Thèse de Jean-Charles Boisson, Lille 1, 2008

Université des Sciences et Technologies de Lille

Modélisation et résolution par métaheuristiques coopératives : de l'atome à la séquence protéique

THÈSE

présentée et soutenue publiquement le 8 décembre 2008

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

(spécialité informatique)

par

Jean-Charles BOISSON

(Obtenu avec la mention très honorable)

Composition du jury

Président :	Ralf Blossey, Directeur de recherche, Université des Sciences et Technologies de Lille
Rapporteurs :	Jin-Kao Hao, Professeur, Université d'Angers Pascal Bouvry, Professeur, Université du Luxembourg
Examinateur :	Pierre Tufféry, Directeur de recherche, Université Paris Diderot
Directeur de thèse :	El-Ghazali Talbi, Professeur, Université des Sciences et Technologies de Lille
Co-encadrant :	Laetitia Jourdan, Chargé de recherche, INRIA-Lille - Nord Europe

Laboratoire d'Informatique Fondamentale de Lille — UMR CNRS 8022

http://doc.univ-lille1.fr

Remerciements

Je tenais tout naturellement à remercier de prime abord mon directeur de thèse El-Ghazali Talbi et mon co-encadrant Laetitia Jourdan pour la confiance dont ils ont fait part à mon éguard ainsi que pour l'aide et les moyens nécessaires à la réalisation de mon travail de thèse qu'ils m'ont fourni. De la même manière, je remercie mes rapporteurs de thèse, les professeurs Jin-Kao Hao et Pascal Bouvry qui m'ont permis de soutenir ma thèse. Je remercie aussi le reste de mon jury, les directeurs de recherche Ralf Blossey et Pierre Tuffery pour leurs remarques et conseils en tant qu'expert sur les aspects non informatiques de mon travail.

Je tenais aussi à dire un grand merci aux permanents de l'équipe projet INRIA DOLPHIN pour leurs précieux conseils. Il est en de même pour les anciens et actuels thésards et ingénieurs de l'équipe. Tout le monde m'a apporté son aide, son soutien ou tout simplement sa bonne humeur. En outre, durant ces années, j'ai pu aussi compter sur le soutien et la confiance sans faille de ma famille et notamment de ma femme.

Enfin, j'ai eu aussi la chance de rencontrer et de collaborer avec de nombreux biologistes, chimistes, bioinformaticiens, … je les remercie pour leur conseils, critiques ou tout simplement sur leur retour concernant mon travail.

Encore merci à tous.

http://doc.univ-lille1.fr

http://doc.univ-lille1.fr

Je dédie cette thèse
à ma femme,
elle qui, compréhensive,
m'a épaulé toutes ces années.

http://doc.univ-lille1.fr

http://doc.univ-lille1.fr

Table des matières

Introduction **1**

Chapitre 1
Métaheuristiques coopératives pour l'optimisation combinatoire

1.1 Définitions 6
1.1.1 Optimisation mono-objectif 6
1.1.2 Complexité 6
1.1.3 Notion de paysage 8
1.1.4 Optimisation multi-objective 10
1.1.5 Méthodes d'optimisation 17
1.2 Les métaheuristiques à base de solution unique 20
1.2.1 La recherche par descente 21
1.2.2 La Recherche Tabou 22
1.2.3 Le recuit simulé 23
1.2.4 La recherche locale itérative 25
1.2.5 Le recherche à voisinage variable 25
1.2.6 Unification des concepts 27
1.3 Les métaheuristiques à base de population de solutions 27
1.3.1 Les algorithmes génétiques 27
1.3.2 La recherche par dispersion 28
1.4 La coopération de métaheuristiques 29
1.4.1 Notion de coopération 29
1.4.2 Niveaux de coopération 29
1.4.3 Modes de coopération 30
1.4.4 Classes hiérarchiques 31
1.5 ParadisEO-MO : une plateforme pour la conception de métaheuristiques à base de solution unique 34
1.5.1 Présentation de la plateforme ParadisEO 34

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.5.2 ParadisEO-MO 38
1.5.3 Synthèse 51
1.5.4 Perspectives 52
1.6 Conclusions 53

Chapitre 2
Nouveaux modèles pour l'identification de protéines et leur résolution par coopération de métaheuristiques

2.1 Définitions 56
2.1.1 Protéomique 56
2.1.2 Acides aminés et protéines 57
2.1.3 Spectrométrie de masse 59
2.2 Etat de l'art 61
2.3 Un nouveau modèle pour l'identification de protéine par empreinte de masse peptidique . 63
2.3.1 La digestion de protéine 63
2.3.2 La génération de spectres 66
2.3.3 Intégration au sein d'un nouveau moteur d'identification : ASCQ_ME 67
2.3.4 Validation de d'ASCQ_ME 71
2.4 SSO (Sequence, Shape, Order) : un nouveau modèle pour le *novo protein sequencing* . . 77
2.4.1 Définition du *de novo protein sequencing* 77
2.4.2 Présentation de SSO 77
2.4.3 S pour « **S**equence » 77
2.4.4 S pour « **S**hape » 82
2.4.5 O pour « **O**rder » 87
2.4.6 Validation de SSO 93
2.5 Conclusions et perspectives 96

Chapitre 3
Nouvelles modélisations multi-objectives pour le docking moléculaire flexible

3.1 Définitions 100
3.1.1 Docking moléculaire 100
3.1.2 Projet ANR Dock 100
3.1.3 interactions énergétiques 101
3.1.4 Notion de surface 105
3.1.5 Energie et champs de force 107
3.1.6 Notion de robustesse 109
3.1.7 Critère de comparaison de conformation : le RMSD 110

http://doc.univ-lille1.fr

3.2 Etat de l'art . . . 111
3.2.1 Classement des méthodes de résolution . . . 111
3.2.2 Modélisations multi-objectives . . . 111
3.2.3 Logiciels existants . . . 112
3.3 Nouvelles modélisations multi-objectives . . . 112
3.4 Etude du paysage énergétique autour des complexes optimaux . . . 113
3.4.1 Protocole de génération . . . 113
3.4.2 Paysages énergétiques avec le ICVFF . . . 114
3.4.3 Paysages énergétiques selon la fonction d'énergie d'Autodock 4.0 . . . 117
3.4.4 Comparaison . . . 119
3.5 Résolution par coopération co-évolutionnaire de bas niveau : AG multi-objectif parallèle hybride . . . 120
3.5.1 Représentation d'une solution . . . 120
3.5.2 Opérateurs . . . 121
3.5.3 Evaluation, sélection et remplacement . . . 124
3.5.4 Critère d'arrêt . . . 126
3.5.5 Parallélisation . . . 126
3.5.6 Validation de l'approche . . . 127
3.6 Docking@GRID . . . 149
3.6.1 Description de la plate-forme . . . 149
3.6.2 Intégration de l'algorithme de docking . . . 149
3.7 Conclusions . . . 150

Conclusions et perspectives **151**

Bibliographie **155**

Glossaire **163**

Annexes

Annexe A
Données complémentaires dédiées à ParadisEO-MO

A.1 Les algorithmes implémentés dans les *moAlgo* . . . 167
A.1.1 *moHC* . . . 167
A.1.2 *moTS* . . . 168
A.1.3 *moSA* . . . 169
A.1.4 *moILS* . . . 170

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

A.1.5 *moVNS* 171

Annexe B
Données complémentaires pour l'identification de protéines

B.1 Preuve de la complétude de l'arbre de digestion 173
B.2 Graphe de dépendance d'ADP 175
B.3 Graphe de dépendance d'ATP 176
B.4 Graphe de dépendance de SpecS 177
B.5 Graphe de dépendance de Ascq_me 178
B.6 Interface web d'Ascq_me 179
B.6.1 Page de soumission de requêtes 179
B.6.2 Page de résultats 180
B.7 Exemple d'une très grosse protéine biaisant les scores d'identification 180
B.8 Exemple de fichiers générés par Ascq_me 182
B.8.1 Fichier de configuration 182
B.8.2 Fichier « default_ascq_me_results.txt » 183
B.8.3 Fichier « default_ascq_me_table_results.txt » 186
B.8.4 Fichier « default_ascq_me_peptide_scoring.txt » 186
B.9 Schéma classique d'un « crossover 1-point » 188
B.10 Matrice de substitution BLOSUM62 189
B.11 Sélection roulette 189
B.12 Données expérimentales de validation de l'approche SSO 190
B.12.1 Séquences FASTA 190
B.12.2 Résultat d'identification de MASCOT (spectres MS/MS) 191

Annexe C
Données complémentaires pour le docking moléculaire

C.1 Longueur et énergie d'un ensemble de liaisons 195
C.2 Les 4 règles de Gillespie 195
C.3 Rayon de Van der Waals d'un ensemble d'élément 196
C.4 Paysages d'instances 197
C.4.1 Instance 2tsc 197
C.4.2 Instance 1htf 203
C.5 Graphes des dépendances pour les différentes parties du docking 209
C.6 Capture d'écran de Docking@GRID 212

http://doc.univ-lille1.fr

Introduction

La bioinformatique est un domaine en perpétuel changement. A chaque amélioration de technologie ou apparition de nouvelles machineries, la quantité et le type de données évoluent. Par conséquent, les méthodes de traitement associées se doivent de suivre ce mouvement. Même si au départ, il s'agissait juste de pouvoir accumuler des données afin de les exploiter directement, aujourd'hui il est nécessaire de mettre en place des méthodes complexes pour extraire l'information recherchée. Ces méthodes sont très coûteuses et il est généralement impossible de fournir des réponses en des temps raisonnables par des méthodes exhaustives. Par conséquent, plusieurs techniques peuvent être utilisées pour permettre d'obtenir des réponse à la plupart des problèmes qui se posent en bioinformatique :

1. la parallélisation (simple) des méthodes : cette approche permet d'accélérer les temps de traitement sans forcément devoir repenser l'approche séquentielle utilisée auparavant car il s'agit de répartir la charge de travail sur un ensemble de machines. Ce type d'approche fonctionne très bien pour l'exploitation de bases de données où la parallélisation peut tout simplement consister à découper la base complète en sous-bases distribuées à plusieurs machines.
2. La formalisation sous forme d'un problème d'optimisation combinatoire : le problème originel est ramené à un problème d'optimisation combinatoire. Ainsi on peut procéder à une analyse complète du problème, des données et des méthodes possibles à utiliser pour le résoudre. Par cette approche, l'utilisation de méthodes dites approchées permet d'obtenir une réponse satisfaisante dans des conditions acceptables là où les méthodes exhaustives ne sont pas réalisables en un temps raisonnable. Cependant, de par leur définition, ce type d'approche n'apporte aucune certitude sur le fait d'avoir trouver la meilleure solution à un problème donné.
3. La coopération de méthodes : ce principe est bien connu dans le monde de la bioinformatique où souvent la résolution de problèmes complexes revient à résoudre un ensemble de sous-problèmes. Mais il existe d'autres modèles de coopération venant du domaine de l'optimisation combinatoire qui permettent non seulement d'apporter une solution à des problèmes complexes mais aussi souvent d'en accélérer la résolution.

Dans ce manuscrit, nous mettons en évidence, à travers deux problèmes bioinformatiques liés à l'étude des protéines, l'intérêt de l'utilisation des approches développées en optimisation combinatoire. En effet, grâce à l'analyse d'un problème bien formalisé, il est possible de choisir les composants adéquats pour mettre en place des méthodes de résolution rapides et efficaces. Nous montrons que les métaheuristiques [1] coopératives sont un des meilleurs représentants de cette famille de méthodes. Ainsi nous détaillons, à travers les deux problèmes étudiés, le processus d'analyse qui nous permet d'arriver à cette conclusion. Cela va du choix de la représentation des solutions, de la mise en place du ou des critères d'évaluation et de l'analyse de ces critères afin de pouvoir choisir de manière la plus objective possible la ou les méthodes d'optimisation à utiliser. Une fois la méthode choisie, il reste encore le travail de paramétrisation de son comportement afin d'obtenir une résolution du problème qui soit efficace et si

[1] Désigne les méthodes approchées génériques i.e. non dédiées à un problème particuliers.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

possible la plus rapide possible. C'est dans ces objectifs que l'utilisation de la coopération de méthodes, notamment les métaheuristiques, et le parallélisme prennent tout leur sens.

Dans le **chapitre 1**, nous définissons ce qu'est un problème d'optimisation combinatoire en nous basant d'abord sur les problèmes à modélisation mono-objectif c'est-à-dire avec un seul critère de qualité à évaluer. Nous détaillons ensuite les notions de complexité et de paysage qui sont des critères de base pour choisir une méthode de résolution pour un problème donné. Les spécificités de comparaison liées aux modélisations multi-objectives sont ensuite présentées afin de mieux comprendre les difficultés de ce choix de modélisation qui semble cependant souvent mieux adapté à modéliser les problèmes réels. Nous proposons également une classification des méthodes de résolution en nous orientant exclusivement vers les métaheuristiques qu'elles soient à base de solution unique ou à population de solutions. En effet, dans ce manuscript, nous ne travaillons pas sur les méthodes dites exactes. Une fois les différents types de métaheuristiques présentés, nous définissons ce qu'est la coopération de métaheuristiques et les modèles de coopération qui y sont généralement associés. Enfin nous présentons ParadisEO-MO qui est une plateforme de développement pour les métaheuristiques à base de solution unique dont j'ai la responsabilité. L'historique de cette plateforme est rappelé avant de faire une comparaison entre la version de 2005 et celle actuelle à laquelle j'ai grandement contribué.

Le **chapitre 2** propose de nouvelles modélisations pour l'identification de protéines. Dans celui-ci, nous commençons par expliquer toutes les notions nécessaires à la bonne compréhension des méthodes d'identification de protéines afin de mieux appréhender notre positionnement par rapport à l'existant. De là, nous présentons la première partie du travail basée sur l'élaboration d'un nouveau modèle d'identification par empreinte de masse peptidique basé sur la création d'une nouvelle fonction d'évaluation permettant de comparer des protéines issues de bases de données avec des spectres masse/intensité obtenus de manière expérimentale. La validation de ce modèle est faite par l'intermédiaire d'un moteur d'identification de protéine, appelé ASCQ_ME, intégrant la nouvelle fonction de comparaison associée. Celle-ci est le point de départ de la seconde partie du chapitre qui consiste à proposer une modélisation complète pour le séquençage de protéine à partir de données expérimentales de différents types. Il s'agit du problème de « de novo **protein** sequencing ». Nous proposons donc, pour ce problème, une découpe en trois sous-problèmes appelée SSO pour « Sequence », « Shape » et « Order ». L'analyse des fonctions d'évaluation et des objectifs de chaque sous-problème de ce modèle a permis de mettre en évidence l'utilisation de trois métaheuristiques coopérant de manière séquentielle. Les premiers résultats d'application de ce nouveau modèle sont détaillés à partir de données expérimentales fournies par le Laboratoire de Chimie Organique et Macromoléculaire de Lille (LCOM) qui est notre principal collaborateur sur cette partie. Ce travail a été soutenu par l'Université des Sciences et Technologie de Lille (USTL) par l'intermédiaire du PPF Bioinformatique.

Le **chapitre 3** présente le travail réalisé avec un autre laboratoire de l'USTL, le laboratoire de glycobiologie structurale et fonctionnelle de Lille 1. Ce travail a lui aussi été soutenu par le PPF Bioinformatique mais fait également parti d'un projet ANR : le projet Dock. Ce travail consiste en la recherche, la mise en place et l'évaluation de nouvelles modélisations multi-objectives pour le problème du docking moléculaire flexible. Nous détaillons en quoi consiste ce problème et précisons la place de mon travail au sein du projet ANR Dock. Par la suite, nous définissons les notions de champ de force, d'énergie, de surface et de robustesse afin faciliter la compréhension des nouveau modèles multi-objectifs présentés. Ceux-ci ont été choisis après examen de modèles existants. Huit modèles sont présentés, analysés et testés. Le but est de mettre en évidence le ou les modèles les plus performants. Nous proposons et testerons également différentes coopérations de métaheuristiques afin de déterminer le (les) meilleur(s) schéma(s) et d'obtenir les meilleurs résultats. Pour ce faire, une analyse statistique de tous les résultats sur toutes

http://doc.univ-lille1.fr

les instances utilisées est proposée afin d'en extraire les meilleurs couples modélisation/configuration.

Enfin, après un rappel des objectifs fixés et des résultats, nous faisons le point sur les perspectives à plus ou moins long terme pour ces travaux.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

1

Métaheuristiques coopératives pour l'optimisation combinatoire

Dans ce chapitre, nous définissons la notion de problème d'optimisation combinatoire ainsi que la manière d'appréhender et de résoudre ce type de problème. Nous présentons ensuite les grandes familles de méthodes en nous attachant plus particulièrement aux métaheuristiques et la manière de les faire coopérer. Nous identifierons ainsi les mécanismes communs dans les métaheuristiques à solution unique afin de mettre en place une plateforme de conception générique indépendante du problème mono-objectif. Enfin, la plateforme ParadisEO-MO dédiée à la conception de métaheuristiques à base de solution unique sera détaillée en insistant notamment sur toutes les améliorations et extensions qui lui ont été apportées par rapport à la version originelle de 2005.

Le travail effectué sur la plateforme ParadisEO-MO a fait l'objet d'une présentation lors de la conférence internationale en recherche opérationnelle EURO XXII en 2007 sous le titre « Software Framework for Metaheuristic » [7] et d'une soumission d'un rapport de recherche INRIA ayant pour titre « ParadisEO-MO : a Framework to Design Single Solution Metaheuristics » en 2008 [13].

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.1 Définitions

Nous présentons ici ce qu'est l'optimisation mono et multi-objectif avant de présenter une classification des méthodes d'optimisation.

1.1.1 Optimisation mono-objectif

Un problème d'optimisation combinatoire est défini par un ensemble fini de solutions dites réalisables (formant l'espace de décision Ω) et une fonction objectif (f) associant à chaque solution de l'espace de recherche un coût [2] : cet ensemble de valeurs constitue l'espace objectif Y. Nous avons donc $f : \Omega \rightarrow Y$. Une solution réalisable x peut être décrite par une vecteur de décision $(x_1, x_2, ..., x_n)$ à laquelle une valeur objectif $f(x)$ est associée. Résoudre un problème d'optimisation combinatoire revient à trouver la ou les solution(s) réalisable(s) de coût minimal (ou maximal) de Ω. Sauf s'il est fait mention du contraire, nous nous placerons généralement dans le cas de la minimisation.

Définition 1. *x^* ($x^* \in \Omega$) est une solution optimale (un optimum global) si et seulement si :*

$$\forall x \in \Omega, \quad f(x) \geq f(x^*)$$

Définition 2. *x^* ($x^* \in \Omega$) est un optimum local si et seulement si :*

$$\forall x \in S \subset \Omega, \quad f(x) \geq f(x^*)$$

Remarque 1. *Un optimum global est toujours un optimum local mais l'inverse n'est pas nécessairement vrai.*

A tout problème combinatoire est associé un espace de recherche ; celui-ci peut être infini ou borné. Sa taille est fortement liée à la représentation des solutions du problème étudié. En effet, si les paramètres sont définis dans des ensembles de valeurs dénombrables et bornés alors l'espace de recherche associé sera aussi dénombrable et borné.

Exemple 1. *Soit un problème dont une solution est représentée par un couple de variables (x_1, x_2) qui sont définies dans l'alphabet $A = \{a, b, c\}$. A est un ensemble dénombrable, par conséquent l'espace de recherche associé est aussi dénombrable, ici $|A| = 2^3$ si les couples symétriques sont considérés comme non équivalents.*

Dans la plupart des cas, les ensembles dans lesquels sont contenus les valeurs des paramètres ne sont pas bornés. Suivant le problème étudié, il peut cependant être possible de discrétiser ces ensembles pour se ramener à des ensembles finis. Mais même dans ce cas, la taille de l'espace de recherche, même si elle est calculable, peut rester excessivement élevée comme cela est montré dans le tableau 1.1.

La taille de l'espace de recherche d'un problème va permettre d'évaluer la complexité d'une méthode qui le parcourt.

1.1.2 Complexité

La complexité d'une méthode de résolution est un indicateur du coût de l'application de celle-ci. La complexité peut-être temporelle et/ou spatiale. Une complexité temporelle correspondra au temps de calcul nécessaire à la finition de la méthode alors que la complexité spatiale définira les besoins physiques d'une méthode (espace de stockage, mémoire vive, ...) pour qu'elle puisse être appliquée. Généralement c'est toujours la complexité temporelle qui est proposée. Cependant, l'étude de la complexité spatiale est très importante car il est souvent possible de réduire la complexité temporelle d'une méthode au détriment de la complexité spatiale.

[2] Cas d'une minimisation sinon on peut parler de gain.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 1.1 – Variation de la taille de l'espace de recherche selon sa description suivant n. TER signifie Taille de l'Espace de Recherche en terme de nombre de solutions.

TER suivant n \ n	**5**	**10**	**50**	**100**	**1000**	**10 000**
n^2	25	100	2500	$1e^4$	$1e^6$	$1e^8$
n^5	3125	$1e^5$	$3.125e^8$	$1e^{10}$	$1e^{15}$	$1e^{20}$
2^n	32	1 024	$1.126e^{15}$	$1.268e^{30}$	$1.072e^{301}$	$2e^{3010}$
$n!$	120	$3.6288e^6$	$3.041e^{64}$	$9.33e^{157}$	$4.0239e^{2567}$	$2.846e^{35659}$

Exemple 2. *Le calcul d'un terme de la suite de Fibonacci repose sur la nécessité de connaître la valeur de tous les termes précédents. Pour rappel, un terme u_n de la suite de Fibonacci est défini de la manière suivante :*

$$u_n = u_{n-2} + u_{n-1}$$

avec $u_0 = 0$ et $u_1 = 1$.
Si on implémente un algorithme récursif pour connaître un terme de la suite de Fibonacci, c'est à dire un algorithme dans lequel la fonction appelée contient elle-même un ou plusieurs appels à elle-même, on peut écrire :

$$Fibonacci(n) = Fibonacci(n-2) + Fibonacci(n-1)$$

Ainsi :

$$\begin{aligned} Fibonacci(5) &= Fibonacci(3) + Fibonacci(4) \\ &= Fibonacci(1) + Fibonacci(2) + Fibonacci(2) + Fibonacci(3) \\ &= 1 + Fibonacci(0) + Fibonacci(1) + Fibonacci(0) + Fibonacci(1) \\ &\quad + Fibonacci(1) + Fibonacci(2) \\ &= 1 + 0 + 1 + 0 + 1 + 1 + Fibonacci(0) + Fibonacci(1) = 5 \\ &= 1 + 0 + 1 + 0 + 1 + 1 + 0 + 1 = 5 \end{aligned} \tag{1.1}$$

On peut remarquer que lorsqu'on déroule les différents appels récursifs, des calculs redondants sont faits. En effet, tout appel à la fonction entraine forcément une pyramide de sous-appels jusqu'aux termes de rang 0 ou 1 pour lesquels la valeur est connue. Toute addition non nécessaire va avoir un impact sur le temps de calcul et donc faire augmenter la complexité temporelle. Cependant, on peut très bien sauvegarder la valeur d'un terme dès que celui est calculé pour obtenir un accès direct à celle-ci au besoin. Ainsi, on diminue grandement la complexité temporelle en contre parti d'une complexité spatiale plus élevée (stockage des valeurs des $n-1$ termes précédents).

Il existe différentes manières d'exprimer la complexité d'une méthode : au mieux, en moyenne ou au pire. C'est généralement la complexité au pire qui est utilisée pour décrire les méthodes car elle est généralement facile à calculer et dans la plupart des problèmes, il est utile d'avoir une borne supérieure que l'on sait non franchissable. Il faut cependant faire attention, car la complexité au pire est souvent fortement surestimée pour des questions de simplification. La complexité moyenne apporte une information plus critique sur le coût d'une méthode mais elle reste très souvent compliquée à calculer. Enfin, la complexité au mieux reste un indicateur peu utilisé car elle correspond à la complexité dans le cas le plus favorable.

Dans toutes les complexités qui seront proposées dans ce rapport, il sera toujours question de la complexité au pire notée $O(x)$ avec x le détail de la complexité.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Lié à la complexité, il est aussi souvent question de la difficulté d'un problème à être résolu. L'ennui de ce type de description est quelle est beaucoup trop vague. En effet, qu'est-ce qu'un problème difficile ? Une réponse à cette question consiste en l'étude de la définition même du problème et de sa structure. On classe généralement les problèmes suivant la complexité des méthodes utilisées pour les résoudre. Les classes les plus connus sont les classes P et NP. Les problèmes de la classe P correspondent aux problèmes dont l'énumération de toutes les solutions existantes mais aussi la vérification de leur validité est possible en temps polynômiale i.e. avec une complexité temporelle et/ou spatiale polynômiale [3]. Les problèmes de la classe NP sont des problèmes pour lesquels la validité d'une solution est vérifiable en temps polynomiale contrairement à la génération de toutes les solutions. Par principe, on a tendance à simplement dire qu'un problème est facile, s'il appartient à la classe P et qu'il est difficile, s'il appartient à la classe NP. C'est pour cela que dès qu'un problème est montré comme difficile, on y associe généralement la preuve de son appartenance à la classe NP.

Il s'agit bien sûr d'un survol de la théorie de la complexité, pour plus d'information notamment sur la manière de définir qu'un problème appartient à la classe NP, on pourra se référer par exemple au livre de Gary and Johnson [48].

1.1.3 Notion de paysage

La notion de paysage permet de décrire visuellement l'ensemble des valeurs que peut prendre une fonction objectif pour une paramétrisation donnée (une instance [4]). Cette description permet notamment de voir s'il existe potentiellement de nombreux optima locaux. La description d'un paysage se fait en analogie avec la notion « géographique » du paysage. Ainsi un paysage peut être décrit comme une vallée, une plaine, un bassin, …. Cela correspond à la forme général. Ensuite, il est possible d'apporter une information sur le nombre d'optima locaux existants. Ainsi, un paysage rugueux ou chaotique correspondra plutôt à un paysage possédant un nombre important d'optima locaux. Au contraire, un paysage tout à fait lisse amènera à priori vers un nombre restreint d'optima locaux. La figure 1.1 propose quelques exemples de forme de paysage.

L'étude du paysage d'une fonction objectif permet de guider le choix de la méthode de résolution à appliquer.

1.1.3.1 Diversification

Le principe de diversification d'une méthode d'optimisation donnée correspond à sa capacité de parcourir aisément l'espace de recherche pour obtenir des solutions très différentes les unes des autres. Dans un paysage donné, cela se traduit par la possibilité de faire des déplacements plutôt horizontaux comme cela est montré dans la figure 1.2. Cependant, comme cela est visible sur cette figure, la diversification seule n'amène pas forcément à des optima locaux mais à des solutions se trouvant potentiellement dans la zone d'un optimum local non encore trouvé jusqu'à maintenant.

[3] Qui s'exprime sous forme d'un polynôme.

[4] Un jeu de valeurs particulières donné à tous les paramètres d'un problème.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

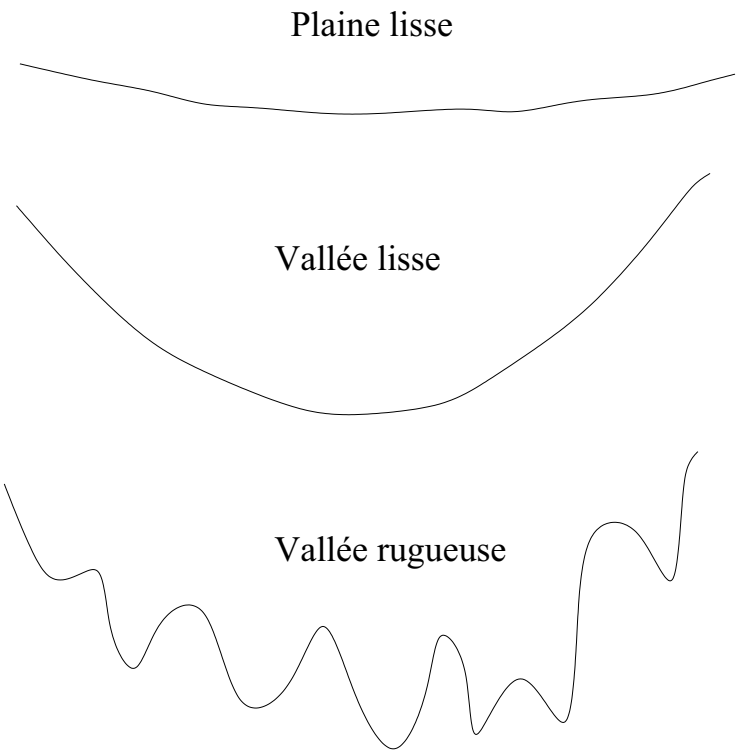


FIG. 1.1 – Exemples de forme de paysage.

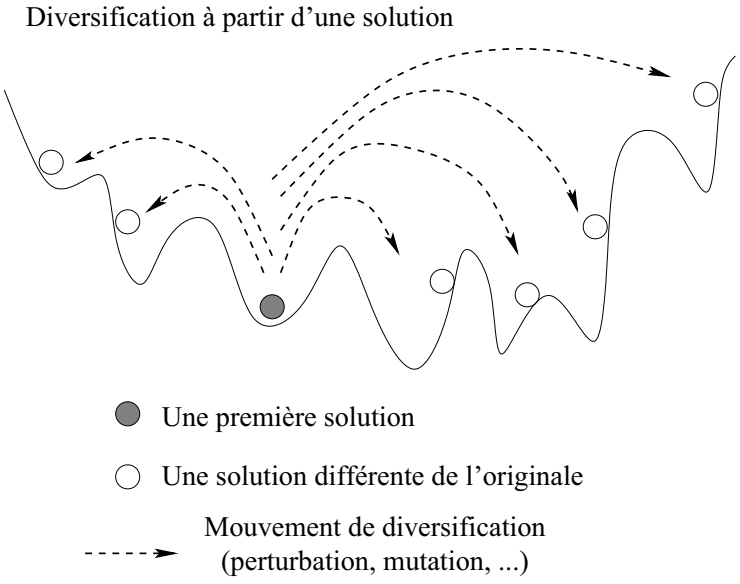


FIG. 1.2 – Processus de diversification d'une solution dans un paysage donné.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.1.3.2 Intensification

Autant le principe de diversification essaye de déplacer les solutions dans d'autres zones de l'espace de recherche, autant le processus d'intensification vise à forcer une solution donnée à tendre vers l'optimum local de la zone à laquelle elle est attachée.

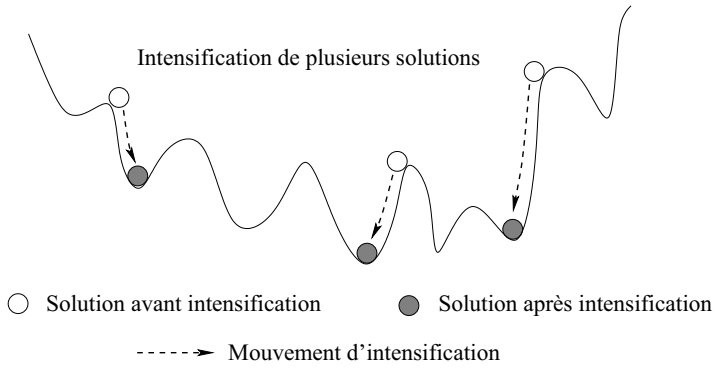


FIG. 1.3 – Processus d'intensification de solutions dans un paysage donné.

La figure 1.3 donne une exemple de processus d'intensification pour des solutions de départ données. Comme on peut le voir, une phase d'intensification correspond plutôt à un mouvement vertical dans le paysage pour accéder à un optimum local (peut-être l'optimum global). La force mais aussi la faiblesse de l'intensification est de rester cantonnée à la zone dans laquelle la solution se trouve sans possibilité de pouvoir sortir de celle-ci. Donc si les solutions que l'on veut intensifier ne sont pas dans la zone de l'optimum global, il n'y a aucune chance que l'intensification seule permettent d'y accéder.

Dans une grande majorité des problèmes réels, il est difficile de qualifier une solution selon un seul critère. Généralement, une bonne solution est une solution de compromis entre plusieurs objectifs a priori contradictoire. Dans ce cas, il est nécessaire d'utiliser une modélisation multi-objective des problèmes et de leur résolution.

1.1.4 Optimisation multi-objective

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

$$(PMO)\ \ Optimiser\ F(x) = (f_1(x), f_2(x), ..., f_n(x)), x \in \Omega$$

où n est le nombre d'objectifs (n $\geq$ 2), $x = (x_1, x_2, ..., x_k)$ est le vecteur représentant les variables de décision, Ω représente l'ensemble des solutions réalisables et $f_i(x)$ les fonctions à optimiser. Sans perte de généralité nous supposerons par la suite que nous considérons des problèmes de minimisation.

Définition 3. *Une solution réalisable $x \in \Omega$ est **Pareto optimale** (ou efficace, ou encore non dominée) si et seulement si il n'existe pas de solution $x \in \Omega$ telle que x domine x^*. On dit d'une solution $y = (y_1, y_2, ..., y_k)$ qu'elle domine une solution $z = (z_1, z_2, ..., z_k)$, dans le cas d'une minimisation d'objectifs, si et seulement si $\forall i \in [1..n]$, $f_i(y) \leq f_i(z)$ et $\exists i \in [1..n]$ tel que $f_i(y) < f_i(z)$.*

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

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Définition 4. *On dit d'une solution* $y = (y_1, y_2, ..., y_k)$ *qu'elle domine* ***faiblement*** *une solution* $z = (z_1, z_2, ..., z_k)$*, dans le cas d'une minimisation d'objectifs, si et seulement si* $\forall i \in [1..n], f_i(y) \leq f_i(z)$.

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, 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.

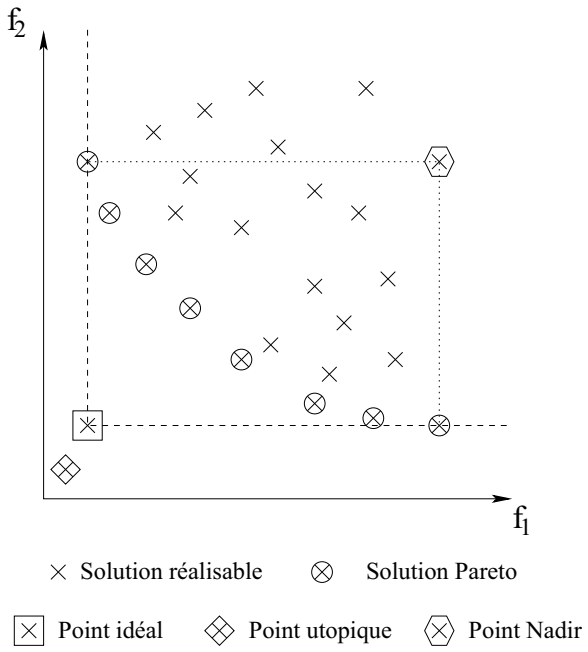


FIG. 1.4 – Description des notions de solution réalisable, solution Pareto, point idéal, point utopique et point Nadir dans un exemple de deux fonctions objectif à minimiser f_1 et f_2.

Le point idéal z^I est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré. z^I tel que $\forall i \in [1..n], f_i(z^I) = opt_{x \in \Omega} f_i(x)$. 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.

A partir du point idéal, on définit le point utopique z^U de la façon suivante :

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

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

Enfin le point Nadir est défini en bi-objectif par :

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

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

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

La figure 1.4 schématise l'ensemble de ces points sur un exemple avec deux objectifs à minimiser.

A l'inverse des modélisations mono-objectives où il est facile de prouver qu'une méthode de résolution est meilleure qu'une autre, en multi-objectif ce genre de raisonnement est plus difficile à mettre en place. Pour ce faire, des indicateurs de performances existent et permettent d'amener de l'information pour permettre des comparaisons.

1.1.4.1 Mesures de performance

De manière générale, il est difficile de mesurer la qualité d'un algorithme multi-objectif et encore plus de le comparer à un autre algorithme multi-objectif. Il existe cependant un certain nombre d'indicateurs (unaire ou n-aire avec n souvent égal à 2) qui permettent d'évaluer la qualité d'un algorithme multi-objectif et de pouvoir le comparer à d'autres sur des critères de convergence et/ou de diversification.

Indicateurs de convergence Les indicateurs de convergence évaluent la qualité des solutions trouvées en fonction de leur distance par rapport au front Pareto optimal. La figure 1.5 montre un exemple simple d'interprétation visuel de la convergence.

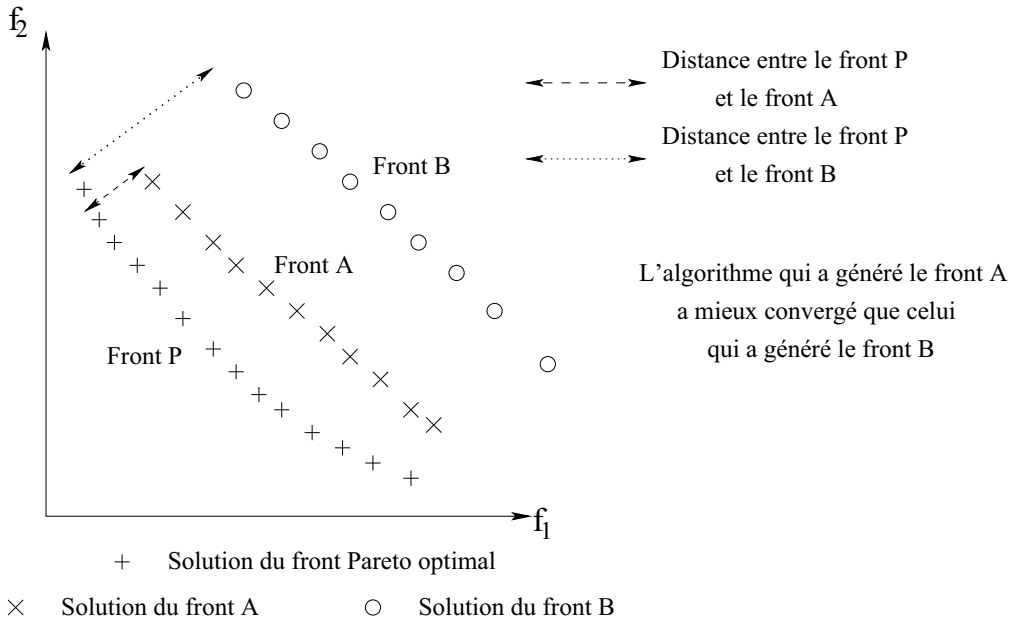


FIG. 1.5 – Indicateur de convergence selon deux fonctions objectif f_1 et f_2 à minimiser.

– Contribution : la contribution est un indice de performance binaire qui permet de déterminer parmi deux fronts P_A et P_B celui qui contribue le plus au front P^* qui correspond à l'ensemble des solutions non-dominées de $P_A \cup P_B$. Si on note P les solutions de $P_A \cap P_B$, D_A (respectivement D_B) les solutions de P_A (respectivement P_B) qui dominent des solutions de P_B (respectivement P_A), SD_A (respectivement SD_B) les solutions de P_A (respectivement P_B) qui sont dominées par des solutions de P_B (respectivement P_A) et enfin NC_A (respectivement NC_B) les solutions non comparables de P_A (respectivement P_B) ; la contribution s'écrit de la manière suivante :

$$Cont(P_A/P_B) = \frac{\frac{||P||}{2} + ||D_A|| + ||NC_A||}{||P^*||}$$

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

avec

$$NC_A = P_A \setminus (P \cup D_A \cup SD_A)$$

La figure 1.6 montre un exemple de calcul de la contribution sur un exemple de minimisation de deux fonctions objectif f_1 et f_2.

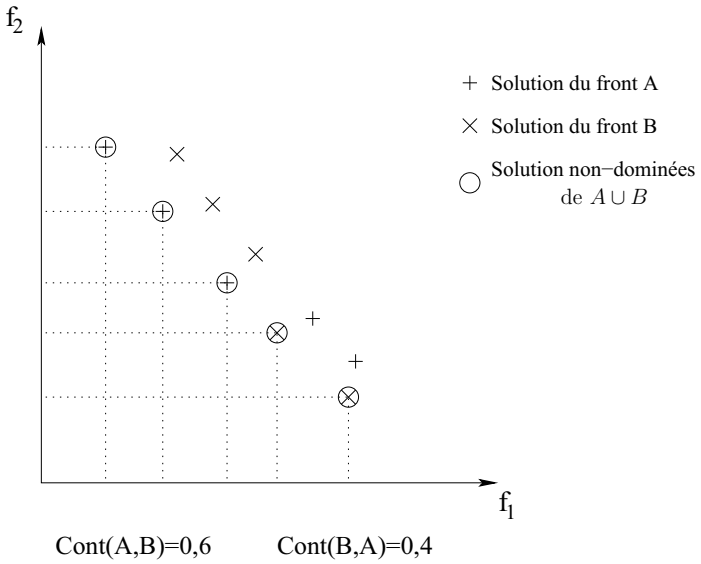


FIG. 1.6 – Comparaison de deux fronts selon l'indicateur de comparaison dans le but de minimiser deux fonctions objectif f_1 et f_2.

L'avantage de cette métrique est qu'elle est facilement interprétable. En effet, si $Cont(P_A/P_B) = 0,8$ cela signifie que 80% des solutions non-dominées proviennent de P_A et que par conséquent $Cont(P_B/P_A) = 0,2$ car $Cont(P_A/P_B) + Cont(P_B/P_A) = 1$. A partir du moment où les solutions d'un front apportent une contribution supérieure à $0,5$ alors on considère que le front correspondant est meilleur en terme de convergence que celui avec lequel il est comparé.

– ϵ-indicator : indicateur binaire qui permet d'obtenir le facteur minimum par lequel un front P_A doit être translaté dans l'espace des objectifs pour dominer faiblement un front P_B. Cet opérateur est défini de la manière suivante :

$$I_\epsilon(A/B) = \min_{\epsilon \in \mathbb{R}} \{\forall z \in B, \exists\, z' \in A : z'_i - \epsilon \leq z_i, \forall 1 \leq i \leq n\}$$

Si la valeur de cet indicateur est négative, cela indique que P_A domine faiblement P_B. Une version unaire de cet indicateur est définie de la manière suivante :

$$I^1_\epsilon(A) = I_\epsilon(A, Z^*_N)$$

où Z^*_N est un front de référence.

– Cardinalité : cette métrique est utilisable uniquement dans le cas où le front Pareto optimal est connu. Elle indique la proportion de solutions qui appartiennent au front Pareto optimal. Soit P_A un front et P^* le front Pareto optimal, la mesure de cardinalité est définie de la manière suivante :

$$I_c(P_A, P^*) = \frac{|P_A \cap P^*|}{|P^*|}$$

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

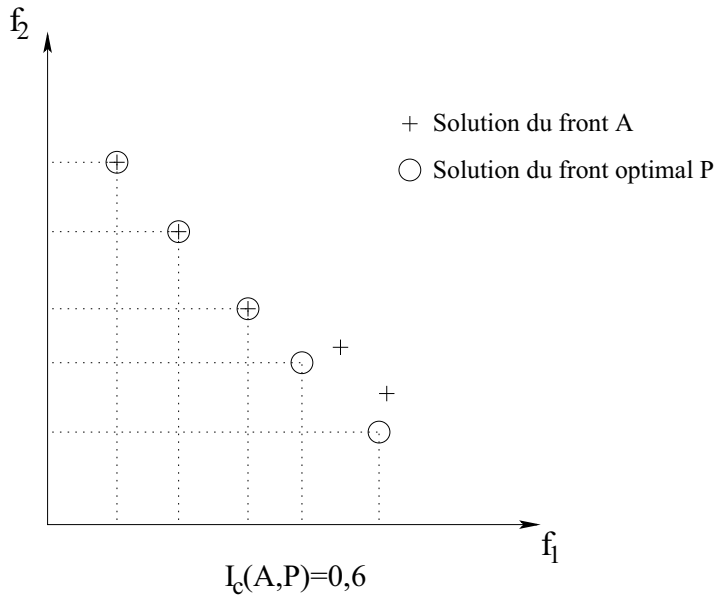


FIG. 1.7 – Comparaison d'un front avec le front optimal selon l'indicateur de cardinalité dans le but de minimiser deux fonctions objectif f_1 et f_2.

La figure 1.7 montre un exemple de calcul de la cardinalité pour la minimisation de deux fonctions objectif f_1 et f_2.

– Distance : comme pour la cardinalité, cette mesure nécessite d'avoir le front Pareto optimal. Elle donne la distance moyenne d'un front P_A par rapport au front Pareto optimal P^*. Elle est définie de la manière suivante :

$$I_d(P_A, P^*) = \frac{\sum_{u \in P_A} \min_{v \in P^*} (\|F(u) - F(v)\|)}{|P^*|}$$

Indicateurs de diversification Les indicateurs de diversification mesurent l'uniformité de la distribution des solutions obtenues, généralement dans l'espace des objectifs. La figure 1.8 montre un exemple de deux fronts : un premier où la diversification est bonne avec une répartition continue des solutions tout au long du front et un second autre moins bien diversifié avec des zones de forte densité de solutions.

