ici. L'idée de ces expérimentations était d'une part de vérifier que le problème de règles d'association se comportait comme un problème d'optimisation "normal" et qu'après la phase de modélisation du problème à travers en particulier la définition du codage et des opérateurs, les mécanismes avancés relatifs au multi-objectif pouvaient être mis en œuvre.

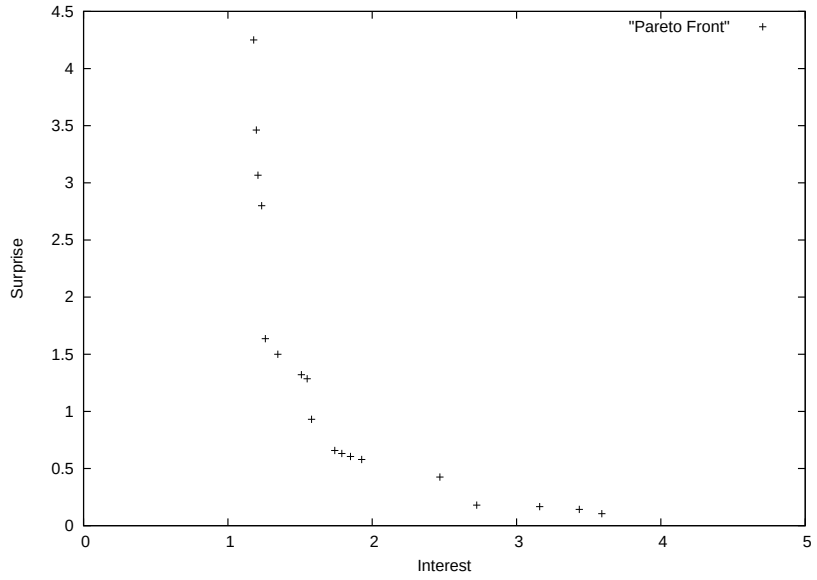


FIG. 4.4 – Front Pareto (Surprise / Intérêt) - YeastDB.

Un autre aspect intéressant à regarder est la structure des fronts Pareto obtenus. Pour cela, les figures 4.4 et 4.5 présentent la projection suivant 2 objectifs d'un front obtenu (il est en effet assez difficile de faire une représentation en cinq dimensions!!).
Deux conclusions peuvent être tirées de ces figures. Tout d'abord, les objectifs choisis sont bien complémentaires, puisque les fronts comportent plusieurs solutions de compromis. Cela valide sur un point l'analyse statistique faite pour la sélection des objectifs. D'autre part, les fronts ne comportent pas un trop grand nombre de solutions, ce qui permet d'offrir au décideur différents choix tout en gardant un nombre raisonnable.

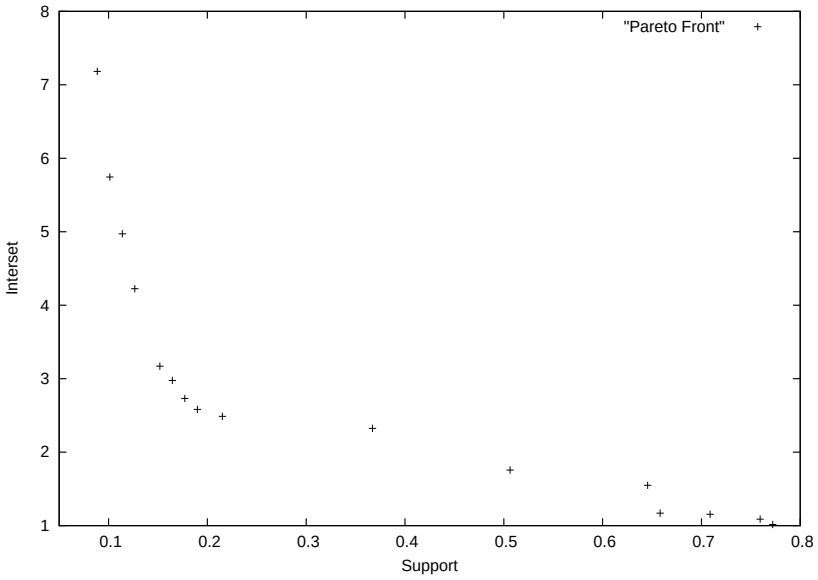


FIG. 4.5 – Front Pareto (Intérêt / Support) - YeastDB.

4.5 ARV : un outil d'aide à la décision

Les résultats fournis par de telles approches Pareto sont composés d'un ensemble de règles. Ces règles ont différentes valeurs pour les différents critères d'évaluation et choisir l'une ou l'autre de ces règles n'est pas chose facile.
Dans le but de faciliter la tâche du choix des règles (qui relève de la théorie de la décision), nous avons pensé réaliser un outil de visualisation de règles multi-objectifs. Ainsi, sous l'encadrement de Laetitia Jourdan, deux étudiants du DESS IAGL de Lille (D. Delautre and S. Demay) ont réalisé le logiciel ARV - *Association Rules Viewer*.

4.5.1 Caractéristiques de ARV

Il existe différentes représentations disponibles pour la visualisation des règles d'association dans la littérature, mais la plupart ne sont pas disponibles librement. L'outil ARV, permet donc, à chacun d'accéder à trois de ces représentations [4] de façon à pouvoir comparer les règles suivant différents critères.

Entrée des données

Utiliser ARV nécessite d'avoir un fichier récapitulant les différentes règles à visualiser et leur valeur pour chacun des critères étudiés. Ces critères ne sont pas nécessairement les critères utilisés dans notre modélisation mais n'importe quel ensemble de critères choisi par

[4] ARV est téléchargeable sur www.lifl.fr/OPAC.

l'utilisateur.
Un format XML a été proposé de façon à rester générique.

Choix des visualisations

Visualisation 3D - Cette visualisation proposée par Wong [86] permet initialement de visualiser un ensemble de règles en représentant les attributs de chaque règle composant la condition (d'une couleur prédéfinie) et la conclusion (d'une autre couleur) et en représentant les valeurs de support et de confiance (voir figure 4.6).
Cette visualisation a été étendue dans ARV de façon à pouvoir visualiser n'importe quel couple de critères.

Visualisation "par lignes" - Cette visualisation simple dans son principe permet de représenter l'ensemble des règles suivant l'ensemble des critères. Chaque ligne représente une règle et chaque critère, une abscisse (voir figure 4.7).
Avec cette représentation, il est aisé de voir si des critères sont corrélés sur l'ensemble de règles étudiées.

Visualisation "Double decker-plot" - Cette visualisation s'intéresse à une seule règle à la fois [83]. Elle représente pour chaque attribut son support (voir figure 4.8).
Cette représentation permet donc de visualiser en détail le rôle joué par chaque attribut composant une règle d'association.

Fonctionalités

Afin de permettre une meilleure analyse de l'ensemble des règles sous étude, différentes fonctionalités sont offertes à l'utilisateur. Celui-ci peut sélectionner un sous-ensemble de règles, les classer suivant leur valeur sur l'un ou l'autre des critères ou encore choisir un sous-ensemble de critères à étudier.
Ainsi l'interface est décomposée en trois parties, une pour chaque visualisation, ce qui permet de proposer des fonctionalités spécifiques par visualisation. Toutes ces fonctionalités sont proposées dans le but de faciliter la phase de choix de la (ou les) règle(s) à étudier en priorité.

Bien sûr des fonctionalités de sauvegarde et d'impression sont offertes.

4.5.2 Exemples de visualisations

Sur un exemple tiré d'une étude en génomique dont nous parlerons au chapitre suivant, voici les visualisations proposées par ARV.

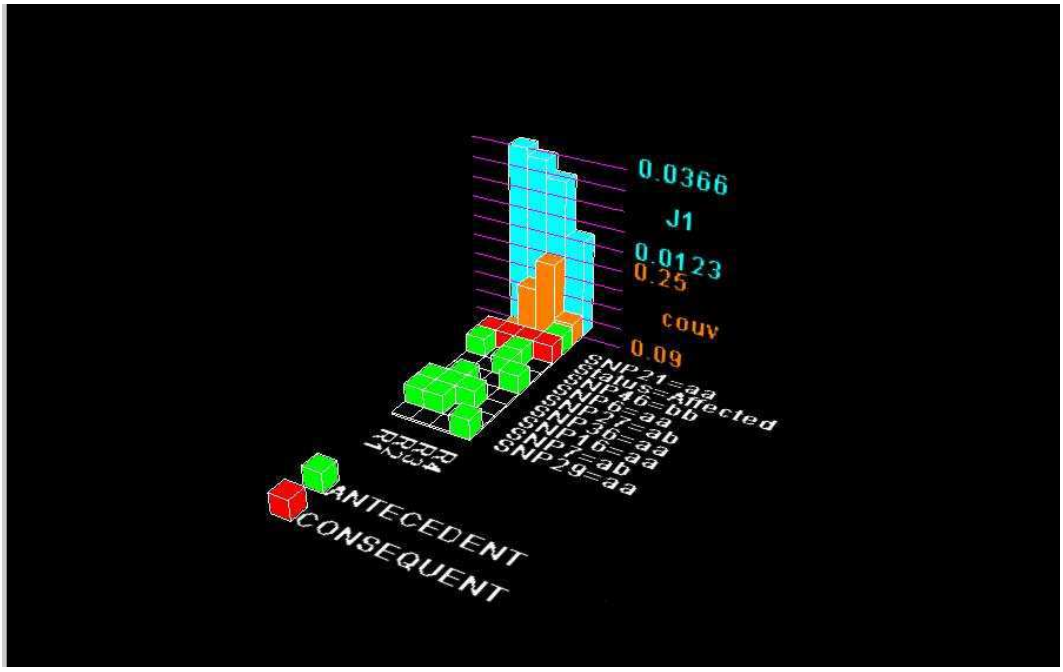


FIG. 4.6 – Visualisation en 3D.

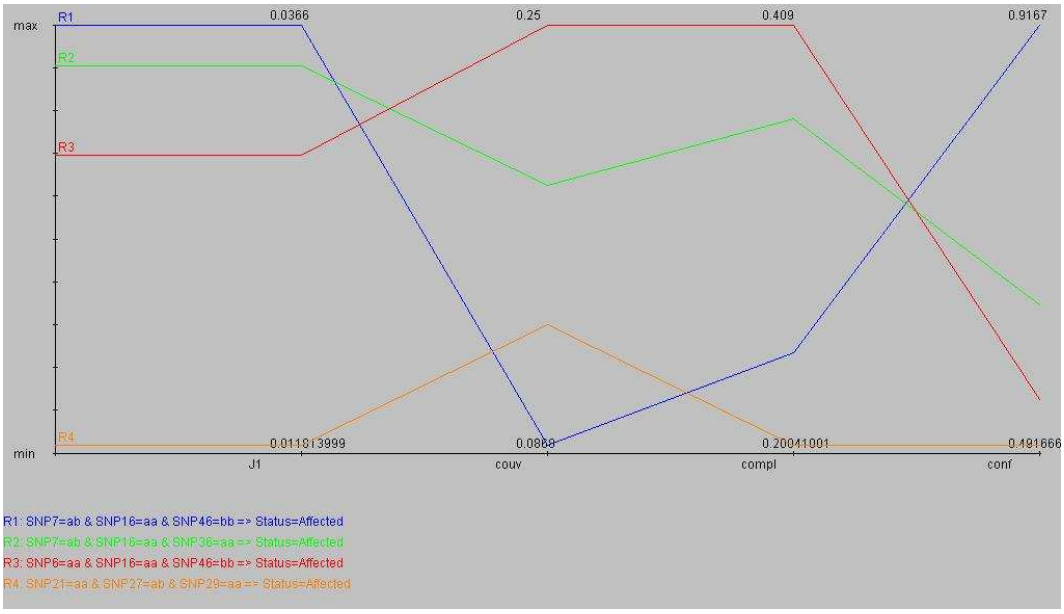


FIG. 4.7 – Visualisation en lignes.

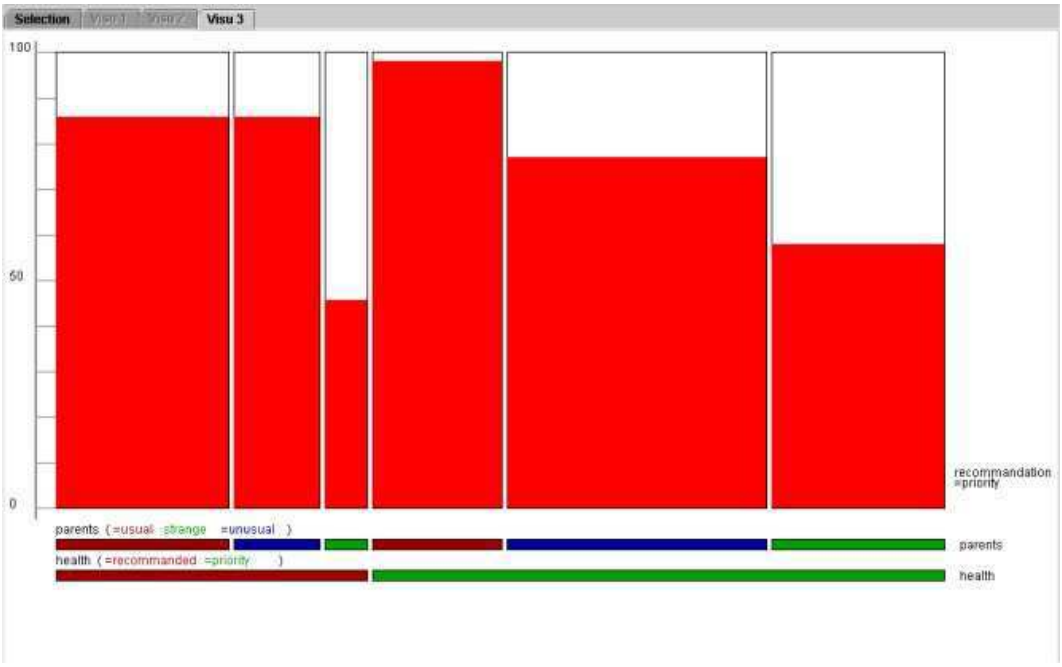


FIG. 4.8 – Double Decker Plot.

4.6 Conclusion et perspectives

Ainsi, il apparaît dans ce chapitre que la recherche opérationnelle et les méthodes d'optimisation peuvent apporter certaines réponses aux problématiques d'extraction de connaissances. Différentes approches ont d'ailleurs déjà été proposées dans ce sens.