– Mesure d'espacement (« spacing ») : métrique qui apporte une information sur la répartition des solutions sur un front donné. Elle est définie de la manière suivante [115] :

$$S = \sqrt{\frac{1}{|P_A|} \sum_{i=1}^{|P_A|} (d_i - \bar{d})^2}$$

avec

$$d_i = \min_{k \in P_A \wedge k=i} \sum_{m=1}^{n} |f_m^i - f_m^k| \quad et \quad \bar{d} = \frac{\sum_{i=1}^{|P_A|} d_i}{|P_A|}$$

P_A est le front Pareto utilisé et n est le nombre d'objectifs. d_i est la valeur minimale des différences absolues des fonctions objectif entre la solution i et toutes les autres solutions de P_A. $\bar{d}$ correspond

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

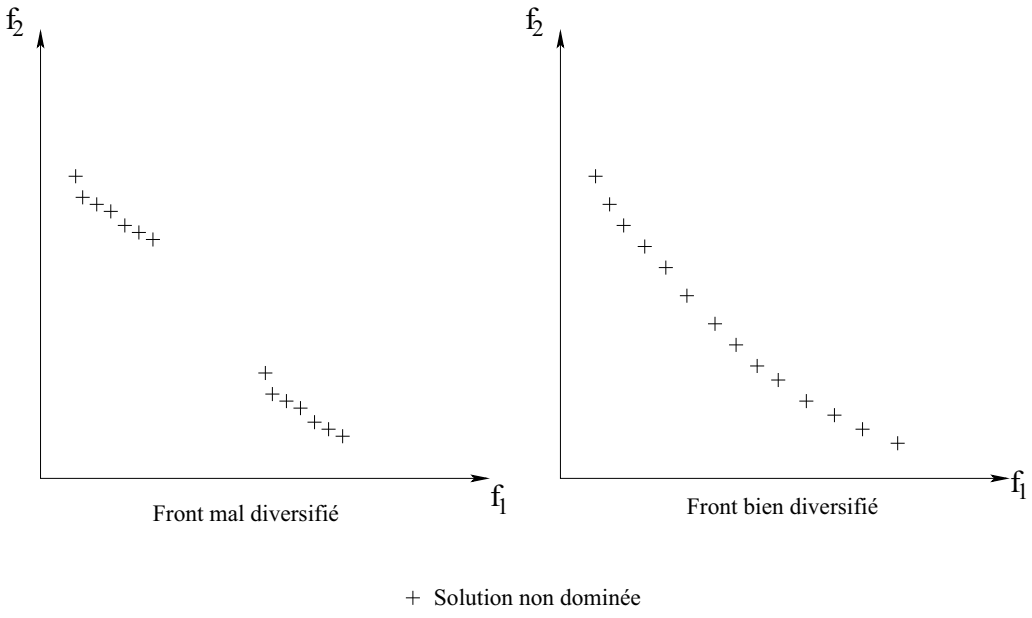


FIG. 1.8 – Indicateur de diversification selon deux fonctions objectif f_1 et f_2 à minimiser.

à la moyenne des d_i. La mesure d'espacement correspond, d'après la formulation précédente, au calcul des écarts types des différentes valeurs de d_i. Meilleure est la répartition des solutions sur le front, plus faible sera la valeur de cette métrique.

– Mesure du maximum de dispersion (« maximum spread ») : cette métrique définie de la manière suivante [139] :

$$D = \sqrt{\frac{1}{|P_A|}\sum_{m=1}^{n}\left(\frac{\max_{i=1}^{|P_A|} f_m^i - \min_{i=1}^{|P_A|} f_m^i}{F_m^{max} - F_m^{min}}\right)^2}$$

permet de calculer la longueur de la diagonale d'une « hyperboîte » obtenue par les valeurs extrêmes des objectifs, F_m^{max} et F_m^{min} étant respectivement la valeur maximale et minimale du m$^{\text{ième}}$ objectif, des solutions d'un front donné P_A. Cette mesure donne une information sur la distance maximum qu'il peut exister entre des solutions mais ne permet pas d'obtenir une information sur la réelle répartition des solutions. Par exemple, il peut exister une zone de forte densité de solutions et une seule solution éloignée de cette zone.

– Mesure de l'entropie : cet indice de performance binaire a pour but de comparer la diversité d'un front P_A obtenu par un algorithme A et la diversité d'un front P_B obtenu par un algorithme B [4, 91]. On notera P^* l'ensemble des solutions non dominées de $P_A \cup P_B$. Pour le calcul de cette métrique, chaque solution est associée à une niche et les solutions appartenant à la même niche sont considérées comme équivalentes. On note N_i le nombre de solutions de $P_A \cup P^*$ se trouvant dans la niche i$^{\text{ième}}$ solution de $P_A \cup P^*$, C_{A*} le cardinal de $P_A \cup P^*$, n_i le nombre de solutions de P_A qui sont dans la i$^{\text{ième}}$ solution de $P_A \cup P^*$. De cette manière l'entropie est décrite par la formule suivante :

$$E(A/B) = \frac{-1}{log\gamma}\sum_{i=1}^{C_{A*}}\frac{1}{N_i}\frac{n_i}{C_{A*}}log\frac{n_i}{C_{A*}}$$

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

avec

$$\gamma = \sum_{i=1}^{C_{A*}} \frac{1}{N_i}$$

γ représente la somme des coefficients affectés à chaque solution. L'entropie permet donc d'avoir une mesure de la diversité relative entre deux fronts. Cependant, elle peut être difficile à interprétée suivant les cas.

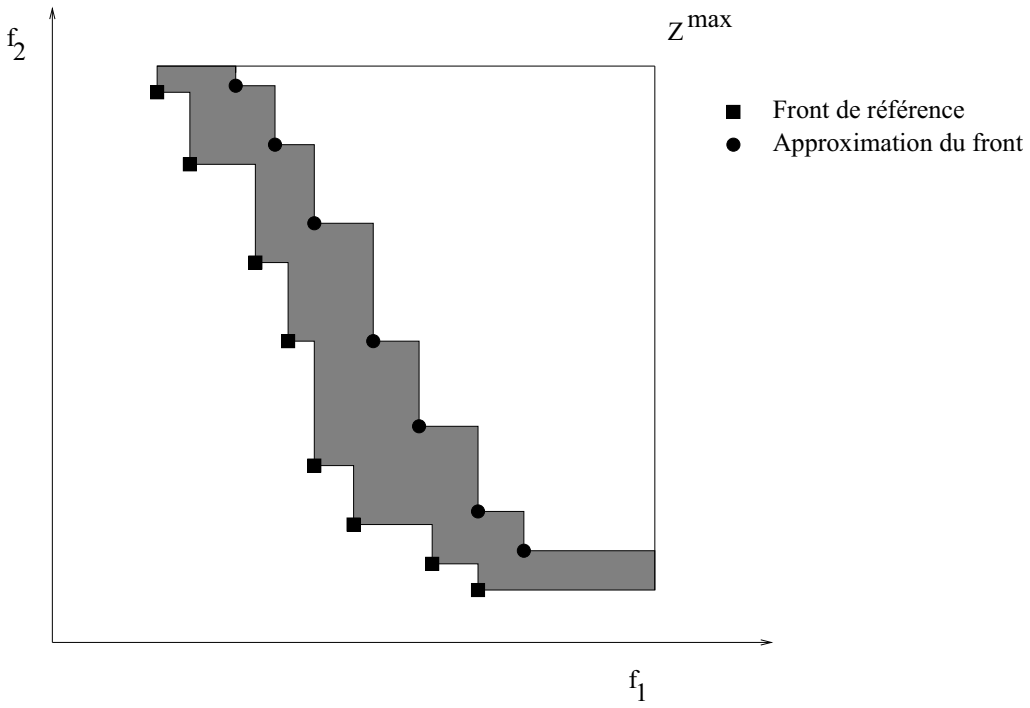


FIG. 1.9 – Représentation de l'hypervolume selon la minimisation de deux fonctions objectif f_1 et f_2. Z^{max} est un point de référence permettant de définir une surface contenant toutes les solutions générées.

Indicateur hybride La métrique S ou hypervolume correspond au calcul de l'hypervolume représenté par les zones de dominance des ensembles de solutions Pareto de deux fronts et un point de référence [139]. L'avantage de cet indicateur est qu'il apporte une information aussi bien au niveau de la diversité que de la convergence. Par contre, il est plus coûteux à mettre en place que d'autres indicateurs. Comme la figure 1.9 le montre, l'hypervolume peut être ramené à calculer la surface existante entre un front de référence et le front à analyser. En effet, lorsque deux fronts sont comparés, le front de référence correspond à la concaténation des solutions non-dominées des deux fronts. On peut ainsi comparer chacun des fronts avec le front de référence pour évaluer l'hypervolume. Plus cette surface est faible, meilleure est l'approximation du front. L'utilisation de cet indicateur a été souvent recommandé, notamment dans [76].

Après avoir analysé puis modélisé un problème, il est temps de choisir une méthode de résolution adéquat parmi toutes celles existantes. Pour cela, il est important de bien définir les différentes classes de méthodes existantes ainsi que leurs avantages et incovénients respectifs.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.1.5 Méthodes d'optimisation

Il existe un grand nombre de méthodes d'optimisation. La figure 1.10 présente une possible hiérarchie des celles-ci.

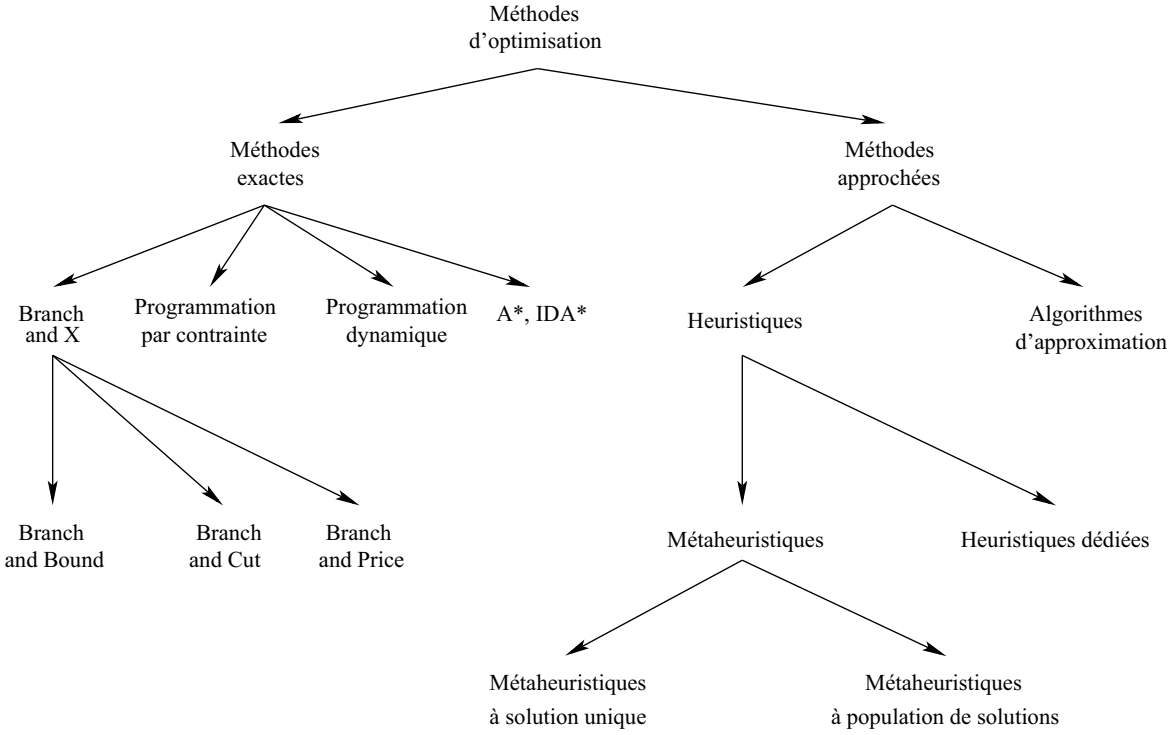


Fig. 1.10 – Une possible hiérarchie des méthodes d'optimisation.

On sépare tout d'abord les méthodes d'optimisation en deux grandes familles de méthodes : les méthodes exactes et les méthodes approchées. Les méthodes de la première famille assurent de trouver l'optimum global pour l'instance considérée avec un coût souvent important alors que les méthodes de la seconde famille ne donnent pas de garantie sur l'optimalité de la solution trouvée mais peuvent assurer des coûts moindres comparées aux méthodes exactes.

1.1.5.1 Méthodes exactes

Les méthodes exactes assurent l'optimalité des solutions car elles parcourent l'ensemble de l'espace de recherche. Par conséquent, plus l'espace de recherche est important, plus l'utilisation des ces méthodes va être coûteuse. Dans leur principe, les différentes familles de méthodes exactes ne diffèrent pas énormément. Elles sont basées sur un parcours « intelligent » de l'espace de recherche pour atteindre la meilleure solution sans pour autant énumérer toutes les solutions possibles. Ces parcours correspondent pour la plupart à des parcours arborescents dans lesquels certains critères permettent de déterminer les branches à développer de celles à éviter. Dans ces arbres, la racine est le problème d'optimisation à résoudre, les noeuds des sous-problèmes du problème d'origine et les feuilles des solutions pour les problèmes « parents ». Le choix des branches à explorer peut se faire selon le calcul de borne (algorithmes « Branch and X » et A*, IDA*), selon une décision partielle correspondant au sous-problème à traiter (programmation dynamique) ou encore selon le respect de contraintes liées aux variables du problème (programmation par contrainte). Dans tous les cas, la réussite de ces méthodes se base sur l'efficacité

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

de la discrimination des branches à parcourir. Plus tôt des branches seront coupées dans le parcours de l'arbre, plus rapide sera l'exploration.

Les méthodes exactes restent cependant limitées aux instances de petite taille même si le passage à des dimensions plus grandes reste très modéremment faisable grâce à l'utilisation du parallélisme. En effet, même si un grand nombre de machines est utilisé en parallèle, il n'est pas forcément possible d'augmenter de la même quantité la taille des instances utilisées. Comme cela avait été évoqué dans le tableau 1.1, l'utilisation du parallélisme à grand échelle ne permet pas d'affronter l'explosion combinatoire du nombre de solutions à énumérer lorsque la taille des instances augmente de manière exponentielle.

Lorsqu'il devient nécessaire d'obtenir des solutions, non forcément exactes, pour des problèmes de grande taille dans des temps impartis restreints, il devient nécessaire d'utiliser les méthodes dites approchées.

1.1.5.2 Méthodes approchées

Comme cela est montré sur la figure 1.10, les méthodes approchées peuvent être divisées en deux sous-familles : les heuristiques et les méthodes d'approximation. La différence entre ces deux familles vient du fait que les méthodes d'approximation assurent une information de garantie sur les solutions obtenues par rapport à l'optimum global [63]. Par contre, les heuristiques ne donnent aucune indication de ce point de vue. Ces dernières peuvent être partagées entre les métaheuristiques qui sont des méthodes génériques non dédiées à un problème en particulier et les heuristiques dédiées qui, comme leur nom l'indique, concernent des méthodes spécifiques à un problème donné. La notion d'heuristique date de 1945 [107] et les premières métaheuristiques, les algorithmes évolutionnaires et les premières recherches locales, datent des années 60. Depuis cette époque, les métaheuristiques ont rencontré de plus en plus de succès et ont vu leur nombre augmenter (plus d'une dizaine de grandes familles d'algorithmes). Cela est dû à leur capacité à résoudre des problèmes de taille importante en des temps raisonnables en apportant des solutions de qualité satisfaisante. Il est bien sûr toujours possible d'obtenir des solutions de meilleures qualités mais cela ne peut se faire qu'au détriment du coût de la méthode utilisée. Deux familles de métaheuristiques existent : les métaheuristiques à base de solution unique et celle à base de population de solutions. Les premières consistent en une amélioration d'une seule solution existante tant que cela est possible, ce sont donc des méthodes avec un fort pouvoir d'intensification. Par contre, les métaheuristiques à base de population permettent de mieux appréhender les grands espaces de recherche de part leur pouvoir de diversification. Même si les métaheuristiques sont applicables sur n'importe quel type de problème, il convient toujours d'analyser les caractéristiques du problème à traiter notamment le paysage associé pour évaluer quelle est la meilleur méthode à utiliser [93].

1.1.5.3 Méthodes approchées multi-objectives

La résolution, même approchée, d'un problème multi-objectif impose des contraintes supplémentaires qui n'apparaissent pas dans l'optimisation mono-objectif. Toute méthode approchée dédiée à la résolution d'un problème multi-objectif doit tenir compte de trois concepts :

1. *L'assignement de la qualité (« fitness ») d'une solution* : ce concept doit détailler comment doit être évaluée la qualité d'une solution. Il existe quatre familles d'assignement de qualité :

   – Approches scalaires : ces méthodes consistent à ramener le problème multi-objectif à un problème mono-objectif ou à un ensemble de problèmes mono-objectif. Cela peut consister en l'agrégation des différents objectifs au sein d'une seule et unique fonction objectif grâce à des

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

poids ou au calcul d'une distance entre les solutions et un point de référence à atteindre. Dans ces deux cas, on ramène le problème initial à un seul et unique problème mono-objectif. Il est aussi possible de travailler sur les différents objectifs de manière indépendante : soit en essayant pour chaque objectif d'atteindre des valeurs cibles, soit encore en transformant les autres objectifs en contraintes. Dans ces situations, le problème originel est tranformé en un ensemble de problèmes mono-objectif. Ces approches nécessitent cependant une bonne connaissance du problème pour être mises en place.

- Approches basées sur la dominance entre solutions : ce type de méthodes permet de ne pas appliquer de transformations sur le problème originel. L'une des plus connues est NSGA-II pour Non Dominated Sorting Genetic Algorithm II [36]. Pour appliquer ces méthodes, il est cependant nécessaire de pouvoir trier les solutions et leur assigner des « rangs ». Le rang peut correspondre au nombre de solutions dominées par la solution considérée (notion apparue dans la famille des algorithmes de type SPEA, « Strength Pareto Evolutionary Algorithm » [141]), au nombre de solutions qui dominent la solution considérée (notion apparue dans l'algorithme MOGA, « Multi-Objective Genetic Algorithm » [44]) ou encore à l'indice de front de solutions non-dominées auquel appartient la solution courante : le front Pareto est le front 1, le front 2 est le front obtenu quand les solutions du front 1 sont enlevées et ainsi de suite (principe utilisé dans NSGA-II).

- Approches basées sur des indicateurs de performance : dans ce type d'approche ce sont des indicateurs de performance binaires qui sont utilisés pour faire l'assignement. La plupart des indicateurs utilisés sont basés sur la notion de dominance. L'algorithme IBEA [140] (Indicator Based Evolutionary Algorithm) est un bon exemple d'algorithme multi-objectif basé sur des indicateurs.

2. *La préservation de la diversité* : la notion de diversité est un concept déjà connu pour l'utilisateur de métaheuristiques à base de populations de solutions. Le but est d'assurer une bonne répartition des solutions dans l'espace de recherche afin de maximiser les chances d'atteindre des zones contenant de meilleurs solutions. En optimisation multi-objectif, le but est d'obtenir une bonne répartition des solutions le long du front Pareto. Afin d'assurer cette diversité, la densité de solutions autour d'une solution donnée est calculée afin de d'évaluer la répartition des solutions les unes par rapport aux autres. Ils existent plusieurs méthodes pour définir cette notion de densité mais la finalité reste la même : pénaliser les zones de trop forte densité pour améliorer la répartition des solutions.

3. *L'élitisme* : l'utilisation de l'élistime doit permettre de conserver les meilleures solutions Pareto trouvées durant la recherche. Pour ce faire, une population de « bonnes » solutions est maintenue en parallèle de celles générées lors de la recherche. Cette population de solutions est appelée « archive ». L'archive peut ne pas être utilisée dans le processus de génération de solutions, on parle alors d'archive « passive » ou au contraire être régulièrement solicitée pour l'obtention de nouvelles solutions (archive « active »). Comme tout utilisation d'élitisme, il est important, lors de l'utilisation d'archive active, de prévenir la convergence prématurée de l'algorithme. La taille de l'archive peut être fixe ou non. La mise à jour des archives est principalement basée sur la notion de dominance et il est possible d'inclure un mécanisme de diversification comme c'est le cas pour les solutions générées lors de la recherche.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Il est important de noter que la plupart des méthodes d'optimisation multi-objectif sont en fait des métaheuristiques à base de population de solutions sous forme d'algorithmes génétiques, de PSO [5] ou de recherches locales (à population) [3, 103]. Il existe cependant des méthodes à base de solution unique mais à ce jour, elles restent moins efficaces.

En mono-objectif, les métaheuristiques à base de solution unique donnent au contraire de bons résultats et sont souvent utilisées comme première approche de résolution pour un problème.

1.2 Les métaheuristiques à base de solution unique

Historiquement, les métaheuristiques à base de solutions uniques sont apparues à peu près en même temps que les algorithmes évolutionnaires comme on peut le voir sur la figure 1.11.

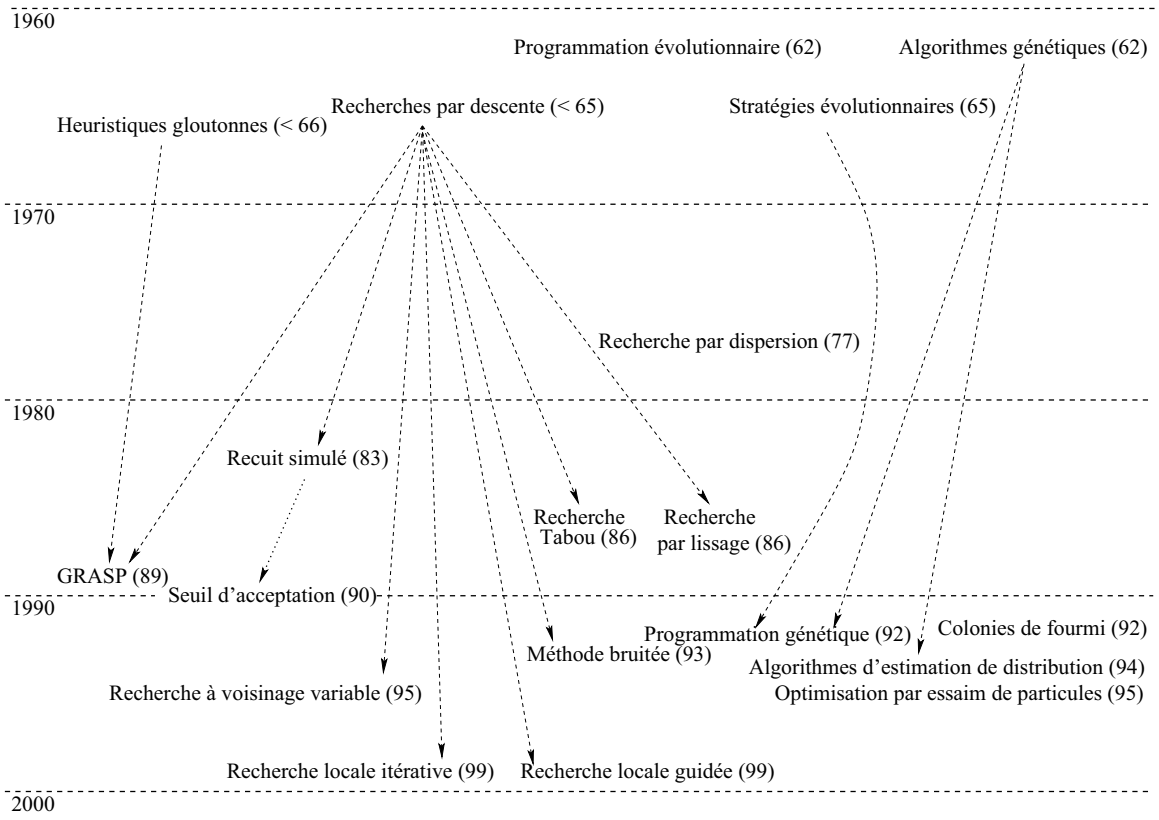


FIG. 1.11 – Historique d'apparition des différentes méthodes d'optimisation.

Elles permettaient de définir aisément des méthodes avec un fort pouvoir d'intensification qui permettent d'obtenir des solutions de bonnes qualités rapidement. Depuis la recherche par descente, les métaheuristiques à base de solution unique ont beaucoup évolué et sont capables aujourd'hui de combiner des phases de diversification et d'intensification et d'utiliser des voisinages multiples.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

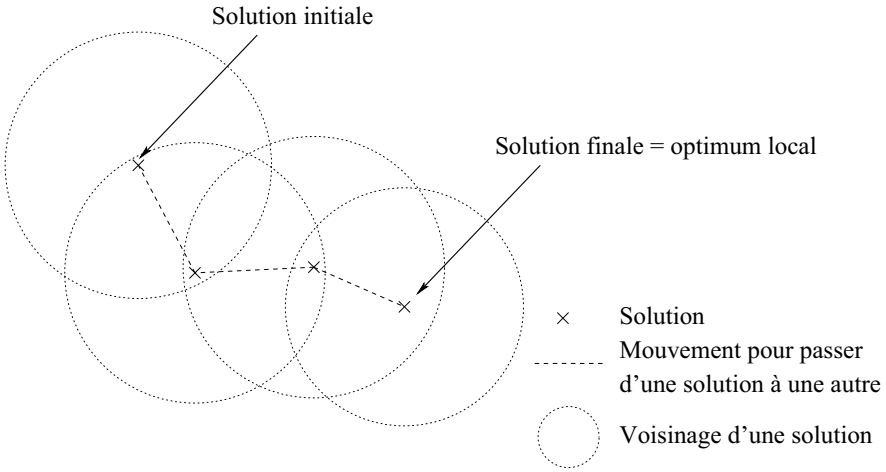


FIG. 1.12 – Processus d'une recherche par descente.

1.2.1 La recherche par descente

La recherche par descente (Hill Climbing) est une des métaheuristiques à base de solution unique la plus utilisée. Elle est facile à mettre en place et donne de bons résultats [101, 102]. Comme toute méthode de recherche à base de solution unique, le but est d'améliorer la qualité (la « fitness », la valeur objectif) d'une solution par exploration de son voisinage. Dans la recherche par descente, tant qu'un voisin de meilleure qualité que la solution courante existe, on utilise ce voisin comme base d'une nouvelle exploration. La figure 1.12 schématise ce processus qui est aussi détaillé par l'algorithme 1.

Algorithme 1 : La recherche par descente.

Données : Une solution initiale « Sol ».
Résultat : Une solution meilleur ou égale à « Sol ».
1 **tant que** *critère d'arrêt non vérifié* **faire**
2 | génération du voisinage de Sol;
3 | **si** *il existe un meilleur voisin* **alors**
4 | | Sol = meilleur voisin
5 | **fin**
6 **fin**
7 **retourner** *Sol*;

Ce processus est déterministe, c'est-à-dire que pour un type de voisinage donné et une même solution initiale, le résultat donné sera forcément le même : un optimum local (qui peut être l'optimum global). Des variations basées sur la modification du principe de sélection du voisin permettent de « casser » le déterminisme de la méthode. Le choix du voisin peut être aussi bien stochastique que basé sur une exploration incomplète du voisinage (premier voisin améliorant). De plus, le choix de la solution initiale restreint la descente dans un sous-espace de l'espace de recherche comme cela est montré dans la figure 1.13. Une fois le processus de descente démarré, la méthode ne remet jamais en compte les choix faits précédemment, par conséquent la descente reste concentrée sur une zone précise de l'espace de recherche.

[5] Particle Swarm Optimization : optimisation par essaim de particules.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

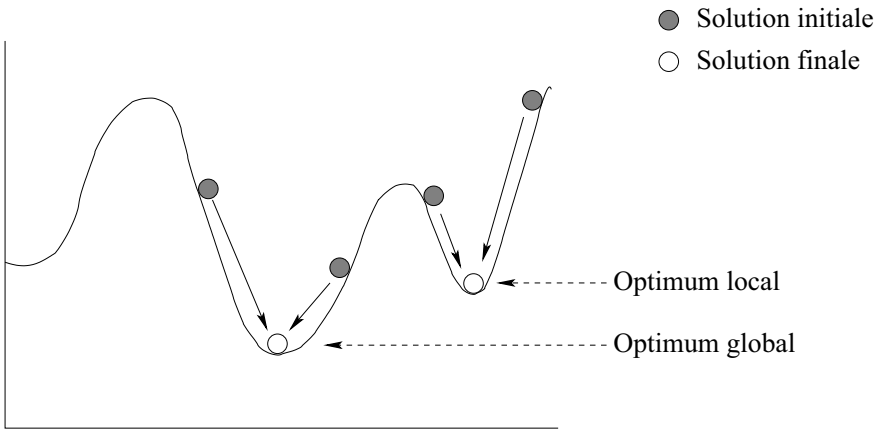


FIG. 1.13 – Impact du choix de la solution initiale sur le résultat d'un recherche par descente.

1.2.2 La Recherche Tabou

L'algorithme de Recherche Tabou a été proposé par Glover [50] et est devenu populaire dans les années 90. A la manière d'une méthode par descente, la voisinage est exploré de manière complète et dès qu'une meilleure solution est trouvée, elle remplace la solution courante. Cependant, lorsqu'il n'y a plus de meilleures solutions dans le voisinage, le meilleur voisin remplace la solution courante même si celui-ci est moins bon puis la recherche recommence. Le risque dans ce type d'approche est l'apparition de cycles. C'est pour cela que la Recherche Tabou possède une liste taboue afin de mémoriser les mouvements déjà effectués dans le voisinage. Cependant, plus cette liste est grande, plus elle prend de place en mémoire et plus il est coûteux de vérifier qu'un voisin appartient déjà à cette liste. C'est pour cela, que les listes taboues sont généralement de taille fixe. De plus, il peut arriver qu'un mouvement particulier, qui appartient à la liste taboue puisse cependant permettre l'obtention d'une meilleure solution. C'est pour cela que la Recherche Tabou possède aussi un critère d'aspiration. Ainsi, une solution est finalement acceptée si le mouvement qui l'a générée n'est pas tabou ou si le critère d'aspiration est vérifié. L'algorithme 2 détaille en pseudo-code l'algorithme classique de Recherche Tabou.

Algorithme 2 : Algorithme de la Recherche Tabou.

Données : Une solution initiale « Sol »
Résultat : Une solution meilleure ou égale « Sol »
1 Initialisation des mémoires à court, moyen et long terme;
2 **répéter**
3 Sol' = meilleur voisin de Sol;
4 Mise à jour de la liste taboue, du critère d'aspiration et des mémoires à moyen et long terme;
5 **si** *critère d'intensification est vérifié* **alors**
6 intensification;
7 **fin**
8 **si** *critère de diversification est vérifié* **alors**
9 diversification;
10 **fin**
11 **jusqu'à** *ce que le critère d'arrêt soit vérifié* ;
12 **retourner** *la meilleure solution trouvée* ;

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Un algorithme de Recherche Tabou comporte trois types de mémoire :

1. La mémoire à court terme : il s'agit de la liste taboue traditionnelle. Le principe est d'éviter de faire des mouvements inverses pour ne pas regénérer des solutions déjà rencontrées. La taille des listes taboues est une donnée critique. C'est pour cela qu'elle peut être fixe ou variable voir même adaptative. Dans ce dernier cas, la taille évolue en fonction des solutions qu'elle contient et du voisinage utilisé.

2. la mémoire à moyen terme : cette mémoire traduit le principe d'intensification pour la Recherche Tabou. Le but est de permettre de tirer la recherche vers des solutions ayant de bonnes propriétés. Le problème est bien sûr de trouver ses bonnes propriétés. Ce type de mémoire est forcément dédié au problème étudié et il faut pouvoir extraire des meilleures solutions trouvées l'information commune qui va guider la recherche.

3. la mémoire à long terme : il s'agit du mécanisme de diversification. Le principe est de pouvoir diriger la recherche vers les régions non encore explorées de l'espace de recherche. Cette étape de diversification est appliquée de temps en temps pour permettre de relancer la recherche.

L'utilisation de ces trois mémoires n'améliore pas à coup sûr la recherche. En effet, l'efficacité des processus d'intensification et de diversification sont intimement liés à la forme du paysage de la fonction d'évaluation utilisée. En effet, si le paysage ressemble à une vallée, un processus de diversification n'a pas d'intérêt et inversement si le paysage est un ensemble d'optima locaux, la phase d'intensification risque d'astreindre la recherche dans certaines zones de l'espace. Donc une étude préalable du paysage du problème doit permettre des choix intelligents du paramétrage de ce type d'algorithme.

1.2.3 Le recuit simulé

L'algorithme de recuit simulé trouve ses origines dans les travaux Kirkpatrick [74] et Cerny [21]. Il doit son succès par son efficacité aussi bien sur des problèmes combinatoires que continus. Le recuit simulé retranscrit le processus physique permettant d'obtenir des structures cristallines. A partir d'une température élevée, un refroidissement contrôlé permet d'abaisser la température jusqu'à l'obtention d'un état d'équilibre. La température initiale et la vitesse de refroidissement sont des critères essentiels de ce processus. Dans l'algorithme de recuit simulé, la température intervient dans la probabilité d'accepter des voisins de moins bonne qualité que la solution courante. Généralement cette probabilité suit la probabilité de Boltzmann :

$$P = e^{-\frac{f(sol')-f(sol)}{T}}$$

$f(x)$ est la valeur de qualité de la solution x et T la température courante. Ainsi, quand le meilleur voisin n'est pas de meilleure qualité que la solution courante, plus grande est la température, plus grande sera la probabilité de l'accepter. Par conséquent, le recuit simulé peut être vu comme un processus d'exploration qui petit à petit tend vers un processus d'intensification à la manière d'une recherche par descente. L'algorithme 3 donne le pseudo-code d'un recuit simulé.

La force du recuit simulé tient dans sa capacité à sortir des optima locaux. La figure 1.14 montre comment la température peut influencer la sortie d'optima locaux. Au début de l'algorithme, il est plus aisé de sortir d'un optimum local (phase d'exploration) mais plus la température descend, plus il devient

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Algorithme 3 : Algorithme de la recherche par recuit simulé.

Données : Une solution initiale « Sol », une température initiale « T ».
Résultat : Une solution meilleure ou égale à « Sol ».

```
1  T = T_max  répéter
2      répéter
3          Sol' = un voisin aléatoire de Sol;
4          δE = qualité de Sol' - qualité de Sol;
5          si δE ≤ 0 alors
6              Sol = Sol';
7          sinon
8              Sol = Sol' avec la probabilité e^(-δE/T)
9          fin
10     jusqu'à ce que l'état d'équilibre soit atteint ;
11     mise à jour de la température;
12 jusqu'à ce que le critère d'arrêt soit vérifié (T < T_min) ;
13 retourner Sol ;
```

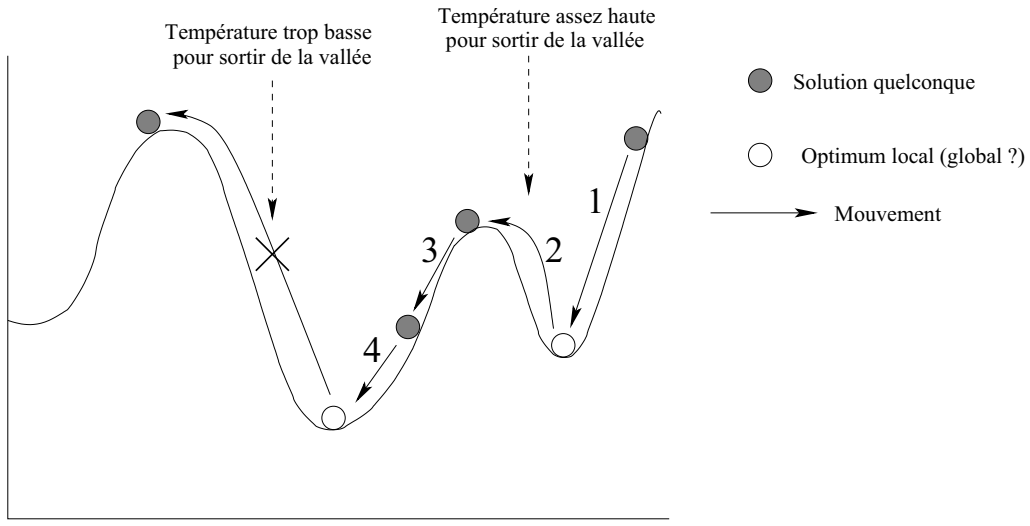


FIG. 1.14 – Exemple de cheminement au sein d'un recuit simulé.

difficile de continuer d'explorer les différentes zones de l'espace de recherche quitte à rester autour d'un optimum local (phase d'intensification).

Le recuit simulé est un processus qui peut se révéler coûteux notamment à cause de la génération de nombres aléatoires et l'utilisation d'exponentiels. Par conséquent, inspirées du recuit simulé, d'autres méthodes ont vu le jour pour permettre une résolution plus rapide sans pour autant perdre la qualité de solution obtenue grâce au recuit simulé. On peut citer la méthode à seuil d'acceptation (« threshold accepting ») [40], le « record-to-record travel » et le « great deluge algorithm » [39] ou encore les « demon algorithms » [135].

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.2.4 La recherche locale itérative

La recherche locale itérative est un modèle de recherche locale qui améliore le principe de recherche locale multi-départ (« multiple start local search ») dans lequel des méthodes de descente sont lancées successivement sur des solutions initiales générées aléatoirement pour contrer l'aspect déterministe de la descente. Ce type de recherche locale a été la première fois utilisé dans [89] mais a été clairement défini dans [82, 120]. Le principe est simple, une recherche par descente est appliquée sur une solution initiale pour générer une meilleure solution. On applique ensuite une nouvelle recherche par descente sur cette nouvelle solution après l'avoir « perturbée ». La solution obtenue est comparée avec la solution initiale pour savoir si elle la remplace ou non. Tout cela représente une itération de la recherche locale itérative. L'algorithme 4 présente en pseudo-code une recherche locale itérative classique.

Algorithme 4 : Algorithme de la recherche locale itérative.

Données : Une solution initiale.
Résultat : Une solution meilleure (ou au moins égale) à « Sol ».
1 Sol_* = résultat d'une recherche locale sur Sol;
2 **répéter**
3 Sol' = résultat d'une perturbation de Sol_*;
4 Sol'_* = résultat d'une recherche locale sur Sol';
5 Sol_* = Sol_* ou Sol'_* selon un critère d'acceptation;
6 **jusqu'à** *ce que le critère d'arrêt soit vérifié* ;
7 **retourner** Sol_* ;

La perturbation est bien souvent un mécanisme plus « intelligent » qu'une simple génération aléatoire de solutions. De ce fait, elle doit modifier suffisamment une solution pour pouvoir accéder à d'autres zones de l'espace de recherche tout en conservant un minimum de caractéristiques qui font la qualité de la solution initiale. Bien sûr, ce principe de perturbation dépend fortement du paysage du problème étudié. Le but étant d'être capable d'obtenir des perturbations intelligentes qui permettent de tendre vers les zones non encore explorées contenant des optima locaux de meilleures qualités voire peut-être même d'atteindre la solution optimale.

Le mécanisme de comparaison qui permet d'accepter ou non une solution obtenue permet aussi d'axer la recherche vers un principe d'intensification ou de diversification. L'intensification se traduit par une acceptation d'une solution seulement si elle améliore la solution courante. Une acceptation constante peut traduire une recherche totalement diversifiée. Il s'agit de deux extrêmes. Le but est de trouver le bon compromis.

1.2.5 Le recherche à voisinage variable

Cette recherche locale, comme la recherche locale itérative, fait partie des méthodes plus récentes basées sur l'exploration successive de plusieurs voisinages pour obtenir des solutions de meilleure qualité [94]. Réussir à trouver un optimum local pour plusieurs voisinages augmente la probabilité de trouver l'optimum global. En effet, l'optimum global est aussi un optimum local sur un voisinage donné. L'algorithme 5 présente en pseudo-code une recherche à voisinage variable classique. L'examen des différents voisinages possibles se fait de manière déterministe selon l'ordre fixé au démarrage de la méthode. Dès que l'optimum local d'un voisinage est obtenu dans un voisinage donné et donc qu'il n'y a plus d'amélioration possible, on passe au voisinage suivant. Dès qu'un nouvel optimum local est trouvé dans un voisinage donné, on recommence la recherche d'optimum local à partir du premier voisinage. Cette

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

phase déterministe se finit dès qu'une solution est un optimum local pour tous les voisinages. Cette aspect déterministe est aussi appelé descente par voisinage variable. A la manière de la recherche locale itérative, un mécanisme de perturbation est appliqué entre deux applications de recherches locales pour essayer de sortir d'un optimum local dans un voisinage donné. C'est ce mélange de descente à voisinage variable et de perturbation entre chaque recherche qui amène à la recherche à voisinage variable. Comme pour les autres recherches locales, un compromis doit être mis en place entre l'aspect intensification de la recherche (recherche locale utilisée) qui apporte de meilleurs optima locaux et l'aspect diversification (mécanisme de perturbation) qui permet d'accéder à d'autres régions de l'espace de recherche.

Algorithme 5 : Algorithme de la recherche à voisinage variable.

Données : Une solution initiale « Sol », un ensemble de voisinage N_k pour $k \in [1, k_{max}]$.
Résultat : Une solution meilleure (ou au moins égale) à « Sol ».

```
répéter
    k=1;
    répéter
        Sol' = une solution aléatoire dans N_k;
        Sol" = résultat recherche locale sur (Sol');
        si Sol" est meilleure que Sol alors
            Sol = Sol";
            k = 1 ;
        sinon
            k = k+1;
        fin
    jusqu'à k = k_max ;
jusqu'à le critère d'arrêt soit vérifié ;
retourner Sol ;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.2.6 Unification des concepts

A partir des métaheuristiques à base de solution unique que nous avons présentées, il est possible de séparer les caractéristiques génériques entre chacune d'entre elles et celles dédiées à un problème donné. La tableau 1.2 résume l'ensemble des caractéristiques communes ou non à toutes ces méthodes.

Tab. 1.2 – Résumé des caractéristiques génériques et dédiées de cinq métaheuristiques à solution unique.

Métaheuristique	**Principes génériques**	**Principes spécifiques (à un problème)**
Méthode par descente	Sélection d'un voisin	Evaluation d'une solution Génération du voisinage
Recherche Tabou	Critère d'arrêt Critère d'aspiration Liste taboue	Evaluation d'un solution Génération du voisinage
Recuit simulé	Système de refroidissement Critère d'arrêt	Evaluation d'une solution Génération du voisinage
Recherche locale itérative	Critère d'arrêt Critère d'acceptation Intensification	Evaluation d'une solution Perturbation
Recherche à voisinage variable	Parcours des voisinages Intensification	Evaluation d'une solution Génération des voisinages

Ce tableau permet de rapidement déterminer ce qu'il convient de définir spécifiquement pour un problème donné de ce qui est générique (et donc réutilisable). Comme nous le verrons plus loin dans ce chapitre, c'est sur ce type de généralisation que ParadisEO-MO, la plateforme de développement de métaheuristiques à base solution unique, a pu être mis en place

Il existe encore de nombreuses recherches locales tel que la recherche locale guidée [132] ou les méthodes dites bruitées [22]. Mais après avoir évoqué les métaheuristiques à base de solution unique et leur grand pouvoir d'intensification, nous allons maintenant nous attarder sur celles à base de populations de solutions et leur capacité à parcourir les grands espaces de recherche.

1.3 Les métaheuristiques à base de population de solutions

Nous allons maintenant détailler les algorithmes qui ne proposent plus une seule mais tout un ensemble de solutions potentielles pour un problème donné.

1.3.1 Les algorithmes génétiques

Les algorithmes génétiques modifient de manière itérative une population de solution en appliquant des opérateurs génétiques [64]. Leur principe s'inpire du principe d'évolution de Darwin [32]. Ce type d'algorithme permet d'imiter l'évolution d'une population dans laquelle l'information va pouvoir être partagée par les individus pour en créer de nouveaux. Des modifications de l'information peuvent parfois avoir lieu de manière spontanée au sein de la population. Et enfin, le principe de sélection naturelle tend à garder les individus les mieux adaptés à leur environnement. Les algorithmes génétiques reprennent ces principes à travers les opérateurs de recombinaison, de mutation et de sélection (voir schéma 1.15).

A partir d'une population initiale et tant qu'un critère d'arrêt n'est pas vérifié, l'algorithme va procéder à la création de nouveaux individus à partir de la population courante. Pour ce faire, un certain

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

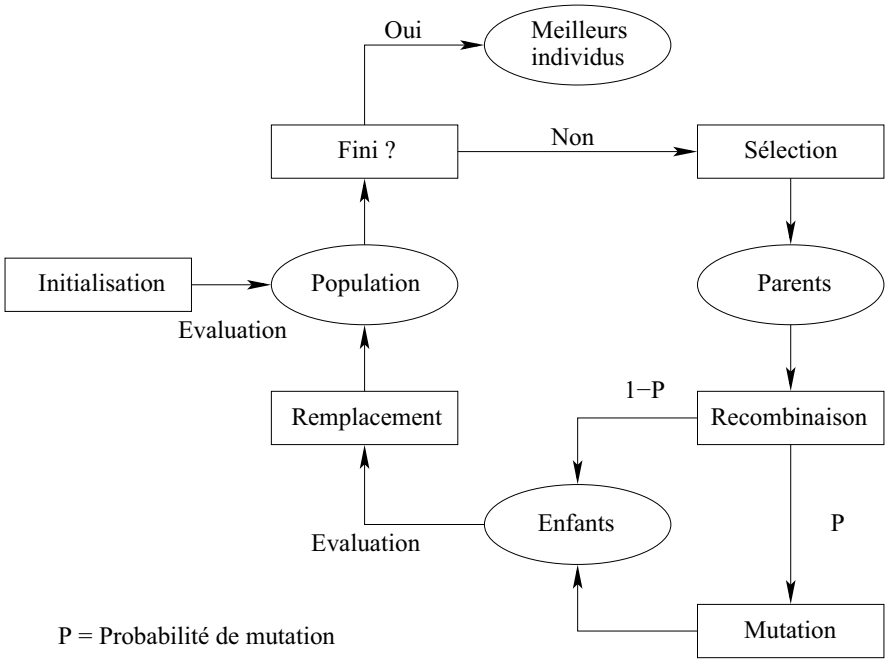


FIG. 1.15 – Schéma global d'un algorithme génétique.

nombre d'individus de la population est sélectionné selon une stratégie de sélection donnée, on les appelle les « parents ». L'opérateur de recombinaison va permettre à partir de ces parents d'obtenir des individus dont l'information génétique consistera en un mélange de l'information présente dans les deux parents : les « enfants ». En plus de ce brassage génétique, de l'information inattendue peut être obtenue grâce à l'opérateur de mutation. Il permet d'obtenir des individus ayant des caractéristiques qui ne pourraient être obtenues par simple recombinaison. Finalement, au terme de l'application des opérateurs de sélection, recombinaison et mutation ; on procède au remplacement de la population selon une stratégie donnée à partir des nouveaux individus créés en les comparant à la population courante.

1.3.2 La recherche par dispersion

La recherche par dispersion (« scatter search ») se base sur la connaissance du problème étudié pour créer des solutions issues de combinaisons de solutions existantes [49]. De ce fait, elle se démarque des algorithmes génétiques qui eux se basent sur des processus stochastiques. Le principe est de générer des solutions diversifiées qui sont ensuite améliorées. De ces solutions améliorées, on extrait un ensemble de solutions de référence qui servira de base pour la génération de nouvelles solutions issues de la combinaison de solutions de l'ensemble de référence. Des variantes des recherches par dispersion ont amené aux algorithmes de « path relinking » [51] dans lesquels des chemins reliant les solutions de qualité sont explorés afin de sélectionner celles qui seront améliorées par des mécanismes d'intensification. Il existe encore d'autres familles d'algorithmes à base de population de solutions comme les colonies de fourmi [38] ou les essaims de particules [73] par exemple.

Comme nous avons pu le voir, que ce soit pour les métaheuristiques à base de solution unique ou celles à population de solutions, le but est toujours de trouver la méthode qui permettra de parcourir efficacement le paysage du problème pour tendre le plus rapidement possible vers la solution optimale (mono-objectif) ou le front Pareto optimal (multi-objectif). Lorsqu'une seule métaheuristique n'est plus

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

capable d'être améliorée, il est toujours possible de la combiner avec d'autres pour aller encore plus loin, c'est le principe de coopération de métaheuristiques.

1.4 La coopération de métaheuristiques

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

1.4.1 Notion de coopération

Pour résoudre un problème d'optimisation donné, on choisit une méthode qui paraît bien adaptée parmi celles existantes (méthode exacte, heuristiques, ...). Ensuite on essaie de l'améliorer en travaillant sur ces paramètres afin d'obtenir la méthode la plus efficace possible. Si c'est possible, on évalue sa qualité en la comparant à d'autres méthodes proposées pour le problème étudié. Malheureusement, d'après les « No Free Lunch Theorems » [134], il n'existe pas de méthode d'optimisation qui, pour un problème donné, sera meilleure que toutes les autres sur toutes les instances possibles. Dans la pratique, il existera toujours des instances pour lesquelles une méthode sera mise en défaut.

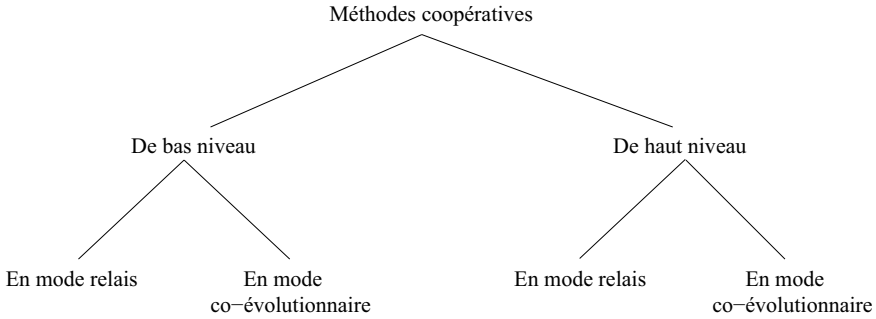


Fig. 1.16 – Hiérarchie des méthodes coopératives.

Quelque soit la méthode choisie, elle présente des avantages et des inconvénients. La coopération peut permettre de trouver de bons compromis où les avantages et défauts de chaque méthode se compensent avec celles avec lesquelles elle coopère. Comme le montre la figure 1.16, nous nous baserons sur une taxonomie hiérarchisée présentée dans [123] pour classifier les différents types de coopération. Dans celle-ci, les critères hiérarchiques pour discriminer les différentes coopérations sont le niveau et le mode de coopération.

1.4.2 Niveaux de coopération

Dans la taxonomie choisie, le premier point pour classifier les coopérations est le niveau de coopération. Il peut être bas (« Low-Level ») ou haut (« High-Level »). Dans le niveau bas de coopération, une méthode devient un bloc fonctionel d'une autre méthode qui l'englobe. A l'inverse, dans le niveau haut de coopération, les méthodes, même si elles peuvent communiquer entre elles, gardent leur intégrité propre. La figure 1.17 schématise la notion de niveau de coopération.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

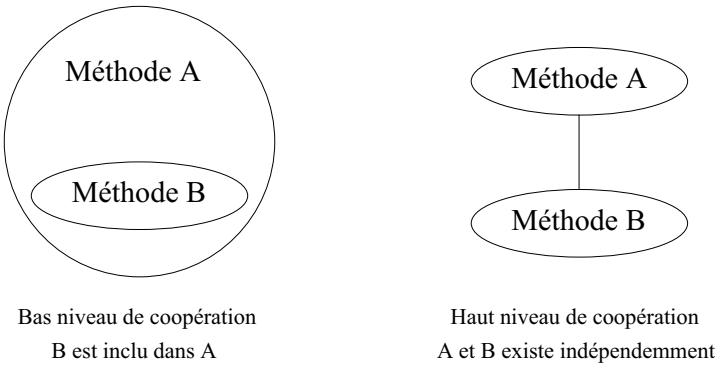


FIG. 1.17 – Les différents niveaux de coopération.

1.4.3 Modes de coopération

Le second point de classification est le choix du mode. Dans le mode relais, les méthodes d'optimisation sont appelées les unes à la suite des autres, le résultat de la précédente servant de point de départ à la méthode courante. On peut parler aussi d'une exécution séquentielle des méthodes. En mode co-évolutionnaire, les différentes méthodes d'optimisation fonctionnent en parallèle pour généralement pouvoir explorer des zones différentes de l'espace de recherche. La figure 1.18 schématise la notion de mode de coopération.

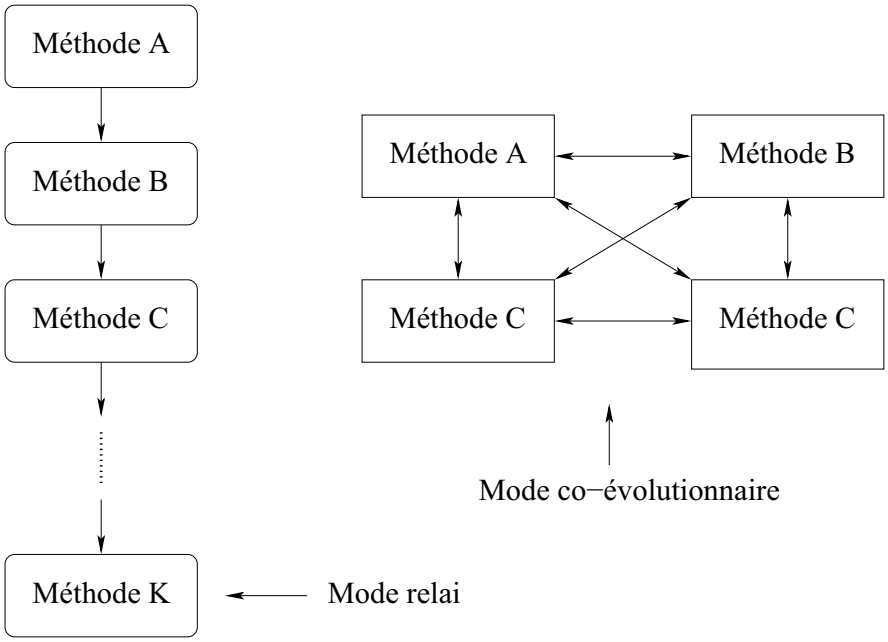


FIG. 1.18 – Les différents modes de coopération.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.4.4 Classes hiérarchiques

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

1.4.4.1 Coopération relais de bas niveau (« Low-level Relay Hybrid »)

Cette classe de coopération (**LRH**) regroupe les méthodes à base de solution unique dans lesquelles une autre méthode d'optimisation est insérée ce qui par conséquent en modifie le comportement. Dans ce type de coopération, les deux (ou N) méthodes utilisées perdent leur intégrité pour produire un nouvel algorithme. Un exemple de ce type de coopération est proposé dans [88] pour résoudre le problème du voyageur de commerce (« Traveling Salesman Problem »). Un recuit simulé est combiné avec une méthode par descente pour pouvoir générer à partir d'un optimum local courant, un nouvel optimum local différent qui soit un candidat pour le recuit simulé. Cela consiste en une perturbation (un « kick » dans [88]) de l'optimum local pour pouvoir lui appliquer par la suite la recherche par descente pour obtenir le nouvel optimum local. Afin de s'assurer de ne pas retomber sur le même optimum local, il peut être nécessaire d'utiliser une perturbation fonctionnant dans un autre voisinage que celui utilisé par le recuit simulé. Il est à remarquer que ce type de coopération rappelle le principe des recherches locales à voisinage multiple tel que celui de la recherche locale itérative (« ILS ») ou de la recherche à voisinage variable (« VNS »).

1.4.4.2 Coopération co-évolutionnaire de bas niveau (« Low-level Teamwork Hybrid »)

Dans cette classe de coopération (**LTH**), on regroupe les métaheuristiques à base de population dans lesquelles un (ou plusieurs) opérateur(s) est (sont) remplacé(s) par une (ou des) méthode(s) d'optimisation. Les algorithmes mimétiques sont un bon exemple de ce type de coopération. Il s'agit d'algorithme génétique dans lequel un opérateur est remplacé par une métaheuristique à base de solution unique. L'avantage de ce type de coopération est de combiner le pouvoir d'exploration des algorithmes à base de population de solutions et celui d'intensification des méthodes à base de solution unique. En effet, les algorithmes à base de population de solutions permettent de mieux appréhender les grands espaces de recherche mais n'ont pas un fort pouvoir d'intensification et peuvent avoir tendance à converger lentement. Inversement, les métaheuristiques à base de solution unique ont une très grande capacité d'intensification mais cela peut entraîner le blocage de la méthode dans des optima locaux suivant la forme du paysage. Par conséquent la coopération de ces deux familles de métaheuristiques permet d'obtenir des algorithmes ayant un grand pouvoir d'exploration et d'intensification. Dans ce type de coopération, généralement l'opérateur de mutation est remplacé, soit par une recherche par descente [121], soit par une Recherche Tabou [43] ou encore par un recuit simulé [14]. Il existe aussi des coopérations où l'opérateur de recombinaison est remplacé par une méthode d'optimisation de type algorithme glouton [52].

1.4.4.3 Coopération relais de haut niveau (« High-level Relay Hybrid »)

Au sein de cette classe de coopération (**HRH**), les méthodes d'optimisation gardent leur intégrité et sont utilisées de manière séquentielle. La (ou les) solution(s) finale(s) de la première méthode devient(deviennent) la (ou les) solution(s) initiale(s) de la méthode suivante. Cette enchaînement de méthodes peut notamment permettre de procéder à une phase de diversification pour couvrir une grande partie de l'espace de recherche (algorithmes génétiques) avant d'arriver à une phase d'intensification pour améliorer les solutions obtenues (recherches locales). Dans [125], un algorithme génétique est suivi d'une Recherche Tabou afin d'améliorer les solutions obtenues. Il est aussi répandu d'utiliser des méthodes rapides et peu coûteuses de type glouton pour générer des solutions initiales de bonne qualité à une

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

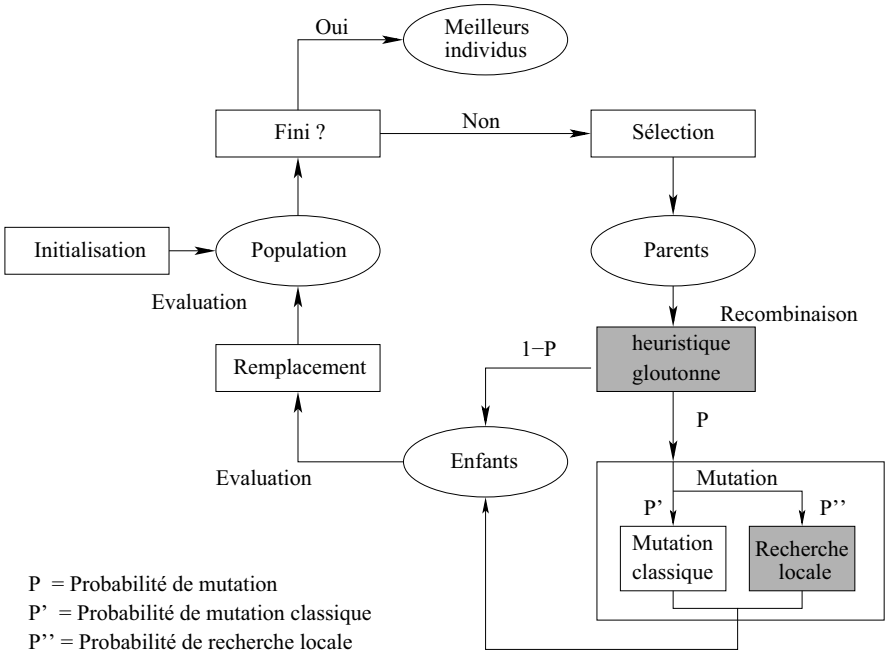


FIG. 1.19 – Schéma d'une coopération de type co-évolutionnaire de bas niveau (« LTH ») : un algorithme génétique est hybridé avec une heuristique gloutonne (recombinaison) et une recherche locale (mutation).

autre méthode plus coûteuse de type métaheuristique à base de population de solutions. La figure 1.20 montre un exemple d'enchaînement de trois méthodes d'optimisation différentes.

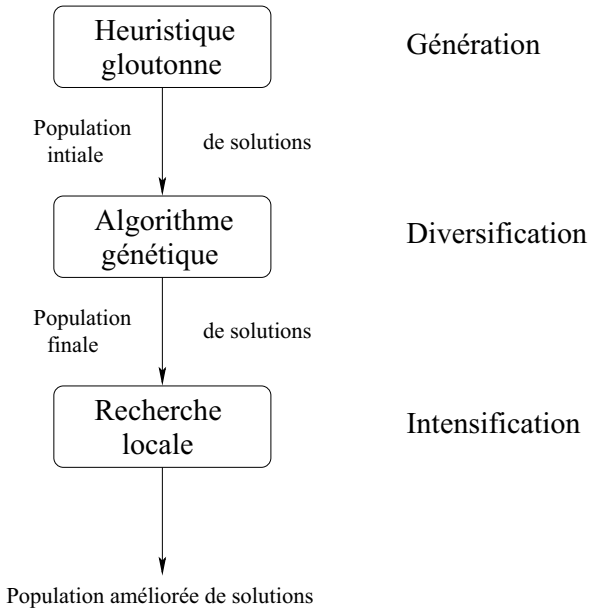


FIG. 1.20 – Schéma d'une coopération de type relais de haut niveau (« HRH ») : une heuristique glouton génère la population initiale d'un algoritme génétique et ensuite la population finale de l'AG est améliorée grâce à l'application d'une recherche locale sur chaque individu de la population.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1.4.4.4 Coopération co-évolutionnaire de haut niveau (« High-level Teamwork Hybrid »)

Cette classe de coopération (**HTH**) correspond à l'exécution en parallèle de méthodes d'optimisation qui peuvent communiquer entre elles durant leur exécution. Un exemple de ce type de coopération est l'exécution d'un algorithme génétique en utilisant un modèle insulaire de parallélisation. Dans ce modèle, la population initiale est divisée en sous-population qui sont réparties entre différentes « îles » sur lesquelles un algorithme génétique est lancée. Ainsi, chaque île correspond à une recherche de solutions dans une partie de l'espace de recherche. De manière périodique, un certain nombre d'individus migrent entre les îles afin que les sous-populations des différentes îles contribuent toutes à produire les meilleures solutions possibles. En effet, de l'information extérieure peut permettre à un algorithme génétique qui n'évoluait plus de pouvoir être relancé afin de tendre vers les meilleures solutions possibles. La figure 1.21 détaille un modèle insulaire d'exécution d'AG dans le lequel la migration d'individu se fait de manière circulaire entre les îles.

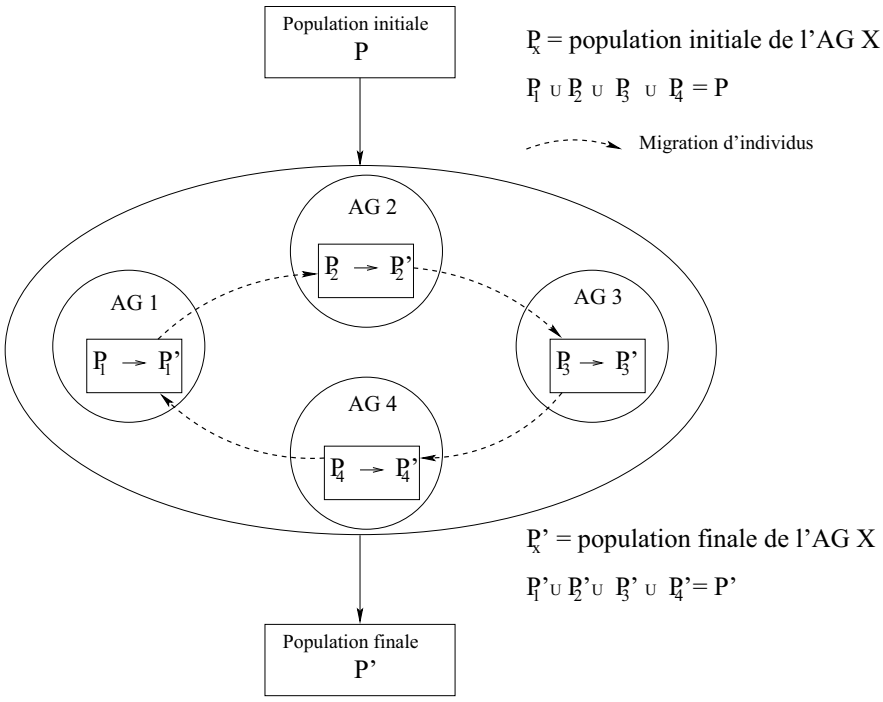


Fig. 1.21 – Schéma d'une coopération co-évolutionnaire de haut-niveau (« HTH ») : un ensemble d'algorithmes génétiques coopèrent selon un modèle insulaire incluant une migration circulaire des individus.

Les travaux basés sur cette classe de coopération utilisent aussi bien des algorithmes génétiques [126], des recuits simulés [34], des Recherches Tabou [35], de la programmation génétique [77] ou encore des stratégies évolutionnistes [130].

La coopération de métaheuristiques permet d'obtenir des algorithmes hybrides de grande efficacité mais la mise au point de telles méthodes a un coût. Il faut mettre au point les métaheuristiques, choisir le mode de coopération le plus adapté ou en tester plusieurs. Par la suite, on peut vouloir changer de métaheuristiques. C'est un travail long et fastidieux. Mais quand on regarde de plus près, un grand nombre de caractéristiques liées à la mise en place de métaheuristiques et à leur coopération est tout à fait générique et non dédié à un problème particulier. C'est sur ce principe que la plateforme ParadisEO

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

a été développé : permettre à l'utilisateur de ne s'intéresser qu'aux parties de ses méthodes de résolution spécifiques à son problème tout en ayant accès à toutes les métaheuristiques actuelles.

1.5 ParadisEO-MO : une plateforme pour la conception de métaheuristiques à base de solution unique

Dans cette partie, nous présenterons la plateforme ParadisEO dans son ensemble avant de nous concentrer sur ParadiSEO-MO et ses caractéristiques.

1.5.1 Présentation de la plateforme ParadisEO

Nous rappelons ici l'histoire de la plateforme de ces débuts, il y a presque 10 ans, jusqu'à sa version actuelle.

1.5.1.1 Historique

ParadisEO est à l'origine une plateforme pour la conception et le déploiement de métaheuristiques parallèles hybrides sur clusters et grilles ; fruit du travail de thèse de Sébastien Cahon [17, 18]. En effet, elle permettait la parallélisation des méthodes développées sur la plateforme **E**volving **O**bjects (EO).

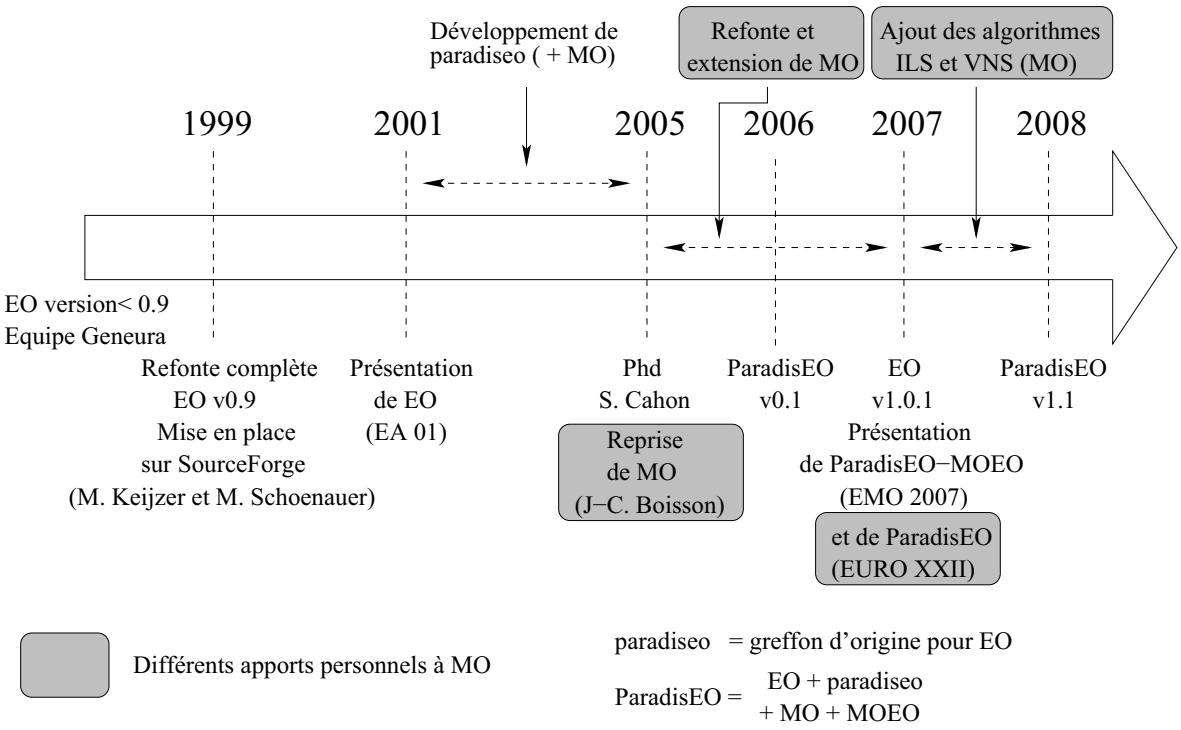


FIG. 1.22 – Chronologie de ParadisEO centrée sur MO : de la première version de EO jusqu'à ParadisEO v1.1.

En suivant la chronologie détaillée par la figure 1.22, on peut voir qu'EO était une plateforme qui a vu le jour à la fin des années 90. EO était (et est toujours) une plate-forme C++ sous licence « GPL » (**G**eneral **P**ublic **L**icence) dédiée à la conception d'applications à base d'algorithmes évolutionnaires.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Elle était le fruit d'une collaboration européenne [72] dont les principales contributions furent de M. Keijzer [6], J.J. Merelo, G. Romero [7] et M. Schoenauer [8]. Elle était définie comme libre de tout paradigme et couvrant les différentes familles d'algorithmes à base de population de solutions : les algorithmes génétiques, les stratégies évolutionnistes, la programmation évolutionnaire et génétique. Un des points fort d'EO était sa flexibilité dans la représentation des solutions. En effet, il suffisait de définir le processus d'évaluation d'une solution et les opérateurs de variations qui y étaient associées (recombinaison, mutation, ...). La plateforme EO fournissait la plupart des représentations classiques ainsi que les mécanismes de variations associés. Elle permettait aussi d'avoir accès à de nombreuses fonctionnalités et services facilitant son exploitation : une visualisation des résultats, le traitement en ligne des paramètres, des mécanismes pour la sauvegarde ou reprise de l'état d'exécution (« checkpointing »), la génération automatique de statistiques, ... La plateforme EO a été exploitée dans divers domaines d'application où des représentations complexes ou opérateurs de variation spécifiques ont dû être mis en oeuvre. On peut citer des exemples liés à l'optimisation de l'architecture de perceptrons multi-couches [20], à l'optimisation de contrôleurs à logique floue [110], à l'ingénierie de voitures [113], à la segmentation de voix [92] et d'image [119], ...

C'est sur cette base que ParadisEO s'est greffé pour permettre à tous les algorithmes développés sous EO d'être utilisables de manière (quasi-)invisible pour l'utilisateur sous des environnements distribués. Pour ce faire, un énorme travail de conception a été nécessaire et a permis d'obtenir différentes versions de ParadisEO suivant le système de message utilisé (MPI, lam, ...) et l'environnement utilisé (cluster, grille hétérogène, ...). Pour simplifier, à l'époque, ParadisEO était donc seulement une « parallélisation » de EO. Avec le temps, EO a été enrichi pour permettre notamment de tenir compte l'optimisation multi-objectif. Cependant, inclure de tels concepts dans EO engendrait forcément des difficultés de maintenance pour les développeurs mais surtout une obligation de compréhension plus poussée de la plateforme pour les utilisateurs (même si des implémentations déjà prêtes et des tutoriaux existaient pour l'aider). Ainsi, il parut clair que mélanger trop de concepts au sein d'une même entité n'était pas forcément la meilleure solution. Comme ParadisEO était dédié au parallélisme, d'autres entités ont vu le jour pour couvrir d'autres domaines de l'optimisation autres que les algorithmes à base de population de solutions. Sébastien Cahon a notamment mis en place une première version d'un greffon dédié aux métaheuristiques à base de solution unique intitulés MO pour « **M**oving **O**bjects ». Fin 2005, il existait donc EO, ParadisEO et aussi MO (même si celui-ci n'était pas connu). Par la suite, et avec l'accord des responsables du projet EO, ParadisEO ne désigna plus le greffon originel pour EO mais une plateforme complète pour la mise en place de métaheuristiques quelles soient séquentielles, parallèles, à base de solution unique, de population de solutions, hybrides, ... Cette plateforme est constituée des trois blocs d'origine : EO devenu ParadisEO-EO [9], MO devenu ParadisEO-MO et ParadisEO devenu ParadisEO-PEO ; auquel s'ajoute un module exclusivement dédié au multi-objectif grâce au travail de thèse d'Arnaud Liefooghe [80].

1.5.1.2 Etat actuel de la plateforme (version 1.1)

La figure 1.23 détaille l'architecture actuelle de la plateforme ParadisEO [10].

[6] Danish Hydraulic Institute

[7] GeNeura Team, Depto. Arquitectura y Tecnologia de Computadores, Universidad de Granada (Spain)

[8] Projet Fractales INRIA, LRI Paris-Sud

[9] Même si la plateforme Evolving Object existe toujours en tant que telle indépendamment à ParadisEO : plus d'informations sur *eodev.sourceforge.net.*

[10] Version actuelle téléchargeable sur http ://paradiseo.gforge.inria.fr

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

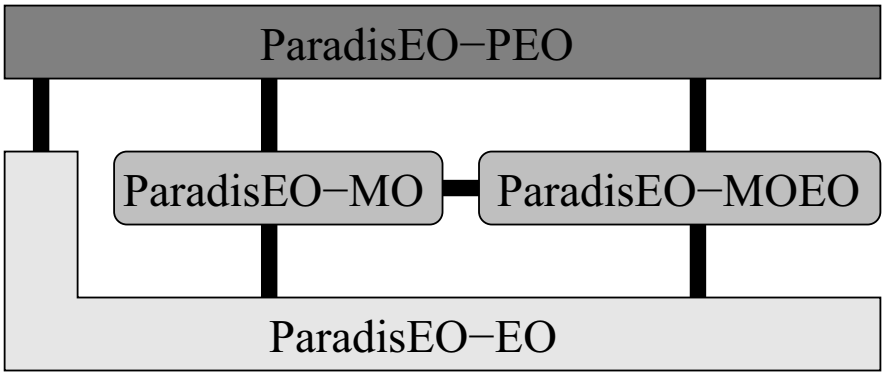


FIG. 1.23 – La plateforme ParadisEO. EO signifie « Evolving Object », MO « Moving Object », MOEO « Multi Objective Evolving Object » et PEO « Parallel Evolving Object ».

Aujourd'hui ParadisEO permet de mettre en place des métaheuristiques à base de solution unique, à base de populations de solutions, en mono ou multi-objectif, en séquentiel ou en parallèle sur tout type de configuration donnée (cluster, grille hétérogène). La tableau 1.3 propose un éventail, non exhaustif, des plateformes existantes pour développer des métaheuristiques afin de pouvoir comparer les caractéristiques de la plateforme par rapport à l'existant.

TAB. 1.3 – Liste des plateformes existantes pour développer des métaheuristiques. RL signifie recherches locales, AE algorithmes évolutionnaires, RD recherche par descente, AG algorithmes génétiques, SS « scatter search », SE stratégies évolutionnaires et PS métaheuristiques à population de solutions.

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

La plupart ont été mises en place pour répondre à un besoin et ont été ensuite proposées à la communauté. C'est pour cela qu'il existe notamment un grand nombre de plateformes dédiées au développement

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

de recherches locales car celles-ci représentent des méthodes simples à mettre en oeuvre et efficaces pour les problèmes d'optimisation mono-objectifs. Sur ce principe, chaque plateforme est à la base développée uniquement pour une thématique telle que les recherches locales mono-objectifs séquentielles, les algorithmes génétiques multi-objectif parallèles, ... mais aucune ne proposait une réelle uniformisation de toutes les méthodes actuelles. Face à cette situation ParadisEO représente une solution performante, flexible, et surtout suivie. En effet, la plupart des plateformes ont été initiée par des personnes qui ont ensuite essayé de maintenir celles-ci mais peu nombreuses sont celles qui font l'objet d'un réel suivi.

Il est clair que c'est le travail conjugué d'un ensemble de personnes qui a permis d'obtenir cette plateforme. Chaque partie de ParadisEO est sous la responsabilité d'une personne (ingénieur ou doctorant) mais tout le monde prend part à l'élaboration et à l'amélioration de l'ensemble de la plateforme. De plus, la plateforme est testée non seulement sur des problèmes scolaires mais est aussi utilisée pour la plupart de travaux de recherches de l'équipe nécessitant l'utilisation de métaheuristiques. Voici ce qu'il est possible de faire à l'heure actuelle avec ParadisEO :

- ParadisEO-EO : développement de métaheuristiques à base de population de solutions telles que les algorithmes évolutionnaires, les algorithmes génétiques, ... mais aussi des algorithmes d'optimisation par essaim de particles (« Particle Swarm Optimisation » ou PSO). Les PSOs sont un apport direct de l'équipe projet INRIA Dolphin dans le noyau EO original.
- ParadisEO-MO : développement de recherches locales appartenant à la plupart des familles de recherches locales actuellement connues : les méthodes par descente, les Recherches Tabou, les recuits simulés, les recherches locales itératives, les recherches à voisinage variable.
- ParadisEO-EO + ParadisEO-MO : mise en place de mécanisme d'hybridation de métaheuristiques à base de population avec une ou plusieurs recherches locales.
- ParadisEO-EO + ParadisEO-MOEO : conception d'algorithmes évolutionnaires multi-objectif.
- ParadisEO-MO + ParadisEO-MOEO : obtention de recherches locales multi-objectif.
- ParadisEO-X (X = EO, MO, MOEO) + ParadisEO-PEO : développement de métaheuristiques parallèles performantes sur tout type d'architecture.

Concernant le parallélisme, ParadisEO-PEO propose un grand nombre des modèles existants de parallélisation (maître/esclave, insulaire) avec autant de techniques de communication possible pour ces méthodes. Le tableau 1.4 présente, pour les plateformes du précédent tableau qui proposait du parallélisme, les différentes méthodes de communication utilisées pour la mise en place des modèles parallèles.

Tab. 1.4 – Architecture utilisée pour le parallélisme pour les plateformes le proposant.

Nom de la plateforme	**Architecture(s) possible pour le parallélisme**
GAlib	PVM
DREAM	Pair-à-Pair, sockets
MALLBA	MPI, Netstream
TEMPLAR	MPI, threads
JDEAL	sockets
ECJ	threads, sockets
Distributed BEAGLE	sockets
ParadisEO	MPI, PThread, Condor, Globus

Ce tableau montre combien ParadisEO est une plateforme exhaustive concernant le parallélisme.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Aujourd'hui, ParadisEO c'est :

1. Une plateforme open-source.
2. Une plateforme de développement utilisable sur les systèmes Unix, Windows et Macintosh.
3. Une documentation complète : code, méthode, concepts, ...
4. Plus de vingt tutoriaux.
5. Des formations proposées en France et à l'étranger.
6. Des utilisateurs de tous les domaines et de toutes les nationalités : étudiants, chercheurs, industriels, ...
7. Une plateforme à fonctionnement professionnel. Chaque partie de ParadisEO subit des tests unitaires. ParadisEO est versionné et a été testé sur tout un ensemble de stations Unix (Mandriva, Ubuntu, Fedora, ...), Windows (XP, Vista) et Mac (Leopard 10, ...) sur des architectures 32 et 64 bits. Une liste de diffusion d'aide existe et à tout problème, une réponse est apportée généralement dans les 48h.

Au sein de mon travail de recherche, j'ai pu utiliser l'ensemble des outils proposés par ParadisEO, je suis moi-même responsable de ParadisEO-MO dédié aux métaheuristiques à base de solution unique.

1.5.2 ParadisEO-MO

Pour bien comprendre mon apport concernant la plateforme, je vais rappeler comme elle était à l'origine avant de détailler tous les changements et extensions que j'y ai apporté.

1.5.2.1 La version originale

Le but est de découvrir la démarche de conception qui a permis d'obtenir ParadisEO-MO tel qu'elle était à l'origine. Aux travers des trois premières métaheuristiques à base de solution unique qu'elle proposait : le recherche par descente, la Recherche Tabou et le recuit simulé ; nous allons comprendre comment elle a été conçue. Le schéma 1.24 montre la hiérarchie des algorithmes sous ParadisEO-MO tel qu'elle était lorsque j'en ai pris la charge en 2005.

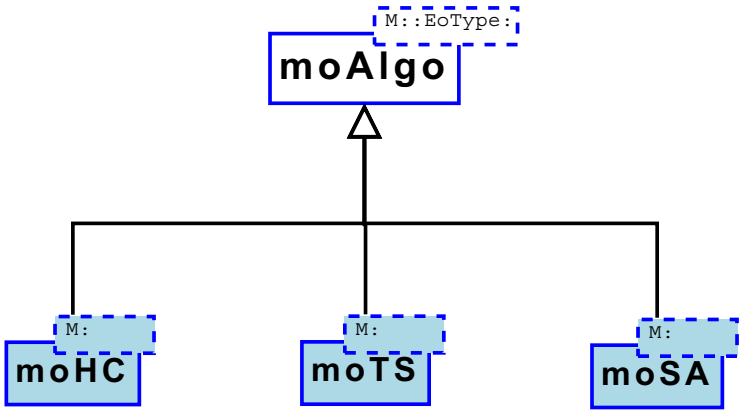


FIG. 1.24 – Schéma des implémentations de type *moAlgo* existant à l'origine dans ParadisEO-MO.

Une métaheuritisque à solution unique sous ParadisEO-MO est un objet de type *moAlgo*. Ce principe a été conservé dans la version actuelle.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Recherche par descente (Hill Climbing) Comme cela a été développé dans la section 1.2.1, la recherche par descente est souvent ce que les acteurs du domaine appellent plus simplement la recherche locale. Le but est d'améliorer une solution courante en explorant son voisinage. Le meilleur voisin remplace la solution courante si celui-ci est meilleur.

Comme toutes les autres recherches locales, le point crucial est l'exploration du voisinage. Plusieurs questions se posent :

1. Comment générer un voisin ?
2. Quand sait-on que l'exploration est finie ?
3. Comment choisir un voisin en particulier ?

A l'origine les recherches locales sont destinées à être utilisées pour leur pouvoir d'intensification, il faut donc aussi répondre à deux autres questions :

1. Comment évaluer la qualité d'un voisin de manière **efficace** ?
2. Quand une recherche locale doit-elle s'arrêter ?

A partir de ces questions, il est aisé de mettre en évidence les bases nécessaires à la mise au point d'une recherche locale. Par conséquent, il faut connaître :

1. Le type de modifications à apporter à la solution courante pour obtenir d'autres solutions.
2. La méthode d'évaluation de qualité d'un voisin en se basant uniquement sur la solution courante et la modification à appliquer pour obtenir celui-ci.
3. Le critère d'arrêt de l'exploration du voisinage.
4. La stratégie de sélection des voisins.
5. La stratégie de remplacement de la solution courante.
6. Le critère d'arrêt de la recherche locale.

Parmi ces six points, les deux premiers dépendent de la représentation de la solution du problème étudié mais les suivants sont tout à fait génériques et dépendent seulement du type de recherche locale que l'on veut effectuer. Ainsi, dans le cas de la recherche par descente, il faut :

1. Explorer complètement le voisinage c'est-à-dire tant que des voisins existent.
2. Sélectionner le meilleur voisin.
3. Remplacer la solution courante par le meilleur voisin sélectionné s'il est de meilleur qualité.
4. S'arrêter lorsqu'il n'existe plus de voisin de meilleur qualité que la solution courante. On est alors dans un optimum qui peut-être local (majorité des cas) ou global. Seul une méthode exacte pourrait permettre de vérifier l'optimalité de la solution trouvée.

Tout ce raisonnement que l'on vient de faire pour extraire les « bonnes » questions à se poser pour concevoir une recherche locale représente la base du travail de conception fait sur ParadisEO-MO. Ainsi, chaque partie d'une recherche locale est symbolisée par un type d'objet. Pour une recherche locale par descente, il existe un objet *moHC*. Il est important de noter que tous les objets de ParadisEO-MO comportent le préfixe -mo- ce qui permet de les différencier des objets des autres parties de ParadisEO (EO,

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

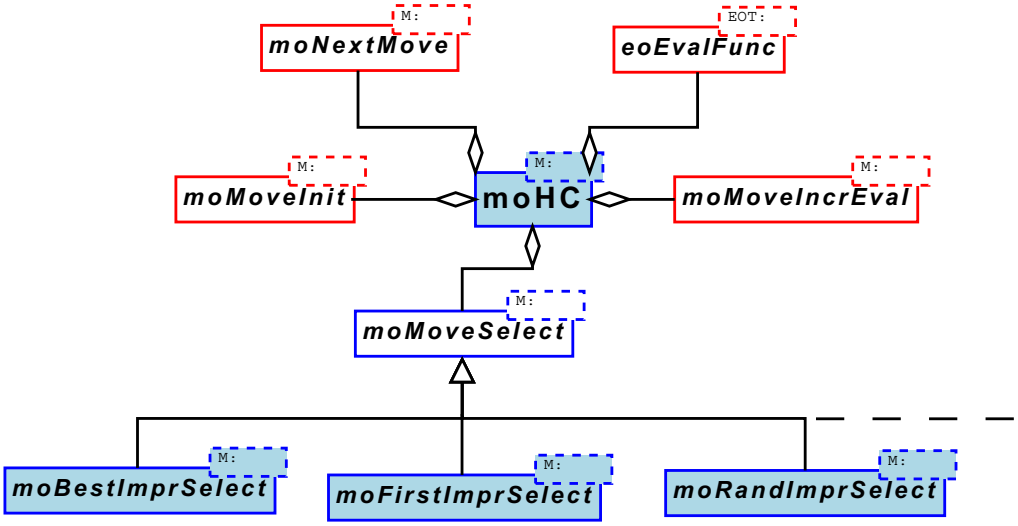


FIG. 1.25 – Schéma d'un objet moHC.

MOEO et PEO). La figure 1.25 présente un schéma de la classe *moHC* et l'algorithme implémenté dans cette classe est visible dans l'annexe A.1.1.

Dans ce schéma et ceux qui suivront, **EOT** correspond au type d'une solution, c'est-à-dire la façon dont celle-ci est représentée. **M** symbolise quant à lui le « mouvement » utilisé. Le mouvement est le point central de ParadisEO-MO, c'est autour de lui que sont articulés les objets. Le principe est simple, pour explorer le voisinage d'une solution réalisable, il suffit d'appliquer des modifications sur cette solution en s'assurant bien sûr que la solution générée est aussi réalisable. Si on modélise une solution et ses voisins dans un espace à deux dimensions, un mouvement peut-être vu comme la ligne qu'il faut tracer pour atteindre une autre solution voisine à partir de la solution courante. Selon la représentation utilisée, un « mouvement » peut être plus ou moins difficile à caractériser. De manière générale, il est nécessaire de définir ce qu'est un mouvement pour sa représentation de solution. Il est à noter qu'un certain nombre de représentations standards existe dans ParadisEO-EO et que par conséquent les mouvements associés existent aussi dans ParadisEO-MO. La seule chose à faire pour inclure un nouveau mouvement dans la hiérarchie de ParadisEO-MO est qu'il hérite de la classe *moMove* qui définit de manière générique la notion de mouvement. Si l'on regarde de nouveau la figure 1.25, on s'aperçoit que pour instancier un objet *moHC*, il est nécessaire de fournir cinq objets différents. Les objets entourés de rouge doivent être définis par l'utilisateur car ils sont spécifiques à la représentation d'une solution. Les objets entourés de bleu sur fond bleu sont prêt à être utilisés directement car ils ne sont pas dédiés à une représentation donnée.

Ainsi les cinq objets nécessaires à l'instanciation d'un objet *moHC* sont :

- **moMoveInit** : cet objet doit permettre de générer le premier mouvement. Cet objet est utilisé à chaque début d'exploration du voisinage.
- **moNextMove** : cet objet peut être vu comme un itérateur dans le voisinage à explorer, c'est-à-dire qu'il permet de savoir s'il y a encore des voisins à explorer et le cas échéant d'accéder au prochain.
- **eoEvalFunc** : cet objet permet d'évaluer la qualité d'une solution. Comme vous pouvez le remarquer, le préfixe de cet objet indique que celui-ci provient de ParadisEO-EO. Il ne s'agit pas de l'objet qui sera utilisé pour évaluer les voisins durant notre exploration de voisinage. Cet objet voit

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

son intérêt dans le cas où l'on donne à l'algorithme une solution initiale non encore évaluée.

- **moIncrEval** : cet objet constitue la réelle évaluation dans une recherche locale sous ParadisEO-MO. En effet, celle-ci se doit d'être efficace car la qualité d'un voisin est déterminée seulement à partir des modifications apportées à la solution courante pour atteindre celui-ci.
- **moMoveSelect** : cet objet permet de spécifier la stratégie de sélection du meilleur voisin. Trois stratégies sont déjà disponibles sous ParadisEO-MO :

  1. **moBestImprSelect** : c'est la stratégie standard. Le voisinage est complètement exploré et le meilleur voisin est choisi.
  2. **moFirstImprSelect** : dans cette stratégie, le premier voisin améliorant la solution courante est choisi. Dans ce cas, tant qu'il existe des meilleures solutions dans le voisinage de la solution courante, l'exploration est incomplète. Cependant, l'atteinte d'un optimum (local ou global) impose une dernière exploration du voisinage qui est complète.
  3. **moRandomImprSelect** : cette troisième stratégie est une variante de la première. En effet, le voisinage est complètement exploré mais le choix du voisin se fait parmi ceux qui améliore la solution courante de manière aléatoire. Ainsi la prochaine solution utilisée serait l'un des meilleurs voisins mais pas nécessairement le meilleur.

  Bien sûr, si l'on veut utiliser une stratégie qui est propre au problème, il est tout à fait possible de la mettre en place en créant l'objet correspondant

  .

Aux vues de ces différents objets, il est aisé de comprendre le fonctionnement d'un *moHC* et de manière plus générale la manière dont fonctionne une recherche locale. Le recherche locale par descente est une métaheuristique et donc par conséquent n'est pas dédiée à un problème. Cependant, il reste important de noter, que certaines sous-parties de cette métaheuristique sont spécifiques au problème étudié et notamment tout ce qui touche à la représentation des solutions. Il semble donc naturel que l'utilisateur ait un effort de codage à faire à ce niveau d'autant plus si sa représentation est très spécifique. L'avantage de cette hériarchie de classe est de permettre une meilleure compréhension de l'algorithme utilisé aussi bien pour le développeur que pour une personne voulant examiner certaines parties du code plus en profondeur.

Un autre avantage non négligeable qui compense largement l'investissement initial est le principe d'évolution du code. Il est clair que mettre au point une recherche locale n'est pas difficile. A partir de là, il est commun de faire diverses optimisations pour améliorer le fonctionnement de celle-ci. Mais si l'on veut modifier complètement un comportement de la méthode, si cela n'avait pas été prévu à la base, ce qui est souvent le cas, un effort supplémentaire de développement et de test de compatibilité sera nécessaire pour s'assurer de la cohérence de la nouvelle méthode obtenue. ParadisEO-MO permet de pallier à ce genre de désagrément. En effet, l'effort très important de conception qui a amené à la mise en place de cette hiérarchie de classe assure une modularité unique pour quiconque désire mettre en place des métaheuristiques à base de solution unique. Si l'utilisateur veut changer une partie de son approche, il change l'objet correspondant sans se préoccuper de problème de compatibilité avec les autres et sans devoir sauvegarder séparément l'ancienne version de son code. Il peut même rendre son approche dynamique en permettant de choisir automatiquement quelle est le meilleur objet à utiliser suivant l'instance qu'il veut résoudre. Ce genre de démarche est fastidieuse dans un code standard mais est naturelle et aisée avec ParadisEO-MO. Tout cela est encore plus vrai pour les objets génériques tel que les stratégies de sélection. Comme ParadisEO-MO fournit trois stratégies de sélection possibles, quand l'utilisateur a mis en place sa recherche par descente, il a en fait trois recherches locales par descente utilisables. Pour ce faire, il n'aura juste qu'une ligne de code à changer par exemple en remplaçant *moBestImprSelect* par

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

moFirstImrpSelect.

Enfin un autre avantage de ParadisEO-MO, et non des moindres, est de permettre une modification aisée de son algorithme mais aussi d'utiliser plus facilement les autres algorithmes de ParadisEO-MO. Ainsi une recherche locale par descente peut devenir aisément une Recherche Tabou.

Recherche Tabou (Tabu Search) Comme cela a été expliqué dans la section 1.2.2, la particularité de la Recherche Tabou est la notion de mémoire. Dans ce type de recherche locale, une liste dite « taboue » est utilisée pour se rappeler quelles sont les solutions qui ont déjà été visitées. Les principales caractéristiques d'une Recherche Tabou sont :

1. une exploration complète du voisinage.
2. la sélection du meilleur voisin.
3. le remplacement de la solution courante par un voisin seulement si celui-ci n'est pas considéré comme tabou ou si le critère d'aspiration est validé.
4. un nombre maximum d'itérations comme critère d'arrêt de la recherche.

Comme la recherche par descente, la Recherche Tabou est représentée sous ParadisEO-MO par un objet spécifique : l'objet *moTS*.

La figure 1.26 détaille la manière dont est constituée une Recherche Tabou sous ParadisEO-MO et son implémentation se trouve dans l'annexe A.1.2.

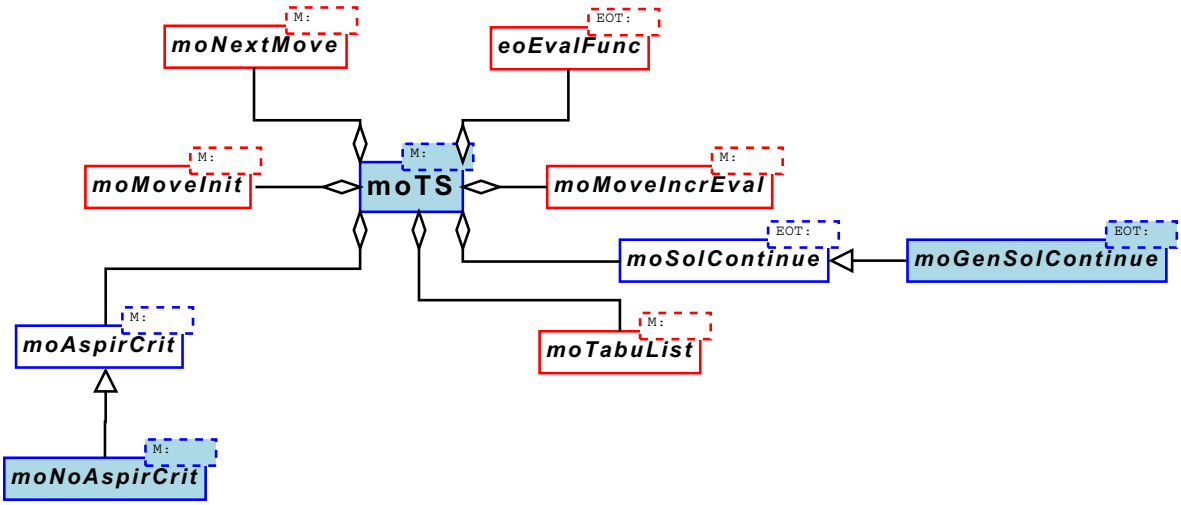


FIG. 1.26 – Schéma d'un objet moTS dans la version originelle de ParadisEO-MO.

Comme pour la recherche locale par descente, des objets sont spécifiques au problème (en rouge) et d'autres sont génériques (en bleu). Si l'on regarde attentivement cette figure et que nous la comparons avec celle de l'objet *moHC* (figure 1.25), on peut remarquer que les objets en rouge sont les mêmes. Donc si les objets en rouge sont ceux que l'utilisateur doit mettre en place lui-même car spécifique à son problème et que celui a déjà mis en place un algorithme de type recherche par descente avec ParadisEO-MO ; il n'a pratiquement rien à faire pour transformer sa recherche par descente en Recherche Tabou. Les différences entre un objet *moHC* et *moTS* sont symbolisés par trois objets génériques :

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

- **moSolContinue** : c'est le critère d'arrêt de la Recherche Tabou. Un seul critère d'arrêt était déjà implémenté dans cette version de ParadisEO-MO : le **moGenSolContinue**. Il s'agit d'un critère qui correspond à un nombre maximum d'itérations.

- **moTabuList** : comme son nom l'indique, cet objet s'occupe de la gestion de la liste taboue. Dans cette version de ParadisEO-MO, il s'agit d'un objet que l'utilisateur doit définir.

- **moAspirCrit** : cet objet correspond au critère d'aspiration de la Recherche Tabou. Deux exemples de critère d'aspiration sont fournis dans ParadisEO-MO :
  1. **moNoAspirCrit** : ce critère très simple rejette tout simplement constamment un mouvement. Cela revient à ne pas avoir de critère d'aspiration.
  2. **moImpBestFitAspirCrit** : la version standard du critère d'aspiration qui compare la qualité de la solution proposée par rapport à la meilleure trouvée actuellement.

Vu que la recherche par descente et la Recherche Tabou sous ParadisEO-MO ne diffèrent que par des objets génériques mise à part la liste taboue, les transformations à apporter pour transformer une recherche par descente en Recherche Tabou ne sont pas énormes. Il faut d'abord définir sa liste tabou. Ensuite il suffit d'instancier un objet *moTS* en choisissant parmi les implémentations proposées le critère d'aspiration voulu et le critère d'arrêt de la recherche. Ce qui représente un intérêt non négligeable pour l'utilisateur.