Nous cherchions ici à montrer que les aspects multi-objectifs étaient aussi dans ce contexte très importants. Pour cela, nous nous sommes focalisés sur une modélisation multi-objectif de la problématique de recherche de règles d'association.
Mais il apparaît également que les approches multi-objectifs pourraient être intéressantes pour d'autres tâches d'extraction de connaissances. En effet, regardons, par exemple, de plus près la problématique de segmentation (classification non supervisée). L'objectif global de ce type de problème est de bien séparer les données différentes en regroupant les données semblables. Intraséquement, le problème est au moins bi-objectif (maximiser l'inter-dissimilarité des groupes tout en maximisant leur intra-similarité). Pour contre-carrer cette difficulté, la plupart des objectifs utilisés pour la résolution d'un problème de segmentation par méthodes d'optimisation sont en fait souvent des combinaisons de plusieurs objectifs. Il serait donc intéressant de voir ce que peuvent apporter les approches multi-objectifs de type Pareto. C'est d'ailleurs ce qu'on commencé à proposer Handl et Knowles [34].

Concernant les règles d'association, nous avons pu, à travers les expérimentations, valider la modélisation multi-objectif proposée (les objectifs sélectionnés sont intéressants pour me-

surer la qualité des règles et sont bien complémentaires). Nous avons également testé un premier algorithme de résolution mettant en œuvre des opérateurs spécifiques pour les règles ainsi que des mécanismes adaptés au multi-objectif. Cet algorithme a montré une certaine convergence et l'introduction de mécanismes supplémentaires a conduit vers une amélioration des fronts proposés. Pourtant, étant donné le très large espace de recherche, une question se pose quant à l'efficacité absolue de la méthode et il semble intéressant de regarder ce que pourrait apporter une approche coopérative. C'est ce que nous proposerons dans le chapitre suivant.

Une difficulté concernant l'aide à la décision multi-objectif concerne le choix des solutions parmi un ensemble de solutions Pareto (ou potentiellement Pareto). Afin de permettre au décideur de visualiser des propriétés des règles d'association, un outil a été proposé. Loin de répondre à toutes les questions, cet outil est un premier pas vers la rencontre entre l'optimisation multi-objectif et l'aide à la décision.
Pourtant il convient de se poser la question de l'utilité d'un tel outil générique. Son avantage est qu'il puisse être utilisé par toute personne manipulant les règles d'association et ce dans n'importe quel contexte. L'inconvénient est qu'il ne tient pas du tout compte du domaine applicatif et ne fait intervenir aucune connaissance spécifique à ce domaine. Dans l'objectif de développer de vrais outils d'aide à la décision, il est maintenant nécessaire de faire intervenir le décideur dans la phase d'analyse d'un tel outil.

Une autre perspective concernant l'optimisation multi-objectif pour l'extraction de connaissances, concerne une reflexion sur l'intérêt de rechercher toutes les solutions Pareto et seulement ces solutions.
En effet, l'intérêt d'avoir tout le front Pareto, y compris les solutions extrêmes est une question qui peut se poser pour tous les problèmes multi-objectifs. Pour certains domaines, cela peut tout de même se justifier. En extraction de connaissances, les critères généralement énoncés ne sont pas relatifs à des coûts que l'on peut précisément mesurer par exemple, mais représentent plutôt des tendances que l'on cherche à rejoindre. Aussi, si plusieurs objectifs sont utilisés, on cherchera en général à trouver les solutions de compromis se trouvant au centre du front Pareto. Ainsi les solutions extrêmes ne correspondent pas forcément à des solutions pouvant intéresser le décideur.
Pour les mêmes raisons sur le type d'objectif utilisé, il est intéressant de se demander si seules les solutions Pareto sont intéressantes ou si des solutions proches du front Pareto ne le sont pas également. Il serait donc intéressant de relacher la définition de dominance afin "d'épaissir" le front Pareto et fournir des solutions proches. Cela pourrait se faire en passant par la notion d'epsilon-dominance, par exemple.

Chapitre 5

Méthodes coopératives pour les règles d'association multi-objectifs

Dans le chapitre précédent nous avons discuté de la modélisation multi-objectif du problème de recherche de règles d'association et avons proposé une première approche de résolution. Étant donné le très large espace de recherche relatif à ce type de problème, une question se pose quant à l'efficacité absolue de la méthode proposée et il semble intéressant de regarder ce que pourrait apporter une approche coopérative. En effet, comme nous l'avons exposé précédemment et visualisé pour le problème de flow-shop bi-objectif, la coopération entre méthodes semblables ou différentes permet pour des problèmes d'optimisation combinatoire à grand espace de recherche d'obtenir de meilleures solutions. Nous cherchons donc à voir si pour des problèmes de type extraction de connaissances, l'approche coopérative est également intéressante.
Deux types d'approches sont considérées dans ce chapitre. Elles concernent soit la coopération de méthodes de même type, soit la coopération de méthodes différentes. Ainsi, nous présentons dans un premier temps une approche coopérative parallèle développée pour le problème de recherche de règles, dans laquelle différents algorithmes génétiques semblables à celui présenté dans le chapitre précédent coopèrent. Puis dans un deuxième temps nous présentons une approche coopérative entre une métaheuristique et un algorithme énumératif. Ces travaux font partie de la thèse de Mohammed Khabzaoui.

5.1 Approche coopérative parallèle

Nous proposons d'utiliser un modèle en îles (voir figure 5.1) où chaque île exécute l'algorithme génétique défini au chapitre précédent et envoie régulièrement quelques solutions de son archive Pareto locale à l'île voisine (topologie en anneau).

5.1.1 Définition de la politique d'échange

Un modèle en îles nécessite la définition de la politique d'échange entre les îles. Les principales questions sont :

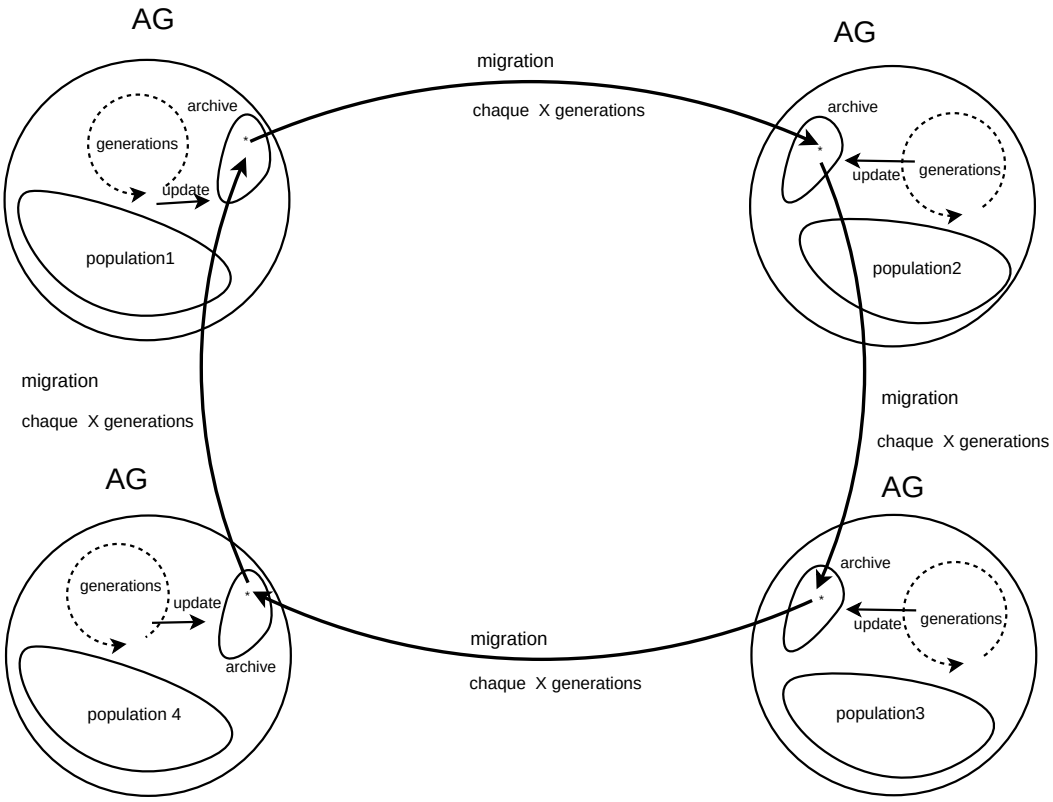


FIG. 5.1 – Modèle en îles.

– Quelle est la structure du voisinage ?
– Comment et quand les solutions doivent-elles migrer ?
– Combien de solutions doivent migrer et comment les sélectionner ?
– Quelles sont les solutions remplacées lors de la réception des solutions de l'île voisine ?

Pour définir la meilleure politique d'échange, nous avons effectué différentes expérimentations afin de déterminer **quand** et **combien** de solutions doivent être envoyées. Pour ces expérimentations, une base de données d'expression génique est utilisée (*yeastDB*). Les paramètres par défaut sont : taille de la population = 300, sélection = 2/3 (200), taux de mutation = 0.5, taux de croisement = 0.8, sélection dans l'archive Pareto (élitisme) = 0.5, nombre minimal de générations = 300. Les résultats représentent des moyennes sur un minimum de 10 exécutions. De nouveau, la contribution et l'entropie sont utilisées. (*Rappel : La contribution indique le ratio des solutions non dominées d'un front par rapport à un autre (une contribution supérieure à 0.5 est une amélioration). L'entropie mesure la diversité du front (plus l'entropie est proche de 1, mieux les solutions sont réparties sur le front)*).

Combien de solutions doivent être échangées ?

TAB. 5.1 – Comparaison de plusieurs scénari sur le nombre de solutions échangées.

	Contribution moyenne (10 exécutions)				
	2%	7%	10%	20%	50%
2%	-	0.47	0.47	0.51	0.51
7%	0.53	-	0.48	0.54	0.54
10%	**0.53**	**0.52**	-	**0.54**	**0.56**
20%	0.49	0.46	0.46	-	0.50
50%	0.49	0.46	0.44	0.50	-

Le tableau 5.1 indique les contributions deux à deux des fronts obtenus à l'aide de différents scénari dans lesquels le pourcentage des solutions de l'archive Pareto envoyées à l'île voisine varie de 2% à 50%. Il apparaît clairement ici que le scénario à 10% est meilleur que tous les autres.

Quand ces solutions doivent-elles être échangées ?

De même, le tableau 5.2 compare différents scénari dans lesquels le nombre d'itérations séparant les migrations varie. Ainsi les migrations ont lieu toutes les 5, 10, 25, 50 ou 80 générations. Une fois encore, un scénario surpasse les autres. Il s'agit du scénario toutes les 50 générations.

TAB. 5.2 – Comparaison de plusieurs scénari sur quand échanger les solutions.

	Contribution moyenne (10 exécutions)				
	5	10	25	50	80
5	-	0.50	0.48	0.46	0.54
10	0.50	-	0.47	0.44	0.50
25	0.52	0.53	-	0.46	0.49
50	**0.54**	**0.56**	**0.54**	-	**0.52**
80	0.46	0.50	0.51	0.48	-

Conclusion sur la politique d'échange

Bien sûr ces expérimentations sont insuffisantes pour conclure précisément sur la valeur que devrait prendre ces paramètres. De plus, il faudrait étendre ces expérimentations à d'autres bases de données.
Néanmoins, ces expérimentations montrent qu'il n'est pas souhaitable, dans le but d'une bonne coopération, d'échanger trop d'informations trop souvent. Il est intéressant de laisser chaque île évoluer entre les partages d'informations.
Pour la suite des expérimentations nous avons choisi de réaliser des échanges toutes les 50 générations de 10% de l'archive Pareto locale.

5.1.2 Validation de l'approche coopérative

Afin de valider l'approche coopérative, trois configurations différentes ont été testées (voir figure 5.2). Ces configurations ont été choisies afin d'avoir une même population globale :

- *Conf 1 :* Un seul algorithme génétique, avec une population de **3 000** individus. L'archive Pareto de l'algorithme est l'archive finale.
- *Conf 2 :* Dix algorithmes génétiques indépendants, d'une population de **300** individus chacun. Les dix algorithmes contribuent à l'archive Pareto finale.
- *Conf 3 :* Dix algorithmes génétiques coopérants, d'une population de **300** individus chacun. Les dix algorithmes contribuent à l'archive Pareto finale.

Le Tableau 5.3 compare en moyenne les fronts obtenus par 10 exécutions de chaque configuration. Le tableau montre clairement que *Conf 3* > *Conf 2* > *Conf 1* que ce soit pour l'efficacité du front ou pour la diversité et atteste de l'intérêt de la coopération de méthodes semblables.

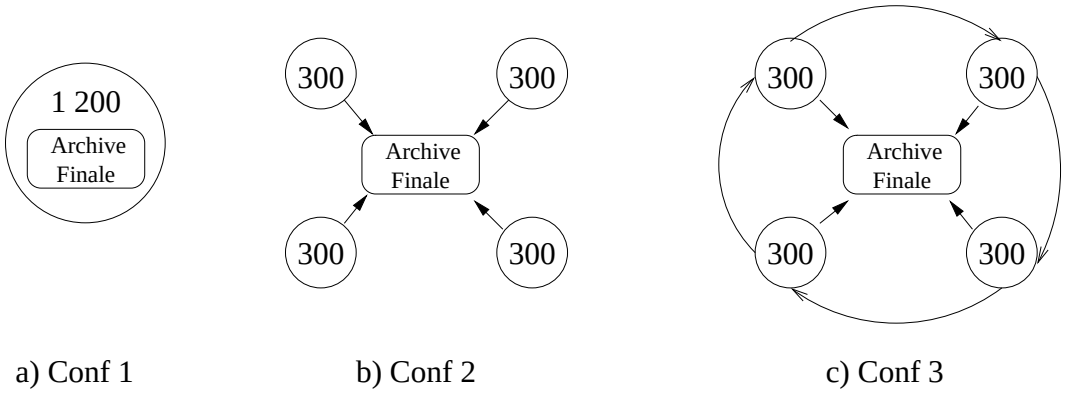


FIG. 5.2 – Les trois configurations testées.

TAB. 5.3 – Comparaison des configurations.

	Contribution moyenne			Entropie moyenne		
	$Conf1$	$Conf2$	$Conf3$	$Conf1$	$Conf2$	$Conf3$
$Conf1$	-	0.39	0.28	-	0.56	0.50
$Conf2$	**0.61**	-	0.40	**0.69**	-	0.53
$Conf3$	**0.72**	**0.60**	-	**0.71**	**0.70**	-

5.2 Coopération avec une approche énumérative

Une autre façon de voir la coopération est d'utiliser différents types de méthodes, afin de tirer partie des avantages de chacune des méthodes. Ici, par exemple nous cherchons à exploiter la capacité d'exploration de l'algorithme génétique et la capacité d'intensification de l'approche énumérative.

5.2.1 Modèle mis en œuvre

Dans cette étude, la procédure énumérative est utilisée comme un opérateur de croisement lorsque le nombre d'attributs différents composant les deux règles parentes n'est pas trop élevé (voir figure 5.3). Ainsi, la procédure énumérative va explorer la région déterminée par les attributs participants aux règles parentes. Le résultat d'un tel opérateur consiste en une archive Pareto locale qui est ensuite utilisée pour mettre à jour l'archive globale et pour générer les enfants.

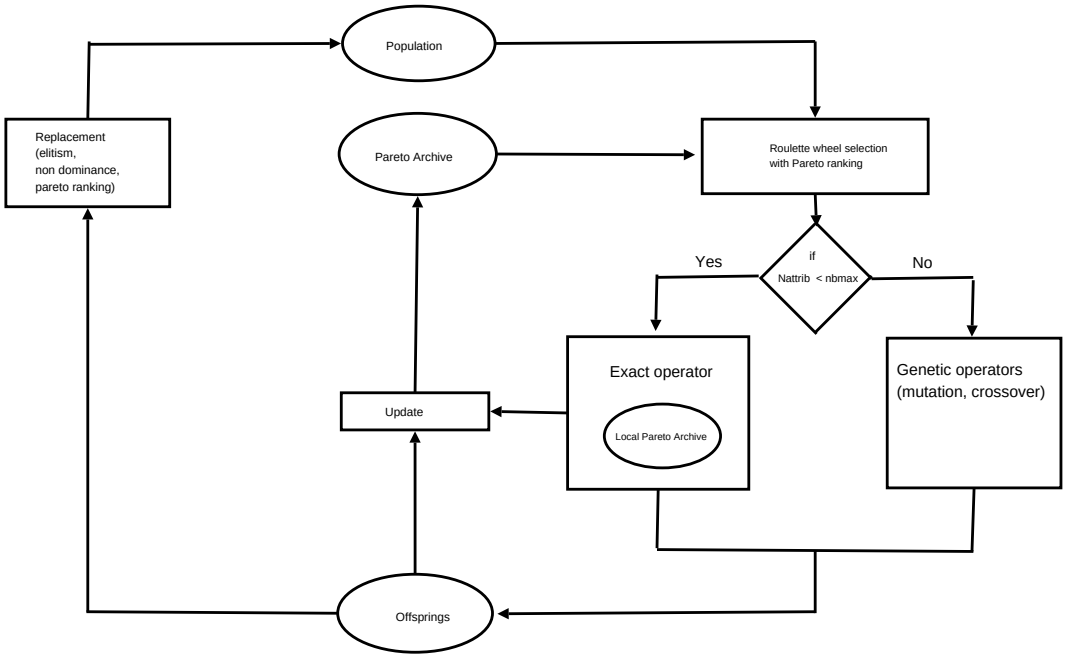


FIG. 5.3 – Coopération méta-exacte.

`Crossover(`$Rule_1, Rule_2$`)`

{
$AttributeSet \leftarrow Attrib_{Rule_1} \cup Attrib_{Rule_2}$ `// Construction de l'ensemble des attributs`
$nb \leftarrow \mid AttributeSet \mid$ `// calcul du nombre d'attributs`

`if (`$nb < MaxNb$`)`
`EnumProc(AttributeSet,nb)`
`else`
`NormalCrossover(`$Rule_1, Rule_2$`)`
}

Différents paramètres sont à définir :
- $MaxNb$ - nombre maximum d'attributs différents que l'on autorise pour lancer la procédure exacte.
- Quand doit-on faire appel à cette procédure ? toutes les générations ? toutes les x générations ?
- Doit-on appliquer cette procédure sur tous les couples formés pour la reproduction ? Ou seulement un sous-ensemble.

5.2.2 Evaluation du modèle

Pour évaluer le modèle, des expérimentations sur le même jeu de données d'expression génique qu'au chapitre précédent ont été réalisées. Mais dans ce cas, un front de référence, a

été utilisé. Ce front a été construit en prenant l'intersection de tous les fronts jamais trouvés au cours des différentes expérimentations que nous avons menées.
Le tableau 5.4 montre la contribution de l'utilisation de l'opérateur exact. Les différentes configurations font varier la fréquence de l'utilisation de cet opérateur depuis 0% (pas d'appel à l'opérateur) jusque 100% (l'opérateur est utilisé toutes les générations). Dans la première colonne, la mesure de contribution est utilisée et une moyenne sur 10 exécutions est indiquée. Il est clair que la version à 100% surpasse les autres versions. Pourtant, comme le montre la figure 5.4, l'accroissement n'est pas linéaire et même une fréquence moyenne permet une nette amélioration de la qualité des fronts obtenus.

TAB. 5.4 – Comparaison des configurations - Contribution.

Fréquence d'application	Contribution moyenne	D-Metric moyenne
0 %	3 %	5,20
20 %	18 %	4,47
50 %	22 %	4,04
100 %	29 %	3,80

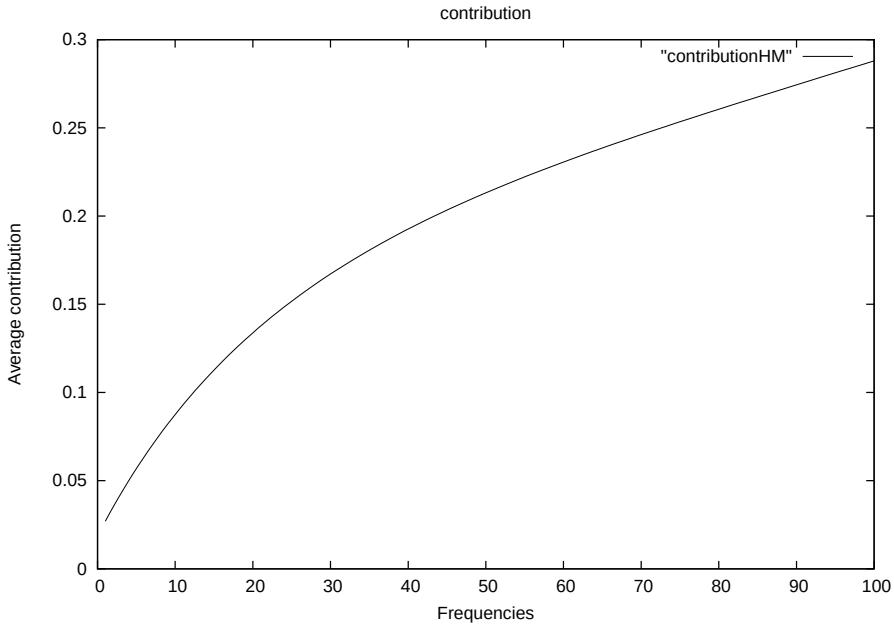


FIG. 5.4 – Evolution de la contribution en fonction de la fréquence d'utilisation de l'opérateur exact.

La deuxième colonne compare les fronts obtenus avec les différentes fréquences à l'aide de la D-métrique. Cette métrique compare deux fronts en utilisant la S-métrique qui mesure l'hyper volume de l'espace objectif dominé par un front. La D-métrique compare la S-métrique

obtenue pour chacun des fronts. Au plus le chiffre donné est petit, au plus l'hyper volume dominé par le front de référence est petit et donc le front comparé est meilleur. Ces résultats confirment bien évidemment la suprématie de la version à 100% avec la même remarque sur la non linéarité de cette augmentation qui est visualisée sur la figure 5.5.

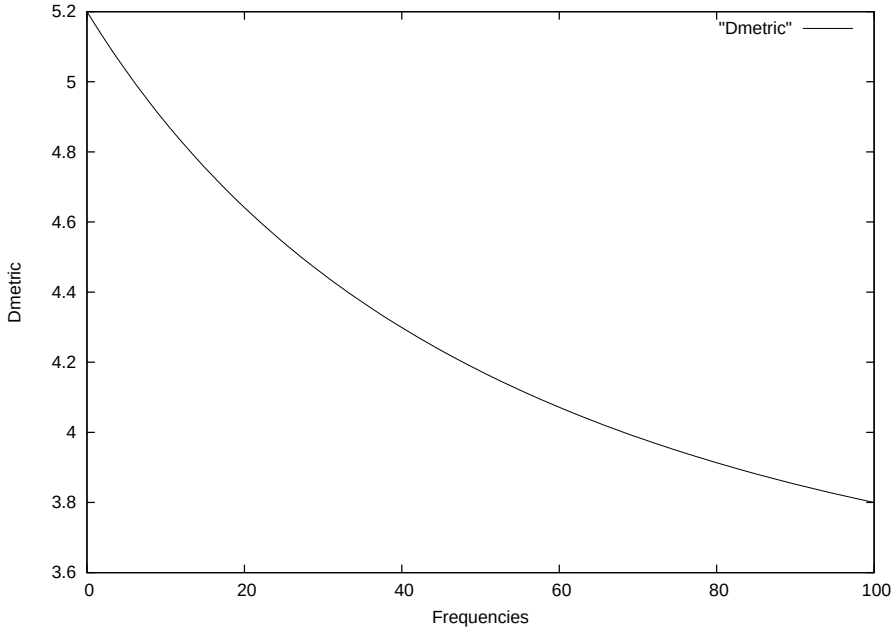


FIG. 5.5 – Evolution de la D-métrique en fonction de la fréquence d'utilisation de l'opérateur exact.

5.2.3 Conclusion sur la coopération avec l'opérateur exact

Ainsi, les expérimentations permettent de confirmer l'utilité de la coopération avec une procédure énumérative. Ces expérimentations montrent aussi qu'il n'est pas forcément nécessaire d'utiliser la procédure d'énumération à chaque itération si le temps d'exécution est un critère important.
En effet, puisque la procédure énumérative est la partie la plus coûteuse, l'utilisation de cette procédure deux fois plus souvent conduira à un temps d'exécution deux fois plus long. Ainsi, dans un soucis de compromis entre temps d'exécution de la recherche et qualité des solutions obtenues, il peut être intéressant de n'exécuter la procédure qu'une fois sur deux, par exemple. Ceci conduit, dans le cas des données étudiées, à une diminution du temps de 50% pour une qualité diminuée d'environ 25%.

5.3 Conclusions et perspectives

L'approche coopérative semble donc très prometteuse même pour des problèmes moins classiques de Recherche Opérationnelle. Dans notre contexte, nous avons montré à la fois l'apport du parallélisme et l'apport de coopération entre méthodes de différents types.

Concernant l'approche coopérative entre méthodes de même type, les principales perspectives concernent l'étude de différents schémas de coopération. De plus, pour le moment chaque île a les mêmes paramètres. Il peut être intéressant de faire des îles différentes, de façon à dédier certaines d'entre elles à l'exploration et d'autres à l'exploitation, par exemple.

En ce qui concerne la coopération avec des méthodes exactes, il est décevant que dans ce contexte (règles d'association multi-objectifs), il ne puisse exister de méthodes exactes plus intéressantes. Cependant il peut être envisageable d'hybrider les méthodes heuristiques avec l'algorithme *A-priori*, qui dans un sous-espace de recherche pourra trouver toutes les règles vérifiant un seuil sur le support et la confiance. Ceci pourra, par exemple, permettre de diversifier la recherche. De plus, il serait intéressant de réflechir à d'autres schémas de coopération entre les méthodes.

Cependant les perspectives les plus intéressantes concernent sans aucun doute l'utilisation conjointe de ces deux modes de coopération. Ainsi, on pourrait imaginer différentes îles coopérant suivant le schéma indiqué plus haut, à la différence que chaque île serait une méthode hybride faisant appel à une procédure d'énumération exacte. Ceci permettrait d'utiliser le parallélisme à la fois pour ces aspects coopératifs (parallélisme de haut niveau) que pour ses aspects performances (en parallélisant au sein de chaque île les appels à la méthode exacte - parallélisme de haut niveau). En poussant plus loin, il serait intéressant de voir comment l'opérateur exact peut être parallélisé pour le rendre plus rapide (parallélisme de bas niveau).

Chapitre 6

Extraction de connaissances en génomique

Notre recherche sur l'extraction de connaissances a été motivée par des applications issues de la génomique et de la post-génomique. Nous expliquons ci-après le contexte d'étude ainsi que certaines définitions nécessaires à la compréhension des deux problématiques étudiées. Puis nous exposons ces problématiques, à savoir l'étude du déséquilibre de liaison et l'étude de données issues d'expérimentations sur biopuces et présentons les approches proposées.

6.1 Contexte d'étude

La recherche génomique est un véritable enjeu pour notre société et de nombreux laboratoires de spécialités différentes (recherche en biologie, en médecine ou en technologie de l'information) se regroupent pour participer à des recherches sur des thèmes précis. En ce qui nous concerne, avec nos différents partenaires (la société IT-OMICS et l'Institut de Biologie de Lille, et en particulier le Laboratoire Génétique des Maladies Multifactorielles), nous étudions les facteurs génétiques de prédisposition à certaines maladies multifactorielles telles que le diabète, l'obésité et les maladies cardio-vasculaires. L'originalité commune des différentes problématiques est de rechercher non pas un seul facteur explicatif, mais bien une ou plusieurs combinaisons de facteurs (pouvant être de différentes natures : facteurs génétiques, facteurs environnementaux...) parmi un ensemble très grand de facteurs potentiels (plusieurs milliers).

Les objectifs scientifiques consistent à formuler des hypothèses quant aux associations susceptibles d'avoir de l'influence sur les maladies étudiées, hypothèses que les biologistes et généticiens devront ensuite vérifier à l'aide d'expérimentations supplémentaires.
Nous nous plaçons en particulier dans le cadre de la génomique et de la post-génomique, caractérisées par un volume de données brutes en augmentation très rapide (grâce aux nouvelles technologies de récolte de ces données). En effet, la difficulté réside aujourd'hui non plus seulement dans l'obtention de ces données, mais également dans leur analyse.

Ainsi, un de nos objectifs consiste à développer des méthodes d'analyse permettant d'extraire un maximum d'informations à partir des données récoltées par les biologistes et généticiens. La première étape concerne donc la modélisation des problématiques exprimées par les biologistes. En ce qui nous concerne, nous les modélisons en des problèmes d'extraction de connaissances que nous transformons ensuite en des problèmes d'optimisation combinatoire. Comme nous l'avons vu précédemement il est donc nécessaire de définir à la fois l'ensemble des solutions admissibles, et donc déterminer quel type de réponse est attendue (classification des instances, relations entre les attributs,...) et le(s) critère(s) à optimiser.
Mais avant de présenter les problématiques, positionnons rapidemment le contexte.

6.2 Bioinformatique / génomique / post-génomique

Afin de mieux situer les données sur lesquelles nous travaillons, il est nécessaire de positionner la post-génomique par rapport à la génomique au sein de la bioinformatique.

6.2.1 La bio-informatique

Ce terme très à la mode peut être rapidemment défini comme le traitement informatique de données issues de la biologie. Différentes définitions peuvent être trouvées :

- **Définition de François Jeanmougin :** Discipline de la biologie spécialisée dans le traitement en masse des données issues d'expériences de biologie.
- **Définition de InfoBiogen :** La bioinformatique a fait son apparition dans les années 1980 avec les premières banques de biomolécules (EMBL et GenBank). Elle propose des méthodes et des logiciels qui permettent de gérer, d'organiser, de comparer, d'analyser, d'explorer l'information génétique et génomique stockée dans les bases de données afin de prédire et produire des connaissances nouvelles dans le domaine ainsi qu'élaborer de nouveaux concepts. Les axes privilégiés autour desquels se focalise la bioinformatique sont :
  - la formalisation de l'information génétique,
  - l'analyse des séquences (biomolécules) et de leur structure (notamment structure 3D),
  - l'interprétation biologique de l'information génétique,
  - l'intégration des données (établissement de cartes et de réseaux d'interactions géniques, d'interactions protéiques ...),
  - la prédiction fonctionnelle.