De la même manière, il est tout à fait possible de passer à un autre type de recherche locale : le recuit simulé.

Recuit simulé (Simulated Annealing) Comme cela a été détaillé dans la section 1.2.3, le recuit simulé est une recherche locale qui s'inpire d'un processus physique. Son principal atout est de permettre, à la différence de la recherche par descente, de faire marche arrière dans l'obtention de meilleurs solutions. En effet, le recuit simulé est capable de choisir selon une certaine probabilité des solutions de moins bonnes qualités. L'intérêt d'une telle possibilité est de permettre, le cas échéant, de sortir d'optima locaux lors du parcours de l'espace de recherche. De part son principe, le recuit simulé peut être vu comme une recherche avec une phase de diversification qui petit à petit conduit à une phase d'intensification en devenant finalement équivalent à une recherche par descente. Les caractéristiques d'un recuit simulé sont les suivantes :

1. Un parcours aléatoire du voisinage durant un certain nombre d'itérations.
2. Une stratégie de sélection gardant le meilleur voisin rencontré.
3. Une stratégie de remplacement de la solution courante par le voisin choisi si celui-ci est meilleur ou selon la probabilité $P(\delta E) = e^{(-\delta E/kt)}$, K étant la constante de Boltzmann, t la température courante et δE la différence de qualité entre la solution courante et le voisin choisi.
4. Un critère d'arrêt correspondant à un seuil minimal pour la température.

Comme la recherche par descente et la Recherche Tabou, le recuit simulé a son équivalent en terme d'objet dans ParadisEO-MO : le *moSA*. La figure 1.27 détaille le schéma correspondant et l'algorithme se trouvant dans l'annexe A.1.3 l'implémentation associée. Comme on peut le remarquer dans ce schéma, il existe un objet rouge que nous n'avons pas encore rencontré. Cela représente donc un objet spécifique au problème que l'utilisateur va devoir mettre en place. En effet, le recuit simulé nécessite un générateur de mouvements aléatoires. Parmi les objets restants, on pourra remarquer le *moCoolingSchedule* qui

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

concerne le principe de refroidissement de la température. La version originelle proposait un seul objet de ce type : le **moLinearCoolingSchedule**. Ce système de refroidissement consiste à retirer une quantité constante à la température. Ainsi, la température à l'itération i serait :

$$T_i = T_0 - i * \beta$$

avec T_0 la température initiale, et β la quantité constante que l'on soustrait. Le peu de modifications à apporter pour obtenir un recuit simulé à partir d'une recherche par descente prouve encore le qualité de la conception mais aussi le recul pris dans la mise en place de ParadisEO-MO.

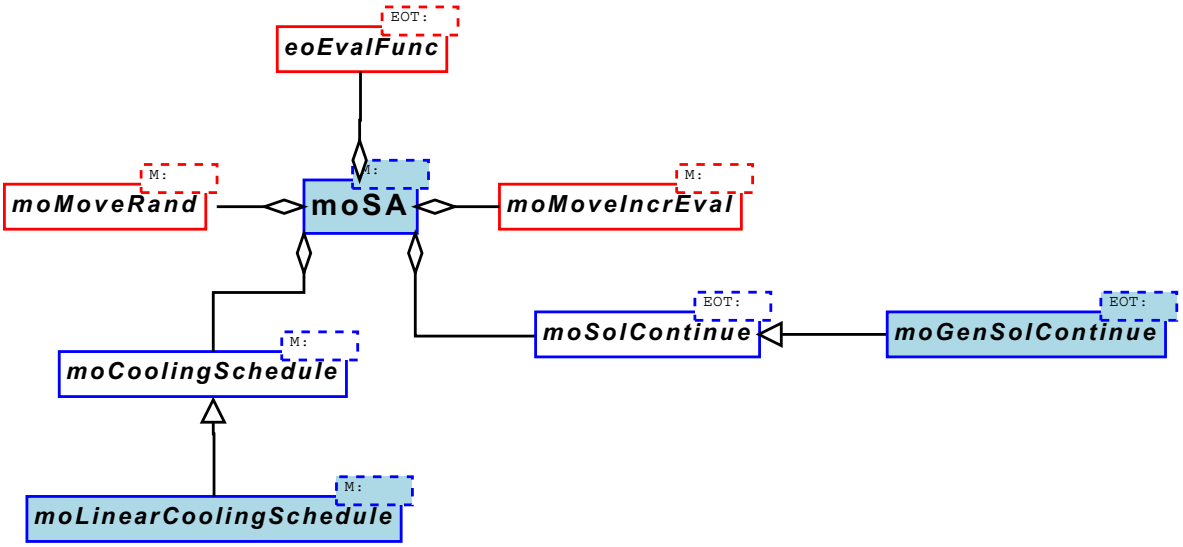


FIG. 1.27 – Schéma d'un objet moSA dans la version originelle de ParadisEO-MO.

Les trois dernières parties ont détaillé un ensemble de trois familles d'algorithme à base de solution unique classique de la littérature. Avant de montrer comment ParadisEO-MO permet aussi de mettre en place des recherches locales avancées, nous allons voir comment la combinaison de EO et de ParadisEO-MO permet de faire de la coopération de méthodes.

Coopération de méthodes (ParadisEO-EO + ParadosEO-MO) ParadisEO-MO représente les métaheuristiques à base de solution unique. Ces méthodes correspondent à des principes d'intensification performants mais qui ont un coût. Ainsi, l'utilisation systématique de telles méthodes ne peut être réaliste pour un certain nombre de problèmes dû notamment à la taille des espaces de recherche associés. Cependant, comme cela a été expliqué dans la section 1.4, leur utilisation avec parcimonie en les combinant à des métaheuristiques à base de population de solutions permet d'obtenir des approches très performantes. Une des hybridations les plus utilisées est celle qui combine un algorithme génétique et une recherche locale. La recherche locale est vu comme un opérateur de mutation qui est appelé avec une certaine probabilité pour intensifier la recherche quand cela devient nécessaire. Il est important de noter que ce type d'hybridation est d'autant plus productive dans le cas d'un algorithme génétique adaptatif qui permet de faire varier les probabilités des différents opérateurs selon leur performance respective.

Comme cela a été évoqué au début de ce chapitre, ParadisEO-EO correspond aux algorithmes à base de populations de solutions. Il est donc tout à fait naturel qu'il possède des objets pour pouvoir mettre

http://doc.univ-lille1.fr

en place des algorithmes génétiques. L'un d'entre eux est le *eoEasyEA* dont un schéma est donné par la figure 1.28.

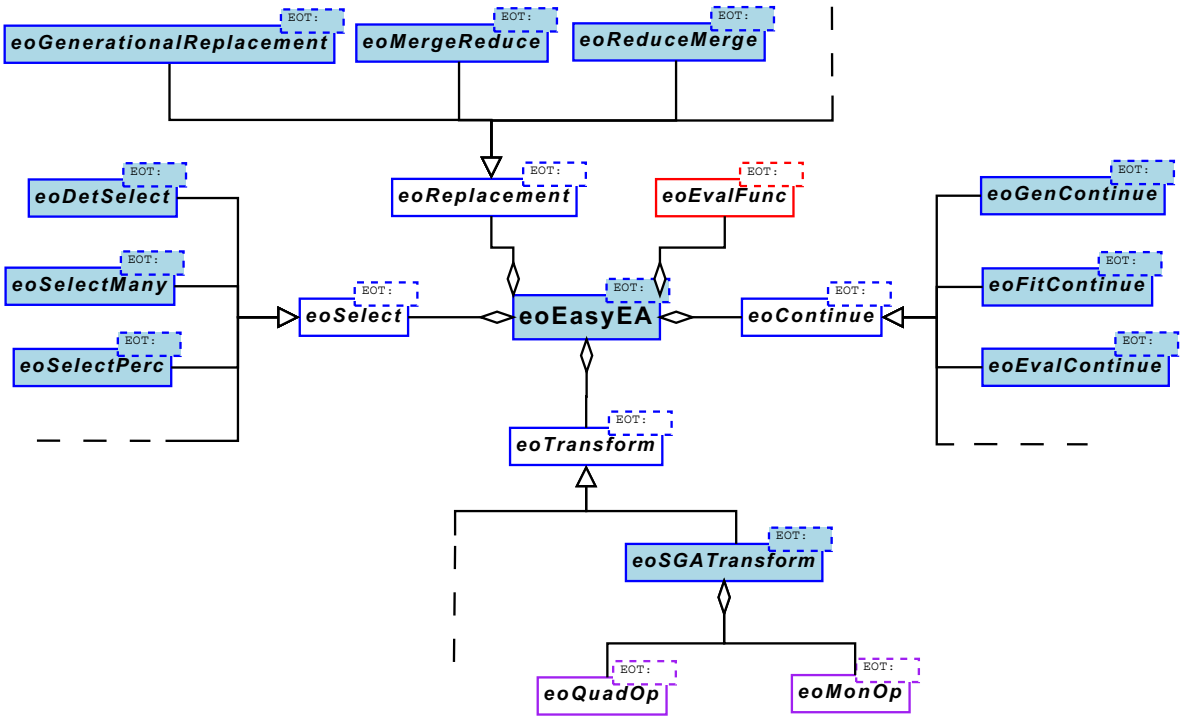


FIG. 1.28 – Schéma d'un objet EasyEA.

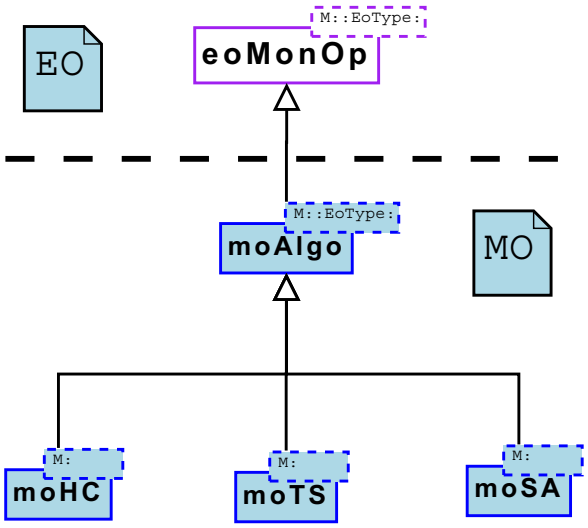


FIG. 1.29 – Schéma explicatif de lien rattachant les parties EO et MO de ParadisEO.

Ce schéma résume tous les composants nécessaires à l'instanciation d'un objet de type *eoEasyEA*. Pour la notion d'hybridation, nous allons seulement nous intéresser à une branche de ce schéma : l'objet *eoTransform*. Ce type d'objet représente tous les objets permettant de modifier une, deux ou plus de solutions en une fois. Comme nous nous intéressons aux algorithmes génétiques, un sous-type de *eo-*

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Transform est imposé : le *eoSGATransform*, dédié à ce type de méthodes. En effet, cet objet a besoin de deux objets spécifiques pour être instancié : un *eoQuadOp* et un *eoMonOp*. Le premier désigne les objets pouvant modifier deux solutions à la fois, c'est-à-dire les opérateurs de recombinaison. Par conséquent, le deuxième correspond à l'autre type d'opérateur de cet famille d'algorithmes, les opérateurs de mutation. Ce type d'opérateur modifie une seule solution pour en produire une nouvelle. C'est donc à l'objet *eoMonOp* que nous allons nous intéresser. En effet, c'est cet objet qui va servir de passerelle entre ParadisEO-EO et ParadisEO-MO. Si on regarde la figure 1.29, on remarque que la classe qui décrit la notion de métaheuristique à base de solution unique dans ParadisEO-MO est la classe *moAlgo* et que celle-ci hérite de l'objet *eoMonOps*. A la vue de ce schéma, on comprend mieux pourquoi il est aisé d'hybrider un algorithme génétique avec une recherche locale. De plus, on se rend compte que l'on peut utiliser n'importe quelle méthode de ParadisEO-MO et non pas seulement la recherche par descente. Ainsi en plus des méthodes proposées par ParadisEO-EO et par ParadisEO-MO, on obtient tout un panel de nouvelles méthodes basées sur des coopérations d'algorithmes par hybridation entre une méthode à base de population et une recherche locale. Il est notamment possible de combiner plusieurs recherches locales à un même algorithme génétique.

Jusqu'ici, tout ce qui a été présenté correspond à ce qu'était ParadisEO-MO en 2005. Depuis, la plateforme a été énormément testée, éprouvée et a été améliorée.

1.5.2.2 Contribution à ParadisEO-MO

Nous allons maintenant reprendre l'architecture de ParadisEO-MO actuelle et la comparer à l'ancienne version. Tout d'abord, le schéma 1.30 indique quelles sont les implémentations actuelles de l'objet *moAlgo* existant dans ParadisEO-MO. On remarque tout de suite que deux nouveaux algorithmes sont apparus, nous les détaillerons plus loin.

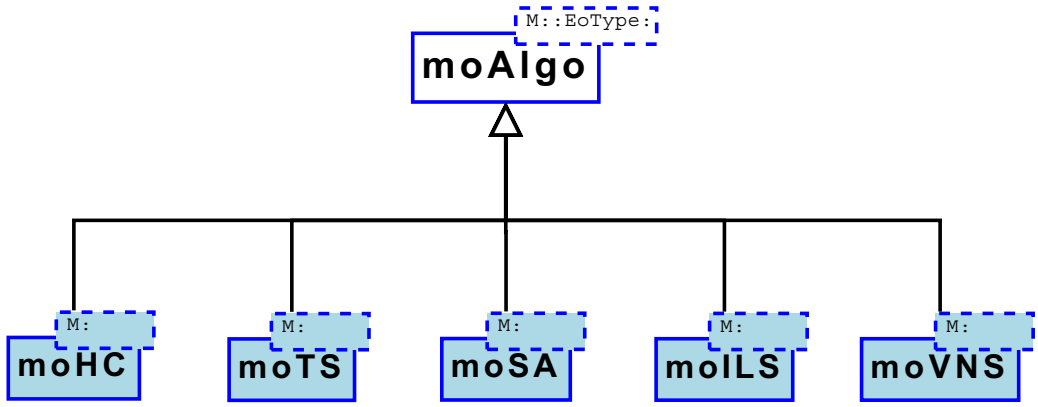


Fig. 1.30 – Schéma des implémentations de type *moAlgo* existant actuellement dans ParadisEO-MO.

Recherche par descente (Hill Climbing) L'algorithme de recherche est un des rares objets de ParadisEO-MO qui dans sa globalité fut gardé tel qu'il était. Cela vient principalement du fait que la simplicité de l'algorithme associé n'impliquait pas de réelles modifications à apporter.

Recherche Tabou (Tabu Search) La figure 1.31 détaille la version actuelle de la Recherche Tabou sous ParadisEO-MO.

http://doc.univ-lille1.fr

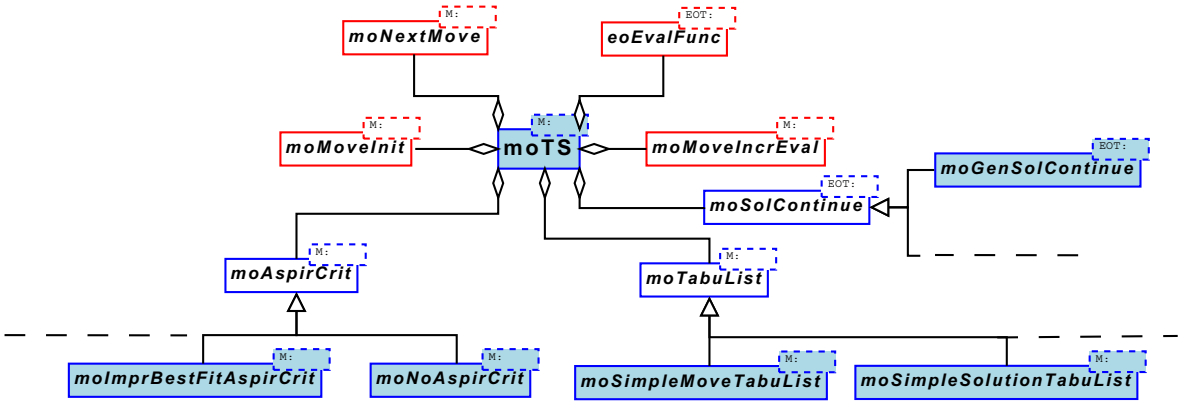


FIG. 1.31 – Schéma d'un objet moTS.

Les différences avec l'ancienne version concernent l'existence de nouvelles implémentations d'objets de type **molSolContinue** et de la proposition de deux implémentations standards de liste taboue pour l'utilisateur :

- **moSolContinue** : c'est le critère d'arrêt de la Recherche Tabou. En plus du **moGenSolContinue**, trois autres critères d'arrêt sont maintenant proposés à l'utilisateur :
  1. **moFitSolContinue** : un autre critère d'arrêt qui permet de stopper l'algorithme quand une valeur cible pour la qualité est atteinte.
  2. **moNoFitImprSolContinue** : cet objet désigne un processus d'arrêt plus évolué dans lequel on précise le nombre d'itérations maximum sans amélioration de la solution courante avant d'arrêter l'algorithme.
  3. **moSteadyFitSolContinue** : ce dernier critère est directement inspiré d'un objet de ParadisEO-EO. D'abord, un nombre minimum d'itération est réalisé. Ensuite seulement le critère d'arrêt devient équivalent à un *moNoFitImprSolContinue*.

- **moTabuList** : comme son nom l'indique, cet objet s'occupe de la gestion de la liste taboue. Deux exemples de listes taboues existent dans la version actuelle de ParadisEO-MO :
  1. **moSimpleMoveTabuList** : c'est la liste classique pour une Recherche Tabou sous ParadisEO-MO. Comment à partir de la solution courante, un mouvement permet-il de reconstruire un voisin, cet objet consiste seulement en une liste de mouvements considérés comme tabou.
  2. **moSimpleSolutionTabuList** : cet autre liste permet de sauvegarder les solutions considérées comme taboues. Cet objet gère donc une liste de solutions et non plus de mouvements seulement.

L'amélioration de la Recherche Tabou sous ParadisEO-MO a permis de la rendre encore plus simple d'utilisation. En effet, maintenant un utilisateur qui a déjà mis en place une recherche par descente n'a plus aucun effort à faire pour passer directement à une Recherche Tabou. Auparavant, il devait définir lui-même sa liste taboue.

Recuit simulé (Simulated Annealing) La figure 1.32 détaille l'architecture actuelle d'un objet *moSA*.

Comme on peut le remarquer sur la figure, en plus du **moLinearCoolingSchedule**, un autre principe de refroidissement de la température a été proposé à l'utilisateur : le **moGeometricCoolingSchedule**.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

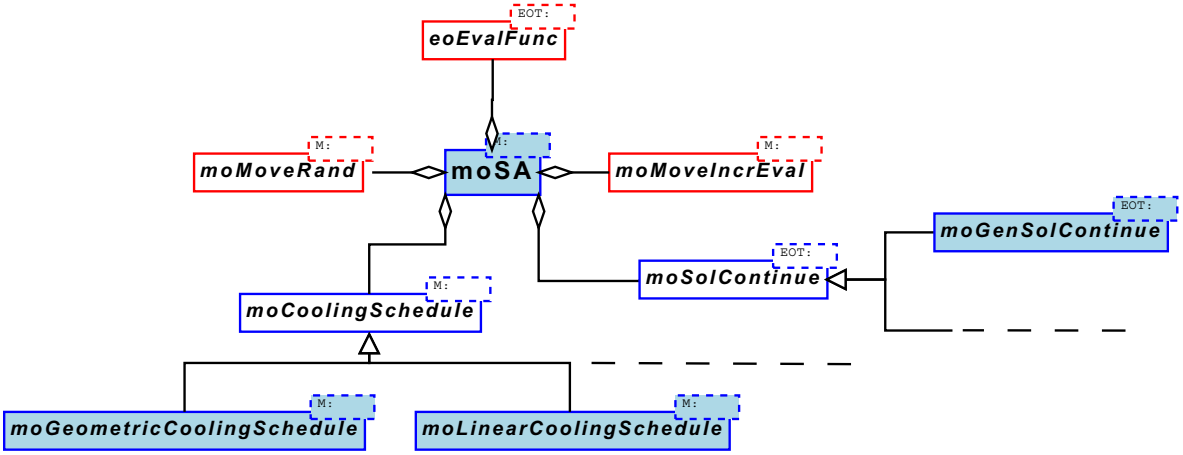


FIG. 1.32 – Schéma d'un objet moSA.

Cet objet propose le principe de refroidissement le plus couramment utilisé. La température est abaissée à l'aide de la formule suivante :

$$T_{next} = \alpha T \quad avec\ \alpha \in]0, 1[$$

Il faut savoir que les valeurs les plus utilisées de α sont comprises entre 0,5 et 0,99.

Une autre modification très importante qui a eu lieu dans la fonctionnement du recuit simulé n'est pas visible directement. Cela concerne la comparaison de valeurs de fonction objectif pour le calcul de l'exponentiel. Dans la version original, le type de valeur associé à la fitness était imposé et ne respectait pas les conventions de comparaison héritées de EO. Tout cela a été généralisé le mieux possible afin d'être beaucoup plus flexible. Cette amélioration a été visible rapidement à travers notamment des tests effectués sur l'implémentation d'un recuit simulé pour le problème du Q3AP [11] qui ne fonctionnait pas avec l'ancienne version de ParadisEO-MO et qui marche avec la version actuelle.

Coopération de méthodes (ParadisEO-EO + ParadisEO-MO) La principale innovation de ParadisEO-MO en terme d'hybridation est de permettre de faciliter le travail de l'utilisateur. En effet, de part son principe, une mutation d'un algorithme génétique peut très bien être considérée comme un mouvement pour une recherche locale. Cependant, l'ancienne version de ParadisEO-MO ne permettait pas de réutiliser une mutation (objet *eoMonOp*) comme base de voisinage et de la même manière un mouvement de recherche locale (objet *moMove*) ne pouvait devenir une mutation sous ParadisEO-EO. Cela entrainait un surcoût de développement de la part de l'utilisateur qui avait l'impression de faire du code redondant. Aujourd'hui, grâce à une reflexion sur la conception de ParadisEO-EO et de ParadisEO-MO, cela est possible et est tout à fait transparent pour l'utilisateur.

Recherche locale itérative (Iterative Local Search) Cette métaheuristique est une des deux nouvelles qui ont été implémentées sous ParadisEO-MO. Comme son nom l'indique, ce type de recherche locale est basée sur le lancement successif de recherches locales et l'ajout d'un mécanisme de pertubation entre deux recherches (voir section 1.2.4). La figure 1.33 présente la diagramme d'un objet *moILS* qui représente ce type de recherche locale dans ParadisEO-MO. L'algorithme de l'annexe A.1.4 détaille l'implémentation de la recherche locale itérative dans ParadisEO-MO.

[11] « 3D quadratic assignement problem »

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

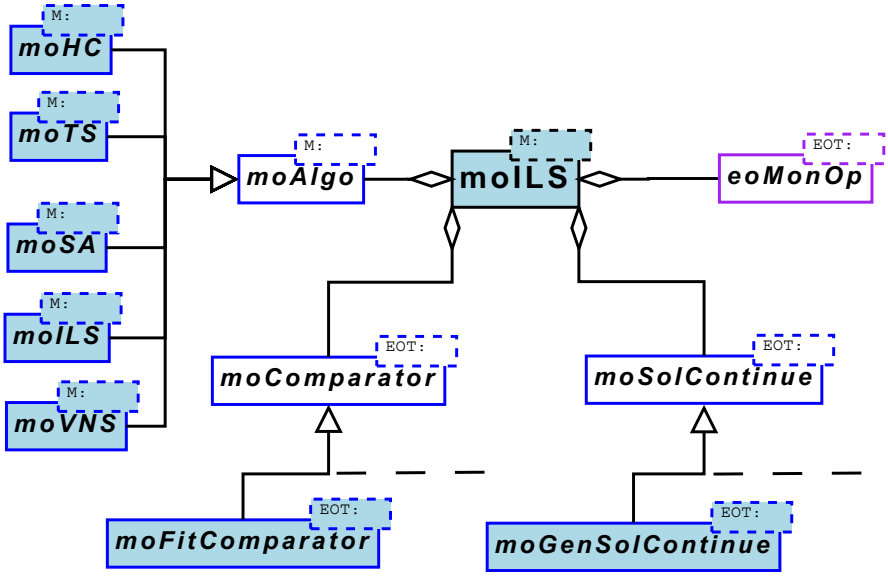


FIG. 1.33 – Schéma d'un objet moILS.

Comme le montre la figure, même si la recherche locale itérative classique est basé sur l'utilisation d'une recherche par descente, un objet *moILS* n'utilise pas un objet *moHC* mais un objet *moAlgo*. L'intérêt d'un tel choix se comprend aisément. Par ce biais, une recherche locale itérative peut utiliser n'importe quel type de recherche locale disponible sous ParadisEO-MO voir même d'autre type de recherche locale. En effet, rien n'empêche l'utilisateur d'utiliser sa propre recherche locale. Cela met encore plus en évidence l'aspect générique et modulaire de la conception de ParadisEO-MO.

Deux objets sont nécessaires à l'instanciation d'un objet *moILS* :

– **moComparator** : cet objet permet de décrire la manière de comparer deux solutions. Ce type d'objet peut paraître superflu, notamment car nous sommes en optimisation mono-objectif et que dans ce cas on peut dire qu'une solution est meilleure qu'une autre au regard de sa seule qualité. Il est clair que ce type de raisonnement est tout à fait juste et c'est pour cela que la première implémentation de *moComparator* fournie par ParadisEO-MO, le *moFitComparator*, se base sur une simple comparaison de la qualité. Cependant, il faut garder à l'esprit que pour un problème donné, il peut très bien y avoir des facteurs supplémentaires rentrant en considération dans la comparaison de solution. Cela peut notamment être le cas d'ajout de contraintes ou de cas spécifiques dans la représentation de la solution. Certains diront que l'on peut en tenir compte directement dans la fonction d'évaluation d'une solution en pénalisant celles qui ne respectent pas telle ou telle contrainte mais il existe des cas où un tel ajout dans le processus d'évaluation n'est pas conceptuellement correct. En effet, le moindre changement de contraintes imposerait de modifier de nouveau la fonction d'évaluation ou de créer d'autres fonctions d'évaluation. Le principe de différents modes de comparaisons est raisonnable et assure une bonne cohérence dans le principe de la recherche locale itérative. L'utilisateur peut très bien rajouter ses propres modèles de comparaisons en implémentant un objet de ce type.

– **eoMonOp** : cet objet symbolise la notion de perturbation dans un objet *moILS*. On pourrait donc se demander pourquoi celui-ci ne s'intitule pas simplement « moPertubation ». Seulement, il est

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

important de réfléchir à ce qu'il se cache derrière le concept de perturbation. Il s'agit grossièrement de « secouer » une solution courante (« shaking »). « secouer » une solution revient à lui appliquer une ou plusieurs modifications. On se ramène donc au concept de mutation de solution tel qu'il a été évoqué dans la section précédente. C'est donc tout naturellement que l'utilisation d'un objet *eoMonOp* paraît comme évident pour permettre la pertubation de solution. Cela permet notamment de réutiliser les mutations existantes qui ont pu être utilisées dans un algorithme génétique développé pour le problème considéré. Une autre conséquence, inattendue, de prime abord, est que l'on peut très bien perturber une solution en utilisant une recherche locale de ParadisEO-MO car les objets *moAlgo* sont aussi des *eoMonOp*. On pourrait se poser la question de l'utilité d'une telle perturbation mais elle est cependant évidente. Si l'on perturbe une solution en utilisant un autre type de mouvement, et donc un autre voisinage, il est fort probable que la solution obtenue, en plus d'être de meilleure qualité (au moins), soit tout à fait différente de la solution originale et donc quelle soit vraiment « perturbée » par rapport au voisinage utilisé initialement.

Par la mise en place de recherche locale avancée, on se rend compte que la conception de ParadisEO-MO ne consiste pas à seulement rajouter des objets de telle manière que ceux-ci correspondent toujours à une caractéristique de recherche locale. L'unification de concepts reste un point très important que ParadisEO-MO, et les autres partie de ParadisEO, se veulent de conserver et de mettre en avant.

Au travers de la recherche locale itérative, on s'est rendu compte qu'il était possible de faire de la recherche locale sur plusieurs voisinages différents. C'est un premier pas vers le principe de recherche à voisinage variable qui est justement une autre recherche locale avancée présente dans la version actuelle de ParadisEO-MO.

Recherche à voisinage variable (Variable Neighborhood Search) Toutes les métaheuristiques présentées dans ParadisEO-MO fonctionnent toutes avec un seul voisinage, sauf peut-être certaines configurations que l'on peut appliquer à un objet de type *moILS* (voir section précédente). En effet, chaque méthode est basée sur un seul type de mouvement pour générer les différents voisins d'une solution donnée. Cependant, les méthodes à base de solution unique avec des voisinages multiples existent (voir section 1.2.5) et par conséquent ParadisEO-MO en propose une implémentation.

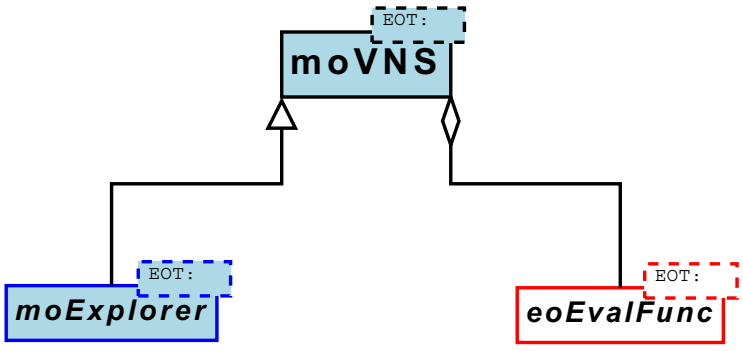


FIG. 1.34 – Schéma d'un objet moVNS.

La figure 1.34 présente le diagramme d'un objet *moVNS* qui représente ce type de recherche locale dans ParadisEO-MO. L'algorithme de l'annexe 14 détaille l'implémentation liée à cet objet. Il est à noté que de part l'utilisation de plusieurs voisinages, un *moVNS* n'est pas lié à un mouvement mais directement au type de la solution de notre problème, d'où la définition suivant le type EOT et non M comme

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

pour les autres méthodes. Comme les autres recherches locales, un objet *moVNS* utilise un objet *eoEvalFunc* qui représente la fonction d'évaluation d'une solution. Par contre, le *moVNS* est le seul objet à utiliser un *moExplorer*. C'est cet objet qui va gérer la notion de voisinage multiple. Chaque voisinage est vu comme un objet de type *eoMonOp*, c'est-à-dire un objet permettant de modifier une solution. Le choix d'une spécification si générique est voulu pour n'avoir aucune limitation dans l'utilisation des objets venant aussi bien de ParadisEO-EO que de ParadisEO-MO. En effet, il faut se rappeler que tout *moAlgo* est aussi un *eoMonOp*.

ParadisEO-MO est une plateforme très complète pour le dévelopement et l'utilisation de métaheuristiques à base de solution unique mais il faut garder à l'esprit qu'elle est censée être utilisable par différents types d'utilisateurs et pas seulement des spécialistes des métaheuristiques. Par conséquent, il est nécessaire que ParadisEO-MO soit correctement documentée.

Documentation La refonte et l'extension de ParadisEO-MO a engendré aussi la nécessité de mettre en place et maintenir toute la documentation associée. La figure 1.35 montre un exemple de schéma automatiquement généré par la documentation du code de ParadisEO-MO qui permet de connaître rapidement quelles sont les implémentations existantes pour les objets génériques.

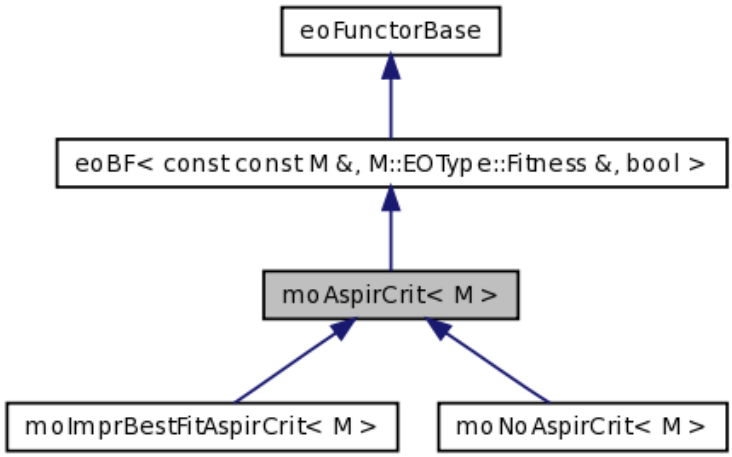


FIG. 1.35 – Schéma provenant de la documentation de l'objet *moAspirCrit*. On peut voir ainsi aisément les implémentations associées qui existent : *moImprBestFitAspirCrit* et *moNoAspirCrit*.

Tous les algorithmes de ParadisEO-MO peuvent être facilement testé sur une implémentation du TSP fournie avec la plateforme. De plus, chaque exemple est accompagné de la leçon correspondante pour guider l'utilisateur dans l'utilisation des différentes classes de ParadisEO-MO. L'implémentation du TSP proposée est compatible avec la TSPLIB [12] pour faire de la comparaison de méthodes facilement.

1.5.3 Synthèse

Etant un des premiers utilisateurs de ParadisEO-MO durant mon master recherche, j'en suis devenu le responsable pendant ma thèse. De ce fait, j'ai du, en marge de mon travail de thèse principal, travailler sur la conception de la version d'origine et les modifications et ajouts à y apporter pour d'un côté la rendre plus facile d'accès et d'un autre qu'elle puisse accueillir les recherches locales avancées

[12] www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

dont j'ai eu besoin durant ma thèse. C'est pour cela que la plateforme a évolué progressivement mais relativement lentement par rapport à d'autres parties de la plateforme (comme ParadisEO-MOEO ou ParadisEO-PEO). Cependant, le travail associé a nécessité un investissement non négligeable (tests, documentation, rapport de recherche, aide aux nouveaux utilisateurs, ...) qui a été largement récupéré lors du dévelopement des méthodes utilisés dans ma thèse et leurs tests. Les grandes étapes d'avancement sur le plan conceptuel ont surtout été obtenues suite à des travaux effectués en collaboration avec d'autres chercheurs de l'équipe et notamment des post-doctorants. Actuellement, une réorientation sur la valorisation de ParadisEO-MO est en cours pour lui permettre d'exprimer tout son potentiel. Cette étude sera principalement réalisée par des doctorants actuels de l'équipe INRIA DOLPHIN avec toujours l'appui des ingénieurs de recherche. Le fil conducteur de cette étude reste la facilité d'utilisation et l'enrichissement de la plateforme par rapport à l'état de l'art actuel.

1.5.4 Perspectives

ParadisEO-MO est une plateforme qui permet de développer des métaheuristiques à base de solution unique pour n'importe quel type de problème mono-objectif qu'il soit discret ou continu. Elle propose aussi bien des recherches locales classiques (recherche par descente, Recherche Tabou et recuit simulé) que des recherches locales avancées (recherche locale itérative et recherche à voisinage variable). Elle a été testée sur des problèmes classiques tel que le le TSP mais aussi sur des problèmes réels basés sur des représentations non triviales à travers mes travaux de thèse. Cela a permis de prouver la capacité de cette plateforme à être utilisable quelque soit le type de problème.

Afin de valoriser les capacités actuelles de cette plateforme, et plus globlement de ParadisEO, un ensemble d'implémentation de problèmes classiques va être bientôt disponible. D'un côté, ces implémentations permettront aux utilisateurs d'avoir une base de travail déjà existante pour se former à la plateforme ; d'un autre cela permettra aussi de faire des mesures de performances de la qualité des méthodes sur des benchmarks bien connus. Par exemple, pour le TSP, les algorithmes développés sous ParadisEO-MO vont être testés sur les instances de la TSPLib. Le but de ces implémentations est de montrer la généricité de la plateforme ParadisEO pour laquelle, quelque soit l'algorithme utilisé, les modifications à appliquer pour passer d'un algorithme à un autre sont minimes et permettent vraiment de se centrer sur les spécificités de chaque approche. Outre ces implémentations de problèmes classiques, la partie contribution de ParadisEO contiendra tous les travaux effectués grâce à cette plateforme dans l'équipe projet DOLPHIN mais aussi les implémentations venant d'acteurs extérieurs. Le but final est de pouvoir obtenir une bibliothèque de problèmes pour lesquels des implémentation sont déjà diponibles. Une telle bibliothèque permettrait un gain de temps non négligeable pour quelqu'un désirant s'attaquer à un des problèmes présents dans celle-ci.

Concernant uniquement ParadisEO-MO, tout son potentiel n'a pas encore été exploité, notamment de part sa grande indépendance par rapport aux autres parties de ParadisEO. En effet, comme MO a introduit le concept de mouvement dans ParadisEO, tous les outils de ParadisEO-MO sont basés sur ce concept et par conséquent ne sont pas fortement liés à d'autres concepts venant des autres parties de ParadisEO. Cette indépendance permet de bien comprendre tous les mécanismes liés aux développements de recherches locales et de ne pas imposer de contraintes trop fortes à l'utilisateur. Cependant, la hiérarchie actuelle impose dans certains cas, un travail supplémentaire d'implémentation qui ne serait pas nécessaire si ParadisEO-MO était plus lié à ParadisEO-EO.

Un autre aspect important est que ParadisEO-MO (et ParadisEO dans sa globalité) tient à respecter dans son évolution la réutilisation de tout code développé. Comme cela a été démontré dans les dernières

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

sections, pour chaque nouvelle méthode à base de solution unique, il n'est pas nécessaire de tout recoder, l'effort à fournir a été voulu comme le plus minime possible tout en restant cohérent. L'intérêt de ce concept est multiple car il permet :

1. de limiter un grossissement non contrôlé de la plateforme. Chaque ajout doit faire l'objet d'une étude de conception pour voir si finalement le nouveau processus que l'on veut ajouter ne correspond pas en fait en la combinaison d'objets déjà existants. De plus, chaque nouvel objet doit être associé à des tests unitaire pour vérifier son intégrité.
2. de faciliter la vie de l'utilisateur. Si une personne développe un algorithme sous ParadisEO, elle sera forcément agréablement surprise si elle peut réutiliser son code aussi bien pour une méthode équivalente (par exemple une autre recherche locale sous MO) que pour un algorithme totalement différent (une recherche locale devenant une mutation sous EO).
3. d'améliorer la compréhension de la plateforme. Si la conception est bonne, la compréhension s'en trouve facilitée.

C'est pourquoi, toutes les parties de ParadisEO doivent être constamment comparées et analysées pour permettre une cohérence la plus parfaite possible sur la plateforme.

Un certain nombre de détails nécessitent aussi d'être améliorés : l'intégration des notions d'intensification et de diversification dans la Recherche Tabou, les mécanismes de recherche dans un voisinage partiel, l'ajout de nouveau mécanisme de refroidissement pour le recuit simulé, ...

Même si ce n'est pas une des priorités actuelles concernant ParadisEO, des outils d'automatisation de construction d'algorithmes sous ParadisEO dirigés par une interface graphique, indépendante de la plateforme en elle-même, serait un atout non-négligeable dans l'aspect accessibilité de la plateforme.

Actuellement, ParadisEO a une bonne visibilité. Le site web de la plateforme a été visité par des visiteurs de plus de 130 pays pour près de 7000 téléchargements de la plateforme [13]. De plus, un dialogue constant entre les utilisateurs et les développeurs via les listes de diffusion associées existe ; ce qui prouve l'intérêt des utilisateurs à développer des méthodes via la plateforme.

1.6 Conclusions

A travers ce chapitre, nous avons donné les définitions essentielles à la compréhension de la thèse du point de vue optimisation. on a pu voir qu'un grand nombre de méthodes d'optimisation existe pour résoudre un problème combinatoire : les méthodes exactes, approchées, des métaheuristiques à base de solution unique en passant par celles à base de population de solution jusqu'à la coopération de métaheuristiques aussi bien en mono qu'en multi-objectif. Suivant les contraintes imposées (temps de calcul, espace, ...), les spécificités du problème et de sa structure, le choix de la meilleure méthode n'est pas a priori aisé et demande souvent une bonne connaissance du problème pour être efficace. C'est pour cela que des plateformes comme ParadisEO existent. ParadisEO permet d'accélérer la développement et le test de la plupart des métaheuristiques tout en restant ouvert à l'ajout de nouvelles. Dans les deux chapitres suivant, nous l'utiliserons notamment pour le test de nouvelles modélisations pour deux problèmes de bioinformatique : le « de novo protein sequencing » et le « docking » moléculaire.

[13] version 1.2 de février 2009

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

2

Nouveaux modèles pour l'identification de protéines et leur résolution par coopération de métaheuristiques

Dans ce chapitre, nous identifions et résolvons deux problèmes de protéomique dont l'objectif est l'identification de protéines. Le premier consiste à retrouver une protéine expérimentale par comparaison entre bases de données et résultats expérimentaux provenant de spectrométrie de masse. Nous proposons pour celui-ci une méthode inédite comparée à celles existantes. Ce travail est en fait la base de la méthode proposée pour le second problème : retrouver la séquence d'une protéine à partir de données de spectrométrie de masse sans l'aide de base de données protéiques. Nous montrons alors en quoi la coopération de métaheuristiques amène une réponse à ce problème là où aucune autre méthode bioinformatique n'a pu encore le faire. Tout d'abord, après une introduction à la protéomique, nous présenterons notre nouveau modèle d'identification de protéine avec utilisation des bases de données. Ensuite, nous présenterons une nouvelle modélisation pour un problème d'identification qui n'utilise pas les bases de données : le « de novo protein sequencing ». Pour ce faire, nous détaillerons toutes les étapes de la modélisation qui nous ont amené à mettre en place une résolution de ce problème par coopération de métaheuristiques.

L'ensemble de ces travaux ont fait l'objet de communications dans des conférences internationales avec comité de lecture :

- *HUPO 4th Annual World Congress, « ASCQ_ME : A New Engine For Peptide Mass Fingerprinting Without Mass List Extraction », 2005 [9].*
- *Recomb, « An online engine for peptide mass fingerprinting from spectrum », 2006 [70].*
- *55th ASMS Conference on Mass Spectrometry, « ASCQ_ME : a New Engine for Protein and Post-Translational Modification Identification from MS and MS/MS Raw Spectrum without Mass List Extraction », 2007 [8].*
- *20th International Conference on Advance Information Networking and Application (AINA), « A Preliminary Work on Evolutionary Identification of Protein Variants and New Proteins on Grids », 2006 [11].*
- *IEEE Congress on Evolutionary Computation (CEC), « Protein Sequencing with an Adaptive Genetic Algorithm from Tandem Mass Spectrometry, 2006 [12]. »*

Ces travaux ont été réalisés en collaboration avec le Laboratoire de Chimie Organique et Macromoléculaire de l'université Lille 1 (LCOM) et ont été soutenus par le PPF Bioinformatique de l'Université des Sciences et Technolgies de Lille (USTL).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.1 Définitions

Dans cette section, nous présentons le domaine de la protéomique avant de préciser les notions de protéine et de spectrométrie de masse.

2.1.1 Protéomique

Depuis le milieu des années 90, le terme de « protéomique » est utilisé pour décrire l'analyse globale des protéines [129]. Il s'agit d'identifier et de quantifier l'ensemble des protéines d'un organisme ou d'un tissu à un moment donné dans des conditions particulières [55].

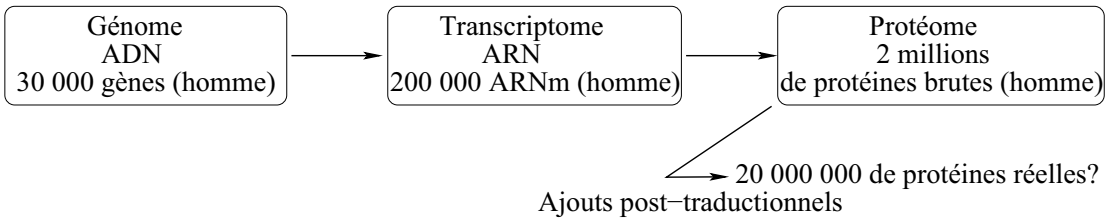


FIG. 2.1 – Génome, Transcriptome et Protéome.

La figure 2.1 permet de situer le protéome par rapport au génome et au transcriptome. Le génome représente le point de départ du dogme central. Le dogme central correspond aux principes du passage de l'information génétique des nucléotides aux polypeptides. Le génome correspond à l'ensemble des gènes dont la séquence nucléotidique est constituée des bases : adénine (A), guanine (G), thymine (T) et cytosine (C). A partir de ces séquences, on peut estimer les *ARNm* produits (acides désoxyribonucléiques messagers) dont l'ensemble représente le transcriptome. De là, un ensemble de protéines brutes peuvent être directement traduites, c'est le protéome. La particularité de celui-ci est que son contenu, à la différence du génome, est dynamique. C'est à dire qu'entre deux instants, les protéines présentes au sein d'un organisme peuvent être différentes. Cela peut notamment traduire un stress subit par l'organisme ou une réponse à un stimuli particulier.

La protéomique est principalement séparée en deux parties : l'identification et de la quantification de protéines. Dans la suite de manuscrit, on se restreindra à l'aspect identification de protéines. Connaître à un instant donné quelle protéine est présente dans une cellule nous apporte directement une information primordiale sur son état et plus généralement sur celui de l'organisme à partir duquel elle a été prélevée. Seulement, il faut bien comprendre toutes les étapes qui sont utiles à l'identification d'une protéine. En effet, il est nécessaire d'extraire la protéine du milieu où elle se trouve en la séparant des autres protéines présentes pour enfin passer à l'étape d'identification. Il y a une quinzaine d'années, cette étape d'identification consistait à trouver la séquence d'acides aminés (technique de dégradation d'Edman) pour ensuite comparer cette séquence à celles déjà connues. Le problème était que cette technique permettait d'obtenir une protéine par jour avec une sensibilité de 10 pmol ($10^{-12} mol$). La mole est une unité de système international pour évaluer les quantités de matière en physique ou en chimie, elle permet de représenter des nombres important d'atomes. Sachant que les échantillons contenaient plusieurs centaines de protéines, on se rend vite compte que la technique entraînait un ralentissement global du processus d'identification et qu'en plus, elle était destructive par rapport aux échantillons utilisés. C'est dans ce contexte que l'apparition des spectromètres de masse a révolutionné le domaine. En effet, ce nouveau système avec l'assistance de la bioinformatique a permis de passer à une identification entre 100 à 1000 protéines par jour avec une sensibilité de l'ordre du femtomole ($10^{-15} mol$). Cela a bien sûr entraînait une

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

accumulation de grandes quantités de données qu'il fallait ensuite analyser. Pour répondre à ce besoin, un grand nombre d'outils ont été développés. Aussi a-t-il fallu convertir les données fournies du spectromètre pour ensuite passer réellement à l'identification. C'est dans cette étape d'identification qu'à partir des données transformées du spectromètre un nombre important de techniques et de logiciels ont été mis en place puis optimisés : MASCOT [106], Profound [138], SEQUEST, ProteinProspector, …. Derrière ce désir d'identifier de manière sûre des protéines est arrivé celui d'identifier de nouvelles protéines ou de nouveaux variants inexistants dans les bases de données utilisées tel que SwissProt, Trembl [14], …

Avant de pouvoir s'attacher à l'identification des protéines en tant que tel, il est nécessaire de rappeler comment elles sont constituées.

2.1.2 Acides aminés et protéines

L'acide aminé est la brique de base d'une protéine ou plus généralement d'un polypeptide. La figure 2.2 présente la forme générale d'un acide aminé où l'on peut voir une structure carbonée composée d'une fonction amine (NH_2) et d'une fonction carboxylique ($COOH$). Tous les acides aminés sont basés sur le même « coeur » et diffèrent seulement par le résidu. La figure 2.2 montre aussi la forme d'un acide aminé au sein d'une chaîne d'acides aminés.

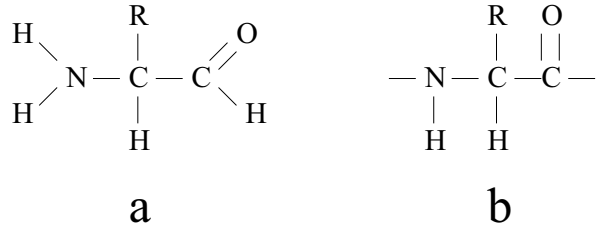


Fig. 2.2 – Forme générale d'un acide aminé seul (a) et au sein d'une chaîne acides aminés (b).

On dénombre aujourd'hui vingt-deux acides aminés. En effet, en plus des vingt acides aminés standards, deux autres sont aussi considérés car ils peuvent être directement obtenus à partir du code génétique :

- la sélénocystéine : c'est la forme biologique majoritaire du sélénium chez les eucaryotes. L'originalité de cet acide aminé réside dans ses mécanismes de biosynthèse et d'incorporation dans les protéines.
- la pyrrolysine : cet acide aminé a été identifié dans les gènes codant des méthyltransferases dans une bactérie méthanogène (Methanosarcina Barkeri).

La figure 2.3 décrit la formule chimique des ces deux acides aminés.

Il existe d'autres acides aminés non standards mais ils proviennent pour la plupart de modifications qu'ont subis les acides aminés standards. Par exemple, l'hydroxyproline provient de l'ajout d'un groupement hydroxyle sur la proline via une modification post-traductionnelle. La table 2.1 indique pour chacun des vingt-deux acides aminés standards : leur nom complet, leur nom court et la formule chimique du résidu qui les caractérise.

Les vingt acides aminés les plus souvent rencontrés sont classés suivant leur type fonctionnels :

- les acides aminés hydrophobique(non polaires) : alanine, isoleucine, leucine, méthionine, phénylalanine, proline, tryptophane et valine.

[14] Le site `www.uniprot.org` permet d'accès à ces bases de données en parallèle

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

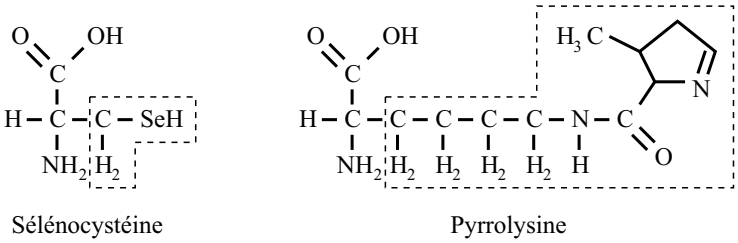


FIG. 2.3 – Formules chimiques de la sélénocystéine et de la pyrrolysine. Le résidu caractéristique est entouré en pointillé.

Nom complet	**Nom court**		**Formule du résidu associé**
	3 lettres	**1 lettre**	
Alanine	Ala	A	CH_3
Arginine	Arg	R	$C_4H_{10}N_3$
Asparagine	Asn	N	C_2H_4NO
Acide aspartique	Asp	D	C_2H_302
Cysteine	Cys	C	CH_3S
Acide glutamique	Glu	E	$C_3H_5O_2$
Glutamine	Gln	Q	C_3H_6NO
Glycine	Gly	G	H
Histidine	His	H	$C_4H_5N_2$
Isoleucine	Ile	I	C_4H_9
Leucine	Leu	L	C_4H_9
Lysine	Lys	K	$C_4H_{10}N$
Methionine	Met	M	C_3H_7S
Phenylalanine	Phe	F	C_7H_7
Proline	Pro	P	C_3H_5
Pyrrolysine	Pyl	O	$C_{10}H_{17}N_2O$
Sélénocystéine	Sec	U	CH_3Se
Serine	Ser	S	CH_3O
Threonine	Thr	T	C_2H_5O
Tryptophane	Trp	W	C_9H_8N
Tyrosine	Tyr	Y	C_7H_7O
Valine	Val	V	C_3H_7

TAB. 2.1 – Tables rassemblant les noms (complets et courts) des 22 acides aminés actuellement utilisés dans les bases de données protéiques ainsi que les formules des résidus associés.

- les acides aminés hydrophiles (polaires) : asparagine, cystéine, glutamine, glycine, sérine, thréonine et tyrosine.
- les acides aminés chargés positivement : arginine, histidine et lysine.
- les acides aminés chargés négativement : acide aspartique et glutamique.

Formé d'un ensemble d'acides aminés (généralement 50 et plus), la protéine est une des molécules essentielle du vivant car elles participent à la plupart des grands systèmes biologiques : catalyse, transport, communication, signalisation, reconnaissance, structure, …. Par conséquent, on comprend mieux

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

pourquoi l'étude de protéome est si importante : elle permet d'en apprendre plus sur le fonctionnement du vivant.

2.1.3 Spectrométrie de masse

La spectrométrie de masse permet d'identifier et de quantifier des molécules mais aussi de faire de l'analyse structurale et d'obtenir des données d'imagerie utiles pour la recherche de biomarqueur. Un spectromètre est consisté de 4 parties :

1. Le système d'introduction de l'échantillon : l'introduction se produit sous forme gazeuse, liquide, solide ou encore par l'association à une méthode séparative (chromatographie en phase liquide, chromatographie en phase gazeuse, électrophorèse capillaire , ...).
2. La source d'ionisation : elle consiste à vaporiser les molécules et à les ioniser. Plusieurs types de sources existent et sont utilisées en fonction du résultat recherché et des molécules analysées. Dans notre cas, la source d'ionisation est une désorption-ionisation laser assistée par matrice (MALDI).
3. L'analyseur : il permet de séparer les ions en fonction de leur rapport masse/charge (m/z). Il existe des analyseurs basse résolution et haute résolution. Dans notre cas, nous utilisons un analyseur haut résolution basé sur le temps de vol (« Time Of Flight » ou TOF). Pour réaliser des expériences de spectrométrie de masse en tandem (MS/MS), des analyseurs peuvent être couplés. Un premier analyseur sépare les ions, une cellule de collision permet de fragmenter les ions, et un second analyseur sépare les ions fragments.
4. Le détecteur et système de traitement : c'est le procédé qui permert de transformer les données mesurées en signal électrique amplifié pour un traitement informatique.

La figure 2.4 montre un exemple de spectre MS que l'on obtient sur le spectromètre que nous utilisons (MALDI-TOF).

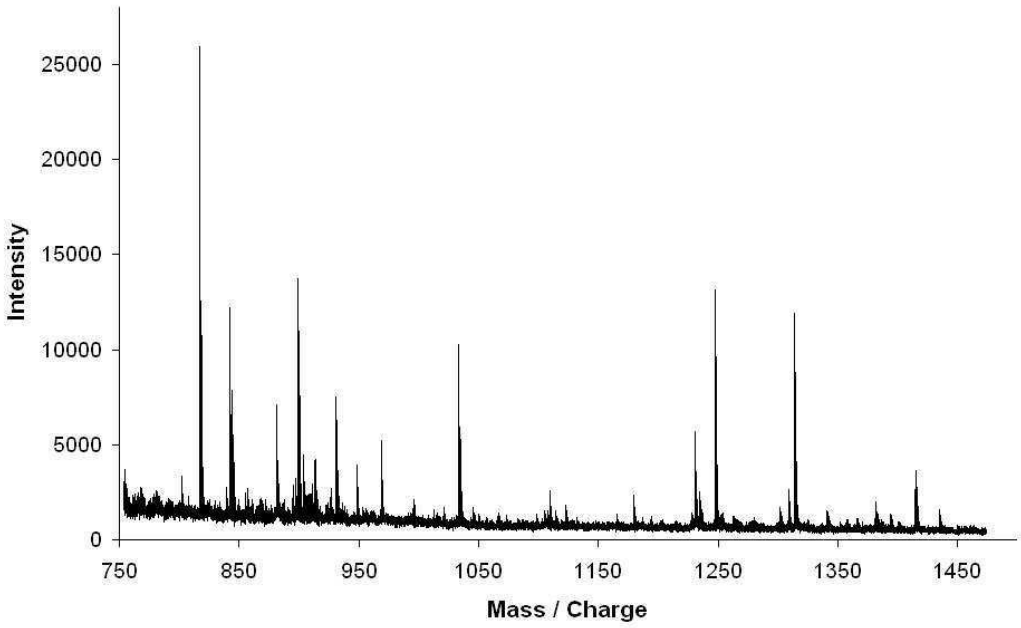


Fig. 2.4 – Spectre MS de la protéine humaine Apo-AI obtenu sur un spectromètre de masse de type MALDI-TOF.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Un spectre MS correspond à ensemble de couples (M,I) avec M=m/z, m la masse et z la charge, et I l'intensité du signal. Ci-dessous le début du fichier décrivant le spectre de la figure 2.4 est donné. Généralement, pour les spectres MS, la charge est de 1 mais cela dépend du spectromètre utilisé.