La **définition de François Rechenmann** [1] permet également de positionner la génomique et la post-génomique. "La bioinformatique a pour but de produire de nouvelles connaissances sur le fonctionnement des cellules des organismes vivants, leur évolution, leur état sain ou pathologique... Pour ce faire, elle s'est tout d'abord limitée à la génomique, qui étudie la structure, le fonctionnement et l'évolution des génomes. Mais il est apparu que la représentation de la cellule donnée par la génomique est statique, et ne permet pas de rendre compte de son évolution au cours du temps. Ainsi est née la post-génomique, qui cherche à savoir quand et dans quelles conditions les gènes vont enclencher la fabrication de protéines,

[1] Source : Dossier Bioinformatique de Interstices : http ://interstices.info/

et comment les protéines fabriquées interviennent dans le fonctionnement de la cellule".

Ainsi la bioinformatique est un domaine très large nécessitant d'importants moyens de gestion et d'analyse de données et dans lequel l'extraction de connaissances a naturellement sa place pour l'analyse des très nombreuses données issues des expérimentations biologiques.

6.2.2 La génomique

Science des génomes, la génomique regroupe un ensemble d'analyses qui vont de l'établissement de cartes du génome (cartographie) à l'identification de nouveaux gènes, à l'étude de leurs fonctions et au séquençage des molécules d'ADN. Ainsi, la génomique étudie le génome c'est-à-dire la molécule d'ADN située dans les cellules d'un être vivant. L'ADN est comme un collier de 4 "perles" différentes (les bases) qui contient les gènes, plans de montage des protéines. La génomique a pour but de trouver l'ordre des "perles", d'identifier les gènes et leur fonction (quelles protéines ils codent) et de déterminer le rôle du reste de l'ADN.

Identification de gènes

Les protéines, qui sont essentielles au fonctionnement des êtres vivants sont fabriquées à l'aide d'un machinerie cellulaire complexe dont les instructions sont issues des gènes (partie codantes de l'ADN). Il est donc fondamental d'identifier ces parties codantes au sein de l'ADN.

Fonction des gènes

Une fois les gènes identifiés, une analyse sur leur fonction est à réaliser. Elle peut se faire par criblage de banques de données en recherchant si des gènes ayant des séquences similaires ont déjà des fonctions connues.

Limites de la génomique

La principale limite de la génomique est que la cellule est vue de façon statique. Même si cette phase d'identification de gènes est indispensable, il est nécessaire ensuite d'étudier les produits d'expression des gènes. C'est la post-génomique.

6.2.3 La post-génomique

La post-génomique tente d'expliquer le fonctionnement de la cellule en regardant dans quelles conditions un gène s'exprime et déclenche la fabrication de protéines. Elle analyse donc le transcriptome (analyse des ARNm - ARN messagers) et le protéome (analyse des protéines).

Le transcriptome

Si au sein d'un organisme donné, le génome est le même dans toutes les cellules, le transcriptome varie selon le stade de développement de la cellule, le type de cellule et sa situation physiologique (état sain ou pathologique).
Ainsi, le transcriptome est dynamique. L'analyser consiste à identifier, à un instant donné, sous des conditions données, les séquences codantes du génome qui sont effectivement exprimées. Comme nous l'expliquerons dans la deuxième problématique, l'étude du transcriptome repose essentiellement sur la technologie des biopuces (puces à ADN).

Le protéome

Ce sont les protéines qui régissent la vie cellulaire. Ainsi connaître la quantité de chaque protéine présente à un instant donné dans une cellule permet de comprendre le fonctionnement cellulaire. C'est l'objet de l'analyse du protéome. Cette analyse est complémentaire du transcriptome, car un ARNm peut coder différentes protéines plus ou moins actives. Ainsi, l'information contenue dans le protéome peut être encore plus précise que celle contenue dans le transcriptome (mais encore plus complexe à décoder) ! !

6.2.4 Positionnement des problématiques étudiées

Dans le cadre de nos études, différentes problématiques ont été étudiées. Nous en exposons deux ici.
La première concerne l'étude du déséquilibre de liaison. Elle étudie les configurations de variants (variation du code de l'ADN) qui peuvent apparaître en certains points particuliers de l'ADN. L'objectif étant de corréler certaines configurations particulières à l'apparition d'une maladie. C'est typiquement une problématique qui relève de la génomique. Ce travail a été réalisé en collaboration avec l'Institut de Biologie de Lille (IBL).
La deuxième problématique exposée concerne l'analyse du transcriptome à travers des expérimentations sur puces à ADN. Ceci relève donc de la post-génomique. Ce travail a été réalisé en collaboration avec la société de biotechnologies IT-Omics (Loos-59).

6.3 Premier exemple : étude du déséquilibre de liaison

L'étude du déséquilibre de liaison (*linkage desequilibrium*) consiste à rechercher des associations préférentielles entre SNPs. Cette étude a été menée avec le Laboratoire Génétique des Maladies Multifactorielles (LGMM) qui cherche à identifier des facteurs (génétiques et autres) de prédisposition à certaines maladies telles que le diabète et l'obésité. La particularité de ces maladies multifactorielles est que ce sont des combinaisons de facteurs qui sont à leur origine. Aussi dans le cadre de la recherche en génomique, l'hypothèse est faite que plusieurs gènes interagissant pourraient être en partie la cause de telles maladies. Deux approches ont été proposées pour aborder ce problème. La première approche consiste à modéliser le problème comme une sélection d'attributs tandis que la deuxième consiste à

rechercher des règles d'association. Avant de présenter ces deux approches quelques mots sur la problématique permettront de comprendre les enjeux liés au problème.

6.3.1 Problématique biologique

Sans rentrer dans les détails liés à la biologie, car la problématique du déséquilibre de liaison mériterait à elle seule plusieurs pages d'explication, voici tout de même quelques notions.

Un **SNP** (*Single Nucleotide Polymorphism*) est une mutation ponctuelle d'un seul nucléotide [2] qui au lieu d'avoir sa valeur habituelle prend une autre valeur. Chaque être humain possèderait 10 millions de SNPs ce qui rend, par combinaison, chaque personne unique. Lorsqu'un SNP est situé sur un gène ou sur une région régulatrice, il peut avoir une influence sur le comportement de ce gène et être à l'origine de troubles fonctionnels. Repérer des SNPs peut donc permettre de découvrir des gènes de prédisposition à certaines maladies.
Un **haplotype**, est un ensemble constitué de plusieurs SNPs. Il peut permettre de suivre non plus un seul gène, mais plusieurs gènes simultanément. Nous comprenons bien, dans le cadre d'étude de maladies multifactorielles, l'intérêt de l'étude des haplotypes. Pour cela, on étudie si les SNPs constituant un haplotype ont statistiquement la même fréquence que s'ils étaient seuls. Ainsi un haplotype composé de SNP-A et SNP-B doit avoir la même fréquence que la fréquence de SNP-A multipliée par la fréquence de SNP-B. Dans le cas contraire, on parle de déséquilibre de liaison, et ce sont ces déséquilibres que l'on cherche à associer avec l'apparition de la maladie.

Les données utilisées pour cette étude sont composées de plusieurs tables, avec entre autre :
– Pour chaque individu (de statut atteint, non atteint ou inconnu) la valeur pour différents SNPs,
– pour chaque SNP, la fréquence de chacun de ses variants (2 variants par SNP).
L'objectif consiste donc à identifier des haplotypes permettant d'expliquer la maladie sous étude. Ces haplotypes peuvent contenir un nombre différent de SNPs.

6.3.2 Modélisation en un problème de sélection d'attributs

Cette étude a fait l'objet du stage de DEA de Gregory Vermeersch et a ensuite été poursuivi dans la thèse de Laetitia Jourdan. Il a fait l'objet d'une présentation à EvoBio 2003 [44].

Motivations

L'idée de cette approche était de se baser sur les mesures utilisées par les biologistes pour évaluer la qualité d'un haplotype. Les biologistes utilisent pour cela deux procédures :

[2] un nucléotide est une unité de construction des acides nucléiques (ADN et ARN). Il existe quatre nucléotides différents pour l'ADN : adénine (A), thymine (T), guanine (G), cytosine (C) et quatre nucléotides différents pour l'ARN : uracile (U), guanine (G), cytosine (C), adénine (A).

– EH-DIALL, qui est une procédure permettant de déterminer la configuration la plus probable d'un haplotype (les variants de chacun des SNPs le composant),
– CLUMP, qui est un programme qui permet d'évaluer la signification des valeurs observées par rapport aux valeurs conditionnelles prévues.

Ainsi le processus d'évaluation d'un haplotype est le suivant :

1. appliquer EH-DIALL sur chacune des populations (affectée / non affectée) afin d'estimer la distribution des variants dans l'haplotype,
2. concaténer les résultats et appliquer CLUMP pour évaluer l'association haplotype-maladie.

Ce qui est important de voir ici est qu'étant donné un haplotype, il est possible à l'aide de ces deux procédures d'obtenir une valeur indiquant la pertinence de l'haplotype par rapport à la maladie. L'objectif est donc d'extraire un (ou des) haplotypes maximisant le critère de qualité. Cela peut être modélisé comme un problème de sélection d'attributs (sélection d'un sous ensemble de SNPs).

Approches de résolution

Pour cette modélisation deux approches sont utilisées. Une approche exacte et une approche heuristique.

L'approche exacte consiste en fait tout simplement en l'énumération des différents haplotypes possibles, car comme nous le verrons juste après, aucune relation permettant la réduction de l'espace d'énumération n'a été trouvée. Or il s'avère que le temps nécessaire au processus d'évaluation augmente rapidemment avec la taille de l'haplotype. Aussi, il n'a été possible d'énumérer complétement que l'ensemble des haplotypes de taille 2, 3 et 4. Ce faisant, cela nous a permis deux choses :
– de réaliser une étude sur la structure du problème,
– d'avoir à disposition les meilleurs résultats pour les plus petits haplotypes et pouvoir les comparer avec les algorithmes proposés.

L'analyse de la structure du problème montre quelque points intéressants :

1. Les meilleurs haplotypes de taille n ne sont pas forcément constitués des meilleurs haplotypes de taille $n-1$. Cette remarque rend impossible l'utilisation de méthodes constructives qui pourraient se baser sur les haplotypes de taille $n-1$ pour rechercher les haplotypes de taille n.
2. Les valeurs de qualité d'haplotypes de tailles différentes ne sont pas comparables. En effet, plus un haplotype est grand plus sa fonction objectif est élevée. Il n'est donc pas possible d'utiliser des algorithmes classiques d'énumération qui vont toujours privilégier les haplotypes de plus grandes tailles.
3. Une visualisation du paysage a montré que celui-ci est de type plat et rugueux. Ce type de paysage est très difficile à appréhender et nécessite le développement de méthodes à fort pouvoir d'exploration.

L'approche par algorithme génétique est spécifique au problème pour pouvoir tenir compte du fait que les haplotypes sont non comparables s'ils ne sont pas de même taille.
Dans cette approche chaque solution représente un haplotype (une sélection de SNPs). Puisque les solutions de taille différente ne sont pas comparables, la population a été divisée en différentes sous-populations contenant chacune les haplotypes d'une taille donnée. Des coopérations entre ces sous-populations sont possibles par le biais des opérateurs de croisement inter-population et de mutation qui ont été définis (voir [44] pour plus de détails). Les aspects intéressants de cette étude ont donc consisté à proposer des opérateurs pouvant interragir avec différentes populations. De plus, les valeurs de la fonction objectif étant dépendantes de la taille de l'haplotype évalué, il a fallu redéfinir les mécanismes d'adaptation, permettant de fixer les taux d'application des opérateurs. Différents mécanismes de diversification et un modèle parallèle de type maître-esclave ont également été proposés.
Les expérimentations ont permis de discuter de l'apport des différents mécanismes mis en œuvre.

6.3.3 Modélisation en la recherche de règles de classification

La même problématique biologique a été modélisée en une recherche de règles de classification.
L'objectif de cette deuxième modélisation est de s'affranchir des fonctions d'évaluation des haplotypes données par les biologistes. L'idée est alors de considérer qu'un haplotype peut être représenté par la partie condition d'une règle et la prédiction être le statut "malade". Ainsi on cherche à prédire les combinaisons de SNPs pouvant prédire la maladie. Cette étude a fait partie de la thèse de Laetitia Jourdan et a donné lieu à une publication dans la revue *Extraction de Connaissances et Apprentissage* [43].
Même si cette modélisation concerne les règles de classification (ou de prédiction, puisque un seul attribut but est considéré), nous l'abordons comme un problème de règles d'association, dans lequel l'attribut but peut être n'importe quel attribut de la base. Cette modélisation, étant définie, il faut donc déterminer la fonction objectif. Puisque cette étude était notre première étude sur les règles d'association, le critère de la J-mesure, couramment utilisé en optimisation, a été choisi. Pourtant c'est cette étude qui nous a ensuite incité à étudier plus précisément les critères existants et nous a mené ensuite vers la modélisation multi-objectif proposée au chapitre 4.
Une fois l'objectif déterminé, ASGARD (*Adaptive Steady state Genetic Algorithm for association Rule Discovery*) est proposé pour la recherche de règles d'association. Cet algorithme adaptatif met en œuvre différents opérateurs pour la recherche des règles. Puis, afin d'appliquer ASGARD précisément à la problématique sous étude, seules des règles de classification sont recherchées (l'attribut à prédire est le statut "malade") et des aspects biologiques sont ajoutés afin de prendre en compte certaines contraintes sur la construction des haplotypes. De plus un traitement des valeurs manquantes (phénomène très courant lorsque l'on travaille avec des bases de données réelles) est proposé.
Cet algorithme montre une bonne convergence et a permis de retrouver certains résultats obtenus à l'aide de la première modélisation. De plus, l'algorithme proposé a l'avantage d'être

rapide et peut donc facilement être utilisé sur des bases de données comportant un plus grand nombre de SNPs.

6.4 Deuxième exemple : analyse de données issues de biopuces

Le deuxième exemple traite de l'analyse du **transcriptome**.

6.4.1 Contexte biologique

L'étude du transcriptome repose en particulier sur la technologie des biopuces, ou puces à ADN, qui cherche à détecter les ARNm (ARN Messagers, produits par l'ADN et menant à la fabrication de protéines) présents dans un mélange donné. Plusieurs dizaines de milliers de résultats peuvent être obtenus simultanément. Comme pour le séquençage du génome, les données sont produites en masse, et nécessitent un traitement des résultats faisant intervenir la bioinformatique.

Définition : Puce à ADN [3]
Technologie employée dans l'étude du transcriptome et basée sur la capacité des molécules d'ADN et d'ARN à s'hybrider entre elles. De courtes séquences d'ADN connues sont fixées sur des supports d'une surface de l'ordre du centimètre carré : les puces. Elles sont mises en présence d'un mélange d'ARN de séquences inconnues. Le système est conçu de sorte à ne détecter que les paires d'ADN/ARN qui se sont hybridées. On en déduit les séquences des ARN présents dans le mélange étudié.

6.4.2 Problématique

Les expérimentations sur puces à ADN produisent un grand nombre de données. Mais ces données sont expérimentales et doivent donc être manipulées avec précaution. En particulier la phase de pré-traitement est très importante. Comparer différentes expérimentations par exemple, nécessite une phase de normalisation entre les données. Cette phase de normalisation, à elle seule représente un intéressant domaine de recherche pour les statisticiens. Pour aider dans cette phase de prétraitement des données, la plupart des fournisseurs de puces à ADN (Affymetrix, par exemple) proposent un logiciel intégrant des fonctionalités permettant de réaliser la plupart des traitement classiques. Lorsque le biologiste produit lui même ses puces (pour se focaliser sur les gènes qui l'intéressent le plus, par exemple) le problème reste entier et il doit mettre lui même en place ces traitements. Notons, que le logiciel R propose des packages à cet effet, mais qu'il n'est pas toujours facile de comprendre les traitements effectivement réalisés.

[3] Source : Dossier Bioinformatique de Interstices : http ://interstices.info/

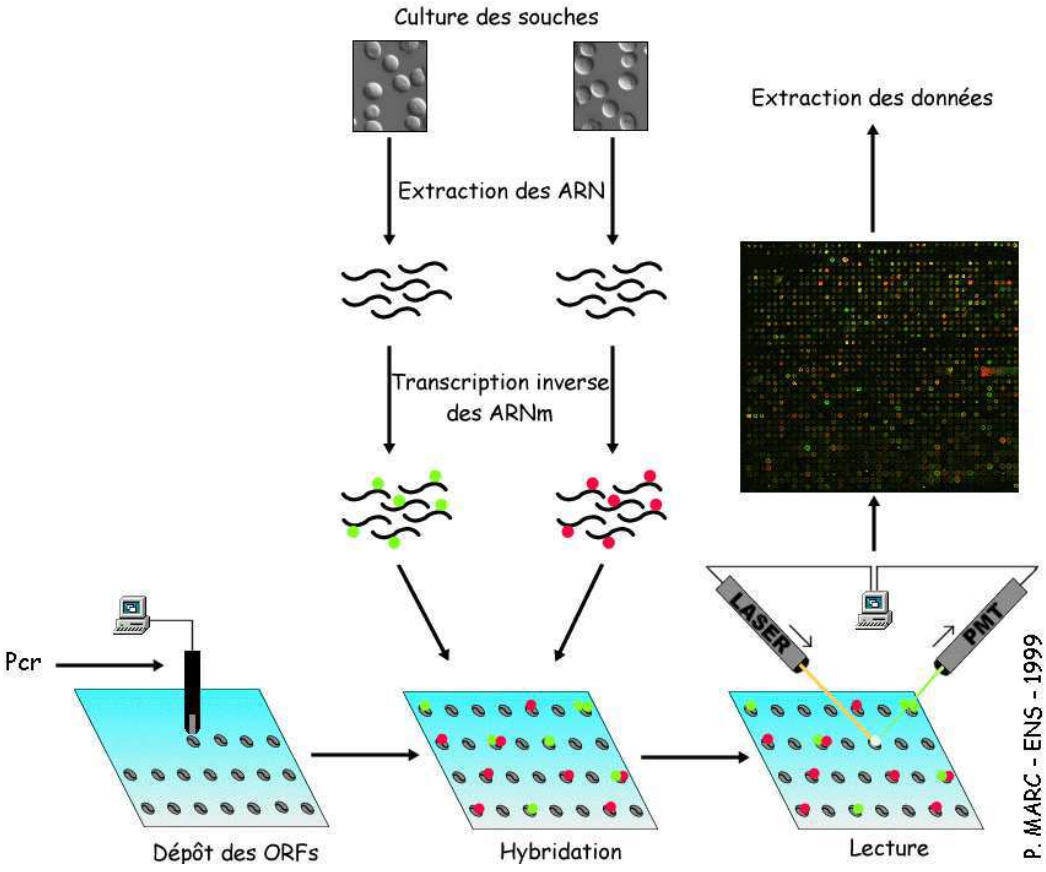


FIG. 6.1 – Puces à ADN : Procédé d'expérimentation.

Ici nous nous focalisons sur la phase d'extraction de connaissances dans des données que nous supposons déjà pré-traitées et donc normalisées. La question classique alors concerne la classification ou la segmentation des gènes en fonction de leurs niveaux d'expression. Les approches classiques varient depuis les approches statistiques [87] jusqu'aux approches évolutionnaires [62] et peuvent pour certaines, utiliser en plus des informations biologiques telles que celles contenues dans la *"Gene Ontology"* [74]. La méthode qui remporte le plus de succès auprès des biologistes reste la segmentation hiérarchique (*hierachical clustering*) de Eisen [23] qui a l'avantage d'être très visuelle. Récemment une approche par bi-clustering a été appliquée [9]. L'avantage de cette approche est qu'elle permet de classer les instances qui partagent les mêmes caractéristiques sur un sous-ensemble d'attributs seulement. Dans le contexte des puces à ADN, cela permet de regrouper des gènes partageant des mêmes schémas d'expression sur un sous-ensemble d'expérimentations (et non nécessairement sur l'ensemble des expérimentations, ce qui peut être limitant). Une approche évolutionnaire a été proposée à cet effet [9].

Ces modélisations en des problèmes de classification et segmentation, bien qu'apportant une réelle illustration de certaines relations existant entre les gènes, peuvent souffrir dans certains contextes de quelques limitations :
- Les classes recherchées doivent être vérifiées sur l'ensemble des expérimentations (même si le bi-clustering assouplie cette condition),
- un gène ne peut appartenir qu'à une seule classe,
- les relations entre les gènes appartenant à une même classe ne sont pas mises en évidence,
- ...

C'est pourquoi nous proposons de traiter ces données issues d'expérimentations sur puces à ADN en utilisant une modélisation par recherche de règles d'association. Ainsi, une relation peut n'être vérifée que sur un sous ensemble d'expérimentations (son support), un gène peut participer à plusieurs règles et une règle, avec sa notion de condition et prédiction permet de mettre en évidence des relations plus précises entre les gènes.

6.4.3 Recherche de règles d'association dans des données d'expression

Nous proposons donc d'aborder le problème par une recherche de règles d'association.

Représentation des données

La première question concerne la représentation des données à utiliser. Il y a communément deux façons de représenter les données [54] :
- **La "Gene table"** dans laquelle les instances sont les gènes et les attributs sont les traitements appliqués (les expérimentations réalisées). Avec cette représentation, il est possible de faire de la segmentation de gènes en fonction de leurs profils d'expression.
- **La "Treatment table"** qui est la transposée de la première et dans laquelle les expérimentations sont les instances et les gènes, les attributs. Cela permet de rechercher des relations entre

les gènes notamment à l'aide de règles d'association.
Nous avons donc choisi la deuxième représentation, même si nous sommes conscients que pour certains jeux de données le nombre d'expérimentations (et donc d'instances) peut être réduit. Ceci devrait être de moins en moins vrai avec le développement de ce type d'expérimentations.

Modélisation

Ce que nous recherchons dans ces données sont des relations entre les niveaux d'expression des gènes. Ainsi, une règle intéressante serait :
Si <gène$_{18}$, sur-exprimé> et <gène$_{3856}$, sur-exprimé> et <gène$_{18534}$, sous-exprimé>
Alors <gène$_{512}$, sur-exprimé>

Dans notre étude nous nous sommes restreints aux règles possédant un seul attribut dans la partie prédiction. Il nous semble que cela permet une meilleure compréhension des relations entre gènes, mais cela peut tout à fait être étendu à des prédictions avec plusieurs termes.

Pour évaluer la qualité des règles nous utilisons l'approche multi-objectif présentée dans le chapitre 4. En effet, dans le contexte de l'analyse de données issues de biopuces, les règles de fort *support* ne sont pas toujours intéressantes puisque l'on recherche plutôt les exceptions (ce qui n'arrive pas souvent et qui peut expliquer certaines maladies). Il est donc nécessaire de réfléchir à la notion de "bonne règle". C'est dans ce cadre, qu'à partir des critères classiques de la littérature, une analyse statistique a été réalisée (voir chapitre 4). Cette analyse a été menée également sur des bases de données issues d'expérimentations sur biopuces et a montré le même type de résultats. Ainsi nous proposons d'utiliser une modélisation multi-objectif du problème.

Approches de résolution

Les approches mises en œuvre pour ce problème ont été présentées dans les chapitres 4 et 5. Nous ne revenons pas ici sur leur présentation, mais reprécisons ce qui nous a amené à les développer.
Comme il a été expliqué dans les chapitres 4 et 5, une approche exacte a été envisagée. Malheureusement, les critères utilisés ne présentent pas de propriétés intéressantes permettant de construire une approche efficace. Seule une énumération exhaustive est possible.
Une approche évolutionnaire est alors proposée pour pouvoir traiter les problèmes de grandes tailles. Des approches coopératives, mélant parallélisme et coopération entre méthodes de différents types sont proposées.

6.5 Conclusion

Les problèmes rencontrés en bio-informatique peuvent être de différentes natures en fonction du niveau de la cellule qui est étudié. De plus, un même type d'étude biologique peut mener

à différents problèmes de bioinformatique en fonction de l'hypothèse que peut formuler le biologiste sur ce qu'il cherche. Il peut guider la recherche vers différents problèmes d'extraction de connaissances. C'est pourquoi, il nous avait semblé opportun d'étudier un même problème de génomique à l'aide de différentes modélisations.

Un point intéressant à retenir est que les méthodes d'optimisation, et en particulier les méthodes évolutionnaires, sont bien adaptées pour l'extraction de connaissances en génomique, et ce pour différentes raisons [24]. Tout d'abord, étant donnée la combinatoire des problèmes, l'espace de recherche associé est souvent très grand, et si l'on veut avoir une vision globale du problème, il faut mettre en œuvre des méthodes capables de gérer ces immenses espaces de recherche. De plus, ces méthodes permettent de prendre en compte des spécificités, des contraintes sur les problèmes. En effet, même si certains schémas de méthodes peuvent sembler similaires, pour être efficace une méthode doit se spécialiser au problème étudié et en utiliser les caractéristiques et les contraintes. Les méthodes d'optimisation sont en général assez souples et permettent de le faire. De plus ces méthodes peuvent intégrer des fonctions particulières liées à la problématique telles que la gestion des valeurs manquantes, la prise en compte de contraintes biologiques sur les solutions proposées ou encore l'utilisation de connaissances connexes sur le domaine.

Conclusions et perspectives

Ce mémoire traite de l'optimisation combinatoire multi-objectif, en montrant l'apport des méthodes coopératives, son utilisation dans le cadre de l'extraction de connaissances et propose des illustrations issues de la bio-informatique.
Chaque chapitre se conclut par une partie perspectives plus ou moins importante. L'idée de cette partie n'est pas de reprendre précisément chacune des perspectives exposées avant, mais de faire une synthèse des contributions et de proposer des directions de recherche pour le futur.

La première partie traite des méthodes coopératives pour l'optimisation combinatoire multi-objectif. Ainsi une première contribution concerne la conception d'une méthode exacte - PPM : Parallel Partitionning Method - permettant de générer le front complet de tout problème bi-objectif. Cette méthode s'inspirant de deux méthodes de la littérature propose des améliorations permettant d'augmenter leurs performances sur des problèmes d'optimisation pour lesquels les problèmes mono-objectifs associés (lorsque l'on ne s'intéresse qu'à l'un des deux objectifs) sont déjà difficiles à résoudre. Le concept parallèle de cette méthode ouvre des perspectives quant aux problèmes pouvant être résolus. Cette nouvelle méthode, avec ses nouvelles performances, peut ainsi être utilisée au sein d'une méthode coopérative faisant intervenir une métaheuristique. Différents schémas de coopération ont été comparés. Le problème académique du flowshop bi-objectif a été utilisé pour tester les différentes méthodes.

Parallèlement à ces études, nous nous sommes intéressés à l'extraction de connaissances en particulier à partir de données génomiques. Notre approche consiste à modéliser des problèmes d'extraction de connaissances en des problèmes d'optimisation combinatoire. Une étude statistique de différents critères de qualité pour le problème de recherche de règles d'association nous a amené à proposer une modélisation multi-objectif pour ce problème. Ainsi, nous avons cherché à appliquer le même type d'approche que pour le flow-shop dans le but de valider l'apport des méthodes coopératives pour d'autres types de problèmes moins classiques en optimisation combinatoire. Concernant la méthode exacte mise en œuvre, l'étude des critères d'évaluation choisis pour le modèle multi-objectif ne nous a pas permis de trouver de propriétés intéressantes et une enumération exhaustive a donc été proposée. Cette recherche a été combinée à un algorithme génétique dédié à la recherche de règles. Egalement, une coopération entre différents algorithmes génétiques a été réalisée. Ces approches ont été appliquées sur des problèmes issus de la bio-informatique et ont montré l'intérêt de la coopération.

Les perspectives de ces travaux sont nombreuses, nous donnerons ici quelques grandes directions concernant différents points traités dans le mémoire.

Nous nous sommes concentrés ici sur l'optimisation combinatoire multi-objectif dont l'objectif est de fournir l'ensemble le plus complet possible des solutions Pareto à un problème multi-objectif. Or après avoir trouvé les solutions du problème, une autre question se pose : il faut, en général, ne sélectionner qu'une seule solution (ou un petit sous-ensemble) parmi les solutions proposées. La solution choisie reflétera ainsi les différents compromis faits par le décideur sur les différents objectifs. Dans ce cadre, les méthodes d'aide à la décision ont pour vocation de modéliser les choix (ou préférences) du décideur. Nous pouvons alors différencier deux grandes théories :

- **la théorie de l'utilité multi-attribut**, où l'on considère qu'en fait chaque décideur cherche implicitement à optimiser une fonction appelée fonction d'utilité [49],
- **la théorie de l'aide à la décision**, où l'on cherche à reproduire le processus de décision du décideur à l'aide d'outils permettant de sélectionner un sous-ensemble de solutions parmi un ensemble complet [71].