```
$
!
754.002019 3045.48
754.020777 2837.28
754.039536 2670.24
754.058295 2337.75
754.077054 2154.88
754.095814 2164.38
754.114573 2212.67
754.133333 2183.37
754.152094 1980.71
754.170854 1694.13
...
```

D'un spectre MS brut est extrait une liste de masses correspondant aux pics monoisotopiques du signal. Un pic monoisotopique correspond au premier pic du profil isotopique, celui qui correspond aux isotopes les plus stables. De cette manière, une liste de masses est obtenue où chaque masse est censée correspondre à celle d'un peptide (sous-partie du polypeptide expérimental donc de la protéine originelle). C'est à partir de cette liste de masse que va ensuite être effectuée la recherche dans les bases de donnée. C'est une identification par empreinte de masse peptidique (« peptide mass fingerprinting » ou PMF).

Lors d'expérience de spectrométrie de masse en tandem, ce sont des spectres MS/MS qui sont obtenus. Dans ce cas, chaque spectre correspond à un seul peptide qui a été fragmenté en ions. Dans ces spectres, les charges sont généralement supérieures à 2. Ci-dessous l'exemple d'un fichier .mgf (mascot generic file) qui rassemble un ensemble de spectres MS/MS.

```
...
BEGIN IONS
TITLE= Cmpd 1, +MSn(350.00000), 0.2 min
PEPMASS=348.104058 13321
CHARGE=2+
295.623590 11907 3+
297.665965 9208 2+
301.741393 9133 3+
303.288158 8672 3+
303.482477 11411 1+
...
END IONS
...
```

Chaque spectre MS/MS est présenté entre les mots clefs « BEGIN IONS » et « END IONS ». C'est à partir de ce genre de fichier que les identifications par empreinte de fragment (« peptide fragment fingerprinting » ou PFF) ou les méthodes par « de novo peptide sequencing » sont réalisées. Les méthodes

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

par PFF sont l'équivalent MS/MS des méthodes par PMF alors que les méthodes dites par « de novo peptide sequencing » se servent de l'information contenue dans chaque spectre MS/MS pour faire du séquençage, généralement partiel, de peptide. Cette dernière méthode est actuellement la plus utilisée en terme d'identification de protéines. Les spectres MS^x avec $x > 2$ sont aussi utilisés mais nous n'en auront pas l'utilité dans les travaux que nous présenterons.

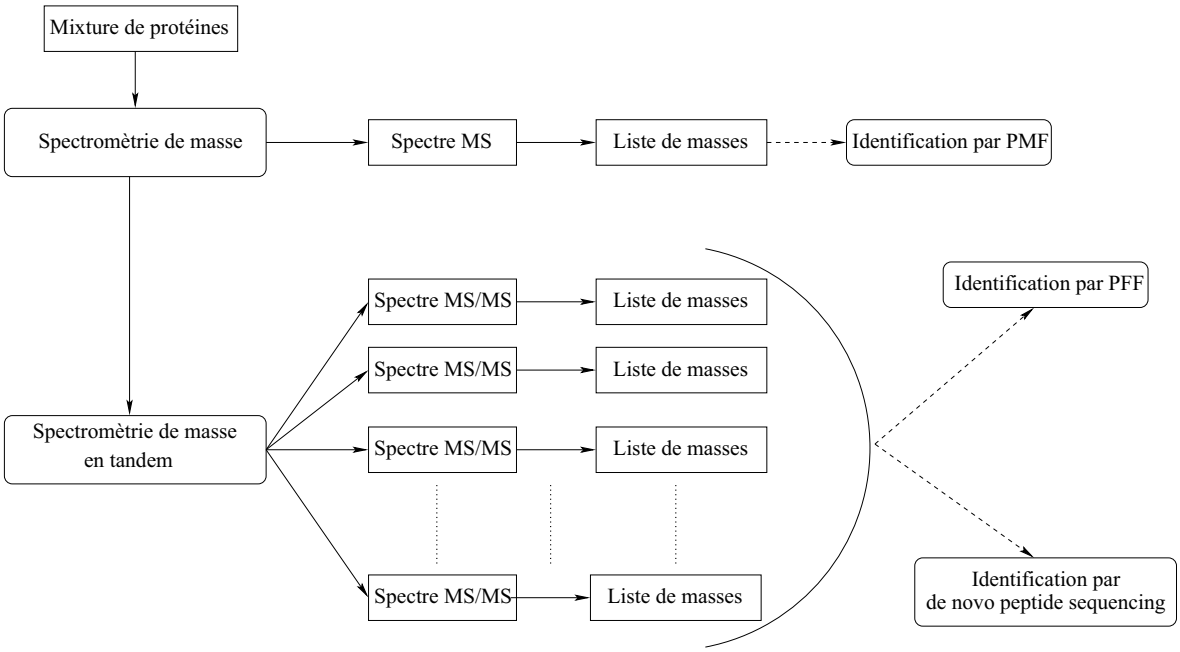


Fig. 2.5 – Synthèse des méthodes d'identification à partir d'une mixture de protéines données. Limitation aux spectres MS et MS/MS.

La figure 2.5 résume de manière schématique les différentes procédures d'identification de protéines à partir de spectres MS et MS/MS.

Grâce aux données de spectrométrie de masse, un grand nombre de méthodes et d'outils ont vu le jour, nous allons maintenant brièvement les présenter.

2.2 Etat de l'art

Les méthodes d'identification par PMF font parties des méthodes d'identification de protéines assistées par spectrométrie de masse les plus anciennes. La principale différence entre ces méthodes vient de leur fonction de comparaison entre la liste de masses obtenue à partir du spectre expérimental et les protéines des bases de données [54, 81, 96, 105]. Ces techniques permettent de trouver les protéines ayant l'empreinte de masse peptidique la plus proche de celle expérimentale. La simple comparaison de deux listes de masse peut paraître triviale mais un grand nombre de paramètres entrent en jeu dans ce calcul. En effet, il faut savoir qu'il y a forcément une différence entre les masses expérimentales et les masses calculées à partir de séquence. Même si l'évolution des spectromètres a permis d'obtenir des mesures de plus en plus précise, la gestion d'un quelconque décalage doit être pris en compte. De plus, l'existence des modifications post-traductionnelles a un impact sur le nombre de masses possibles à comparer.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Ainsi, il arrive que dans certains cas, il ne soit pas possible de départager deux protéines, c'est la limite des méthodes par PMF, elles ont un niveau de précision limité.

Les données de type MS/MS apportent une information sur la séquence des peptides, chose que les spectres MS ne peuvent faire. Ils ont donc une meilleure précision et permettent par PFF de différencier des protéines que les méthodes par PMP ne pouvaient faire [2, 24, 25, 26, 86]. Comme évoqué plus tôt, le « de novo peptide sequencing » reste aujourd'hui une des méthodes les plus utilisées à l'heure actuelle pour faire de l'identification de protéines [23, 31, 46, 83, 87, 117]. Malheureusement, comme toutes données biologiques, les spectres MS/MS sont souvent bruités et/ou incomplets et donc, on ne peut obtenir que des séquences partielles de peptides. Ces ensemble de séquences partielles peuvent être suffisantes pour obtenir des identifications de bonne qualité ou pour prouver l'existence de certains types de caractéristiques ne pouvant être décelées à partir de spectres MS. Par exemple, on peut déterminer la présence de modifications post-traductionnelles non prévisibles ou découvrir des variants de protéines.

Les méthodes d'optimisation pour la résolution du problème du « de novo peptide sequencing » sont nombreuses, il y a les méthodes qui se basent sur une analyse directe des listes de masses [60], d'autres qui transforment les données sous forme de séquence spécifiques [58, 95, 117] ou qui créent des graphes [31] pour faire apparaître des « TAG » dans les données à séquencer. Concernant les méthodes d'optimisation qui ont été utilisées pour ce problème, on va retrouver :

1. La programmation dynamique [23, 46, 83].
2. Le branch and bound [15].
3. Les algorithmes génétiques mono-objectif [61] ou multi-objectif [87].

Concernant les outils existants exploitant les données de spectrométrie de masse en tandem, on pourra citer sans être exhaustif : MASCOT [106], Profound [138], une version modifiée de Sequest [85], Peaks [84], VEMS [90] et ProbIDtree [137].

Aujourd'hui, il existe énormément de méthodes et d'outils pour procéder au séquençage de peptides à partir de spectrométrie de masse en tandem. Cependant, derrière le terme de « de novo sequencing », la plupart des travaux ne font pas la différence entre le « de novo **peptide** sequencing » et le « de novo **protein** sequencing ». Le « de novo protein sequencing » correspond au fait de trouver la séquence d'une protéine à partirde données expérimentales sans utilisation d'autres connaissances. C'est donc bien une méthode dite « de novo ». Mais le fait de séquencer un ensemble de peptides, et donc faire du « de novo peptide sequencing », ne signifie pas que l'on a séquencé la protéine correspondante. En effet, les bouts de séquence sont peut-être présents mais la protéine n'est pas connu pour autant car seul un ordonnancement des peptides permettrait l'accès à cette information. A l'heure actuelle [15], aucune approche bioinformatique ne propose une telle méthode. Le principe du « de novo protein sequencing » s'inscrit dans les enjeux actuels de la protéomique : découvrir des séquences de protéines là où les banques de données de protéines provenant de donnée génomique ne sont d'aucune aide, c'est-à-dire les organismes pour lesquels les génomes sont inconnus comme ceux des plantes par exemple.

[15] Depuis 2005 jusqu'à la dernière révision dernier trimestre 2009

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.3 Un nouveau modèle pour l'identification de protéine par empreinte de masse peptidique

Pour faire de l'identification par empreinte de masse peptidique (PMF), la majorité des modèles utilise des fonctions de comparaisons basées sur des listes de masses : une extraite du spectre expérimentale et l'autre des bases de données considérées. Ces approches transforment donc les données d'entrées qui sont dans des formats différents en un format comparable qui est la liste de masses. La nouvelle approche proposée consiste à uniquement modifier les informations en provenance des bases de données pour effectuer une comparaison directe avec le spectre expérimentale. L'intérêt d'une telle approche est claire, elle permet d'éviter une quelconque perte d'information dûe à la phase d'extraction de données en provenance du spectre expérimental.

Deux étapes principales sont nécessaires pour transformer une protéine, représentée dans la base de données sous forme d'une chaîne d'acides aminés, en un spectre comparable à l'expérimental :

1. la digestion : cette étape, déjà utilisée dans les modèles existants, consiste à simuler l'action d'une enzyme sur la protéine pour obtenir des sous-ensembles d'acides aminés : les peptides. On utilise la même enzyme que celle utilisée durant le processus d'obtention du spectre expérimentale.
2. la simulation de spectres : à partir d'un ensemble de peptides représentés par leur formules chimiques, un ensemble de spectres indépendants vont être sommés pour obtenir un spectre complet équivalent simulé d'un spectre MS.

On obtient ainsi deux spectres qu'il faut ensuite comparer. Pour ce faire, une méthode de comparaison spectrale a été mise en place : elle se base sur l'analyse point à point des deux spectres pour déterminer un score de corrélation.

2.3.1 La digestion de protéine

A partir d'une enzyme donnée, il est possible de simuler la digestion d'une protéine par cette enzyme. Pour ce faire, il faut tout d'abord associer à chaque enzyme une grammaire de découpe qui lui est propre et qui traduit le plus fidèlement possible son comportement dans la réalité. Par exemple, la trypsine, une des enzymes les plus utilisées, peut avoir comme grammaire associée (simplifiée) « couper la chaîne après l'acide aminé lysine (K) ou arginine (R) s'il n'est pas suivi de l'acide aminé proline (P) ». Ensuite, dès que l'enzyme est choisie, on applique la grammaire de découpe sur la séquence d'acides aminés et on obtient les sous-séquences d'acides aminés désirées que sont les peptides. De ce fait, la digestion peut être implémentée sous forme d'un algorithme linéaire dans le nombre d'acide aminés formant la protéine. Cependant, dans le processus réel de digestion, il arrive que l'enzyme « rate » un ou plusieurs points de coupe. Dans ce cas, on parle de clivage manqué (« missed cleavage »). Ce facteur augmente le nombre de peptides générés lors de la digestion. En effet, si une protéine de n acides aminés possèdent k points de coupure possibles ($k << n$ généralement), une digestion standard i.e. sans clivage manqué va générer $k + 1$ peptides. Si un clivage manqué est autorisé au maximum, cela ajoute k nouveaux peptides qui correspondent à tous les cas de clivage manqué unique.

La figure 2.6 détaille un exemple de l'arbre de digestion complet que l'on obtient pour une protéine fictive qui peut avoir au maximum 4 clivages manqués. Comme on peut le constater sur cette figure, prendre en compte les peptides avec des clivages manqués revient à fusionner les peptides sans clivage manqué correspondant. Ainsi si on veut obtenir un peptide avec k clivage manqué, il suffit de fusionner $k + 1$ peptides contigus sans clivage manqué. Construire de manière incrémental les peptides avec des

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

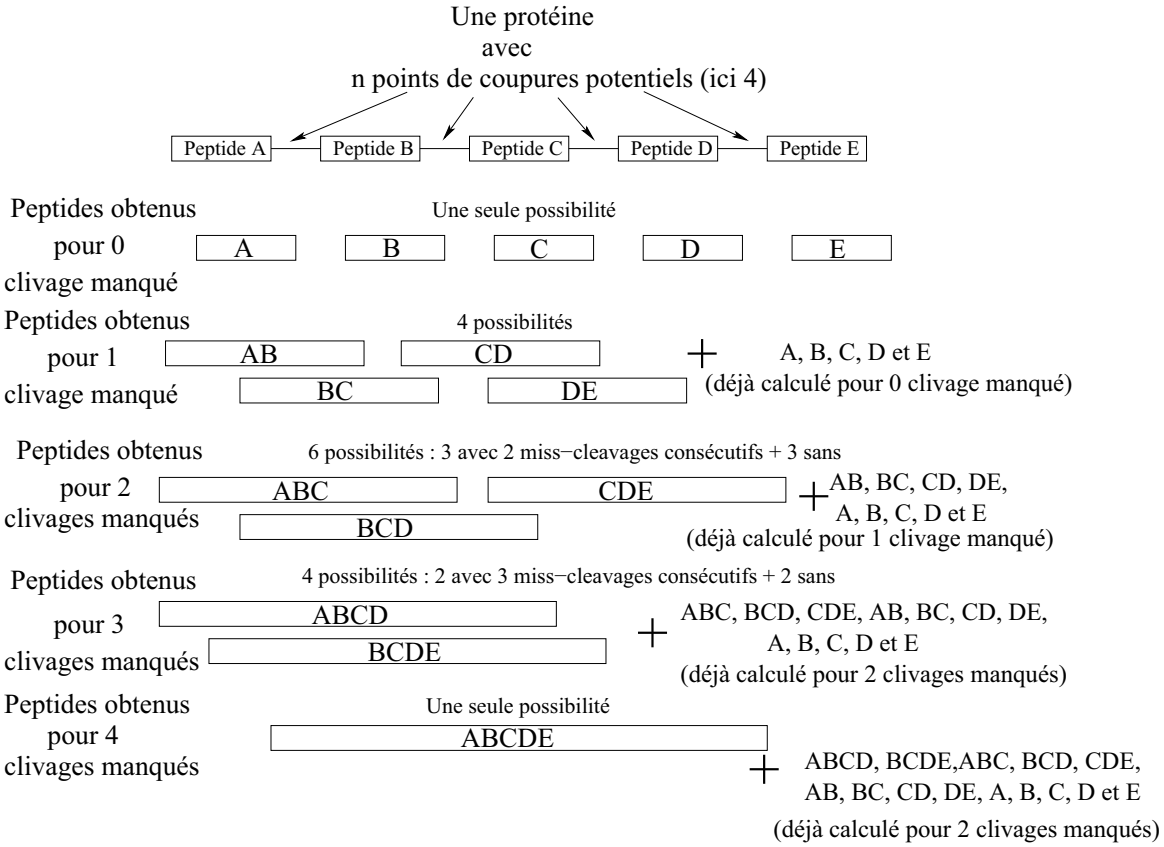


FIG. 2.6 – Parcours de l'arbre de digestion sur un exemple avec 4 points de clivages possibles.

clivages manqués assure de parcourir complètement l'arbre de digestion. La preuve de la complétude de l'arbre de digestion peut être consulté dans l'annexe B.1. En plus des clivages manqués, un autre phénomène très important est à prendre en compte lorsque l'on digère une protéine. Il s'agit des modifications post-traductionnelles. Ces modifications ont un impact considérable sur les protéines car elles peuvent activer, modifier ou inhiber les fonctions de celles-ci voir même fortement modifier leur structure. La prise en compte des modifications post-traductionnelles ont un impact important sur la digestion d'une protéine. En effet, suivant le type de modification que l'on essaye de simuler lors de la digestion, la modification peut être qualifié de « fixe » ou « variable ». Dans le premier cas, les modifications sont appliquées sur tous les emplacements compatibles possibles après digestion. Ainsi, le nombre de peptides résultant de la digestion reste inchangé. Dans le second cas, pour chaque emplacement de modification possible, on génère les deux peptides potentiels associés à la mise en place ou non de ce type de modification. Ainsi, chaque peptide se retrouve être la racine d'un arbre binaire de nouveaux peptides engendrés à cause des modifications post-traductionnelles. L'utilisation de modifications post-traductionnelles fixes conserve la linéarité de l'algorithme de digestion. Dès lors que des modifications post-traductionnelles variables sont activées, cette linéarité est perdue.

Le processus complet de digestion est englobé dans ADP qui signifie **A**utomatic **D**igestion **P**rocess, c'est-à-dire "processus automatique de digestion". On pourra remarquer l'analogie avec l'ADP, l'adénosine diphosphate, qui est le résultat de l'hydrolyse de l'ATP, l'adénosine tryphosphate, qui est la réserve d'énergie de la cellule. En effet, c'est une réaction exergonique i.e. qui fournit de l'énergie. « Notre »

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

ADP correspond donc au principe complet de digestion de protéine tel que nous l'utiliserons dans le reste de ce chapitre. Selon la configuration de la digestion, elle peut se dérouler en 1 à 4 étapes pour être complète, résumée dans l'algorithme 6 :

1. La découpe de la protéine par l'enzyme est simulée et un ensemble de peptides (sans clivage manqué) est obtenu. Il s'agit de la digestion standard.
2. Si un nombre de clivage manqué est demandé, k, on procède à la génération de tous les peptides avec un nombre de clivage manqués compris entre $[1, k]$ à partir de ceux de l'étape 1.
3. Si des modifications post-traductionnelles **fixes** sont activées, on les applique sur tous les sites de peptides obtenus lors de la dernière étape effectuée (1 ou 2).
4. Si des modifications post-traductionnelles **variables** sont activées, on génère les nouveaux peptides correspondant à la prise en compte de ces modifications à partir des peptides obtenus précédemment (étape 1, 2 ou 3).

Algorithme 6 : la digestion complète d'une protéine.

Données : Une séquence d'acides aminés (la protéine).
Résultat : Un ensemble de sous séquence avec de l'information supplémentaire si nécessaire (les peptides).

```
1  Recherche des points de clivages;
2  Digestion sans clivages manqués = génération de N peptides;
3  si k clivages manqués sont autorisés alors
4  |  pour i=1 à k faire
5  |  |  positionnement sur le premier peptide sans clivage manqué;
6  |  |  tant que i+1 peptides contigu sans clivage manqué sont à disposition faire
7  |  |  |  fusion de i+1 peptides contigu sans clivage manqué;
8  |  |  |  décalage à droite d'un peptide pour la fusion suivante;
9  |  |  fin
10 |  fin
11 |  Génération de M peptides supplémentaires;
12 sinon
13 |  M=0;
14 fin
15 si Modification post-traductionnelles fixes alors
16 |  pour i=1 à (N+M) faire
17 |  |  Application des modifications sur chaque site possible;
18 |  fin
19 fin
20 si Modification post-traductionnelles variables alors
21 |  pour i=1 à (N+M) faire
22 |  |  Génération des combinaisons de présence ou non de la modification.
23 |  fin
24 |  Obtention de P nouveaux peptides;
25 sinon
26 |  P=0;
27 fin
28 retourner (N+M+P) peptides;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

ADP est aussi un outil autonome pour procéder à la digestion de banques de données de protéines au format FASTA. C'est un outil « open-source » développé sous la licence française CeCILL-C [16]. L'organisation interne d'ADP est visible dans l'annexe B.2.

Un fois qu'il est possible de générer les peptides résultant de la digestion, on peut en déduire les formules chimiques associées et ainsi pouvoir passer à la génération de spectres.

2.3.2 La génération de spectres

Le but de la génération de spectres est de pouvoir obtenir des données permettant une comparaison directe avec l'expérimental. La génération de spectres est inspirée d'une méthode basée sur les Transformées de Fourier Rapides (TFR) proposé par Rockwood [111, 112]. Celle-ci a été étendue pour permettre de générer un spectre complet à partir d'un ensemble de formules chimiques. Pour comprendre cette méthode, il est tout d'abord nécessaire de comprendre comment la distribution isotopique d'un seul élément est générée. Ainsi générer la distribution isotopique de C_5 nécessite 3 étapes :

1. La distribution isotopique initiale, dans notre cas celle de C_1, est transposée dans l'espace de Fourier par TFR.
2. Dans cet espace, le signal est multiplié successivement par lui-même pour arriver à la puissance i, i correspondant à l'abondance de l'élément considéré, dans notre exemple i vaut 5.
3. Par TFR inverse, le signal est transposé dans l'espace Euclidien et on obtient ainsi la distribution isotopique désirée.

Notre générateur de spectres étend cette approche de calcul de distribution isotopique pour calculer des spectres de protéines complets. Un spectre théorique (ou simulé) est obtenu en quatre étapes schématisées par la figure 2.7 :

1. Une protéine est digérée en peptides.
2. Pour chaque peptide, la formule chimique de chaque acide aminé est calculée.
3. Pour chaque formule chimique :
   - Pour chaque élément de la formule, la distribution isotopique correspondante est calculée.
   - Une fonction Gaussienne est utilisée pour donner une largeur aux pics.
   - Les différentes distributions sont sommées pour générer un seul spectre.
4. Les spectres des peptides sont additionnés pour générer le spectre théorique de la protéine.

De par l'utilisation des TFR, qui ont une complexité en $O(Nlog_2N)$, ce processus de génération de spectres peut être admis en $O(Nlog_2N)$ avec N le nombre de points caractérisant un spectre. Le coût associé est temporel mais aussi spatial car les tous les calculs sont basés sur des multiplications de nombres complexes et que la plupart des spectres sont stockés en mémoire pendant le processus. En effet, les distributions isotopiques des éléments de bases jusqu'à une certaine abondance sont gardés en mémoire pour optimiser les calculs et éviter des redondances. Le processus complet de génération des spectres simulés est englobé dans SpecS qui signifie **S**pectrum **S**imulator, c'est-à-dire "simulateur de spectres". SpecS est un outil autonome pour générer des spectres simulés à partir de données au format FASTA. A la manière d'ADP, c'est un outil « open-source » développé sous la licence française CeCILL-C. L'organisation interne de SpecS est visible dans l'annexe B.4.

Avec ADP et SpecS, nous sommes capables de générer un spectre à partir de n'importe quelle protéine au format FASTA.

[16] Plus d'information sur http ://www.cecill.info/

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

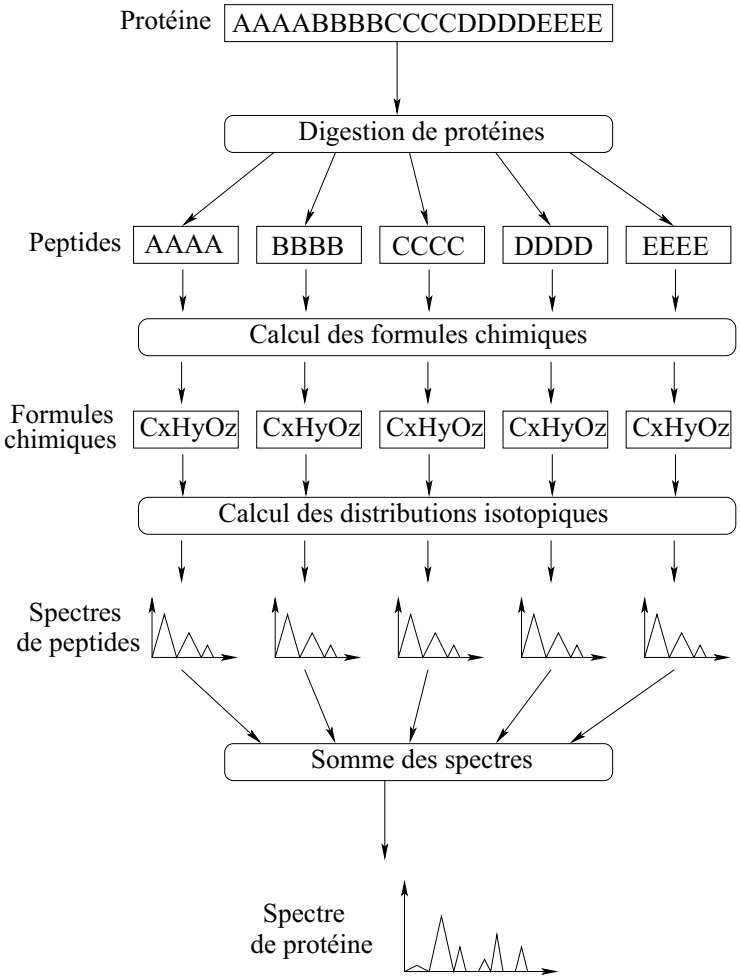


FIG. 2.7 – La génération de spectres en 4 étapes.

2.3.3 Intégration au sein d'un nouveau moteur d'identification : ASCQ_ME

Ascq_me est un moteur d'identification par empreinte de masse peptidique basé sur une nouvelle fonction d'évaluation pour comparer les protéines des bases de données face à un spectre expérimental. Ascq_me est à la base une application en ligne de commande fonctionnant avec un fichier de configuration et fournissant en sortie un ensemble d'information concernant le résultat de l'identification. Par un souci de validation de la part de la communauté mais aussi pour proposer notre modèle au plus grand nombre, nous avons mis en place une version online du moteur d'identification. Cette version, allégée, permet de faire des requêtes à la manière de ce qu'il peut se faire avec les autres outils d'identification de protéines par PMF. Le site web d'Ascq_me est consultable à l'adresse suivante :

$$https : //www.genopole - lille.fr/logiciel/ascq_me/$$

Il présente le modèle, l'application, et permet d'avoir accès à l'interface de requête d'identification ainsi qu'aux sources puisqu'Ascq_me est un outil open-source. Des captures des différentes étapes de l'envoi d'une requête d'identification avec Ascq_me sont disponibles en annexe B.6.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Nous allons maintenant expliquer en quoi Ascq_me est différent des autres approches et comme fonctionne sa fonction de comparaison spectrale.

2.3.3.1 Schéma global

Le schéma 2.8 décrit les différences qu'il y a entre Ascq_me et les autres approches existantes. L'organisation interne d'Ascq_me et les caractéristiques de la version actuelles sont respectivement visibles dans l'annexe B.5.

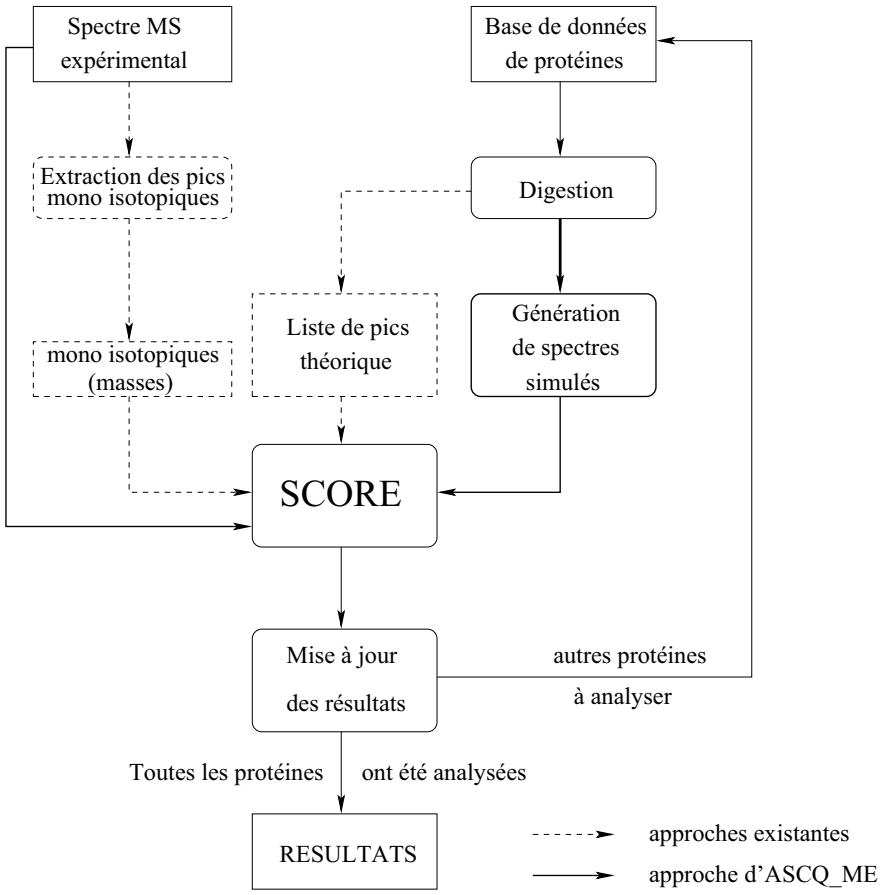


FIG. 2.8 – Identification par empreinte de masse peptidique : comparaison des approches.

Comme nous l'avions précédemment expliqué et comme cela est visible sur la figure 2.8, les méthodes existantes d'identification par PMF comparent les protéines des bases de données et les spectres expérimentaux sous à l'aide d'un format commun : la liste de masses. Côté base de données, la génération de ces listes de masse n'engendre pas de perte d'information par contre côté expérimental, la génération de la liste de masses représentative entraîne forcément une perte d'information. Pour pallier ce souci, nous laissons brut les données expérimentales et transformons les protéines des bases de données dans un format équivalent aux données expérimentales. Ainsi, d'un côté comme de l'autre, l'information est complètement présente sans perte. C'est un avantage non négligeable lorsque les données biologiques en elles-mêmes sont déjà bruitées.

Nous allons maintenant détailler notre fonction de comparaison spectrale.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.3.3.2 La comparaison spectrale

Une fois des spectres simulés obtenus, il fallait mettre en place une procédure d'évaluation de ceux-ci face aux spectres expérimentaux. Cette évaluation doit pouvoir donner une réponse à différents problèmes liés à la comparaison de spectres :

- Comment différencier le bruit et l'information dans le spectre expérimental ?
- Selon quelle mesure de qualité doit-on évaluer le spectre simulé par rapport à l'expérimental ?
- Quel information va renvoyer l'évaluation finale ?

La première question est toute naturelle puisque d'une part nous travaillons sur des données biologiques qui par définition sont forcément dans une certaine mesure bruitées et d'autre part, utilisé directement le spectre brut apporte une plus grande somme d'information que la simple liste de masses. Cependant plus d'information inclut aussi forcément plus de bruit. La séparation entre bruit et information n'est pas faisable de manière sûre automatiquement. Nous avons choisi une méthode simple et rapide qui permet de faire un minimum de distinction tout en espérant limiter l'impact du bruit. Il n'est pas grave de garder un peu de bruit dans les données par contre il n'est pas raisonnable de perdre de l'information. C'est pour cela que nous avons mis en place un seuil d'intensité qui est calculé dynamiquement à partir de la configuration d'Ascq_me et du spectre expérimental utilisé. Ce seuil est calculé à partir d'un examen de l'ensemble du spectre pour extraire une intensité moyenne répartie le long de celui-ci. Ce seuil est peu discriminant mais permet d'avoir une borne minimale lors de la comparaison.

Concernant la mesure de qualité pour comparer le spectre simulé et l'expérimental, la seule information réelle que l'on possède est que les méthodes basées sur les listes de masse n'ont pas l'intensité des pics expérimentaux. C'est la seule information fiable qui est disponible. En effet, l'intensité des spectres simulés générés par SpecS n'est pas une donnée fiable car il ne s'agit que de l'addition de différentes distributions isotopiques. L'intensité obtenue n'est pas un reflet du processus expérimental tel qu'il se passe à l'intérieur du spectromètre de masse. Il y a deux raisons à cela : premièrement, il existe de nombreux types de spectromètres qui ont des fonctionnements différents. Par conséquent, il faudrait pouvoir configurer le simulateur en fonction de type de spectromètre utilisé ce qui impose une connaissance approfondie de leur fonctionnement et la manière dont est calculée l'intensité. Ce qui amène à la deuxième raison : il n'existe pas à l'heure actuelle de modèle fiable pour simuler la réelle intensité des pics d'un spectre. Par souci de facilité, nous avons décidé de baser nos calculs de score de corrélation uniquement sur les intensités en provenance du spectre expérimental.

Enfin, la comparaison spectrale doit renvoyer une information traduisant la qualité de la corrélation entre le spectre théorique et l'expérimental. Comme le spectre simulé est un spectre « parfait », il ne pourra jamais correspondre à 100% au spectre expérimental aussi bien préparé soit-il. Dans un spectre MS, à chaque peptide va correspondre un pic monoisotopique mais aussi la distribution qui lui est associé. Le but de notre comparaison est de pouvoir affirmer pour un peptide en particulier s'il a son équivalent, à un décalage près, côté expérimental. Pour apporter cette information, un score est donné à chaque peptide suivant une étude point à point de son spectre simulé face à celui expérimental.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

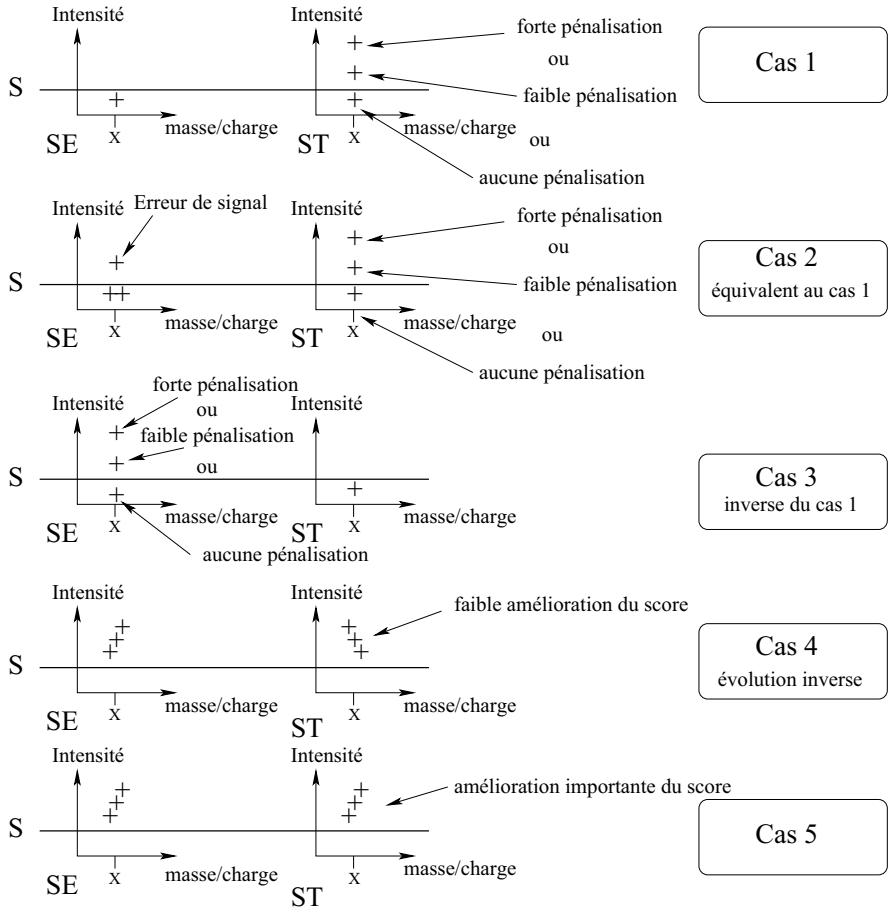


FIG. 2.9 – Calcul de la corrélation spectrale : cinq cas sont possibles. "+" correspond à un point du spectre, SE signifie spectre expérimental, ST spectre théorique et S est le seuil d'intensité différenciant le bruit de l'information.

Durant cette comparaison, cinq cas sont possibles comme schématisés dans la figure 2.9, on note S l'intensité seuil entre le bruit et l'information dans le spectre :

1. Le spectre expérimental a une intensité plus petite que S. Dans ce cas, le score est pénalisé par l'intensité du spectre simulé.
2. Le spectre expérimental a une intensité plus grande que S mais les points avant et après ont une intensité plus petite que ce seuil. Dans ce cas, le point est considéré comme un artefact ou du bruit dans le spectre. Le score est par conséquent aussi pénalisé par l'intensité de spectre simulé. On peut considérer ce cas comme équivalent au précédent.
3. Le spectre simulé a une intensité plus petite que S alors que le spectre expérimental a une intensité supérieure à ce seuil. Dans cette situation, le score est pénalisé avec l'intensité du spectre expérimental.
4. Les deux spectres ont une intensité supérieure à S. Cependant, leur évolution respective diffère. Un des spectres à une évolution croissante et l'autre une évolution décroissante. Dans ce cas, l'intensité du spectre expérimental est ajoutée au score. En effet, même si les évolutions sont différentes, il se peut très bien que l'on est à faire à la même distribution isotopique mais avec un décalage. Par conséquent le score n'est pas pénalisé.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

5. Les deux spectres ont une intensité supérieure à S et leur évolution respective coïncide. Dans ce dernier cas, l'intensité du spectre expérimental multipliée par une constante (typiquement 1.5) est ajoutée au score.

Certains peptides n'ont pas de score car ils ne correspondent pas aux critères de configuration choisis et sont donc été rejetés. Dans ce cas, leur score a pour valeur $-MAXIMUM_SCORE$. C'est une valeur spéciale unique à chaque couple spectre expérimental / fichier de configuration. C'est une borne maximum pour le score d'un peptide qu'il n'est pas possible d'atteindre. Cela permet de « tagger » les peptides rejetés. Selon son score, un peptide va être classé dans une catégorie particulière selon un score seuil (par défaut 0). Ce seuil est un paramètre modifiable qui permet de plus ou moins accepter des décalages entre le spectre simulé et l'expérimental. Ainsi, les peptides sont classés dans trois familles selon leur score :

- Les « bons » peptides : ils ont un score au moins égal au seuil (par défaut un score au moins positif). Ils sont considérés comme ayant une bonne corrélation avec le spectre expérimental.
- Les « mauvais » peptides : ils ont un score inférieur au seuil (par défaut un score négatif). Ils sont considérés comme ayant peu ou pas de corrélation avec le spectre expérimental.
- Les peptides « null » : ils ont un score égal à $-MAXIMUM_SCORE$.

Par conséquent, plus le seuil de différenciation sera élevé, moins le nombre de « bons » peptides sera important.

A partir de la répartition des peptides dans chaque famille, le score finale du spectre simulé correspond au nombre total de bons peptides BP par rapport au nombre total de peptides TP (bon et mauvais) sans compter les peptides redondants RP dus aux modifications post-traductionnelles variables si elles ont été activées.

$$score = (BP/(TP - RP)) \times 100$$

L'utilisation d'une valeur de type pourcentage a été motivée par tous les essais réalisés sur des bases de données réelles qui ont montré que la taille des protéines avait un impact direct sur les versions antérieures de notre score. L'annexe B.7 montre l'exemple d'une protéine bien connue des bases de données qui, par sa taille et le nombre de peptides qu'elle fournit, entraîne forcément de fausses corrélations et donc un bon classement dans les résultats d'identification. Le score actuel permet de pallier à ce type de protéine.

La fonction de comparaison représente la dernière étape du processus permettant la comparaison d'une protéine au format FASTA avec un spectre MS expérimental. Nous allons maintenant présenter les tests réalisés avec Ascq_me pour évaluer le comportement de notre fonction de comparaison spectrale.

2.3.4 Validation de d'ASCQ_ME

Afin de valider Ascq_me, nous avons procéder à deux séries de tests et comparer les résultats d'Ascq_me avec d'autres moteurs d'identification utilisés. Le but est de prouver que notre comparaison spectrale fonctionne.

2.3.4.1 Protocole expérimental

Dans cette section, nous détaillons, rapidement, comment les spectres fournis par le LCOM sont produits. Les différentes étapes sont :

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

1. Électrophorèse SDS PAGE : l'appareillage utilisé pour l'électrophorèse est un « EPS 301 » de GE Healthcare Biosciences. L'électrophorèse est réalisée dans un tampon contenant 0,025M de tribase, 0,192 de glycine et 0,1%(w/v)SDS.
2. Séparation en deux dimensions du gel : le première dimension est réalisée par un « Ettan IPGphor II » avec les protéines immobilisées sur des bandes. Ensuite, ces bandes sont placées tout en haut de gels de polyacrylamide pour la séparation selon la seconde dimension sur un système « Etta Daltsix ».
3. Coloration du gel : les gels sont colorés par un brassage d'une nuit dans une solution de complexe $\overline{RU(BPs)_3}$, une solution de coloration équivalente à la solution de coloration par fluorescence commerciale Ruby, après une étape de fixation.
4. Gel images : les gels sont scannés dans un scanner à fluorescence *Typhoon* 9000 avec un laser bleu 2 (λexc = 488nm / λem= 610nm BP 30) d'une résolution de 200 microns.
5. Digestion par la trypsine : les protéines peuvent être manuellement ou automatiquement digérées à l'aide d'un robot de digestion *Ettan*. Les gels sont automatiquement excisés et exposés à la trypsine (5ng/μL) pendant 4 heures. Ensuite les morceaux de gel sont laissés pendant 6 heures dans du bicarbonate d'ammonium.
6. Analyse par spectrométrie de masse : tous les spectres sont obtenus sur un MALDI-TOF Voyager STR DE (Applied Biosystems, Darmstadt, Allemange) en mode réflectron. Les identifications sont finalement réalisées grâce à ProteinProspector ou encore MASCOT.

Il s'agit du protocole standard et sauf mention du contraire dans les résultats, toutes les données produites suivent ce protocole.

2.3.4.2 Données

Pour la première série de tests, les protéines utilisées proviennent de kits de protéines standards. Les kits comportent des protéines de masse variable. Ceux utilisés proviennent de GE Healthcar Biosiences (Uppsala, Suisse). Le tableau 2.2 résume les protéines présentes dans ces kits ainsi que leur masse et leur proportion respective. L'analyse expérimentale se compose en une série de dilution dans l'eau de 1 μg à 0,5ng de protéines sur gels.

Tab. 2.2 – Protéines de différentes masses provenant du kit moléculaire utilisé. Les masses des protéines sont exprimées en Dalton (Da) et les quantités en microgrammes (μg).

Nom	**Masse moléculaire (Da)**	**Quantité (μg)**
Phosphorylase-b	97000	67
Albumine	66000	83
Ovalbumine	45000	147
Anhydrase carbonique	30000	83
Inhibiteur de trypsine	20100	80
Lactalbumine	14400	116

Pour la seconde série de tests, 16 spots d'un gel 2D de plasma humain ont été analysés par des outils standards d'identification de protéines afin de comparer les résultats avec ceux obtenus avec Ascq_me. En effet, dans ce test en conditions réelles, les protéines à trouver ne sont a priori pas connues.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.3.4.3 Résultats

Cette section présente les résultats obtenus avec Ascq_me lors des tests de dilution de protéines et de l'analyse du plasma humain.

Tests de dilution Lors de ces premiers tests, nous avons plus particulièrement examiné trois protéines du kit moléculaire : l'albumine bovine, l'ovalbumine de poulet et l'anhydrase carbonique bovine. Leur identifiant dans la base de données Swiss-Prot sont **P02769** pour l'albumine, **P01012** pour l'ovalbumine et **P00921** pour l'anhydrase carbonique. Le but de cette série de tests est de connaître la précision d'identification que possède Ascq_me. Ces protéines sont représentées par vingt-trois spectres correspondant à l'analyse de vingt-trois bandes [17] : huit pour l'albumine (bandes 14 à 20 et 20'), sept pour l'ovalbumine (bandes 21 à 27, pas de bande 25) et huit l'anhydrase carbonique (bandes 28 to 35). Les tableaux 2.3 et 2.4 montrent respectivement les paramètres de configuration de MASCOT et d'Ascq_me pour nos tests. A la vue de ces tableaux, on pourra remarquer que seul le nombre de clivages manqués diffère pour l'albumine et l'anhydrase carbonique. Cette différence s'explique par la volonté de configurer les outils afin qu'ils donnent les meilleurs résultats possible sans donner l'avantage à l'un ou à l'autre en particulier. De plus, concernant Ascq_me, une masse estimée de la protéine présente dans le spectre expérimental est donnée. Cette information, optionnelle dans la version « offline », est donnée dans ce cas car elle permet de simuler l'interrogation d'Ascq_me comme si c'était un utilisateur de l'interface web qui la faisait.

Tab. 2.3 – Configuration de MASCOT. CM correspond au nombre de clivages manqués autorisés et MOD aux modifications post-traductionnelles activées. Pour ces dernières, CMV signifie carbamidomethylation (C) variable et OMV oxydation (M) variable.

Nom	CM	Espèce	MOD
Albumine	1	BOVIN	CMV & OMV
Ovalbumine	1	CHICK	CMV & OMV
Anhydrase carbonique	1	BOVIN	CMV & OMV

Tab. 2.4 – Configuration d'ASCQ_ME. CM correspond au nombre de clivages manqués autorisés et MOD aux modifications post-traductionnelles activées. Pour ces dernières, CMV signifie carbamidomethylation (C) variable et OMV oxydation (M) variable. ME correspond à masse expérimentale.

Nom	CM	Espèce	MOD	ME
Albumine	0	BOVIN	VCM & VOM	69293
Ovalbumine	1	CHICK	VCM & VOM	42750
Anhydrase carbonique	0	BOVIN	VCM & VOM	28983

Le tableau 2.5 montre les résultats d'identification obtenus avec MASCOT et Ascq_me. Concernant Ascq_me, une identification est considérée comme un échec (noté "non") quand le rang de la bonne protéine est supérieur à 50. Dans le cas d'un rang entre 10 et 50, le résultat n'est pas vraiment concluant (notée "oui/non") et nécessite une analyse des fichiers de résultats. Si le rang est inférieur à 10, la protéine est estimée comme bien identifiée (noté "oui"). Ce classement en trois catégories a été justifié par la manière d'interpréter les résultats qu'ont les techniciens avec qui nous avons travaillé. En effet, ils ont l'habitude de parcourir le classement des protéines et sont capables de faire rapidement la différence entre les protéines qui apparaissent dans le classement mais qui ne sont pas intéressantes et celles d'intérêt.

[17] Désigne l'endroit où les données expérimentales étaient placées.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 2.5 – Résultats d'identification des 23 spectres. IM signifie identification par MASCOT, IA identification par Ascq_me et PPI le pourcentage de peptides identifiés.

Bande	IM	PPI	IA	Rang	PPI
14	oui	9.4%	oui	**2**	**16.8%**
15	oui	5.8%	oui	**3**	**7.2%**
16	oui	6.5%	oui	**9**	**8%**
17	oui	**5.5%**	non	> 50	∅
18	oui	**5.2%**	non	> 50	∅
19	oui	1.3%	oui/non	41	**9.6%**
20	non	∅	non	> 50	∅
20'	oui	1.3%	oui/non	47	**1.6%**
21	oui	11.9%	oui	**1**	**19%**
22	oui	12.6%	oui	**1**	**17.1%**
23	oui	9.1%	oui	**1**	**13.3%**
24	oui	11.9%	oui	**1**	**24.8%**
26	oui	2.8%	oui/non	24	**5.7%**
27	non	∅	oui/non	20	**6.7%**
28	oui	19%	oui	**1**	**22.2%**
29	oui	15.5%	oui	**1**	**22.2%**
30	oui	20.7%	oui	**1**	**29.6%**
31	non	∅	oui/non	24	**25.9%**
32	non	∅	non	> 50	∅
33	non	∅	non	> 50	∅
34	non	∅	non	> 50	∅
35	oui	**20%**	non	> 50	∅

C'est sur cette expérience d'analyse que nous avons basé notre répartition des résultats. Pour MASCOT et Ascq_me, le pourcentage de peptides identifiés correspond au nombre de peptides considéré comme présents dans le spectre expérimental divisé par le nombre total de peptides analysés (moins les peptides redondants dus aux modifications post-traductionnelles variables).

Le tableau 2.6 fait une synthèse des résultats provenant du tableau 2.5. A l'examen de ces deux tableaux, on remarque que le pourcentage de peptides identifiés par Ascq_me est plus important que celui de MASCOT. Cela peut notamment s'expliquer par le fait que la comparaison spectrale permet une plus grande flexibilité que les méthodes standards par analyse de listes de masses n'ont pas forcément. Cependant, on peut aussi voir que globalement, l'identification des protéines reste équivalente entre les deux

Tab. 2.6 – Comparaison des résultats d'identification entre MASCOT et Ascq_me. PMPIM correspond au pourcentage moyen de peptides identifiés par MASCOT, PMPIA à celui de ceux identifiés par Ascq_me et diff=PMPIM-PMPIA.

Nom	PMPIM	PMPIA	diff
Albumine	5%	**8.6%**	**+3.6%**
Ovalbumine	9.7%	**14.4%**	**+4.7%**
Anhydrase carbonique anhydrase	15.5%	**25%**	**+9.5%**

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

outils. Certaines protéines ne sont pas détectées par MASCOT mais le sont par Ascq_me et inversement. Le but n'est pas de prouver qu'Ascq_me est meilleur que MASCOT mais de prouver que son comportement et ses résultats sont cohérents et exploitables.

Nous avons précédemment évoqué la possibilité d'une analyse plus poussée des fichiers de résultat d'Ascq_me. En effet, même si l'interface web d'Ascq_me fournit tous les résultats d'identification nécessaires à l'utilisateur, elle ne correspond en fait qu'à l'extraction d'information en provenance de fichiers générés par Ascq_me. En effet, lors de son exécution, Ascq_me fournit trois fichiers différents :

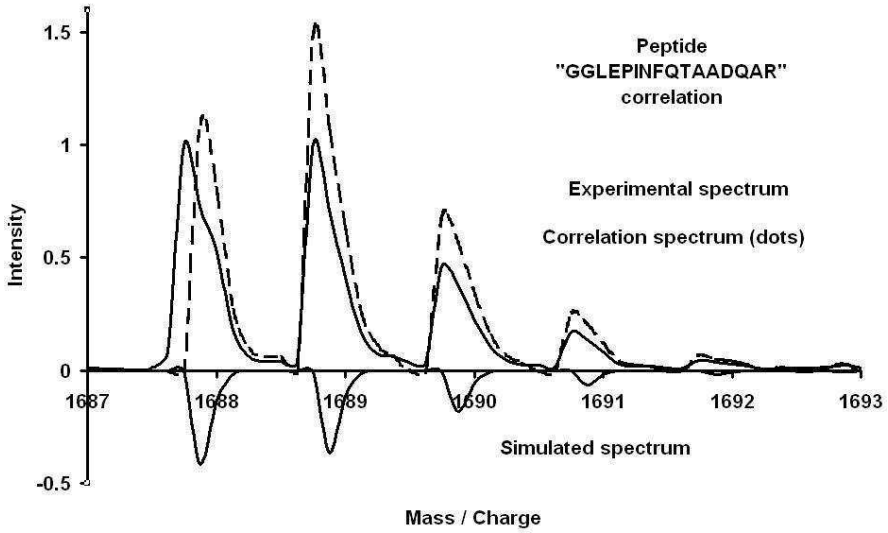


Fig. 2.10 – Analyse des résultats de la bande 21. Grâce au fichier « ascq_me_table_results », il est possible de tracer le spectre expérimental, le spectre simulé et le spectre de corrélation pour avoir une analyse visuelle des résultats. Il est aussi tout à fait possible de zoomer sur les peptides ayant de fortes corrélations.

– ascq_me_results : ce fichier contient les résultats de l'identification sous une forme lisible par l'utilisateur.

– ascq_me_table_results : ce fichier contient les résultats sous un format mis au point pour être ensuite utilisé dans un autre logiciel pour pouvoir tracer les différents spectres générés durant l'analyse. Il s'agit du spectre expérimental, du spectre simulé de chaque protéine classée mais aussi du spectre de corrélation entre ces deux spectres. Le spectre de corrélation est le résultat visuel de la fonction de comparaison spectral. C'est une information très intéressante notamment car cela permet une interprétation plus accessible des résultats pour l'utilisateur. Un exemple du type de schéma que l'on peut en extraire est montré par la figure 2.10. Cette figure a été obtenue un lisant simplement le fichier brut sous Excel et en sélectionnant directement les colonnes d'intérêt à dessiner. De plus, le nombre de points représentant les spectres dans ce fichier peut être adapté au logiciel de visualisation choisi. C'est un détail intéressant lorsque l'on sait que certains outils ne peuvent gérer des fichiers de spectres trop long. Par exemple, le tableur Excel ne peut gérer en un seul passage des données à tracer de plus de 32 000 points. Cela permet encore de réduire la tache de l'utilisateur.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

– ascq_me_peptides_scoring : ce fichier indique pour chaque protéine classée, le nombre de peptides qui ont effectivement était identifié lors de l'analyse avec leur score partiel et enfin le score global de la protéine.

Un exemple de chacun de ces fichiers est proposé dans l'annexe B.8.

Nous allons maintenant présenter les résultats de la seconde série de tests qui va confirmer la cohérence de notre fonction d'évaluation spectrale.

Identification des protéines du plasma humain Alors que les tests de dilution ont été spécialement réalisés pour tester Ascq_me, cette seconde série de test a été faite sur des données utilisées actuellement par le LCOM sur l'étude du protéome de l'homme. Les techniciens ont procédés à l'identification de protéines sur 16 spots d'un gel contenant des protéines du plasma humain et ont obtenu des résultats sur 7 d'entre eux grâce à plusieurs outils d'identification (MASCOT et ProteinProspector entre autres). Nous avons donc analysé ces mêmes spectres avec Ascq_me pour savoir s'il trouvait des résultats équivalents ou non.

Ainsi, à la manière des tests de dilution, le tableau 2.7 détaille des identifications réalisées et le tableau 2.8 fait une synthèse de ces résultats.

Tab. 2.7 – Résultats d'identification sur 7 spots où des protéines ont été trouvées. Id correspond à l'identifiant Swiss-Prot de la protéine que les autres outils ont classée en premier, PPIO au pourcentage de peptides identifiés par les autres outils, IA à l'identification d'Ascq_me et PPIA au pourcentage de peptides identifiés par Ascq_me.

Bande	Id	PPIO	IA	Rang	PPIA
D8	P02753	**17%**	non	> 50	∅
E7	P02753	**9.1%**	non	> 50	∅
B7	P02787	**11.5%**	oui	**1**	7.5%
B9	P02675	**5.6%**	non	> 50	∅
F5	P06727	**22%**	oui	**1**	11.1%
F7	P02647	**37.5%**	oui	**1**	30.8%
F8	P01009	6.8%	oui/non	17	**7.7%**

Globalement, Ascq_me arrive à identifier les protéines sur 4 spots sur 7 ce qui n'est pas négligeable puisqu'il est comparé à plusieurs outils d'identification et non pas seulement un seul. De plus, le pourcentage de peptides identifiés par Ascq_me est légèrement plus faible que celui des autres outils.

Tab. 2.8 – Comparaison des résultats d'identification entre Ascq_me et les autres moteurs d'identification utilisés (ProteinProspector and MASCOT). PMPIO correspond au pourcentage moyen de peptides identifiés par les autres outils, PMPIA à celui d'Ascq_me et diff = PMPIA-PMPIO.

PMPIO	PMPIA	diff
15.6%	14.3%	**-1.3%**

Cela tend à prouver que même si Ascq_me fonctionne, il n'est pas encore, dans sa version actuelle, assez performant pour être complètement exploité seul pour identifier des protéines. Il est important de se rappeler que la mise en place d'Ascq_me avait pour but premier de valider notre fonction de comparaison spectrale. Au vu des résultats, on peut affirmer que notre objectif est atteint.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Après cette première application de cette fonction de comparaison, nous allons maintenant exploiter tout son potentiel sur le problème de « de novo protein sequencing ».

2.4 SSO (Sequence, Shape, Order) : un nouveau modèle pour le *novo protein sequencing*

Afin de bien comprendre en quoi consiste le modèle SSO, nous commencerons par définir ce que nous entendons par « de novo protein sequencing » pour ensuite détailler ce que chaque étape de ce modèle peut apporter pour donner une solution à ce problème.

2.4.1 Définition du *de novo protein sequencing*

Par homologie avec la méthode dite de « de novo **peptide** sequencing », le « de novo **protein** sequencing » consiste en la prédiction de séquence de protéines à partir de données expérimentales sans utilisation ni a priori, ni a posteriori de bases de données. Aucune méthode existante ne fait à l'heure actuelle de réel « de novo **protein** sequencing ». Elles utilisent des résultats de « de novo **peptide** sequencing » standards sur des spectres MS/MS qu'elles exploitent. Le but peut être d'obtenir un meilleur séquençage (plus complet) [61] ou de caractériser une portion précise de séquence par homologie grâce à une base de séquences connues [117]. Dans tous les cas, ces méthodes ne peuvent réaliser un réel « de novo **protein** sequencing » car il leur est impossible de remonter jusqu'à la protéine originale. Cela vient du fait que les données MS/MS ne peuvent pas tout apporter notamment au niveau de l'ordonnancement des peptides qui forme la protéine d'intérêt.

2.4.2 Présentation de SSO

Le modèle SSO propose une méthodologie en trois étapes pour, à partir de spectres MS et MS/MS, retrouver la séquence originelle d'une protéine sans aide de bases de données. La figure 2.11 synthétise le principe. A partir de spectres MS/MS, un « de novo **peptide** sequencing » standard est utilisé. On obtient ainsi des portions de séquences correspondant à des peptides partiels, c'est l'étape « Sequence ». Ensuite, à l'aide d'un spectre MS, on va essayer de compléter les séquences partielles de peptides pour obtenir une empreinte de masse peptidique cohérente face à l'expérimentale, c'est l'étape « Shape ». Enfin, un autre spectre MS, issu d'une digestion différente, est utilisé pour retrouver le bon ordre des peptides qui amènera à la bonne séquence de protéine, c'est l'étape « Order ».

Nous allons maintenant détailler chacune de ces étapes en rappelant son principe, les processus biologiques qui y sont liés et les expérimentations que nous avons réalisées pour choisir de manière objective les meilleurs moyens de les implémenter.

2.4.3 S pour « Sequence »

Cette première étape est basée sur l'utilisation de spectres MS/MS. Chaque spectre MS/MS correspond à la fragmentation d'un peptide. Il est possible à partir des différents types d'ions identifiables dans ce type de spectre de retrouver la séquence du peptide originel. Le but de cette étape est de produire des séquences d'acides aminés de peptides à partir de ces spectres. Le principe de fragmentation de peptides est rappelé afin de mieux comprendre comment il est possible de séquencer ceux-ci à partir de l'identification des ions présents dans le spectre.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

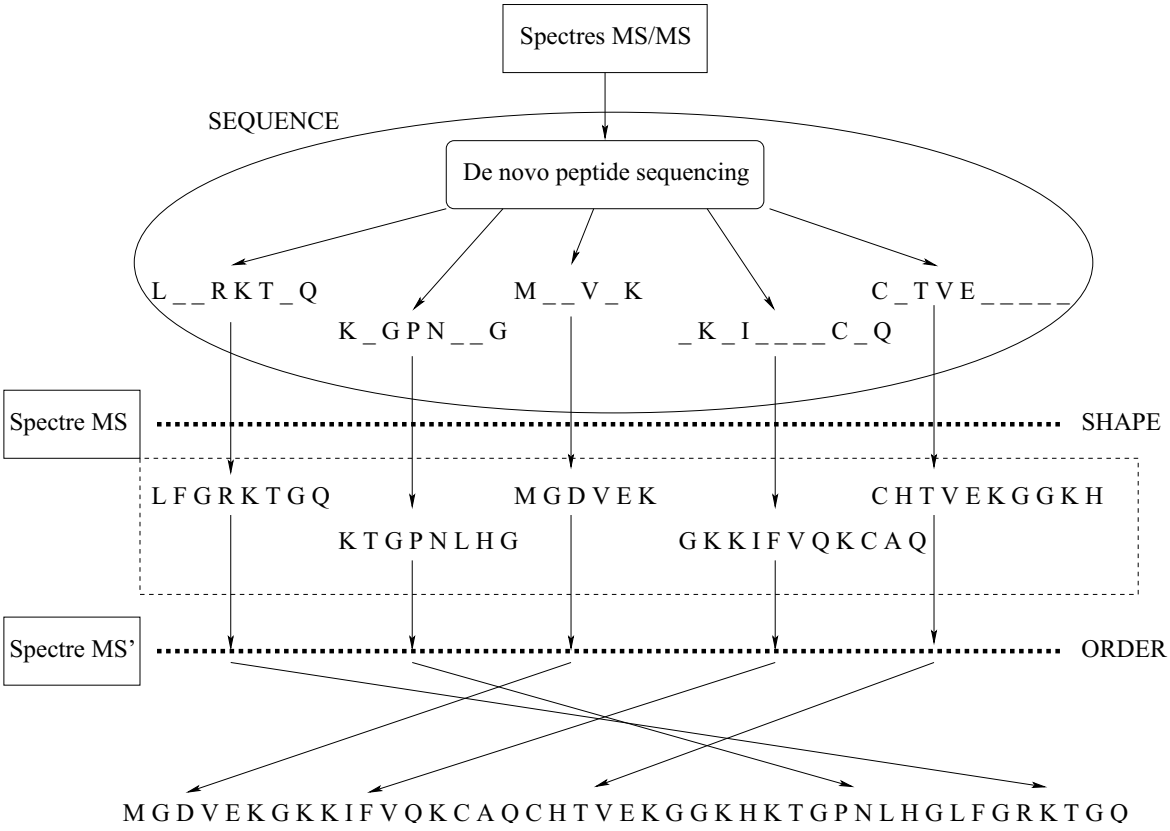


FIG. 2.11 – Schéma générique de l'approche SSO : Sequence, Shape, Order.

2.4.3.1 La fragmentation de peptides

La nomenclature actuelle utilisée pour les ions issus de la fragmentation a été proposée à l'origine dans [122] et a été modifiée ensuite dans [68].

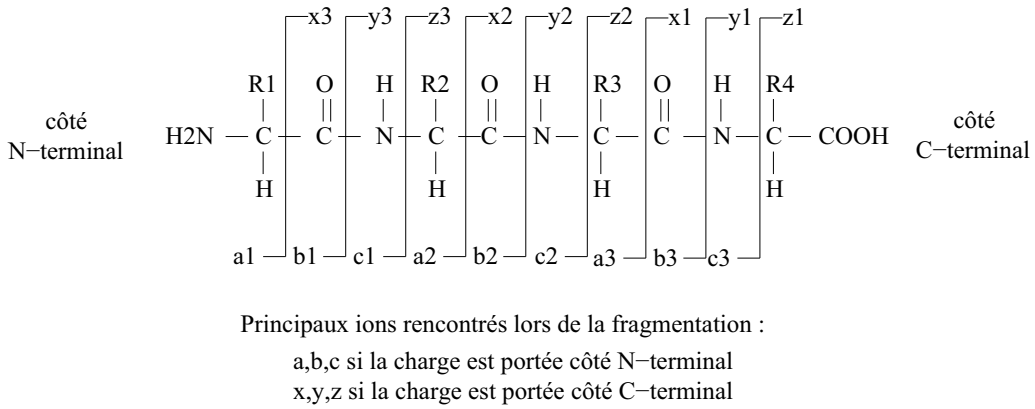


FIG. 2.12 – Règle de fragmentation des peptides en ions pour les principaux types existants (a, b, c, x, y et z).

La figure 2.12 présente les règles de coupure permettant d'obtenir les principaux types d'ions. Le

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

type d'ion obtenu dépend du sens et de l'endroit de la fragmentation. En notant M_{raa}, la masse totale des résidus des acides aminés d'un ion et M_X la masse de tous les atomes constituant la formule X, on obtient les formules suivantes pour calculer les masses des différents types d'ion :

$$ion\ a = M_H + M_{raa} - M_{CO} \quad ion\ b = M_H + M_{raa}$$
$$ion\ c = M_H + M_{raa} + M_{NH_3} \quad ion\ x = M_{OH} + M_{raa} - M_{CO}$$
$$ion\ y = M_{OH} + M_{raa} - M_{H_2} \quad ion\ z = M_{OH} + M_{raa} - M_{NH}$$

En notant M_{R_i} la masse du résidu de rang i, on peut déduire des formules de transition pour passer d'un type d'ion à un autre :

$$ion\ a_i\ vers\ ion\ b_i = M_{a_i} + M_{CO}$$
$$ion\ b_i\ vers\ ion\ c_i = M_{b_i} + M_{NH} + M_{H_2}$$
$$ion\ c_i\ vers\ ion\ a_{i+1} = M_{c_i} + M_{R_{i+1}} - M_{NH} - M_{CO} - M_{H_2}$$
$$ion\ z_i\ vers\ ion\ y_i = M_{z_i} + M_{NH} + M_{H_2}$$
$$ion\ y_i\ vers\ ion\ x_i = M_{y_i} + M_{CO} - M_{H_2}$$
$$ion\ x_i\ vers\ ion\ z_{i+1} = M_{x_i} + M_{R_{i+1}} - M_{NH} - M_{CO}$$

Grâce à ces formules, on peut mettre en place un algorithme linéaire de fragmentation de peptides sachant qu'il nécessite que la digestion ait été réalisée auparavant.

Le processus complet de fragmentation est englobé dans ATP qui signifie **A**utomatic fragmen**T**ation **P**rocess, c'est-à-dire « processus automatique de fragmentation ». Comme pour la digestion, on pourra remarquer l'analogie avec l'ATP, l'adénosine triphosphate, qui est la source d'énergie de la cellule. ATP correspond donc ici au principe complet de fragmentation de protéine tel que nous l'utiliserons dans le reste de ce chapitre.

ATP est aussi un outil autonome pour procéder à la fragmentation de banques de données de protéines au format FASTA. C'est un outil « open-source » développé sous la licence française CeCILL-C comme ADP et Ascq_me. L'organisation interne d'ATP est visible dans l'annexe B.3.

2.4.3.2 Objectif et modélisation

Le but de l'étape « Sequence » est d'obtenir de manière rapide de l'information sur la séquence de chaque peptide à partir du spectre MS/MS correspondant. Comme pour un « de novo **peptide** sequencing » standard, nous nous basons sur la liste de masses représentant les ions détectés lors de la fragmentation du peptide. Une fois les ions identifiés, il est possible de retrouver les acides aminés originaux du peptide. En effet, il suffit de retrouver deux types d'ions consécutifs et de faire la différence de masses. Cette différence équivaut à la masse d'un résidu d'acide aminé comme cela est montré ci-dessous sur les ions de type a :

$$ion\ a_{i+1} - ion\ a_i = (M_H + M_{raa} - M_{CO}) - (M_H + M_{raa} + M_{R_{i+1}} - M_{CO}) = M_{R_{i+1}}$$

avec M_X la masse de tous les atomes constituant la formule X, M_{raa}, la masse totale des résidus des acides aminés d'un ion donné et M_{R_i} la masse du i^{ieme} résidu d'un peptide donné.

Nous avons donc mis en place une heuristique gloutonne qui à partir d'une masse donnée essaye de déterminer le prochain acide aminé par analyse des masses voisines. Grâce à celle-ci, nous essayons d'obtenir un modèle type pour chaque peptide dans lequel un maximum d'information est rassemblé.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.4.3.3 Une heuristique gloutonne pour le séquençage de peptides

L'heuristique utilisée dans cette étape a pour but de trouver des informations sur la séquence de chaque peptide représentée par une donnée de type MS/MS différente. Le but est de diminuer l'espace de recherche à parcourir en essayant de trouver le maximum d'acides aminés de chaque peptide. Dans le cas où un peptide ne peut pas être séquencé entièrement, l'heuristique se propose de joindre au peptide un maximum d'information le concernant. Cela consiste principalement en un système de patron pour décrire la séquence du peptide qui servira de base à la prochaine étape. La séquence d'un peptide pourra donc être constituée de quatre motifs différents :

- 'X' : X correspond à un acide aminé considéré comme étant identifié dans la séquence et par conséquent bien placé. C'est une information sûre qui ne sera jamais remise en cause.

- '[X_1, X_2, …, X_n]' : ensemble de n acides aminés pouvant potentiellement se trouver à la position correspondante dans la séquence. C'est une information de moins bonne qualité que la précédente mais qui diminue tout de même l'espace de recherche d'autant plus si n est petit.

- '_' : motif indiquant l'emplacement d'un seul acide aminé sans pour autant pouvoir apporter de l'information sur son identitité.

- '*' : motif indiquant que le nombre d'acides aminés entre le précédent motif et le suivant n'est pas connu mais qu'il est supérieur ou égal à 1. C'est la moins bonne information que l'heuristique peut donner. En effet, n'importe quel peptide peut au pire être décrit comme étant simplement le mot « * », c'est-à-dire un nombre inconnu d'acides aminés non identifiés.

L'algorithme 7 présente les grandes lignes de l'heuristique. Tout est basé sur mise en place et la mise à jour des motifs décrivant les acides aminés durant le fonctionnement de l'heuristique grâce une information que possède chaque spectre MS/MS que l'on analyse : la masse du peptide parent. En effet, tout spectre MS/MS provient de la fragmentation d'un peptide qui avait une certaine masse. Chaque spectre MS/MS rappelle cette information. Par conséquent, grâce à cette masse, il est tout fait possible de calculer une borne minimale et maximale pour le nombre d'acides aminés formant le peptide considéré. La borne minimale s'obtient en considérant le peptide comme formé essentiellement d'acides aminés de masse de résidu importante portant le cas échéant une ou plusieurs modifications post-traductionnelles. La borne maximale est calculée en fonction du nombre maximum d'acides aminés de taille minimale que l'on peut mettre pour obtenir la masse du peptide parent. Lorsqu'aucune information concernant la séquence du peptide n'est connue, ces bornes sont exagérées. Mais à chaque modification de la séquence, ces bornes sont mises à jour et petit à petit elles deviennent de plus en plus précises. Voilà un exemple fictif du type de peptide que l'heuristique propose comme résultat de séquençage :

$$AF_G[PUY] * P * K$$

Cette séquence comporte cinq acides aminés bien identifiés, un acide aminé pour lequel trois possibilités subsistent, un emplacement unique où l'acide aminé n'est pas connu et enfin deux emplacements pouvant accueillir un nombre non connu d'acides aminés.

Nous avons donc mis défini et mis en place une heuristique qui permet de faire un « de novo **peptide** sequencing » pour essayer de faire diminuer l'espace de recherche de notre problème. De plus, le format de sortie proposé correspond à des modèles pour la complétion potentielle des séquences partielles. Cela va être détaillé dans la seconde étape de la méthode SSO : « Shape ».