Dans le cadre de cette deuxième théorie, l'objectif de l'optimisation multi-objectif est donc de fournir au décideur cet ensemble complet. Malheureusement, ces deux phases sont souvent traitées séparément sans interconnexion. Pourtant, dans un souci d'efficacité des méthodes de recherche et de respect des choix du décideur il semble intéressant de gérer ces deux phases conjointement. Cela nécessite l'intervention du décideur pendant le processus d'optimisation. L'avantage est que le décideur peut au cours de la recherche affiner ses choix et dans le cadre d'applications spécifiques, faire intervenir des critères non mesurables et donc non optimisables par une fonction automatique. La difficulté de cette approche réside dans la nécessité de présenter au décideur un panel de solutions intéressantes, c'est-à-dire dans ce contexte, des solutions suffisemment différentes entre-elles et de bonnes qualités par rapport aux objectifs affinés au cours de la recherche. Il est donc nécessaire de proposer des schémas d'interaction entre l'algorithme de recherche et le décideur ainsi que des visualisations des solutions. Cette perspective, bien que très générale, sera difficilement réalisable de façon générique car le domaine applicatif au centre des recherches doit être pris en compte.

A propos des méthodes exactes pour l'optimisation combinatoire multi-objectif, différentes perspectives sont envisageables. Tout d'abord les plus immédiates concernent l'amélioration de performances de ces méthodes avec notamment une utilisation plus extensive du parallélisme ce qui permettrait de résoudre des problèmes de plus grandes tailles et également d'utiliser ces méthodes de façon plus intensive dans des schémas de coopération. Une autre perspective concerne le développement de méthodes exactes pour des problèmes à plus de deux objectifs. Même si théoriquement la méthode PPM semble s'y préter, les limites concernant les temps d'éxécution devront être repoussées. Enfin, une perspective intéressante concerne l'intégration de ce type de méthode dans des plateformes proposant des méthodes de résolution pour problèmes d'optimisation combinatoire. En effet les schémas proposés sont génériques et peuvent être appliqués à tout problème d'optimisation combinatoire multi-

objectif. La difficulté résidera néanmoins dans la séparation entre les aspects liés au problème à optimiser et les aspects liés à la méthode, elle même.

Les méthodes coopératives étudiées dans ces travaux restent d'un schéma très classique. En général un algorithme évolutionnaire appelle un fonction d'intensification (recherche locale ou opérateur exact, dans notre cas) en fin d'exécution ou comme opérateur de recherche. Il devient absolument indispensable, dans le but de toujours améliorer les qualités des solutions produites, d'innover en proposant de nouveaux schémas spécifiques pour l'optimisation multi-objectif. L'utilité de la coopération n'est plus à démontrer mais les performances peuvent encore être améliorées. Pour cela, une voie intéressante concerne la coopération avec d'autres types d'approches. En effet, les méthodes exactes utilisées en coopération sont souvent des approches par séparation et évaluation. Or, la programmation linéaire ou la programmation par contraintes peuvent offrir d'autres voies pour intensifier la recherche dans un sous-espace de recherche et permettre dans le cadre de méthodes coopératives d'obtenir de meilleurs résultats.

L'extraction de connaissances offre des problèmes d'optimisation combinatoire et c'est un véritable challenge pour la communauté de Recherche Opérationnelle. Sans prétendre pouvoir résoudre tous les problèmes, l'approche par optimisation combinatoire permet tout de même d'apporter un autre éclairage. En proposant une modélisation multi-objectif pour la recherche de règles, nous avons mis en évidence la non existence de critère unique. Ce constat peut être fait pour d'autres problématiques (telles que la segmentation, par exemple). Aussi, il serait intéressant d'étudier ces autres problématiques et voir ce que peut apporter l'optimisation multi-objectif.
A l'inverse, le dual est également vrai, puisque l'extraction de connaissances, par le biais de méthodes d'apprentissage, par exemple, peut apporter aux méthodes d'optimisation des informations importantes permettant de guider les recherches.

Les applications en bio-informatique ont montré l'importance de la partie modélisation. En effet, certains problèmes peuvent être modélisés de différentes façons menant à différents résultats. Dans ce domaine un aspect important concerne l'analyse de données hétérogènes (venant de différentes sources) puisque un grand nombre de données de différents types sont maintenant disponibles. Les recherches actuelles avancent dans ce sens. Deux voies sont possibles pour traiter ces données de différents types : soit toutes les données sont utilisées dans le processus d'extraction de connaissances, soit l'extraction se fait sur un type de données (données de biopuces, par exemple) puis les résultats sont enrichis/validés à l'aide d'autres données.

Bibliographie

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Intl. Conference on Very Large Databases, Santiago, Chile*, 1994.

[2] H. Azzag, C. Guinot, N. Monmarché, M. Slimane, and G. Venturini. Classification hiérarchique par des fourmis artificielles : application à la construction de sites portails. In *Cinquième congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF'03)*, pages 124–125, Avignon, France, 2003.

[3] J.E. Baker. Adaptive selection methods for genetic algorithms. In *Int. conference on Genetic Algorithms and their application*, page 101, 1985.

[4] M. Basseur. *Conception d'algorithmes coopératifs pour l'optimisation multi-objectif : Application aux problèmes d'ordonnancement de type Flow-shop*. PhD thesis, Université des Sciences et Technologies de Lille, 2005.

[5] M. Basseur, J. Lemesre, C. Dhaenens, and E-G. Talbi. Cooperation between Branch and Bound and Evolutionary Approaches to solve a Biobjective Flow Shop Problem. In *Workshop on Efficient and Experimental Algorithms (WEA'04)*, volume LNCS 3059, pages 72–86, Rio de Janeiro, Brazil, May 2004.

[6] M. Basseur, F. Seynhaeve, and E.-G. Talbi. Design of multiobjective evolutionnary algorithm : application to the flowshop scheduling problem. In *Congress on Evolutionnary Computation (CEC'02)*, IEEE Press, pages 1151–1156, 2002.

[7] M. Basseur, F. Seynhaeve, and E.-G. Talbi. Adaptive mechanisms for multi-objective evolutionary algorithms. In *IMACS multiconference, Computational Engineering in Systems Applications (CESA'03)*, IEEE Press, 2003.

[8] P.J. Bentley and J.P. Wakefield. *Soft Computing in Engineering Design and Manufacturing*. Springer Verlag, 1997. Chapter Finding acceptable Pareto-optimal solutions using multiobjective Genetic Algorithms.

[9] S. Bleuler, A. Preli, and E. Ziztler. An EA framework for biclustering of gene expression data. In *Congress on Evolutionnary Computation (CEC'04)*, IEEE Press, pages 166–173, 2004.

[10] R.L. Carraway, T.L. Morin, and H. Moskowitz. Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research*, 44 :95–104, 1990.

[11] V. Chantreau. Approche multi-critère hybride pour les règles d'association : Application l'analyse des données de puces à adn. Master's thesis, Université des Sciences et Technologies de Lille, 2004.

[12] C.A. Coello Coello. Using a min-max method to solve multiobjective optimization problems with genetic algorithms. In *IBERAMIA'98*, pages 303–314, 1998. LNCS vol 1993, Springer Verlag.

[13] C.A. Coello Coello and G.B. Lamont. *Applications of Multi-Objective Evolutionary Algorithms*. World Scietific Publishing co., 2004.

[14] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont, editors. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Press, 2002.

[15] C. Cotta, J.F. Aldana, A.J. Nebro, and J.M. Troya. Hybridizing genetic algorithms with branch and bound techniques for the resolution of the TSP. In R.F. Albrecht D.W. Pearson, N.C. Steele, editor, *Artificial Neural Nets and Genetic Algorithms 2*, pages 277–280. Springer-Verlag, Wien New York, 1995.

[16] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley and sons, 2001.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm : NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6(2) :182–197, 2002.

[18] Société DECISIA, editor. *SPAD 5.5*. 2002.

[19] F. Degoutin and X. Gandibleux. Un retour d'expérience sur la résolution de problèmes combinatoires bi-objectifs. Programmation Mathématique MultiObjectif (PM2O), mai 2002.

[20] C. Dhaenens. Recherche opérationnelle et optimisation : quelles perspectives pour le data mining. In *Sixième congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF'05)*, pages 9–15, 2005. Invited speaker.

[21] F.Y. Edgeworth. *Mathematical Physics*. P. Keagan, London, 1881.

[22] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR specktrum*, 22 :425–460, 2000.

[23] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of National Academy of Sciences,95(25) :14863-8*, 1998.

[24] G. Fogel and D. Corne. *Evolutionary Computation in Bioinformatics*. Morgan Kaufmann, 2002.

[25] C. Fonseca and P. Fleming. Multiobjective genetic algorithms made easy : selection, sharing and mating restrictions. In *IEEE Int. Conf On Genetic Algorithms in Engineering System : Innovations and Applications*, pages 45–52, Scheffield, UK, 1995.

[26] C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1) :1–16, 1995.

[27] A.A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer, 2001.

[28] A.A. Freitas. *Advances in evolutionary computing : theory and applications*, chapter A survey of evolutionary algorithms for data mining and knowledge discovery, pages 819–845. Springer-Verlag New York, Inc., 2003.

[29] K. Fujita, N. Hirokawa, S. Akagi, S. Kimatura, and H. Yokohata. Multi-objective optimal design of automotive engine using genetic algorithm. In *ASME Design Engineering Technical Conferences (DETC'98)*. Lawrence Erbaum, 1987.

[30] A.M. Geoffrion. Proper efficiency and the theory of vector minimization. *Journal of Mathematical Analysis and Applications*, 22 :618–630, 1968.

[31] D.E. Golberg. *Genetic algorithms in Search, Optimization and Machine learning*. Addison-Wesley Publishing Company, 1989. Reading, Massachussets.

[32] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimisation. In *ICGA'2, Second Int. Conf on Genetic Algorithms*, pages 41–49. Lawrence Erbaum, 1987.

[33] Y. Haimes, L. Ladson, and D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on System, Man and Cybernetics*, 1 :296–297, 1971.

[34] J. Handl and J. Knowles. Exploiting the trade-off - the benefits of multiple objectives in data clustering. In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume LNCS 3410, pages 547–560. Springer-Verlag, 2005.

[35] M.P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations of non dominated set. Tech. Rep., Institute of Mathematical modeling, Tech. Univ of Denmark, 1998. IMM Tech. Rep. IMM-REP-1998-7.

[36] A. Hertz, B. Jaumard, C.C. Ribeiro, and W.P. Formosinho Filho. A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO Operations Research*, 28(3) :303–328, 1994.

[37] J.H. Holland. *Adaptation in natural and artificial systems*. Michigan Press Univ, 1975.

[38] T.P. Hong, H. Wang, and W. Chen. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of heuristics*, 6 :439–455, 2000.

[39] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *First IEEE Conference on Evolutionnary Computation*, IEEE Press, pages 82–87, 1994.

[40] 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, Inc., 2002.

[41] C.A.R. Jahuira and E.C. Vargas. Solving the TSP by mixing GAs with minimal spanning tree. *Sociedad Peruana de Computacion*, II-3 :123–133, 2003.

[42] L. Jourdan. *Métaheuristiques pour l'extraction de connaissances : Application à la génomique*. PhD thesis, Université des Sciences et Technologies de Lille, November 2003.

[43] L. Jourdan, C. Dhaenens, and E-G. Talbi. ASGARD : un algorithme génétique pour les règles d'association. *Extraction de Connaissances et Apprentissage (ECA - Hermès)*, 16(6) :657–683, 2002.

[44] 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 LNCS 2611, pages 66–75, Colchester, England, UK, avril 2003. Springer-Verlag.

[45] 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)*, pages 29–32, Paris, France, 2003.

[46] L. Jourdan, C. Dhaenens, E.G. Talbi, and S. Gallina. A data mining approach to discover genetic and environmental factors involved in multifactoral diseases. *Knowledge Based Systems (Elsevier)*, 15(4) :235–242, 2002.

[47] N. Jozefowiez. *Modélisation et résolution approchée de problèmes de tournées de véhicules*. PhD thesis, Université des Sciences et Technologies de Lille, December 2004.

[48] H. Kargupta and P. Chan, editors. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press - The MIT Press, 2000.

[49] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives : preferences and value rtadeoff*. Cambridge University Press, 1993.

[50] M. Khabzaoui, C. Dhaenens, A. N'Guessan, and E-G. Talbi. Etude exploratoire des critères de qualité des règles d'association en datamining. In *Journées Françaises de Statistique*, pages 583–587, 2003.

[51] M. Khabzaoui, C. Dhaenens, and E-G. Talbi. A Multicriteria Genetic Algorithm to analyze DNA microarray data. In *Congress on Evolutionary Computation (CEC'04)*, volume II, pages 1874–1881, Portland, USA, juin 2004. IEEE Service center.

[52] J.D. Knowles and D.W. Corne. Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary Computation Journal*, 8(2) :149–172, 2000.

[53] J.D. Knowles and D.W. Corne. On metrics for comparing non-dominated sets. In *Congress on Evolutionnary Computation (CEC'02)*, IEEE Press, pages 711–716, 2002.

[54] P. Kotala, P. Zhou, S. Mudivarthy, W. Perrizo, and E.Deckard. Gene expression profiling of DNA microarray data using peano count trees. In *Online Proceedings of the First Virtual Conference on Genomics and Bioinformatics. URL : http ://midas-10.cs.ndsu.nodak.edu/bio/.*, 2001.

[55] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3) :263–282, 2002.

[56] L. Lebart, A. Morineau, and M. Piron. *Statistique Exploratoire Multidimensionnelle*. Dunod, Paris, 1995.

[57] J. Lemesre, C. Dhaenens, and E-G. Talbi. A parallel exact scheme to solve bicriteria problems. In *Multiobjective Programming and Goal Programming (MOPGP'04)*, Hammamet, Tunisia, 2004.

[58] J. Lemesre, C. Dhaenens, and E.G. Talbi. An exact parallel method for a bi-objective permutation flowshop problem. *European Journal of Operational Research*, To appear, 2005.

[59] M.H. Mabed, M. Rahoual, E-G. Talbi, and C. Dhaenens. Algorithmes génétiques pour les problèmes de flow-shop. In *3ème conférence Francophone de MOdélisation et SIMulation (MOSIM)*, pages 843–849. Troyes, France, 2001.

[60] S. Mardle, S. Pascoes, and M. Tamiz. An investigation of genetic algorithm for the optimization of multi-objective fisheries bioeconomic models. *International Transaction of Operation research*, 7 :33–49, 2000.

[61] M.L. Mauldin. Maintaining diversity in genetic search. In *Nat. Conf. on artificial intelligence*, page 247, 1984.

[62] P. Merz. Clustering gene expression profiles with memetic algorithms. In *7th international conference on Parallel problem Solving from Nature (PPSN VII)*, pages 811–820. LNCS, 2002.

[63] N. Monmarché, C. Guinot, and G. Venturini. Fouille visuelle et classification de données par nuage d'insectes volants. *Extraction de Connaissances et Apprentissage (ECA - Hermès)*, 16(6) :729–752, 2002.