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Algorithme 7 : une heuristique pour le « de novo **peptide** sequencing ».

Données : Des spectres MS/MS.
Résultat : Un ensemble de séquences formées de lettres et de motifs censés être des modèles pour la génération de peptides complets.

```
1  pour chaque spectre MS/MS faire
2      Calcul des bornes pour la séquence à partir de la masse du parent;
3      Génération du premier acide aminé;
4      si trouvé alors
5          valeur ajoutée dans la séquence;
6      sinon
7          si plusieurs possibilités alors
8              toutes sont enregistrées;
9          sinon
10             valeur '_' placée dans la séquence;
11         fin
12     fin
13     Mise à jour des bornes;
14     tant que un autre acide aminé est trouvé faire
15         si unique alors
16             valeur ajoutée;
17         sinon
18             toutes les valeurs sont enregistrées;
19         fin
20         Mise à jour des bornes;
21     fin
22     Génération du dernier acide aminé;
23     si trouvé alors
24         valeur ajoutée dans la séquence;
25     sinon
26         si plusieurs possibilités alors
27             toutes sont enregistrées;
28         sinon
29             si précédent = '_' alors
30                 valeur '*' à la place de '_';
31             sinon
32                 valeur '_' placée dans la séquence;
33             fin
34         fin
35     fin
36     Mise à jour des bornes;
37 fin
38 retourner les séquences descriptives de peptides;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

2.4.4 S pour « Shape »

Basé sur la même idée que dans [61], cette étape est censée améliorer le résultat d'un séquençage à partir de spectres MS/MS. La différence ici est que cette amélioration se base sur un autre type de données, le spectre MS. L'utilisation de ce type de données peut paraître étonnante car les spectres MS ne peuvent pas donner d'information sur la séquence de la protéine. Cependant, l'étape de séquençage permet d'obtenir pour chaque spectre MS/MS, une séquence d'acides aminés la plus complète possible. Ainsi, le nombre de peptides, perçu par le spectromètre, ainsi qu'une partie de la séquence de chacun d'entre eux sont déjà connus. A partir de ces peptides séquencés, on peut générer les spectres simulés associés et les comparer avec le spectre MS de la protéine expérimentale. Le but est de compléter les différentes séquences de peptides pour obtenir la distribution isotopique associée la plus proche de celle expérimentale. Bien sûr, même si on complète un peptide de manière à ce que sa distribution isotopique soit proche de l'expérimentale, il se peut très bien que la séquence ne soit tout de même pas la bonne. C'est pour cela, qu'a priori nous voulions obter pour un algorithme de la famille des métaheuristiques à base de population de solutions. En effet, cela permettrait de proposer un ensemble de protéines avec des peptides potentiellement proche de ceux expérimentaux. Pour un peptide donné, plus il y a de complétion proche de l'expérimental, plus la chance de trouver la bonne complétion est grande. Cela est d'autant plus vrai que le peptide est de petite taille.

Nous allons présenter la fonction de comparaison utilisée pour pouvoir compléter une séquence de peptide par comparaison avec un spectre MS. Nous pourrons ensuite l'analyser via une étude de paysage pour finalement arriver au choix de la métaheuristique à utiliser et son implémentation.

2.4.4.1 Modèle d'évaluation

La fonction d'évaluation utilisée pour évaluer une protéine reprend le principe de la comparaison spectrale présentée dans la section 2.3. Pour chaque peptide, sa formule chimique puis sa distribution isotopique sont calculées afin d'obtenir le spectre simulé correspondant. Ensuite, ce spectre est comparé au spectre expérimental et un score lui est attribué. Cependant, pour notre fonction d'évaluation, les scores des peptides ne sont pas classés par famille car de par leur construction, ils sont déjà censés être corrélés avec le spectre expérimental. C'est pour cela que la qualité d'une solution est représentée par le formule suivante , n étant le nombre de peptides au sein de la protéine :

$$\sum_{i=1}^{n} score_peptide_i$$

Le but de cette fonction d'évaluation est donc de maximiser la corrélation de l'ensemble des peptides. Nous allons examiner en quoi cette définition de l'évaluation a un impact direct sur le paysage associé.

2.4.4.2 Etude de paysage

Nous détaillerons d'abord le protocole de génération des paysages afin d'analyser les résultats.

Protocole de génération des paysages La fonction d'évaluation se base sur une méthode de comparaison spectrale qui se sert d'un spectre MS. Celui-ci décrit l'empreinte de masse peptidique d'une protéine donnée. Pour générer le paysage associé nous avons utilisé deux spectres MS : celui du cytochrome C bovin (CytC) et celui de l'apolopoprotéine AI (ApoAI). Afin de générer des protéines comparables à ces

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

spectres, nous nous sommes basés sur les masses respectives de CytC et d'ApoAI en prenant comme valeur moyenne 11500 Daltons pour la première et 30750 Daltons pour la seconde. A partir de ces masses, la génération se fait en 3 étapes :

1. Génération de séquences protéiques aléatoires par ajout d'acide aminé un à un jusqu'à atteindre la masse voulue.
2. Evaluation de la protéine obtenue.
3. Application d'une méthode par descente basée sur des substitutions d'acides aminés.

Cette génération de paysage assure que toute solution générée aura été intensifiée pour tendre si possible vers un optimum local.

Résultats et conclusions Pour chacune des protéines utilisées, le paysage est formé de cent protéines générées selon le protocole évoqué précédemment. Pour faciliter leur interprétation, nous avons donné les valeurs inverses des qualités des solutions pour avoir des paysages qui soient comparables à ceux évoqués dans le chapitre 1. Pour le CytC, nous obtenons le paysage représenté sur la figure 2.13 Pour l'ApoAI, le paysage obtenu est celui de la figure 2.14. Comme on peut le remarquer, les paysages obtenus sont très chaotiques avec la présence d'un grand nombre d'optima locaux. A la vue d'un tel paysage, il paraît évident que pour maximiser les chances d'obtenir des solutions dans les zones contenant les meilleurs optima locaux, l'utilisation d'algorithmes à base de population est conseillée. En effet, de par leur pouvoir exploratoire, ils vont être capables d'appréhender l'espace de recherche pour essayer d'atteindre les meilleures solutions.

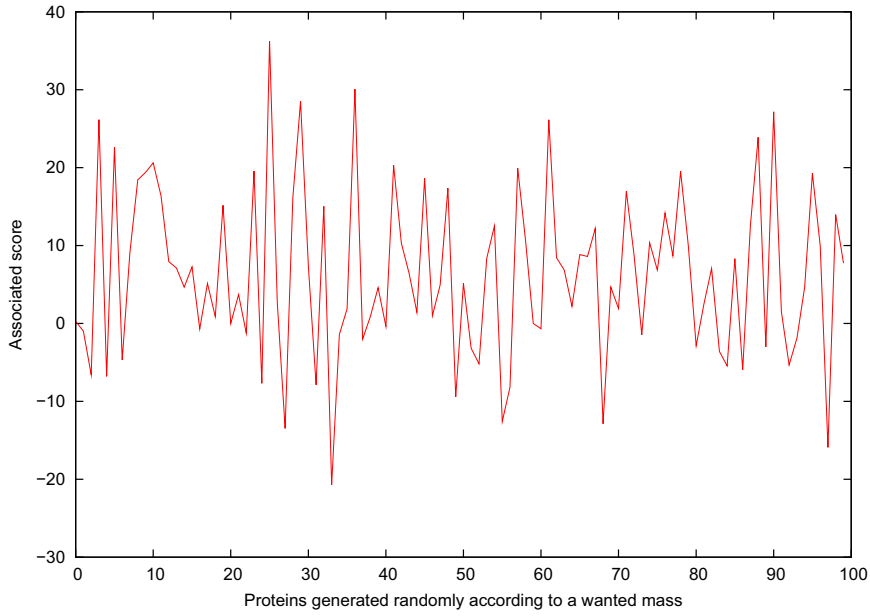


Fig. 2.13 – Paysage de la fonction d'évaluation de l'étape « shape » basé sur l'utilisation d'un spectre MS de cytochrome C bovin.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

La forme de ces paysages se justifie par le fait que la fonction d'évaluation de l'étape « shape » correspond à la somme des scores partiels des peptides. Chaque score partiel est le fruit d'une comparaison point à point entre le spectre simulé et le spectre expérimental. Par conséquent, la moindre petite modification a un impact important sur le score associé.

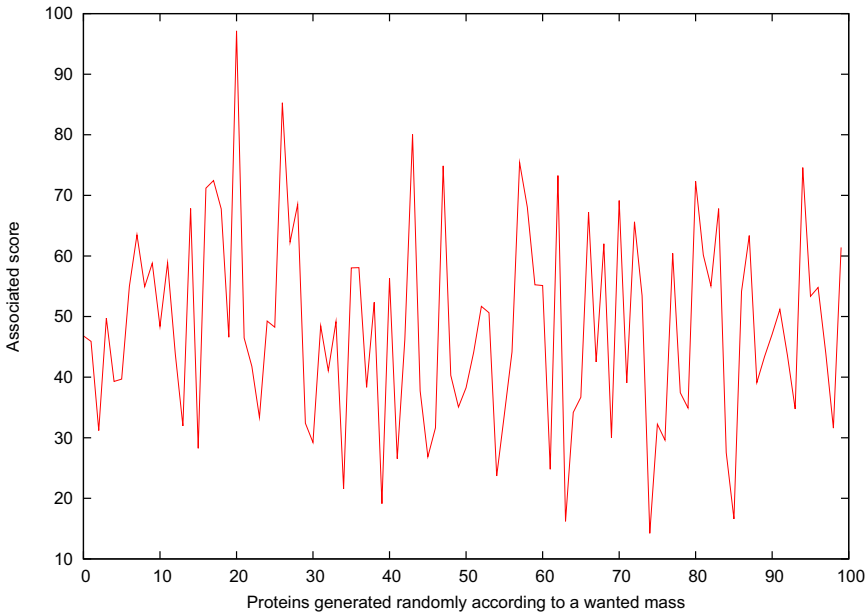


Fig. 2.14 – Paysage de la fonction d'évaluation de l'étape « shape » basé sur l'utilisation d'un spectre MS d'apolipoprotéine AI.

De par son pouvoir de diversification mais aussi d'appréhension des grands espaces de recherche, nous avons décidé d'utiliser un algorithme génétique pour l'étape « shape ».

2.4.4.3 Un algorithme génétique adaptatif parallèle pour la complétion de peptides

L'étape « shape » va permettre d'approfondir le travail de l'étape « Sequence » en complétant les séquences de peptides pour obtenir des protéines avec une empreinte de masse peptidique bien corrélée avec le spectre MS expérimental.

Représentation d'une solution Nous avons choisi de prendre une représentation naturelle pour notre individu : la liste de peptides. L'avantage de la liste de peptides est que la digestion n'est faite qu'une fois pour toutes et que ce format est assez flexible pour soit regénérer la séquence de la protéine complète, soit étudier la fragmentation d'un peptide en particuliers. Cependant, ce n'est pas une simple liste de petites séquences : pour chaque peptide est rappelé la masse expérimentale du peptide parent qui a généré le spectre MS/MS utilisé au séquençage ; le nombre minimum et maximum d'acides aminés pouvant composer ce peptide et le nombre de clivages manqués qu'il possède. Chaque acide aminé d'un peptide comporte en plus de sa lettre, un type « de novo » hérité de l'étape « Sequence » ainsi que le nombre de modifications post-traductionnelles qu'il porte. Toutes les informations portées par

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

un individu permettent d'assurer que son évolution restera conforme aux contraintes imposées durant le séquençage notamment lors des mutations.

Opérateurs Dans un algorithme génétique, deux types d'opérateurs existent : l'opérateur de recombinaison et celui de mutation. L'opérateur de recombinaison, aussi appelé opérateur de « crossover », est un opérateur binaire qui permet de créer deux nouveaux individus « enfants » en mélangeant l'information contenue dans deux individus « parents ». Un des opérateurs les plus connus et que nous utilisons dans notre algorithme est le « crossover un point ». Dans ce type de recombinaison, un point de coupure identique est choisi chez les deux parents et la génération des nouveaux individus se fait par mélange de l'information située de part et d'autre de ce point de coupure. Un schéma de « crossover-1-point » est visible en annexe B.9. Pour notre représentation, le point de coupure se situe entre deux peptides et permet d'obtenir des nouveaux individus valides.

L'opérateur de recombinaison réordonne l'information contenue dans une population donnée mais ne permet pas d'apporter de l'information extérieure à celle-ci. C'est le rôle de l'opérateur de mutation qui va apporter de l'information nouvelle en appliquant des modifications à un ou plusieurs individus de la population. Dans notre cas, quatre types de mutation ont été mis en place :

1. insertion d'un acide aminé : cette mutation ajoute de manière aléatoire un acide aminé sans pour autant dépasser la masse maximale et la taille maximale du peptide tous deux déterminés lors de l'étape « Sequence ».
2. délétion d'un acide aminé : cette mutation supprime de manière aléatoire un acide aminé sans passer en-dessous de la masse minimale et de la taille minimale du peptide tous deux déterminés lors de l'étape « Sequence ».
3. substitution d'un acide aminé par un autre : cette mutation échange un acide aminé par un autre suivant le type « de novo » auquel il est associé. Si celui-ci est de type '*' ou '_', la susbstitution se fait avec un autre acide aminé de manière aléatoire. Si son type est '$[X_1, X_2, ..., X_N]$', la substitution de l'acide aminé se fait avec un autre acide aminé choisi aléatoirement dans l'ensemble $[X_1, X_2, ..., X_N] \setminus X_i$, X_i étant le type actuel de l'acide aminé.
4. ajout d'une modification post-traductionnelle : ajout ou suppression d'une molécule sur un acide aminé ou sur le peptide en entier. Cette mutation n'a lieu que dans le cas de modifications post-traductionnelles variables car les modifications fixes sont toujours placées dès qu'un site est dipo-nible.

Remarque 2. *Historiquement, une version plus intelligente de la substitution d'un acide aminé avait aussi été mise en place. Il s'agissait d'une mutation guidée à l'aide de matrices de substitution. Une des matrices les plus utilisées est BLOSUM62. Une version reformatée pour notre utilisation est donnée en annexe B.10. Le principe était de se baser sur la mutation de type substitution d'acide aminé tel qu'elle peut se produire dans la nature. En effet, il est reconnu que certains types de substitution sont plus fréquents que d'autres. Les matrices de substitution renseignent donc sur la fréquence de chaque substitution possible. On peut ainsi simuler des mutations des acides aminés plus en accord avec ce qui est biologiquement plus probable. Cependant, dans notre principe de mutation actuel, la substitution est seulement dépendante de type « de novo » de l'acide aminé concerné. Par conséquent, l'utilisation de matrice de substitution n'est plus utile actuellement.*

L'application de ces opérateurs doit permettre de garder les informations de bonnes qualités entre les individus (recombinaison) tout en permettant la découverte de nouveaux individus potentiellement de meilleure qualité (mutation).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Adaptivité Dans le paragraphe précédent, nous avons présenté les opérateurs utilisés par notre AG. Concernant leur propabilité d'application, les probabilités de recombinaison et de mutation sont les valeurs généralement choisies dans la littérature. Mais au vu du nombre d'opérateurs de mutation, il est difficile de savoir a priori lesquels vont avoir le plus d'impact et surtout à quel moment. C'est pour répondre à cette question, que nous avons mis en place un mécanisme adaptatif pour faire varier la probabilité d'application associé à chaque type de mutation suivant le gain que celle-ci apporte. De nombreux auteurs ont déjà travaillé sur des systèmes de mises à jour automatiques de probabilité de mutation [33, 62, 71]. Dans [65], les auteurs ont proposé de calculer le nouveau taux de mutation en sa basant sur le progrès de la j^{ième} application de la mutation M_i et analysant l'impact correspondant sur la qualité de l'individu qui a été muté. Le progrès est donc défini de la manière suivante dans le cas d'une maximisation :

$$progress_j(M_i) = Max(Eval(individu_original), Eval(individu_mute)) - Eval(individu_original)$$

Si le progrès est positif, le taux d'application de la mutation correspondante sera augmenté, s'il est négatif, il restera fixe et si celui-ci est nul, le taux est augmenté mais moitié moins que dans le cas d'un progrès positif.

Initialisation de la population L'initialisation de la population sa base sur le résultat de l'étape « Sequence » dans le respect de la coopération relais de haut niveau. Les séquences de peptide obtenues lors de cette étape servent de modèle pour générer l'ensemble de la population initiale. Chaque individu de la population est créé en parcourant peptide par peptide les séquences modèles pour obtenir des séquences complètes. Par exemple, si un peptide à une séquence la forme :

$$AF_G[PUY] * P * K$$

Quelques séquences potentielles pour ce peptide peuvent être :

AF*M***G**[*U*]*TQ***P***L***K**

AF*L***G**[*P*]*TR***P***LQ***K**

AF*C***G**[*Y*]*NOQ***P***MP***K**

AF*I***G**[*U*]*KK***P***LNU***K**

AF*R***G**[*P*]*GHIL***P***QE***K**

Sachant que le nombre et le type d'acides aminés pouvant prendre place sur les types « de novo » '*' dépendent de la masse du peptide parent et des bornes minimale et maximale pour la taille du peptide.

Sélection, remplacement et critère d'arrêt Pour faire évoluer notre population, il est nécessaire, à la manière du modèle biologique, de sélectionner des « parents » qui vont permettre d'engendrer des « enfants ». Cette étape de sélection est importante car elle permet d'axer la manière vers laquelle nous voulons que notre population évolue. Nous nous sommes basés sur la sélection élitiste qui donne la priorité aux individus de meilleure qualité. Ainsi plus un individu aura une qualité importante, plus sa chance d'être sélectionné comme parent sera importante. Cette sélection est connue sous le nom de sélection roulette dont un schéma est visible en annexe B.11. Une fois des individus sélectionnés et après recombinaison et mutation, il est temps de procéder au remplacement de la population. Il s'agit à partir de l'ancienne population et des nouveaux individus de générer la nouvelle population. Historiquement, les « enfants » remplaçaient toujours les « parents » dans un AG générationnel. Depuis, un grand nombre de stratégies de remplacement ont été mises en place et peuvent être combinées ensemble. Par exemple,

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

dans [87], à chaque génération, 10 % de la population est remplacée selon différentes stratégies (du choix aléatoire au choix des plus mauvais individus). Dans notre cas, nous avons décidé d'appliquer une stratégie de type « steady-state », c'est-à-dire que les meilleurs individus parmi les « parents » et les « enfants » remplacent les pires individus de la population. Nous sommes toujours dans l'esprit d'une stratégie élitiste. Il n'est pas possible a priori de savoir combien de générations l'AG va effectuer pour obtenir les meilleurs individus. Il est donc nécessaire d'avoir un critère d'arrêt flexible. C'est pour cela que nous avons choisi le nombre de générations sans amélioration. Il s'agit simplement de sauvegarder le nombre de générations entre deux améliorations de la meilleure valeur de la fonction d'évaluation. Dès que ce nombre de générations dépasse un certain seuil, l'algorithme s'arrête. Pour éviter une convergence prématurée, ce critère d'arrêt n'est activé qu'après un minimum de générations.

Parallélisation Comme tout algorithme à base de population, l'évaluation des individus d'un AG reste une étape coûteuse. La parallélisation est un bon moyen de faire diminuer ce coût. En effet, les AGs sont aisément parallélisables à plusieurs niveaux. Dans notre cas, nous avons opté pour le choix du paradigme maître/esclave pour l'évaluation des individus. Un processus gère le fonctionnement de l'AG et délègue l'évaluation des individus à d'autres processus. On obtient ainsi une évaluation en parallèle des individus. Le choix du nombre de processus « esclave » à utiliser dépend de deux paramètres : le temps d'évaluation d'un individu et le temps de communication entre processus. Afin d'être évalué par un autre processus un individu doit être « codé » par le maître afin d'être envoyé à un processus esclave. Celui-ci devra « décoder » l'individu, l'évaluer, le réencoder afin de l'envoyer au maître. Ce dernier devra donc le « décoder » avant de pouvoir l'utiliser. A cause de cet enchaînement de petites étapes, choisir un nombre de processus esclaves égal au nombre d'individus de la population n'est pas forcément le choix le plus efficace. L'important est de choisir si possible un nombre suffisant de processus esclave pour que le premier processus évaluant un individu n'est pas terminé son évaluation avant que le maître ait envoyé le dernier individu à évaluer. Le but étant de plutôt privilégier une attente du maître plutôt qu'une attente d'un esclave.

2.4.5 O pour « Order »

Comment est-il possible d'ordonnancer des peptides avec seulement de l'information en provenance d'un spetre MS ? C'est la question à laquelle se propose de répondre cette étape. Si une séquence de protéine est connue ; grâce à ADP et SpecS, il est facile de générer les spectres MS simulés associés à l'action de différentes enzymes sur cette protéine. Chaque spectre MS sera donc l'empreinte de masse peptidique de la protéine pour une enzyme donnée. Ainsi, si plusieurs enzymes sont utilisées, elles amèneront chacune à une empreinte de masses peptidiques différente. Par conséquent, si on a une protéine dont les spectres simulés pour différentes enzymes sont bien corrélés avec ceux expérimentaux pour les mêmes enzymes ; cela indique que ce sont les mêmes peptides qui sont générés pour chaque digestion et donc que la séquence de la protéine est correcte. Les protéines issues de l'étape 2 comportent des peptides qui permettent d'avoir une empreinte de masse peptidique proche du spectre MS utilisé. A partir du moment où les peptides ont été générés, leur ordre n'a pas d'importance pour générer l'empreinte de masse peptidique. Cependant, si l'on veut comparer ses peptides face à un spectre MS résultat d'une autre digestion, il est important de revenir à la séquence de la protéine, de la digérer en fonction de l'enzyme voulue et de comparer l'empreinte de masse peptidique obtenue avec l'expériementale. Si la séquence de la protéine n'est pas bonne, les peptides engendrés lors de cette nouvelle digestion n'amèneront pas à une empreinte de bonne qualité comparée à l'expérimental. Au contraire, si les peptides sont bien ordonné, la comparaison avec un autre spectre MS amènera à une bonne corrélation. Cela est d'autant plus fiable

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

que le nomre de digestion différentes augmente. Cela permet d'évaluer la qualité de la séquence.

Le principe de cette dernière étape est donc simple : pour chaque solution de l'étape « Shape », trouver la combinaison de peptides qui permet de maximiser l'évaluation de la protéine correspondante face au nouveau spectre MS utilisé.

Comme pour l'étape précédente, c'est une étude de paysage qui va orienter notre choix de métaheuristique.

2.4.5.1 Etude de paysage

Pour l'étude de paysage, nous allons utiliser le même protocole que pour l'étape « Shape ».

La particularité de la fonction d'évaluation de l'étape « Order » est d'être extrêmement discriminante. Pour pouvoir obtenir un score différent de 0, il faut obtenir au moins un peptide dont le score partiel est lui-même supérieur à un seuil donné. Ce seuil, qui a pour but la différenciation entre les bons et les mauvais peptides, est par défaut fixé à 0. Dans ce cas, un bon peptide est un peptide dont le score est positif. De par sa définition, le nombre de protéines pouvant espérer obtenir un score supérieur à 0 est limité. Cela s'est vérifié lors de la génération des protéines aléatoires. En effet, afin d'obtenir cent protéines ayant un score différent de 0 afin de les optimiser, il a fallu générer 230 931 protéines pour le CytC et 2140 pour l'ApoAI.

Les figures 2.15 et 2.16 montre combien les zones contenant des protéines intéressantes est faible. Cette remarque est d'autant plus vraie que la protéine est petite. Cette forme de paysage montre combien cette fonction d'évaluation est bien adaptée pour l'étape « Order ». Comme dans cette étape, l'infor-

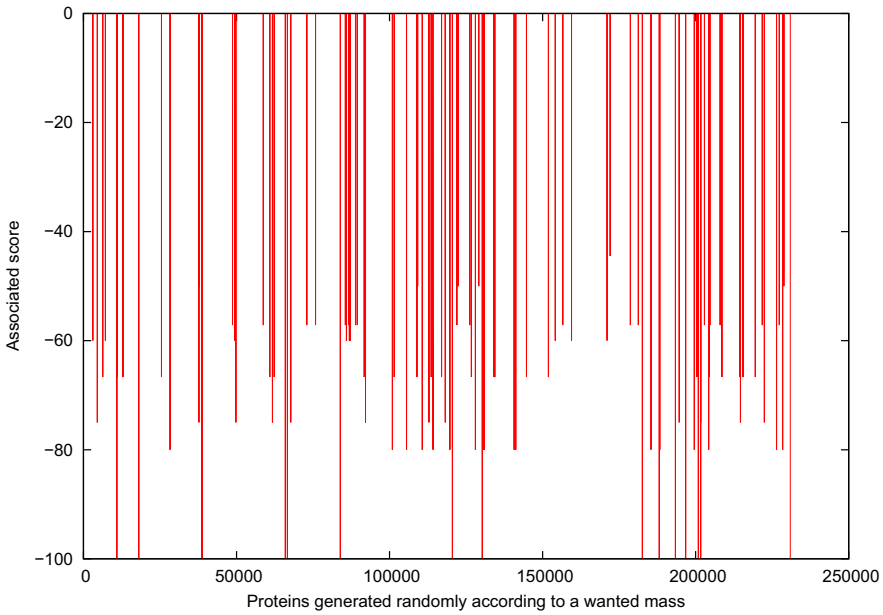


Fig. 2.15 – Paysage de la fonction d'évaluation de l'étape « order » basée sur l'utilisation d'un spectre MS de cytochrome C bovin. 230 931 protéines aléatoires ont été utiles pour générer cent protéines ayant un score non nul.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

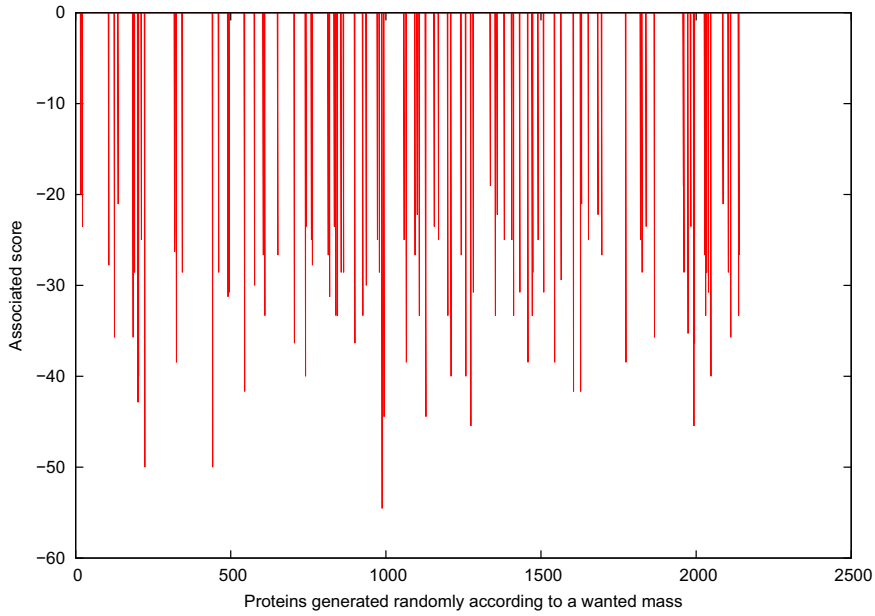


FIG. 2.16 – Paysage de la fonction d'évaluation de l'étape « order » basée sur l'utilisation d'un spectre MS d'apolipoprotéine-AI. 2140 protéines aléatoires ont été utiles pour générer cent protéines ayant un score non nul.

mation de la séquence de la protéine expérimentale est déjà présente à un réordonnancement près des peptides, les protéines que nous avons générées dans les étapes précédentes ont déjà leur empreinte de masse peptidique a priori peu éloignée de celle à trouver. Ainsi, grâce aux étapes précédentes, nous nous

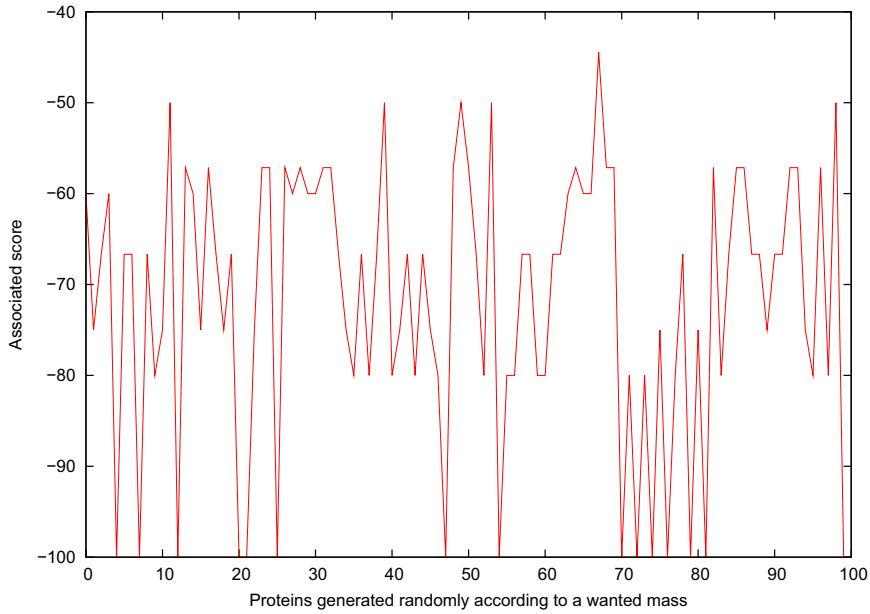


FIG. 2.17 – Paysage de la fonction d'évaluation de l'étape « order » basée sur l'utilisation d'un spectre MS de cytochrome C bovin.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

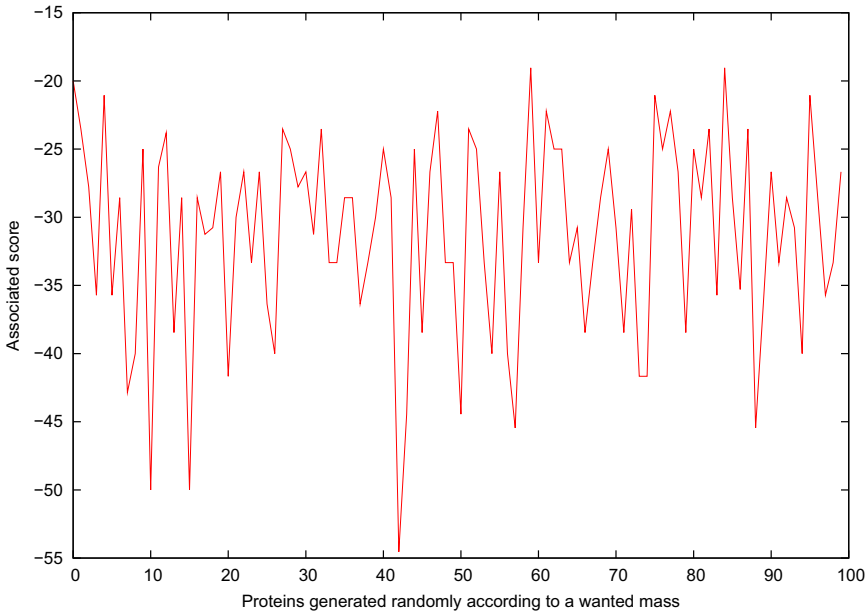


FIG. 2.18 – Paysage de la fonction d'évaluation de l'étape « order » basée sur l'utilisation d'un spectre MS d'apolipoprotéine-AI.

ne travaillons pas sur les paysages tels qu'ils sont montrés dans les figures 2.15 et 2.16 mais plutôt dans les paysages des figures 2.17 et 2.18. Ces figures représentent les paysages formés par seulement les protéines ayant un score différent de 0. Ces deux paysages sont assez chaotiques même si on peut remarquer qu'ils sont légèrement différents. En effet, la figure 2.17 a un paysage moins torturé que le celui de la figure 2.18. Cela est au nombre de peptides qui décrivent une protéine. Plus celui-ci est petit, plus il est facile d'améliorer la solution courante par quelques échanges de peptides bien choisis. Au contraire, lorsqu'il y a un grand nombre de peptides, le nombre de mouvement nécessaire pour espérer trouver des améliorations est beaucoup plus élevé. Ainsi, lorsque l'on examine paysage du CytC (figure 2.17), il paraît facile d'arriver sur des optima globaux (score à -100 sur la figure) à partir de divers endroits du paysage à partir du moment où il est possible de sortir des quelques optima locaux que l'on rencontre. Un paysage de cette forme a tendance à inciter à utiliser des métaheuristiques à base de solution unique qui ont un pouvoir d'extraction des optima locaux. On pensera notamment au recuit simulé, à la recherche locale itérative et à la recherche à voisinage variable.

Grâce à cette étape nous avons un ensemble de protéines potentiellement bien séquencées à un réordonnancement près. Au vu de l'étude de paysage effectuée, une forte intensification permettrait d'arriver au bon séquençage. C'est pour cela que nous allons utiliser une métaheuristique à base de solutions uniques.

2.4.5.2 Une recherche locale pour l'ordonnancement de peptides

Vouloir utiliser une métaheuristique à base de solution unique est une chose, savoir laquelle choisir en est une autre. Nous allons donc, après avoir détaillé le voisinage utilisé, comparer différentes types de recherches locales.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Modélisation Dans cette étape, nous avons besoin de faire du réordonnancement de peptide. Comme notre représentation de solution est la liste de peptides, il semble logique de prendre comme base de génération de nouvelle solution l'échange de peptide. Grâce à la plateforme ParadisEO-MO, qui a été présenté dans la section 1.5, il est aisé d'implémenter et de tester plusieurs modèles de recherche locale pour cette étape. Nous avons décidé de mettre en place la notion de mouvement qui est à la base de tout métaheuristique sous ParadisEO-MO, ce sera le « peptide swap ».

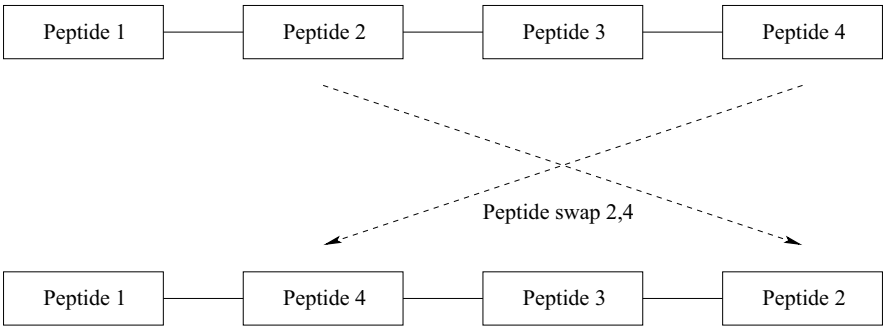


Fig. 2.19 – Exemple d'application du mouvement « peptide swap » sur une protéine formée de 4 peptides.

Comme son nom l'indique, il s'agit simplement d'une inversion de deux peptides donnés (voir figure 2.19). Ainsi, il est possible à partir de ce mouvement de générer toutes les combinaisons possibles de peptides. Cependant, à partir d'une solution donnée, suivant le nombre de peptides la composant, le nombre de voisins potentiels peut être élevé. Par conséquent, pour la génération de voisins, j'ai mis en place sous ParadisEO-MO un objet qui permet de faire des recherches dans des voisinages partiels. Cet objet est encore en phase de test ce qui explique qu'il ne soit pas encore dans la version actuelle de ParadisEO-MO. Son nom, pour le moment, est *moPartialSearchFirstImprSelect*. Il s'agit d'un critère de sélection dérivé de l'objet *moFirstImprSelect*. En effet, il s'agit de sélectionner le premier voisin rencontré l'améliorant lors de la génération du voisinage, sachant qu'un nombre maximum de voisins est généré. Si ce nombre est atteint sans avoir trouvé un voisin de meilleure qualité que la solution courante, un voisin aléatoire est retourné. Afin de ne pas avantager certains voisins en particulier, la génération des voisins est un processus stochastique. Ainsi pour une protéine donnée, les voisinages générés sont nécessairement différents mais de taille maximum fixe.

Chaque solution résultant de l'application de l'AG de l'étape précédente va subir une recherche locale. A priori, il n'est pas possible de savoir combien de temps chaque recherche locale va durer et donc son coût. Afin de minimiser les coûts, il est tout à fait possible de paralléliser l'évaluation des solutions. Grâce à ParadisEO-PEO, nous pouvons lancer un ensemble de recherches locales en parallèle. Ainsi, le coût de l'étape « order » dépendra seulement du nombre de processus disponibles. De plus, si le nombre de processus est supérieur ou égal au nombre de solutions à intensifier, dans ce cas, le coût total ne dépendra que de la solution la plus longue à intensifier.

Comparaison de métaheuristiques Afin de tester ces différentes recherches locales, nous allons générer une liste de peptide correspondant à une digestion selon une enzyme particulière et nous allons ensuite simuler un spectre MS correspondant à la même protéine mais selon une digestion par une autre enzyme. Le but est de tester la capacité des recherches locales à réussir à réordonner les peptides pour maximiser la corrélation au niveau de l'empreinte de masse peptidique. Au final, les recherches locales seront comparées selon deux critères : le pourcentage de recouvrement de la séquence de la protéine à

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

trouver donné grâce à blastp [18] et la valeur moyenne de la fonction d'évaluation obtenue lors des recherches. Nous effectuerons dix recherches sur une liste de peptides correspondant au Cytochrome-C bovin (CytC) digéré par l'enzyme GluC. Le spectre MS de comparaison correspondra quant à lui à une digestion de type trypsine.

Le tableau 2.9 détaille les résultats concernant le critère d'alignement pour le CytC.

Tab. 2.9 – Résultats d'alignement après dix essais de chaque recherche locale sur l'exemple du cytochrome C bovin. MD signifie méthode par descente, RT Recherche Tabou, RS recuit simulé et RLI recherche locale itérative. Min et max correspondent respectivement aux valeurs minimum et maximum d'alignement obtenues parmi les dix tests.

Recherche Locale	Pourcentage d'alignement moyen	Ecart-type	Min	Max
MD	84,4	10,91	65	98
RT	87,4	10.69	63	98
RS	**92,4**	6.02	79	**100**
RLI	92,1	**5,7**	**83**	**100**

Comme l'étude de paysage le laissait supposer, les résultats d'alignement concernant le CytC sont bons puisque toutes les métaheuristiques ont été capables au moins une fois d'accéder à une valeur de 98% de recouvrement de la séquence. On peut cependant remarquer que deux groupes sont visibles : celui formé par la recherche par descente et la recherche taboue et celui formé par le recuit simulé et la recherche locale itérative. C'est un résultat qui paraissait prévisible au vu du paysage du CytC. En effet, le recuit simulé et la recherche locale itérative sont capables de s'extraire des optima locaux et étaient donc mieux préparés à explorer l'espace de recherche. Le tableau 2.10 compare les métaheuristiques en fonction du score qu'elles ont obtenues.

Tab. 2.10 – Qualité de l'évaluation après 10 essais de chaque recherche locale sur l'exemple du cytochrome C bovin. MD signifie méthode par descente, RT Recherche Tabou, RS recuit simulé et RLI recherche locale itérative. Min et max correspondent respectivement aux valeurs minimum et maximum obtenues parmi les dix tests pour la fonction d'évaluation.

Recherche Locale	« Fitness » moyen	Ecart-type	Min	Max
MD	90,33	3,1	85,71	90
RT	80,95	4,1	85,71	95,24
RS	94,74	3.51	**90,48**	**100**
RLI	**97,09**	**3,36**	**90,48**	**100**

Au vu de ce second tableau, les écarts entre les métaheuristiques sont plus importants. Ainsi la recherche locale itérative se révèle être la méthode avec le plus grand pouvoir d'intensification suivi par le recuit simulé, la méthode par descente et enfin la Recherche Tabou.

Finalement, au vu des pourcentages de recouvrement et à la valeur de la fonction d'évaluation, il semblerait que la recherche locale itérative donne de meilleurs résultats. Cependant, au vu des résultats donnés par le recuit simulé, il pourrait être envisagé de combiner celui-ci avec la recherche locale itérative étant donné que celle-ci nécessite une recherche locale en son sein et que ParadisEO-MO facilité la réutilisation des divers *moAlgo*.

[18] Version de blast dédiée à la comparaison de séquences protéiques.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Pour l'étape « Order », nous utiliserons une recherche locale itérative représentée par un objet *moILS* dans ParadisEO-MO.

2.4.6 Validation de SSO

Après avoir détaillé chaque métaheuristique de l'approche SSO, nous allons maintenant valider leur coopération sur des données expérimentales fournies par le LCOM. Après avoir décrit les données utilisées, nous présenterons les premiers résultats obtenus.

2.4.6.1 Données

L'approche SSO combine des données spectrales de type MS et MS/MS. Il faut au minimum un spectre MS et les spectres MS/MS correpondants venant d'une digestion donnée et un spectre MS venant d'une digestion différente. Le LCOM nous a proposé des données concernant trois protéines à partir desquelles des spectres MS et MS/MS ont été générés. Les deux types de digestion utilisés pour ces données furent la pepsine et la trypsine. Les protéines que nous avons récupérées sont considérées être de l'albumine de bovin dont l'identification par « de novo **peptide** sequencing » aura montré qu'il s'agirait de la protéine d'identifiant P02769, viennent ensuite le cytochrome C bovin d'identifiant P62894 et l'obalbmine de poulet possédant l'identifiant P01012.

Le tableau 2.11 présente pour chaque protéine son nom, son identifiant SwissProt, la taille de sa séquence, sa masse et le pourcentage de recouvrement de séquence obtenu lors de l'identification par « de novo **peptide sequencing** » par les méthodes traditionnelles. Cette dernière valeur correspond à la taille séquence que l'information des spectres MS/MS permet d'obtenir une fois comparée aux bases de données. Cette valeur vient de l'identification par MASCOT des protéines à partir des mêmes spectres que nous utiliserons dans l'étape « Sequence ». L'annexe B.12 présente pour chaque protéine, sa séquence FASTA ainsi qu'une capture d'écran du début de la page d'identification générée par MASCOT qui rappelle le taux de recouvrement.

Tab. 2.11 – Nom, identifiant SwissProt, taille de séquence et masse des données utilisées lors de la validation de l'approche SSO. La dernière colonne intitulée PRM correspond au pourcentage de recouvrement trouvé par MASCOT à partir des données MS/MS.

Nom	Id	Taille de la séquence	masse (Da)	PRM
Cytochrome-C bovin (CYC_BOVIN)	P62894	105	11 704	72
Sérum d'albumine bovine (ALBU_BOVIN)	P02769	607	69 293	15
Ovalbumine de poulet (OVAL_CHICK)	P01012	386	42 881	9

2.4.6.2 Résultats

Evaluation de l'efficacité de la parallélisation Nous avons mener une étude de « speed-up » et d'efficacité sur notre parallélisation. Pour rappel, elle consistait à distribuer les individus de la population sur un ensemble de machines pour qu'ils soient évalués. Ainsi, à chaque génération, chaque machine parallèle évaluera un ou plusieurs individus suivant la taille de la population et le nombre total de machines utilisées. Nous avons d'abord testé l'algorithme génétique seul sur deux instances données : l'Apo-AI et le CytC. Le « speed-up » et l'efficacité sont calculés de la manière suivante :

$$S_N = \frac{Ts}{Tp} \qquad\qquad Eff = \frac{S_n}{N}$$

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Avec Ts le temps séquentiel d'exécution et Tp le temps d'exécution en parallèle avec N esclaves. Le tableau 2.12 montre les résultats de cette étude sur l'algorithme génétique. Ils proviennent de tests sur le site de Lille de Grid5000 [19], ceux du tableau 2.13 proviennent d'exécutions sur un cluster de soixante-quatre processeurs qui appartiennent à l'équipe projet DOLPHIN. Grid5000 est une infrastructure distribuée sur 9 sites sous forme d'une grille hétérogène dédiée à la recherche pour des calculs hautes performances. Elle a pour but de tester les techniques de passage à l'échelle pour des problèmes réels dont la résolution est un challenge. Cela reste une grille de recherche avant tout.

Nb	**Apo-AI**			**Cytc**		
Machines	**Temps**	S_N	**Eff**	**Temps**	S_N	**Eff**
1	5530	1	1	14712	1	1
4	3430	1,61	0,4	**3164**	**4.65**	**1.16**
8	1641	3,37	0,42	2055	7,16	0,9
16	1215	4,55	0,28	1443	10,2	0,64
32	759	7,29	0,22	1307	11,26	0,35
40	947	5,83	0,14	1020	14,42	0,36

Tab. 2.12 – Temps d'exécution (en secondes), « speed-up » (S_N) et efficacité (Eff) sur Grid5000 pour deux instances selon le nombre d'esclaves utilisés (Nb machines).

Nous avons ensuite procédé au test de parallélisation de l'approche en entier. Pour ce faire, nous avons utilisé CytC dans une configuration permettant une exécution assez rapide en séquentielle (< 25min). En utilisant les mêmes critères que précédemment, la tableau 2.13 présente les résultats de cette étude.

Nb	**Cytc**		
Machines	**Temps**	S_N	**Eff**
1	1275	1	1
2	503	**2,53**	**1,27**
4	168	**7,6**	**1,9**
8	153	**8,3**	**1,038**
16	119	10.7	0,67
32	86	14.8	0,46
48	70	18.2	0,57
54	75	17	0,31

Tab. 2.13 – Temps d'exécution (en secondes), « speed-up » (S_N) et efficacité (Eff) sur cluster pour CytC selon le nombre d'esclaves (Nb machines).

Au vu des résultats de ces tableaux, on peut remarquer que jusqu'à seize processeurs, l'efficacité reste supérieure à 0,5 ce qui est intéressant. Par contre, dès que l'on augmente encore le nombre de processeurs, l'efficacité passe en-dessous des 0.5 et le gain n'est plus significatif. Cela est lié à la complexité de nos fonctions d'évaluation d'une part et au coût des communications d'autre part. Nous avons en gras des valeurs pour lesquelles le « speed-up » est supra-linéaire. Il n'y a a priori pas d'explication aisée à ce résultat, cela peut-être dû à l'obtention de solutions de bonnes qualités concernant l'ordre des peptides dès la sortie de l'étape « Shape ».

[19] Plus d'information sur https ://www.grid5000.fr/

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Résultat de séquençage Comme le présentait le tableau 2.11, la seule information des spectres MS/MS a permis à MASCOT de trouver des correspondances avec des peptides des bases de données permettant l'identification des protéines associées. De cette connaissance des peptides en est déduit un pourcentage de recouvrement de la séquence de la protéine. C'est sur cette base que nous allons comparer nos résultats. Pour évaluer la qualité de nos résultats, nous allons procéder à une comparaison de toutes les séquences générées par l'approche SSO pour chacune des trois instances (CytC, BSA, Ovalbumin) face à la séquence FASTA objectif (P62894, P02769 et P01012). Ainsi, nous pourrons savoir quelles portions des protéines correspondent ou pas et obtenir une estimation de la portion de la séquence qui a été trouvée.

Le tableau 2.14 rappelle pour chaque instance le pourcentage de recouvrement de la séquence indiqué par MASCOT et indique le nombre d'acides aminés auquel ce pourcentage correspond.