[64] M.A. Muharram and G.D. Smith. The effect of evolved attributes on classification algorithms. In T.D. Gedeon and L.C.C. Fung, editors, *AI 2003, Advances in Artificial Intelligence, 16th Australian Conference on AI*, LNAI, no 2903, pages 933–941. Springer, 2003.

[65] M.A. Muharram and G.D. Smith. Evolutionary feature construction using information gain and gini index. In *7th European Conf. on Genetic Programming, EuroGP 2004, Portugal*, LNCS, no 3003, pages 379–388. Springer, 2004.

[66] T. Murata and H. Ishibuchi. A multi-objectives genetic loacal search algorithm and its application flow-shop scheduling. *IEEE Transaction System*, 28(3) :392–403, 1998.

[67] V. Pareto. *Cours d'économie politique*. Rouge, Lausanne, 1896.

[68] N. Pech-Gourg and J.-K. Hao. Métaheuristiques pour la classification de bouchons naturels en liège. *Extraction de Connaissances et Apprentissage (ECA - Hermès)*, 16(6) :785–806, 2002.

[69] F. Picarougne, C. Fruchet, A. Oliver, N. Monmarché, and G. Venturini. Recherche d'information sur internet par algorithme génétique. In *Quatrième congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF'02)*, pages 175–151, Paris, France, 2002.

[70] A. Przybylski, X. Gandibleux, and M. Ehrgott. Seek and cut algorithm computing minimal and maximal complete efficient solution sets for the biobjective assignment problem. In *6th Int. Multi-Objective Programming and Goal Programming conf (MOPGP'04)*, 2004.

[71] B. Roy and D. Bouyssou. *Aide multicritère à la décision : méthodes et cas*. Economica, 1993.

[72] E. Sandgren. *Advances in design optimization*. Chapman and Hall, 1994. chapter Multicriteria design optimization by goal programming.

[73] J.D. Schaffer. Multiple objective optimisation with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *ICGA Int. Conf on Genetic Algorithms*, pages 93–100. Lawrence Erbaum, 1985.

[74] N. Speer, C. Spieth, and A. Zell. A memetic co-clustering algorithm for gene expression profiles and biological annotation. In *Congress on Evolutionnary Computation (CEC'04)*, IEEE Press, pages 1631–1638, 2004.

[75] M.K. Sreenivas, K. AlSabti, and S. Ranka. *Advances in Distributed and Parallel Knowledge Discovery*, chapter Parallel Out-Of-Core Decision Tree Classifiers, pages 319–338. AAAI Press - The MIT Press, 2000.

[76] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetics algorithms. *Evolutionary Computation*, 2(3) :221–248, 1994.

[77] B.S. Stewart and C.C. White. Multiobjective A*. *Journal of the ACM*, 38(4) :775–814, 1991.

[78] E-G. Talbi, M. Rahoual, M.H. Mabed, and C. Dhaenens. New genetic approach for multicriteria optimization problems : Application to the flow shop. In *Evolutionary Multi-criterion Optimization (EMO)*, volume LNCS 1993, pages 416–428. Zurich, Switzerland, 2001.

[79] P-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eight ACM SIGKDD conference, Edmonton, Canada*, 2002.

[80] P-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Eight ACM SIGKDD conference*. Edmonton, Canada, 2002.

[81] V. T'Kindt and J.C. Billaut. *L'ordonnancement multicritère*. Presse Universitaire de Tours, 2000.

[82] E.L. Ulungu and J. Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundation of computing and decision science*, 20 :149–156, 1995.

[83] A. Unwin. Visualisation for data mining. In *International Conference on Data Mining, Visualization and Statistical System*, Séoul, Korea, 2000.

[84] D. Van Veldhuizen. *Multiobjective Evolutionary Algorithms : Classifications, Analyses, and nex Innovations*. PhD thesis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[85] L. Vermeulen-Jourdan, C. Dhaenens, and E.G. Talbi. A parallel adaptive genetic algorithm for linkage disequilibrium in genomics. In *Workshop on Nature Inspired Distributed Computing (NIDISC'2004)*, pages 29–32, Santa Fe, USA, 2004. IEEE IPDPS. 8 pages.

[86] P.C. Wong, P. Whitney, and J. Thomas. Visualizing association rules for text mining. In *INFOVIS '99 : Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 120, Washington, DC, USA, 1999. IEEE Computer Society.

[87] K.Y. Yeung and W.L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17 :763–774, 2001.

[88] M.J. Zaki. *Advances in Distributed and Parallel Knowledge Discovery*, chapter Hierarchichal Parallel Algorithms for Association Mining, pages 339–376. AAAI Press - The MIT Press, 2000.

[89] M.J. Zaki and C.T. Ho, editors. *Large-Scale Parallel Data Mining*, volume 1759 of *LNAI*. Springer, 2000.

[90] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization : Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[91] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms : A comparative case study and the strenth pareto approach. *IEEE Trans. on Evolutionary Computation*, 3(4) :257–271, 1999.

[92] E. Zitzler, L. Thiele, M. Laumanns, C.M Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers : an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2) :618–630, 2003.

Troisième partie

Annexes

Curriculum Vitae Détaillé

Curriculum Vitae synthétique

Adresse professionnelle
Laboratoire LIFL
Bâtiment M3
59655 Villeneuve d'Ascq cedex
dhaenens@lifl.fr
www.lifl.fr/∼dhaenens

Adresse personnelle
2, rue Nelson Mandela
59790 Ronchin (F)

Etat civil
Née le 28/11/1972
à Tourcoing (59)
Mariée
2 enfants (2000, 2001)

Maître de conférences en informatique
Ecole Polytechnique Universitaire de Lille (Polytech'Lille - Ex EUDIL)
Laboratoire d'Informatique Fondamentale de Lille (L.I.F.L)

Formation

1995-1998 Doctorat à l'INP-Grenoble. Monitrice de l'enseignement supérieur.
Thèse soutenue le 30 octobre 1998 : Mention très honorable avec félicitations.
Laboratoire Leibniz-IMAG (Grenoble - 38).
Sujet : "Optimisation d'un réseau de production et de distribution"
Directeurs : M. Lionel Dupont (Professeur 61^e),
M. Gerd Finke (Professeur 27^e).
Jury : M. C. Proust, M. P. Dejax, M. A. Guinet, M. J.-Y. Talon

1994-1995 DEA de Recherche Opérationnelle - Grenoble - Mention Très Bien (1/30).

1991-1994 EUDIL (École Universitaire D'Ingénieurs de Lille - 59).
Département Informatique-Mesures-Automatique, spécialité Informatique.

93-94 Échange ERASMUS - University of SUSSEX at Brighton (U.K.).
School of Engineering & School of Cognitive Science.

Activités d'enseignement - Récapitulatif

2002 → Titulaire de la prime d'encadrement et de recherche (PEDR).

1999 → Maître de conférences en informatique à Polytech'Lille.
Enseignements en **Recherche Opérationnelle** (Théorie des graphes, RO, Ordonnancement, Gestion de production,..) et en **informatique** (Algorithmique, projets bases de données...).
Mise en place de certains de ces cours pour la filière G.I.S. (Génie Informatique et Statistique) ouverte en 1999.

1998-1999 Attaché Temporaire d'Enseignement et de Recherche (1/2 poste d'A.T.E.R) à l'ENSGI (École Nationale Supérieure de Génie Industriel) - INP Grenoble.
Enseignements en **informatique** et en **gestion de production**.

1995-1998 Monitrice en informatique à l'Université Joseph Fourier, Grenoble.

Langues

Anglais	Parlé, lu et écrit couramment. 93-94 : Troisième année d'école d'ingénieurs passée à l'University of Sussex at Brighton (U.K.).
Allemand	Parlé, lu et écrit. 1993 : Obtention du Goethe Zertifikat, avec mention bien.

Principales responsabilités

2006 →	Secrétaire de l'association ROADEF (Association Française de Recherche Opérationnelle et d'Aide à la Décision).
2005-2006	Co-organisatrice de la conférence MOPGP'06 (Tours, juin 2006) - Édition du livre des résumés et publications des articles longs (LNEMS - Springer).
2005-2006	Co-organisatrice de la conférence ROADEF'06 (Lille, février 2006) - Organisation générale et responsable du budget.
2005	Co-organisatrice de la conférence EA'05 (Lille, octobre 2005) - Organisation générale et responsable du budget.
2004 →	Animatrice du groupe de travail national PM2O (Programmation Mathématique MultiObjectif).
2004 →	Membre du Comité de Direction du GDR I3 (Information - Interaction - Intelligence).
2004 →	Membre du conseil pédagogique de Polytech'Lille.
2003 →	Membre élu du conseil du laboratoire du L.I.F.L.
2002 →	Animatrice scientifique du Centre Intégré de Bioinformatique (CIB) de la génopole de Lille.
2002 →	Membre de la commission communication du L.I.F.L.
2000 →	Membre titulaire de la CSE 27e section de L'USTL (Université de Lille I). Réélue en 2004.
2000-2003	Responsable des stages de la filière GIS de Polytech'Lille.
1999 →	Coordinatrice des relations internationales au sein de la filière GIS de Polytech'Lille.
1998-1999	Co-organisatrice de la conférence ROADéF'99 (200 participants, 140 présentations) - Élaboration du programme et édition du livre des résumés.
1996-1999	Représentante des nons-permanents au conseil du laboratoire Leibniz-IMAG.
1996-1998	Membre du conseil pédagogique du CIES - Centre d'Initiation à l'Enseignement Supérieur - de Grenoble I (Université Joseph Fourier).
1993-1994	Vice présidente communication de la Junior Entreprise de l'EUDIL.
1992-1993	Membre du conseil d'administration de l'EUDIL.

Enseignements et responsabilités pédagogiques

Cadre général

Mon arrivée au sein de **Polytech'Lille** (Ex EUDIL) a coïncidé avec la création de la nouvelle filière **G.I.S.** (Génie Informatique et Statistique). C'est donc dans ce contexte que j'ai commencé mes enseignements. L'équipe pédagogique d'informatique existait déjà (de par l'existence du département Informatique-Mesures-Automatique), mais certaines orientations allaient être différentes dans ce nouveau département.
Dès mon arrivée, j'ai eu la responsabilité de mettre en place de nouveaux cours tels que, "Graphes et combinatoire" (première année), "Recherche Opérationnelle" (deuxième année). De plus, j'ai pris en charge la **gestion des stages** (concernant essentiellement les 2^e et 3^e années) et suis la correspondante pour les **relations internationales** au sein du département. Pour ces deux charges, une prime pédagogique m'a été accordée pour l'année 2001-2002.

Détails des principaux cours dispensés récemment

Ci-dessous sont décrits les principaux cours que j'ai été amenée à dispenser et pour lesquels j'ai quelques responsabilités. Bien sûr cette liste n'est pas exhaustive et ne tient pas compte, par exemple, des différents encadrements de projets et stages que j'ai pu effectuer.

Graphes et combinatoire

Public :	GIS1 - $1^{ère}$ année d'école d'ingénieurs.
Horaire étudiant :	20 h de Cours, 18h de TD, 10h de TP.
Responsabilité :	Responsable de l'ensemble du module. Réalisation d'un polycopié de 70 pages. Rédaction des sujets de TD et de TP.
Contenu :	Concept de graphe, représentation, cheminement-connexité, arbres/arborescences, plus court chemin, ordonnancement simple, flot maximum.

Recherche opérationnelle

Public :	GIS2 - $2^{ème}$ année d'école d'ingénieurs.
Horaire étudiant :	20 h de Cours, 14h de TD, 14h de Tutorat.
Responsabilité :	Ce cours est divisé en deux parties. Responsable de la deuxième partie (10 h de Cours, 8h de TD) et du tutorat. Réalisation d'un polycopié de 45 pages. Rédaction des sujets de tutorat.
Contenu :	$2^{ème}$ partie : Dualité et analyse de sensibilité, problème de transport, programmation linéaire en nombres entiers, programmation dynamique, programmation par but, satisfaction de contraintes.

Datamining

Public :	GIS3 - $3^{ème}$ année d'école d'ingénieurs.
Horaire étudiant :	(15 h de Cours,) 10h de TP.
Responsabilité :	Chargée de TP. Mise en place des TP et rédaction des sujets.
Contenu :	Manipulation de logiciels (Weka, Sipina). Développement d'algorithmes classiques.

Optimisation Combinatoire

Public :	IMA2 - $2^{ème}$ année d'école d'ingénieurs.
Horaire étudiant :	8 h de Cours, 8h de TD.
Responsabilité :	Responsable de l'ensemble du module Réalisation d'un polycopié de 55 pages.
Contenu :	Concept de graphes, cheminement connexité, arbres/arborescences, plus court chemin, ordonnancement simple (PERT), notions de complexité, algorithmes exacts, méthodes heuristiques.

Gestion de production

Public :	Option transversale - $3^{ème}$ année d'école d'ingénieurs.
Horaire étudiant :	24 h de Cours-TD.
Responsabilité :	Responsable de l'ensemble du module. Réalisation d'un polycopié de 70 pages.
Contenu :	Gestion des stocks, planification de la production, programmation linéaire en planification, méthodes MRP, méthode Kanban, ordonnancement de projet.

Responsabilités pédagogiques

Stages

De 2000 à 2003 j'ai été responsable des stages pour la filière GIS (stages des $2^{ème}$ et $3^{ème}$ années). Cette responsabilité consistait à :
– aider les étudiants dans leur recherche de stage,
– valider les sujets,
– répartir les stages entre les tuteurs de l'école,
– planifier les soutenances (sessions de juin et septembre),
– animer le jury de stage.
Bien sûr il convient également de gérer les problèmes particuliers pouvant arriver en stage.

Dans le cadre de cette reponsabilité, j'ai participé à un groupe de travail sur les stages au niveau de l'école et ai contribué, de façon majoritaire, à la rédaction d'un guide des stages

au sein de Polytech'Lille.

Relations internationales

Depuis 1999, je suis coordinatrice des relations internationales pour la filière GIS. Ainsi, avec l'appui du service des relations internationales de l'école, nous proposons aux étudiants de faire soit un de leur stage à l'étranger, soit tout ou une partie de leur $3^{ème}$ année. Chaque année quelques étudiants choisissent cette voie. Ainsi, 5 éudiants sont partis en 2001-2002, 2 en 2002-2003, 4 en 2003-2004, 6 en 2004-2005 et 8 le projettent pour 2005-2006.
Etre coordinatrice consiste principalement à épauler les étudiants dans leur recherche de l'université d'accueil, à valider les programmes d'études suivis, à les aider dans leurs démarches administratives (avec le service des relations internationales) et à faciliter leur retour et leur ré-intégration dans le cursus français.

Dans le cadre de cette responsabilité, une reflexion importante à propos de la mise en place des crédits ECTS (*European Credit Transfer and Accumulation System*) est menée au sein de l'école. Je participe activement à la commission en charge de ce dossier.

Encadrements

L'encadrement d'étudiants fait intégralement partie de la tâche de l'enseignant-chercheur, qu'il s'agisse d'étudiants de $3^{ème}$ cycle (Thèse de doctorat, de master) ou bien de $2^{ème}$ cycle (Ecole d'ingénieur, maîtrise...).

Thèses de doctorat (3)

2000-2003 **Laetitia Jourdan**, *"Métaheuristiques pour l'extraction de connaissances : application à la génomique"*, soutenue en novembre 2003, encadrement à 50%, E-G. Talbi 50% (METMBS'01, MIC'01, KBS'02, ECA'02, JOBIM'02, EvoBio'03, ECCB'03, NIDISC'04, EvoCop'04, IJFCS'05).

Résumé : Le travail présenté dans cette thèse traite de l'extraction de connaissances à l'aide de métaheuristiques et de ses applications à des problématiques en génomique. Après un état de l'art sur les métaheuristiques utilisées pour l'extraction de connaissances, sont présentées deux problématiques issues d'une collaboration avec l'Institut de Biologie de Lille autour de la recherche de facteurs génétiques de prédisposition à certaines maladies multifactorielles (diabète de type II, obésité). Une modélisation de ces problèmes en problèmes d'extraction de connaissances est proposée. Puis, les différentes tâches d'extraction de connaissances identifiées comme des problèmes d'optimisation sont traitées et un schéma d'algorithme génétique possédant des mécanismes avancés d'intensification et de diversification pour les résoudre est proposé. Des connaissances du domaine biologique sont intégrées afin de répondre aux problématiques posées. Cette intégration s'effectue aussi bien au niveau des fonctions d'évaluation proposées qu'au niveau de certains mécanismes utilisés. Enfin, différents modèles de parallélisme sont utilisés.

2002- ? ? **Mohammed Khabzaoui**, *"Modélisation et résolution multi-objectifs des règles d'association : application à l'analyse de données biopuces"*, 50%, E-G. Talbi 50% (ROADEF'03, JDS'03, SIAM-KDD'04, CEC'04, ROADEF'05).

Résumé : Le dévelopement de l'informatique facilite l'acquisition de données en très grand nombre. La difficulté ne réside donc plus dans l'acquisition des données mais dans leur analyse. C'est l'objectif des problèmes d'extraction de connaissances qui cherchent à décrire le comportement actuel et/ou futur d'un procédé. Une de ces problématiques, concerne la recherche de règles d'associations qui consiste à extraire un ensemble de formules logiques conditionnelles permettant de déduire la valeur d'un attribut but à partir des valeurs d'autres attributs.

La plupart des méthodes développées à ce jour pour la recherche de règles associatives se basent, pour mesurer la qualité des règles, sur un critère unique alors qu'il existe plusieurs critères pertinents et aucun consensus sur le meilleur critère. Une modélisation multi-objectif va permettre de combiner différents critères antagonistes et de trouver des règles explicatives plus intéressantes et complètes.
Le terrain d'application concerne l'analyse de données issues de biopuces (puces à ADN). La recherche de règles d'association au sein de ces données permettra de trouver des relations liant certains gènes. Pour cela des schémas de résolution parallèles hybridant algorithmes évolutionnaires et approches exactes sont proposés.

2003- ? ? **Julien Lemesre**, *"Méthodes parallèles exactes pour l'optimisation multi-objectif"*, 50%, E-G. Talbi 50% (PMS'04, MOPGP'04, ROADEF'05, CIRO'05, EJOR).
Résumé : Les problèmes de décision s'appuyant sur plusieurs critères rencontrent une attention sans cesse croissante.
Ici, nous nous intéressons à la conception et au développement de méthodes exactes capables de générer l'ensemble des solutions efficaces (optimales au sens de Pareto) pour des problèmes d'optimisation discrets multi-objectifs.
Le fait de rechercher un ensemble de solutions en travaillant dans plusieurs sous espace décisionnels fait penser à la mise en place d'un processus de génération parallèle et coopératif. Une approche fondée sur l'utilisation des algorithmes parallèles semble particulièrement adaptée. Le but n'est pas d'utiliser le parallélisme en vue seulement de produire des résultats non encore atteints par une algorithmique séquentielle, mais d'exploiter la propriété nativement parallèle du problème de génération de l'ensemble des solutions.
Il ne semble pourtant pas réalisable de résoudre les problèmes de très grandes tailles, avec cette seule approche. Une réflexion intéressante doit donc porter sur les possibilités de coopération entre méthodes exactes et (méta-) heuristiques, ces dernières permettant en effet de traiter des problèmes de grandes tailles, mais pas toujours de façon optimale. Allier la force d'exploration des métaheuristiques et le pouvoir d'obtention des meilleures solutions des méthodes exactes devrait permettre de mieux résoudre les problèmes de grandes tailles.

Mémoires de DEA (8)

1997-1998 Haris Gavranovic, DEA de ROCO-IMAG, Grenoble, *"Partitionnement en cliques pour le poinçonnage de tôles"*, 30%, M-L. Espinouse 30%, N. Brauner 30%, G. Finke 10% (FRANCORO'98, CO'98, ROADEF'99).

Denis Lheytienne, DEA de ROCO-IMAG, Grenoble, *"Ordonnancement préemptif sur machines parallèles non liées en vue de minimiser le flot moyen"*, 100%.

1998-1999 Karima Skiba, DEA de ROCO-IMAG, Grenoble, *"Ordonnancement préemptif sur machines identiques en vue de minimiser le flot moyen"*, 100%.

1999-2000 Laetitia Jourdan, DEA Informatique, Lille, *"Datamining pour la bio-informatique"*, 50%, E-G. Talbi 50% (EMGM'2001).

2000-2001 Grégory Vermeersch, DEA Informatique, Lille, *"Algorithmes génétiques pour la bio-informatique"*, 80%, E-G. Talbi 20%.

2001-2002 Mohammed Khabzaoui, DEA Informatique, Lille, *"Algorithmes évolutionnaires pour la recherche de règles associatives : application à la génomique"*, 50%, E-G. Talbi 50%.

2002-2003 Julien Lemesre, DEA Informatique, Lille, *"Algorithmes parallèles exacts pour l'optimisation multi-objectif"*, 50%, E-G. Talbi 50%.

2003-2004 Vanessa Chantreau, DEA Informatique, Lille, *"Approche multi-critère hybride pour les règles d'association : Application à l'analyse des données de puces à ADN"*, 30%, L. Jourdan 40%, E-G. Talbi 30%.

2004-2005 Julien Garet, Mastère recherche Informatique, Lille, *"Définition et optimisation des zones de localisation d'un réseau de téléphonie mobile"*, 50%, E-G. Talbi 50%.

Projets étudiants

2001	Polytech'3	N. Nachit et G. Le-Goff, *"Application des algorithmes génétiques pour le problème de Clustering"*, co-encadrement avec L. Jourdan
2002	IUP3 - Calais	J-M. Wyngaert et S. Achiba, *"Analyse génomique par Datamining"* , co-encadrement avec L. Jourdan, N. Melab.
2002	IUP3 - Lille	C. Arbelaiz et S. Lardjoun, *"Interface graphique pour la génomique"*, co-encadrement avec L. Jourdan.
2002	DESS Bioinformatique	J. Eteve, *"Evaluation d'un outil de datamining pour la recherche d'interaction entre gènes et facteurs d'environnement"*, co-encadrement avec L. Jourdan, S. Gallina (IBL).
2002	Polytech'3	J-B. Bavugilije, *"Création d'une base de données Oracle et d'une application d'aide à la réalisation de schémas de puces à ADN"*.
2002	Polytech'2	F. Blondel, *"Sur l'utilisation de l'algorithme Apriori"*.
2002	IUP3 - Lille	T. Longuemart, A. Wasson, *"Etude des critères d'évaluation des règles d'association"*, co-encadrement avec L. Jourdan.
2003	Polytech'3	H. Bendali, J. Gallant, G. Tyrou, *"BibClust : Bibliothèque de méthodes de clustering"*, co-encadrement avec L. Jourdan.
2003	Polytech'3	S. Dederen, T. Lam, *"Chaines de Markov cachées : Théorie et Application"*.
2003	DESS IAGL	D. Delautre, S. Demay, *"Visualisation multicritère de règles d'association"*, co-encadrement avec L. Jourdan.
2003	Polytech'2	J-P. Nirel, *"Approche bi-critère du Covering Tour Problem"*.
2003	IUP3 - Lille	Y. Delalande, G. Westrelin, *"Etude de critères du clustering"*, co-encadrement avec L. Jourdan.
2004	Polytech'3	M. Roussel, Y. Fourdrain, *"Challenge ROADEF'2005 : The car sequencing problem"*, co-encadrement avec L. Jourdan.
2004	Polytech'3	M. Kejeiri, *"Méthode exacte pour la recherche de règles d'association multicritères"*.
2005	Polytech'3	E. Larose, B. Leclercq, *"Manipulation de données hétérogènes en génomique"*.

Remarque : Polytech'3 indique troisième année du cycle ingénieur (BAC+5) de l'école Polytechnique Universitaire de Lille (Polytech'Lille).

Participation à des projets et administration de la recherche

Participation à des projets

• **Porteuse** d'un projet JEMSTIC (*"Méthodes d'optimisation pour l'extraction de connaissances en génétique"*), soutenu en 2001 et 2002 par le département STIC du CNRS.

• **Participation** au projet MOST (*"Méthodologies pour l'Optimisation dans les Systèmes de Transport et de Télécommunications"*) (2000-2003) de l'opération TACT du Contrat Plan Etat Région.

• **Participation** au projet NOMéBIO (*"Nouvelles méthodologies Bioinformatiques pour les pathologies multifactorielles et pour la protéomique*) (2001-2002) du Contrat Plan Etat Région (Génopole de Lille).

• **Participation** à un projet GENHOMME (*"Plate-forme d'extraction de connaissances à partir de données hétérogènes d'intérêt pour les maladies cardio-vasculaires"*) (2002-2003) du ministère de la recherche avec la société IT-OMICS.

• **Participation dans le cadre du CIB** à une ACI NanoScience (*"Puces Nano3D"*) (2004-2006) du ministère de la recherche sur le design expérimental des puces, en collaboration de l'IEMN (Institut d'Electronique et de Microélectronique et de Nanotechnologie), de l'Institut de Biologie de Lille et de l'Institut Pasteur de Lille.

• **Participation à une soumission** (coordinatrice pour l'université de Lille) d'un projet Européen STREPS - Fet open. (*"Surprise : Bio-Inspired Algorithms for Data Mining with Controlled Surprise from biological Data"*) en collaboration avec l'université de Malaga (Espagne), l'université Libre d'Amsterdam (Pays Bas), l'université du Kent (UK), l'université de Newcastle (Australie) et l'université de Paris-Sud.

Participation à des groupes de travail

• Membre du groupe de travail **Bermudes** (Groupe de travail "Ordonnancement"), groupe de travail du GDR MACS.

• Membre du groupe de travail **Gotha** (Groupe de recherche en Ordonnancement Théorique et Appliqué).

• Membre du groupe de travail **META** (Métaheuristiques : Théorie et Applications), groupe de travail du GDR MACS.

• Depuis 2004, animatrice du groupe **PM2O** (Programmation Mathématique Multi-Objectif), GT 1.6 de l'axe 1 du GDR I3 également soutenu par la ROADEF.

Administration de la recherche

• **Organisation d'une session** sur le thème de *Méthodes d'optimisation pour l'extraction de connaissances*, lors de la conférence *ROADEF'02 : quatrième congrès de la Société Française de Recherche Opérationnelle et Aide à la décision*. 4 papiers sélectionnés.

• **Editeur invité** de la revue *"Extraction de Connaissances et Apprentissage" - Hermes* pour l'organisation d'un numéro sur *Méthodes d'optimisation pour l'extraction de connaissances*. 7 articles sélectionnés. Volume 16, numéro 6, 2002.

• **Organisation d'une session** sur le thème de *Extraction de connaissances*, lors de la conférence *ROADEF2003 : cinquième congrès de la Société Française de Recherche Opérationnelle et Aide à la décision*. 5 présentations.

• **Organisation de sessions** sur le thème de *Programmation Multi-Objectif*, lors de la conférence *ROADEF2005 : sixième congrès de la Société Française de Recherche Opérationnelle et Aide à la décision*. 12 présentations.

• **Editeur invité** avec E-G. Talbi et P. Siarry de la revue *"RAIRO - OR"* pour l'organisation d'un numéro sur *Cooperative methods for multi-objective optimization*, 2005-2006.

• **Editeur invité** avec E-G. Talbi et F. Semet de la revue *"EJOR - European Journal of Operational Research"* pour l'organisation d'un numéro sur *Cooperative Combinatorial Optimization*, 2005-2006.

• Membre du jury de la thèse de B. Estève sur "Des problèmes d'ordonnancement multicritères de type juste-à-temps : Modélisation et résolution" - Laboratoire d'Informatique de Tours - juin 2005.

Activités administratives

• Co-organisation d'une journée du groupe de travail Bermudes (septembre 1997 à Grenoble).

• Responsable au sein de l'équipe Recherche Opérationnelle du Laboratoire LEIBNIZ de la **coordination de la rédaction du rapport d'activités** et de la mise en place des pages **WEB** présentant l'équipe (1999).

• Membre du **comité d'organisation de la conférence ROADEF'99** (**organisation du programme scientifique** (environ 140 présentations) et **édition du livre des résumés**).

• Co-organisation d'une journée de travail commune entre les groupes Bermudes et META (Février 2003 à Lille).

• Co-organisation d'une journée de travail commune entre les groupes META et GRID2 (Mars 2003 à Lille).

• Co-organisation d'une journée de travail commune entre les groupes PM2O et MCDA (Euro Working Group of Decision Aiding) (Avril 2004 à Brest).

• Co-organisation d'une journée de travail commune entre les groupes META et PM2O (Mai 2005 à Lille).

• Membre du **comité d'organisation de la conférence EA'05** (Evolution Artificielle /Artificial Evolution 2005) à Lille en octobre 2005 : **Organisation générale et budget**.

• Membre du **comité d'organisation de la conférence ROADEF'06** (septième congrès de la Société Française de Recherche Opérationnelle et Aide à la décision) à Lille en février 2006 : **Organisation générale et budget**.

Arbitrages

Différents arbitrages me sont régulièrement demandés.

– Soit pour des conférences nationales et internationales : EuroPar, International Conference on Artificial Evolution (EA), Congress on Evolutionary Computation (CEC), Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la décision (ROADEF), Metaheuristics International Conference (MIC)...

– Soit pour des journaux internationaux : European Journal of Operations Research (EJOR), Annals of Operations Research (AOR), Journal Européen des systèmes automatisés (JESA), Computers and Operations Research (CaOR), Parallel Computing, International Journal of Production Economics (IJPE), RAIRO Operations research (RAIRO-OR)...