Tab. 2.14 – Instance, pourcentage de séquence identifiée par MASCOT (PRM) et équivalence en nombre d'acides aminés.

Instance	**PRM**	**Nombre d'acides aminés reconnus**
CytC	72	75
BSA	15	91
Ovalbumin	9	34

On peut remarquer que sauf pour le CytC, le recouvrement donné par MASCOT correspond à des portions de séquences séparées non contiguës qui correspond à des peptides. Le tableau 2.15 présente pour chaque instance la taille de séquence maximum générée corrélée à la protéine cible.

Tab. 2.15 – Instance, pourcentage de séquence corrélée avec l'expérimentale et équivalence en nombre d'acides aminés.

Instance	**PRM**	**Nombre d'acides aminés reconnus**
CytC	71,4	68
BSA	12	73
Ovalbumin	**28**	**108**

Il est important de noter que la valeur donnée pour chaque instance est le maximum de séquence corrélées trouvés sur un individu de la population de solution sans chevauchement. En effet, il se peut qu'une portion de la séquence d'individu soit corrélée avec plusieurs endroits de la protéine expérimentale. Dans ce cas, cette portion n'est comptée qu'une seule fois. Ces séquences corrélées correspondent à des ensembles de peptides **contigus**. Cette première constatation permet d'affirmer que le principe de **protein** sequencing est atteint puisque que nous arrivons à reconstruire des portions de la séquence de la protéine expérimentale qui ne sont pas des peptides isolés. De plus, les résultats obtenus sur l'ovalbumine montre que l'information cumulée des spectres MS/MS et MS a permis de générer des séquences qui ne sont pas identifiée avec seulement les données de type MS/MS.

Le tableau 2.16 donne les intervalles de séquences reconnues par les deux méthodes.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 2.16 – Comparaison des portions de séquences reconnues par MASCOT et l'approche SSO. Pour chaque instance, les intervalles de séquences correspondants sont indiqués. Pour SSO, le nombre de peptides contenus dans l'intervalle est donné.

Instance	**Intervalles MASCOT**	**Intervalles SSO**	**Nombre de peptides sans clivage manqué dans l'intervalle**
CytC	[25,100]	[34,98]	12
BSA	[179,188] ; [243, 257]	[11, 33]	5
	[498, 410] ; [431, 446]	[355, 377]	5
	[479, 483] ; [512, 525]	[520, 528]	3
	[553, 566] ; [692, 698]	[582, 599]	3
Ovalbumin	[44, 59] ; [100, 114]	[146, 204]	6
	[358, 365]	[340, 368]	2

A l'aide de ce tableau, on se rend compte combien l'information du nombre des peptides contenus dans un intervalle pour l'approche SSO est importante car elle exprime le fait que les étapes précédentes ainsi « Order » ont fonctionné.

Ces résultats sont encourageants mais nécessitent encore du travail et de la validation sur d'autres ensembles de données expérimentales.

2.5 Conclusions et perspectives

Au sein de ce chapitre, nous avons montré comment la coopération de métaheuristiques peut permettre d'obtenir des modèles performants pour répondre à des problèmes pour lesquels aucune réponse n'était actuellement donnée. Nous avons aussi insisté sur l'importance de la modélisation du problème, des fonctions d'évaluation associées et sur l'étude du paysage. Connaître le paysage d'un problème permet de savoir a priori quelles sont les métaheuristiques qui seront le plus à même de le parcourir de la manière la plus efficace.

Concernant le moteur d'identification par PMF Ascq_me, il a permis de valider une fonction d'évaluation innovante. Même si à l'heure actuelle, il nécessite encore d'être amélioré pour être utilisable par la communauté, Ascq_me est déjà un outil qui a beaucoup muri et qui s'adapte déjà aux besoins des utilisateurs. Ainsi la version actuelle d'Ascq_me propose la recherche en mode decoy pour obtenir un indicateur de confiance pour les résultats de l'identification. Beaucoup d'autres travaux sont en cours autour d'Ascq_me, une version pour faire du PFF a déjà été mise en place et testée pour faire de l'identification avec des spectres MS/MS mono-chargés ; une version avec un z-score comme indice de confiance est en cours de test également.

Le modèle SSO reste quant à lui très prometteur, tout le travail effectué jusqu'à maintenant ayant permis de mettre en place une première monture. Maintenant, il reste encore des comparaisons de métaheuristiques à faire comme par exemple l'essai de recherches locales à voisinages multiples pour l'étape « Order ». Il est aussi important de rajouter une étape d'extraction d'une protéine consensus à partir de l'ensemble des solutions potentielles afin d'avoir de meilleurs résultats. Il est aussi question de procéder, comme pour Ascq_me, à la mise en place d'une plateforme de tests pour permettre à des acteurs du domaine de critiquer et de valider notre approche. Un avantage de notre modèle coopératif est sa modularité. Il est tout à fait possible de remplacer l'étape « Sequence » ou/et l'étape « Shape » par des

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

approches ou logiciels existants pour procéder à de nouveaux essais. En effet, l'étape « Order » est la partie manquante à beaucoup de méthodes pour arriver à faire du « de novo **protein** sequencing ». Enfin, concernant l'avenir du « de novo **protein** sequencing », le LCOM mais aussi d'autres laboratoires sont déjà intéressés par le séquençage de protéines de plantes dès que le modèle sera bien stabilisé.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

3

Nouvelles modélisations multi-objectives pour le docking moléculaire flexible

Dans ce chapitre, nous allons présenter une étude sur différentes nouvelles modélisations multiobjectives possibles pour un problème du domaine de l'analyse structurelle des molécules et de leurs intéractions : le docking moléculaire flexible. Notre but est d'essayer de trouver la modélisation qui soit la plus fidèle au processus biologique. Nous commençerons par décrire le problÃ¨me en détaillant les interactions énergétiques existantes au sein des molécules. De là nous présenterons les différents critères que nous utiliserons dans nos modélisations. Celles-ci seront prÃ©sentÃ©es après un état de l'art des modèles existants au sein de la littérature. Les nouvelles modélisations proposées seront testées et comparées à travers des algorithmes génétiques hybrides. Nous montrerons notamment l'impact du choix du champ énergétique sur les résultats obtenus. La présentation de la plateforme Docking@GRID dans laquelle notre travail est inclus cloturera ce chapitre.

*Le travail effectué sur le docking moléculaire flexible a fait l'objet d'une présentation lors de la conférence internationale avec comité de lecture IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) sous le titre « Parallel multi-objective algorithms for the molecular docking problem » en 2008 et a reçu le « **IEEE Best Student Paper award** [10] ».*

Ce travail a été réalisé en collaboration avec le laboratoire de glycobiologie structurale et fonctionnelle de Lille 1. De plus, il s'incrit au sein du projet ANR Dock et est soutenu par le PPF Bioinformatique de l'Université des Sciences et Technolgies de Lille (USTL).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.1 Définitions

Dans un premier temps, nous présentons le docking moléculaire puis nous détaillons les objectifs du projet ANR Dock au sein duquel notre travail s'inscrit. Par la suite, les connaissances nécessaires à la bonne compréhension des modélisations présentées dans les sections suivantes seront proposées.

3.1.1 Docking moléculaire

Le problème du docking moléculaire consiste à prédire le complexe pouvant être formé par deux molécules données. Généralement, il s'agit d'une plus petite molécule, le ligand, qui se lie à une plus grosse, le récepteur. La zone où le ligand se fixe au récepteur se nomme le site (actif). Savoir prédire une telle interaction permet d'améliorer le processus de découverte de médicaments. En recherche médicamenteuse, la découverte des « bons » ligands pouvant intéragir avec le récepteur d'intérêt représente souvent un goulot d'étranglement. Parmi des bases de données de milliers de ligand, il faut trouver ceux ayant une affinité avec le récepteur dont le docking avec lui pourra avoir un effet sur celui-ci pour modifier ou inhiber sa fonction. La recherche du ou des ligands qui ont un potentiel de liaison avec le récepteur est longue et fastidieuse. Comme cela est expliqué dans [42], la modélisation de docking moléculaire assistée par ordinateur permet de faire ce qu'on appelle du « virtual screening » qui correspond à une analyse automatique des bases de données de ligand pour extraire les ligands qui peuvent potentiellement se lier au récepteur. Par conséquent l'étape expérimentale s'en trouve diminuée. Un autre intérêt du docking moléculaire est de permettre de trouver une explication à certaines interactions moléculaires. En effet, il est très important de savoir comment se forme un complexe pour espérer mieux le comprendre pour pouvoir notamment le modifier si nécessaire.

3.1.2 Projet ANR Dock

La prédiction de conformations énergétiquement stables de molécule est le fondement des méthodes assistées par ordinateur pour l'étude de l'activité biologique et des propriétés des molécules [98]. De telles prédictions ont un intérêt économique certain car elles facilitent la découverte de composants avec les caractéristiques recherchées. L'énumération de tous les minima d'énergie d'intérêt pour une géométrie donnée est un problème prouvé NP-complet [1, 19, 27] avec de dix à des milliers de degrés de liberté. Par conséquent, ce problème nécessite l'utilisation du calcul distribué. De plus, le paysage énergétique est extrémement chaotique avec des murs énergétiques du aux répulsions de Van Der Waals [56, 57] ce qui entraîne entraine d'énormes modifications d'énergie au moindre décalage dans la position des atomes aussi minimes soient-ils. C'est pour cela que les méthodes d'optimisation multimodales actuelles ont été utilisées sur le problème de l'énumération de conformation [99, 116, 30]. Pour les petites molécules, d'au plus 10 degrés de liberté, les programmes d'énumération systématique ou les simulations de type Monte Carlo sont utilisés. Mais dans le cas de macromolécules, ces mêmes méthodes sont utilisées en partant de molécule déjà en partie repliée de manière expérimentale. Le repliement de protéines de taille modeste ab initio i.e. en ignorant les contraintes expérimentales est encore aujourd'hui un challenge. En effet, les simulations de dynamique moléculaire actuelles ne sont pas assez efficaces pour traverser les barrières énergétiques et il n'est pas aisé de faire ce type de simulation de manière efficace. Des travaux récents [104] ont montré que les algorithmes évolutionnaires et plus spécialement les algorithmes génétiques peuvent être rendus plus efficaces pour le problème d'énumération de conformation par hybridation avec d'autres heuristiques. Cela permet d'obtenir des outils avec un bon comportement face aux barrières énergétiques, une bonne diversité face aux solutions trouvées et une convergence relativement rapide vers les zones de faible énergie. De plus, les algorithmes génétiques sont bien adaptés pour

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

le fonctionnement en parallèle [124].

Le premier but de ce projet est d'obtenir des techniques viables d'optimisation multimodales pour le calcul sur grille dédiées à l'énumération de conformation et au docking moléculaire. Le second but est d'arriver à une définition plus précice d'un champ de force efficace pour obtenir les zones d'énergie minimum sur un ensemble de molécules données (peptides, sucres, ligands) et qui définit un espace énergétique aussi lisse que possible pour permettre de maximiser les chances de succès pour les approches d'énumération de conformations et de docking.

Le projet ANR Dock est axé autour de 4 étapes :

1. La spécification et la validation de modèles mathématiques pour l'énumération de conformations et le docking moléculaire.
2. La mise en place de métaheuristiques coopératives parallèles pour exploiter les nouveaux modèles proposés.
3. L'évaluation des différentes stratégies de déploiement sur grille des métaheuristiques développées.
4. La validation d'outils développés en utilisant des benchmarks connus de la littérature et travaillant sur des problèmes réels.

Au sein de ce projet, le travail présenté dans cette thèse a pour but d'étudier les différentes modélisations possibles pour ce problème.

3.1.3 interactions énergétiques

Nous allons détailler les interactions atomiques existant au sein d'une molécule. Ces interactions sont à la base de la détermination de la stabilité d'un molécule donnée.

3.1.3.1 Entre atomes liés

Ce type d'interaction correspond aux relations qu'il y a entre des atomes qui de manière topologique sont liés selon 1, 2 ou 3 liaisons. On présentera donc l'énergie de liaison (2 atomes et 1 liaison), angulaire (3 atomes et 2 liaisons) et de torsion (4 atomes et 3 liaisons).

Energie de liaison L'énergie de liaison correspond à l'interaction qui existe entre 2 atomes reliés par une liaison dite de covalence. Ces liaisons sont le résultat de la mise en commun d'orbitales électroniques. Par conséquent, les liaisons de covalence peuvent se rompre ou se créer entre des molécules au cours de réactions chimiques par exemple. Les orbitales, nuages d'électron autour du noyau atomique, peuvent être « liantes » si elles sont partiellement remplies i.e. un électron cherche à se lier, ou « non liantes » si elles sont complètement remplies. Ces liaisons de covalence peuvent être de différents ordres (simple, double ou triple) et sont donc plus ou moins flexible selon leur ordre. En effet, les liaisons simples, les plus courantes, sont relativement flexibles en tant qu'axe de rotation. Au contraire, les liaisons multiples ne peuvent être que dans 2 états : *cis* et *trans*. L'état « cis » correspond à la présence des atomes voisins de la liaison dans le même plan alors que dans l'état « trans », ils sont dans des plans opposés. La figure 3.1 montre un exemple d'états cis et trans pour une molécule de stilbène composée de deux anneaux de benzène.

La figure 3.2 décrit le modèle utilisé pour simuler l'énergie de liaison entre deux atomes.

Ce modèle est basé sur le système du « ressort » car les forces en jeu sont de types vibratoires. Sur ce modèle, l'énergie de liaison se traduit par l'équation 3.1.

$$E_{liaison} = K_l(d - d_0)^2 \tag{3.1}$$

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

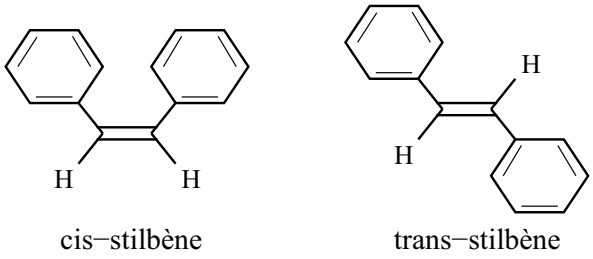


FIG. 3.1 – Forme cis et trans du stilbène.

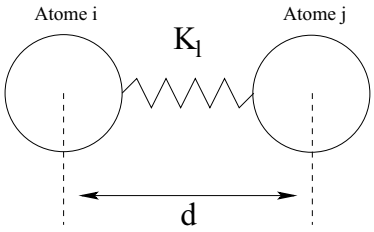


FIG. 3.2 – Schéma décrivant le modèle énergétique d'une liaison. d est la distance réelle entre les atomes i et j. K_l est la constante de force associée.

Dans cette équation, K_l est la constante de raideur associée au modèle (400 kJ/mol/Å^2), d la longueur réelle de la liaison et d_0 la longueur idéale de la liaison. L'annexe C.1 présente un exemple des longueurs idéales ainsi que l'énergie associée de quelques liaisons.

Energie d'angle Autour de chaque atome, les orbitales électroniques tendent à occuper l'espace au maximum. Par conséquent, elles ont tendance à se repousser les unes les autres. Cela correspond à une des 4 règles de Gillespie (cf annexe C.2). Comme pour l'énergie de liaison, l'énergie d'un angle est modélisée sur le principe du ressort comme cela est montré sur la figure 3.3.

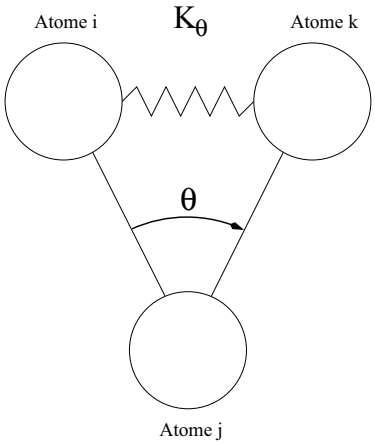


FIG. 3.3 – Schéma décrivant l'énergie d'un angle. θ est l'angle réel entre les atomes i et k. K_θ est la constante de force associée.

L'équation 3.2 décrit la formule de calcul de l'énergie correspondante.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

$$E_{angle} = K_\theta(\theta - \theta_0)^2 \tag{3.2}$$

K_θ est la constante de force associé ($40kJ/mol/deg^2$), θ est l'angle réel entre les 3 atomes et θ_0 l'angle idéal. On remarquera que la constante de force est plus faible que dans le cas de l'énergie de liaison, ce qui permet la possibilité d'oscillations plus importantes.

Energie de torsion Une torsion est un quadruplet d'atomes relié deux à deux par des liaisons. La figure 3.4 montre un exemple de torsion en mettant en évidence la liaison qui sert d'axe de rotation pour celle-ci. Même s'il n'a pas d'interprétation théorique, ce terme amène à des résultats empiriques plus cohérent. Il est utilisé pour modéliser les modifications d'énergie qui ont lieu dans une partie de la molécule qui tourne selon une liaison comme axe de rotation.

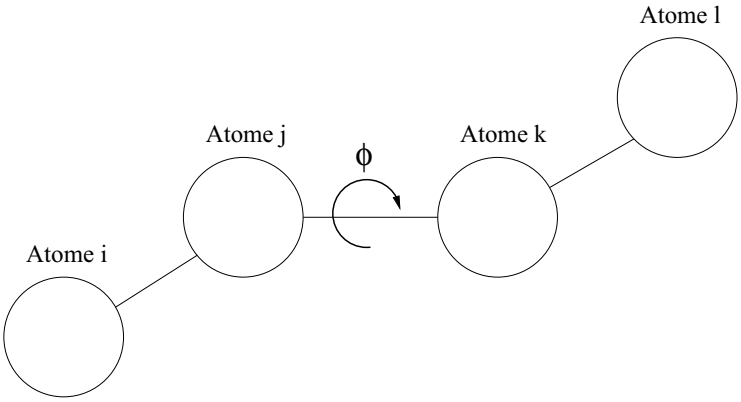


Fig. 3.4 – Schéma décrivant l'énergie d'une torsion. ϕ est l'angle de rotation principal de la rotation.

Ce terme énergétique se traduit par l'équation 3.3 dans laquelle ϕ est l'angle de la torsion, ϕ^0 prend les valeurs 0 ou π, n vaut 2, 3 ou 4 et K_ϕ est la constante de force associée.

$$E_{torsion} = K_\phi(1 + cos(n\phi - \phi^0) \tag{3.3}$$

3.1.3.2 Entre atomes non liés

Les interactions entre atomes non liés correpondent aux énergies mises en jeu qui ne sont pas basées sur la notion de liaison. Ces types d'interactions sont essentiels car ce sont eux qui assurent la stabilité des molécules et leur cohésion au sein de leur environnement.

Energie de Van der Waals Les interactions de type Van der Waals ont lieu entre des atomes proches non liés. Plus la distance entre deux atomes non liés est petite, plus les électrons des deux noyaux correspondants ont tendance à s'attirer. Mais une force de répulsion du à l'impossibilité de recouvrement des nuages électroniques compense cette attraction. On arrive donc à une distance d'équilibre des forces d'attraction et de répulsion, cette distance correspond au rayon de Van der Waals. Les rayons de Van der Walls d'un ensemble d'éléments sont donnés en annexe C.3. La figure 3.5 montre où ont lieu les interactions de Van der Waals selon la topologie des atomes.

l'évolution de la contribution de Van der Waals est présentée dans la figure 3.6. Cette contribution varie selon la distance entre les deux atomes concernés.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

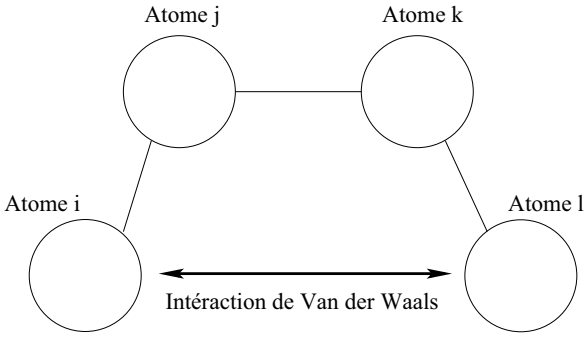


FIG. 3.5 – Schéma décrivant où les interactions de Van der Waals peuvent avoir lieu.

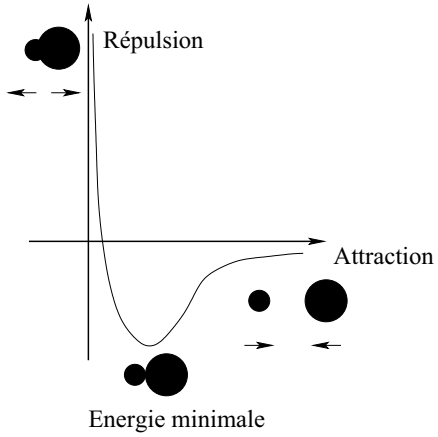


FIG. 3.6 – Schéma décrivant l'énergie de Van der Waals.

L'équation 3.4 décrit le calcul de cette contribution.

$$E_{Van_der_Waals} = \frac{K_1}{d^{12}} - \frac{K_2}{d^6} \tag{3.4}$$

K_x est la constante de force liée à l'atome x et d est la distance entre les noyaux des atomes 1 et 2. Il est important de noter que le calcul des interactions de type Van der Waals, ainsi que des autres interactions entre atomes non liés, est coûteux du au nombre d'ensembles d'atomes non liés existant au sein d'une même molécule.

Energie de Coulomb L'existence de différents niveaux d'électronégativité entre les noyaux atomiques entraine une polarisation des liaisons au sein d'une molécule. La présence des charges associées à cette polarisation se traduit par des interactions coulombiennes dont l'évaluation se fait via l'équation 3.5.

$$E_{Coulomb} = \frac{q_1 q_2}{4\pi\epsilon d} \tag{3.5}$$

q_x est la charge associée à l'atome x, d est la distance entre les atomes 1 et 2 et ϵ est la constante diélectrique du solvant.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

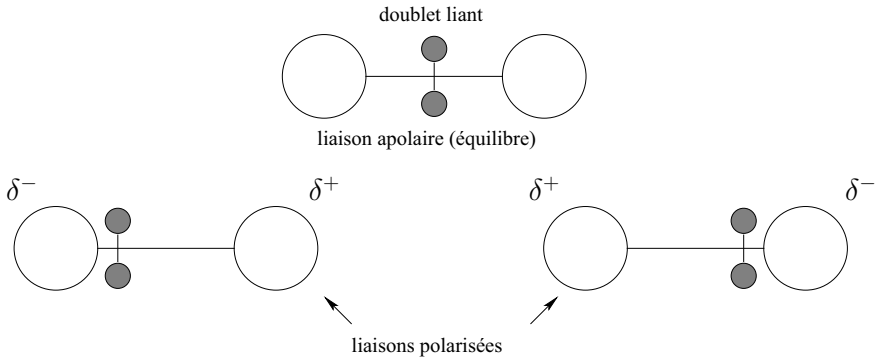


FIG. 3.7 – Schéma décrivant la polarisation d'une liaison suivant le déplacement du doublet liant.

L'énergie reste le critère nécessaire pour évaluer la stabilité d'une molécule. Cependant, il existe d'autres critères qui ne sont pas énergétiques. La surface moléculaire, que nous allons utiliser dans nos travaux, en est un exemple.

3.1.4 Notion de surface

La surface peut être classée comme un critère géométrique car elle donne une indication, indirecte, sur la forme des molécules. Dans le cas du docking, nous utiliserons cette notion pour qualifier le degré de pénétration du ligand au sein du récepteur. Il existe plusieurs types de surface, nous allons les présenter avant d'indiquer laquelle nous utiliserons dans nos travaux.

3.1.4.1 Définition

En deux dimensions, le calcul de la taille du pourtour d'une figure donnée est appelé calcul de périmètre. En passant à trois dimensions, le calcul de la zone formée par la couche extérieure d'un objet est appelée ici calcul de surface. Par conséquent, pour un complexe moléculaire donné, le calcul de sa surface correspondra à la mesure de la taille de la couche extérieure de son volume. De ce principe, la surface des molécules peut être décrite de trois manières différentes. On parlera de surface **de Van der Waals** qui est la plus simple à représenter mais que ne tient pas compte de la présence de solvent. Il existe aussi celle dite **accessible au solvent** qui elle gère la notion de solvent autour des atomes mais de manière simplifiée. Enfin il existe aussi la surface de **Connolly** qui en plus de simuler la présence du solvent, elle le fait de manière plus fine que la surface précédente.

Un atome peut être schématisé telle une sphère à l'aide de son rayon de Van der Waals. En sommant

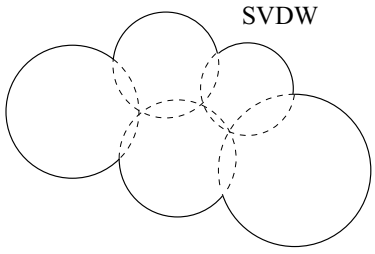


FIG. 3.8 – Surface de Van der Waals (SVDW) pour une molécule composée de cinq atomes.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

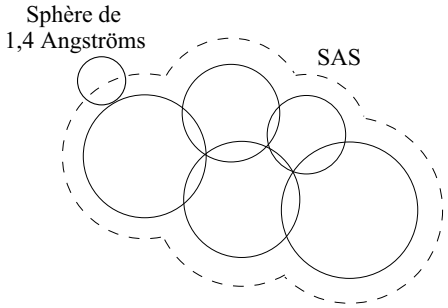


FIG. 3.9 – Surface accessible au solvent (SAS) d'une molécule composée de cinq atomes. La sphère qui permet le tracé de la surface modélise l'action du solvent, elle est censée contenir une molécule d'eau.

les surfaces des sphères de chaque atome composant une molécule, sans tenir compte des surfaces en collision, on obtient la surface de Van der Waals d'une molécule. La figure 3.8 montre un exemple de cette surface pour une molécule de cinq atomes. La surface accessible au solvent et plus tard la surface de Connoly furent définies par Lee and Richards dans [79] et [109] respectivement. Pour la surface de Van der Waals, la molécule est considérée comme étant dans le vide mais cela n'est pas un modèle réaliste. Quelle que soit la molécule, elle est naturellement entourée d'un solvent. Pour la surface accessible au solvent et la surface de Connoly, ce solvent est modélisé par une sphère « roulant » sur le pourtour de la molécule dont on veut connaître la surface. Pour la surface accessible au solvent, ce parcours suit le mouvement du centre de la molécule symbolisant le solvent. Cette molécule de solvent a généralement un rayon de 1.4 Å (1 angstrom(Å)= 0.1 nanomètre) car elle doit être capable de contenir une molécule d'eau qui est un des solvents standard. La figure 3.9 présente la surface accessible au solvent sur le même exemple de molécule à cinq atomes que précédemment.

Pour la surface de Connoly, la surface tracée par la sphère de solvent représente tous les points de la sphère touchant les surfaces des sphères des atomes. Un cas particulier intervient quand la sphère de solvent touche deux autres sphères en même temps, dans ce cas la surface est tracée en suivant le contour de la sphère de solvent orientée vers la molécule. La figure 3.10 présente la surface de Connoly sur le même exemple de molécule à cinq atomes que précédemment.

Différentes méthodes pour calculer ses surfaces existent, des exemples de telles méthodes peuvent être trouvé dans [45, 47, 114, 131].

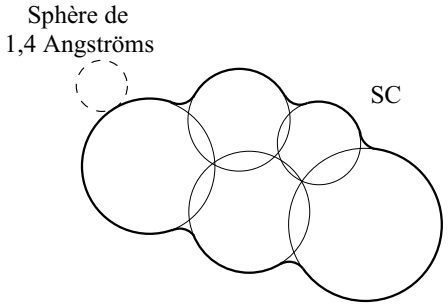


FIG. 3.10 – Surface de Connoly (SC) d'une molécule composé de cinq atomes. La sphère qui permet le tracé de la surface modélise l'action du solvent, elle est censée contenir une molécule d'eau.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.1.4.2 Surface utilisée

Comme nous l'avons indiqué prédemment, la notion surface va être utilisée dans nos modélisations afin de qualifier le degré de pénétration du ligand dans le site. Il nous a donc fallu choisir quel type de surface utiliser. Comme nous voulons trouver le modèle qui simule au mieux le processus de docking moléculaire, il semblait clair que la surface de Van der Walls ne pouvait être choisie car elle ne permet pas la prise en compte du solvent. Restait donc à désigner quelle surface utilisée parmi celle accessible au solvent et celle de Connoly. Les deux prennent en compte le solvent mais celle de Connoly permet le tracé d'une surface plus respectueuse de la réalité. L'objectif du critère de surface étant l'obtention de la valeur de la surface et non son tracé, le gain de traitement à utiliser celle accessible au solvant plutôt que celle de Connoly, plus complexe à calculer, nous a finalement décidé. Par conséquent dans le reste de ce chapitre, lorsque nous évoquerons la surface d'une molécule, cela signifiera la surface accessible au solvent. En effet, nous calculons cette surface via un algorithme l'approximant. Ce type de surface représente un bon compromis qualité/coût. En effet, elle modélise la présence du solvent mais son calcul reste moins coûteux comparé à la surface de Connoly. L'algorithme utilisé a été présenté la première fois dans [53] mais a aussi été utilisé plus récemment dans [78]. Il approxime la méthode Shrake et Rupley [118].

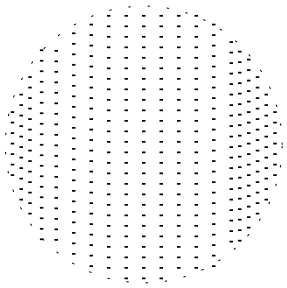


Fig. 3.11 – Représentation d'un atome comme une sphère de points.

Selon cette méthode, chaque sphère d'atome est représentée par un ensemble de points comme schématisé dans la figure 3.11. Chaque point est codé sous forme d'un bit pour indiquer si celui-ci est en interaction avec le solvent (1) ou non (0). Ainsi une surface sphérique est représentée comme une chaîne de bits. Par cet encodage, calculer l'aire de chaque atome revient à des « et » logique. Des tables de conversions sont utilisées pour accélérer le calcul en sauvant les intersections des points atomiques sous forme de masques Booléens.

3.1.5 Energie et champs de force

Quelles ques soient les interactions énergétiques utilisées, les calculs sont basés sur des ensembles de constantes liées à chaque type d'interaction. Ces constantes dépendent fortement du champ de force utilisé pour les définir. C'est notamment pour cela qu'il n'est pas possible de comparer les valeurs énergétiques venant de deux champs de force différents. A l'origine, nous avons travaillé sur une évaluation de l'énergie selon un champs de force dérivé du « Consistent Valence Force Field » car c'est ce qui avait été déjà utilisé par notre partenaire chimiste lors de précédentes études. Cependant nous avons aussi testé celui qui est utilisé à l'intérieur de la suite d'outils Autodock qui est à l'heure actuelle un des outils les plus utilisés pour le docking moléculaire.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.1.5.1 L'énergie inspirée du « consistent valence force field »

Pour éviter toute confusion, on désignera ce champs de force par ICVFF pour « Inspired by the Consistent Valence Force Field ». L'énergie dans le ICVFF somme les différents types d'interaction énergétiques tels que nous les avons présentés précédemment. Ainsi on retrouvera l'énergie des atomes liés décrits dans l'équation 3.6.

$$\begin{aligned} E_{atomes_lies} = & \sum_{liaisons} K_b(b - b_0)^2 + \\ & \sum_{angles} K_\theta(\theta - \theta_0)^2 + \\ & \sum_{torsions} K_\phi(1 - cos\ n(\phi - \phi_0)) \end{aligned} \tag{3.6}$$

K_b, K_θ et K_ϕ sont des constantes de force liées à chaque type de constribution que sont les liaisons, les angles et les torsions. b_0, θ_0 et ϕ_0 sont des valeurs empiriques pour la longueur, l'angle et la différence de phase. b, θ et ϕ sont les valeurs réelles pour la longueur, l'angle et la différence de phase. Pour la contribution de type torsion, n est la périodicité liée au type de la liaison central de la torsion (double ou triple).

L'équation 3.7 décrit qu'elles sont les termes énergétiques rentrant en considération dans les interactions entre atomes non liés.

$$\begin{aligned} E_{atomes_non_lies} = & \sum_{Van\ der\ Waals} \frac{K^a_{ij}}{d^{12}_{ij}} - \frac{K^b_{ij}}{d^6_{ij}} + \\ & \sum_{Coulomb} \frac{q_i q_j}{4\pi\epsilon d_{ij}} + \\ & \sum_{desolvatation} \frac{K q_i^2 V_j + q_j^2 V_i}{d^4_{ij}} \end{aligned} \tag{3.7}$$

Dans cette équation, q_i est la charge de l'atome i ; d_{ij} est la distance entre les atomes i et j ; V_i est une mesure volumétrique pour l'atome i ; K et K^x_{ij} sont des constantes de force liées à la contribution considérée. La contribution de type Van der Waals permet de qualifier les forces d'attraction et de répulsion rentrant en jeu entre deux atomes selon la distance entre leurs centres. La contribution de type Coulomb décrit comment des différences d'électronégativité au sein d'une molécule entre des atomes de taille et de masse différente peuvent avoir un impact sur l'énergie correspondante. Ces différences produisent des charges pouvant être attractives ou répulsive. Le terme de désolvation modélise l'action du solvent autour de la molécule.

L'énergie d'une molécule se traduit donc selon l'équation suivante :

$$\begin{aligned} E_{totale} =& E_{atomes_lies} + E_{atomes_non_lies} \\ =& E_{laisons} + E_{angles} + E_{torsion} + E_{Van\ der\ Waals} + E_{Coulomb} + E_{desolvatation} \end{aligned} \tag{3.8}$$

Le ICVFF correspond au paramétrage d'un ensemble de facteurs qui sont liés aux différentes interactions. Ce champ de force est toujours en phase d'amélioration. Il se base sur l'étude d'un ensemble de molécules dont la conformation de plus basse énergie à l'état naturel est connue. Le but est de trouver le jeu de paramètres pour le champ de force qui permet de générer les conformations de molécule les plus proches de celles considérées comme optimales. Ainsi le champ de force doit permettre de simuler au mieux les variations d'énergie existant au sein d'une molécule.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.1.5.2 L'énergie dans Autodock

L'équation 3.9 présente tous les termes de la fonction d'énergie utilisée dans Autodock [97].

$$\begin{aligned}\Delta G =& \Delta G_{vdw} \sum_{i,j} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + \\ & \Delta G_{hbond} \sum_{i,j} E(t) \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} + E_{hbond} \right) + \\ & \Delta G_{elec} \sum_{i,j} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} + \Delta G_{tor} N_{tor} + \Delta G_{sol} \sum_{i,j} S_i V_j e^{-r_{ij}^2/2\sigma^2}\end{aligned} \tag{3.9}$$

Dans cette équation, on va retrouver des termes similaires à ceux du ICVFF comme l'énergie de Van der Walls ΔG_{vdw}, de liaison ΔG_{hbond}, de Coulomb ΔG_{elec}, de torsion $\Delta G_{tor} N_{tor}$ et de désolvatationΔG_{sol} même s'ils ne sont pas exactement définis de la même manière. Le terme de désolvatation donné dans l'équation est celui d'Autodock 3. Dans Autodock 4, le terme de désolvatation est défini de la manière suivante [66] :

$$\Delta G_{desolv} = W_{desolv} \sum_{i,j} (S_i V_j + S_j V_i) e^{\frac{-r_{ij}^2}{2\sigma^2}} \tag{3.10}$$

Où i est l'indice des atomes dans le ligand, j celui des atomes dans le site, W_{desolv} un poids associé au terme de désolvatation, S_i le terme de solvatation de l'atome i, V_i le volume du fragment atomique de l'atome i, r_{ij} la distance entre les atomes i et j en Angström et σ une constante de distance gaussienne (3,5 Å).

Comme pour le ICVFF, le champ de force utilisé dans Autodock 4.0 a été optimisé sur un ensemble de complexes ligand/récepteur dont on connaissait la conformation optimale.

L'information énergétique d'une molécule nous donne une indication sur sa stabilité. Cependant, elle ne permet pas de savoir si elle est plus ou moins sensible à de petites variations de conformations. A partir de cette constatation, nous avons décidé d'en tirer un nouveau critère d'évaluation de qualité : la robustesse.

3.1.6 Notion de robustesse

Cette notion se base sur un calcul de type entropie sur un couple ligand/récepteur donné. Elle doit permettre de déterminer si un couple ligand/récepteur est capable de résister à des modifications de la position du ligand par rapport à sa position actuelle dans le récepteur (rotation et/ou translation). Ce calcul nécessite l'échantillonnage des solutions autour d'un complexe donné. Ainsi le nombre de voisins $nbConf$ et leur énergie associée, le critère de robustesse est donné par l'équation 3.11.

$$\begin{aligned} Robustesse &= -\frac{1}{\beta} * ln \sum_{i=1}^{nbConf} -\beta * Energie[i] \\ avec\, \beta &= \frac{1}{kT} \end{aligned} \tag{3.11}$$

k est la constante de Boltzmann et T la température. La définition de cette équation maintient la formule de Boltzmann. Le but est d'estimer si un complexe ligand/récepteur est plus probable qu'un

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

autre. Un complexe relativement flexible est dans une large vallée de potentiels. Un complexe rigide est dans un puit de potentiels très serré. L'hypothèse traduite par cette équation viendrait du fait que même si on a souvent comme objectif de choisir un complexe de plus faible énergie possible, il est possible qu'en réalité, le complexe choisi ait une énergie plus haute mais plus robuste à de petites variations. Cependantcette hypothèse reste à vérifier et sera donc testée dans de notre étude de nouveaux modèles multi-objectifs pour le docking.

Peu importe les critères d'évaluation utilisés, une des grandes difficultés dans le docking moléculaire est de pouvoir comparer les résultats d'exécution avec d'autres travaux. En effet, à moins d'utiliser exactement les même champs de force avec les mêmes paramétrages, comparer des énergies n'a pas de sens. C'est pour cela que le RMSD a été utilisé.

3.1.7 Critère de comparaison de conformation : le RMSD

Ce critère est pour les problèmes de docking le seul qui peut-être appliqué pour comparer les différents modèles et méthodes existantes à partir du moment où les mêmes intances sont utilisées. RMSD signifie **R**oot **M**ean **S**quare **D**eviation. Selon [128], le RMSD est défini de la manière suivante :

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n}(dx_i^2 + dy_i^2 + dz_i^2)}{n}} \tag{3.12}$$

Dans l'équation 3.12, n correspond au nombre total d'atomes (sans considérer les hydrogènes). dx_i, dy_i et dz_i sont les différences de coordonnées atomiques entre le ligand dans sa position obtenue par l'algorithme et sa position considérée comme optimale.

Le RMSD permet de définir la distance entre une conformation donnée et une conformation objectif. Pour le docking, ce critère correspond à évaluer la position du ligand d'une solution finale par rapport à sa position dans la solution considérée comme optimale. Cependant, de par sa définition, il n'est pas considéré comme un critère robuste. En effet, plus une molécule sera importante moins un décalage dans la position des atomes aura un impact sur le RMSD associé. Inversement, plus une molécule sera petite, plus le moindre décalage aura un impact conséquent. Un autre problème avec ce critère est qu'il ne gère pas la notion de symétrie. Cela est schématisé par la figure 3.12.

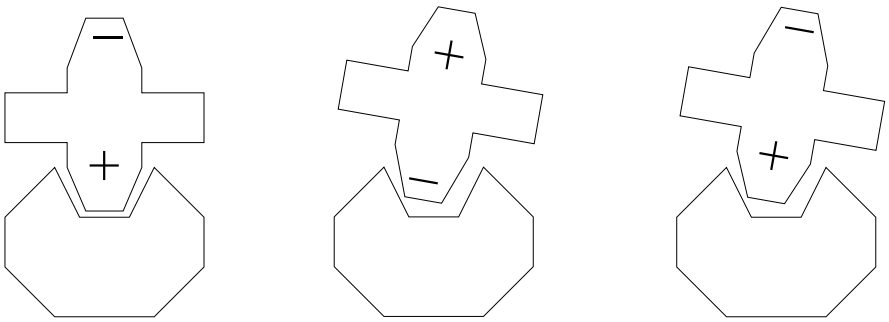


Fig. 3.12 – Exemple de symétrie qui a un impact non négligeable sur le calcul du RMSD.

Cela permet d'indiquer que même si une molécule a une RMSD de mauvaise qualité (généralement > 2.0 Å), une analyse plus poussée est toujours nécessaire pour valider ou non un résultat de docking. Cependant, il est important de remarquer que le RMSD est un critère de comparaison et que donc il

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

nécessite d'avoir une molécule de référence pour pouvoir être calculé. Pour la validation du docking, ce critère nécessite donc l'information du complexe ligand/récepteur optimal venant de données crystallographiques.

3.2 Etat de l'art

Dans cette section, nous différencions les différents modèles utilisés pour le docking moléculaire en nous concentrant sur les modélisations multi-objectives.

3.2.1 Classement des méthodes de résolution

Lors de la simulation de docking, les molécules utilisées peuvent être considérées comme flexible ou non. Par flexible, on désigne le fait qu'une molécule puisse avoir sa conformation interne modifée. Les premiers modèles de docking se basaient sur des modèles non ou peu flexibles (différentes conformations pour le ligand). Aujourd'hui les derniers travaux proposent des approches semi-flexibles voir flexibles mais avec des degrés de liberté limités. Le nombre de degrés de liberté correspond généralement au nombre de torsions considérées comme flexibles dans une molécule. Les principaux problèmes du docking sont :

- La taille de l'espace de recherche liée à toutes les conformations possibles pour le ligand et le récepteur.
- Les conformations les plus stables du site et du ligand n'amèneront pas forcément au meilleur complexe.
- Le bon complexe ne sera pas forcément celui le plus stable (énergétiquement parlant).

Au niveau de leur modélisation, deux familles ont été extraites : les modèles énergétiques purs et les modèles mixtes. La première famille correspond aux premiers modèles mono-objectif proposés mais aussi aux modèles multi-objectif basés sur une séparation des différents termes énergétiques. La seconde famille de modèles, généralement multi-objectif, permet de tenir compte de critère non énergétique au sein de la simulation du docking. Quelle que soit la famille utilisée, trois types de docking sont possibles suivant la flexibilité des molécules utilisées :

1. rigide : les deux molécules sont considérées comme structuralements fixes. Il n'y a pas de modifications possibles de leur conformation.
2. semi-flexible : une des deux molécules, généralement le ligand, peut voir sa conformation modifiée.
3. flexible : les deux molécules peuvent avoir leur conformation modifiée. Autodock 4 permet ce type de docking en spécifiant quelles parties du récepteur sont flexibles.

De part notre travail au sein du projet ANR Dock et de nos objectifs à atteindre, nous nous sommes concentrés sur les modélisations multi-objectives pour le problème du docking moléculaire flexible.

3.2.2 Modélisations multi-objectives

La modélisation la plus utilisée est celle basée sur une évaluation bi-objective de l'énergie héritée des problèmes de prédiction de structures de protéine selon les 2 termes suivants : énergie des atomes liés et énergie des atomes non liés [29, 28, 127]. Elle est d'autant plus facilitée que les librairies telles que CHARMM décomposent déjà l'énergie selon ces deux termes. Dans [100], une modélisation à trois

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

objectifs est proposée, les deux premiers objectifs sont les mêmes que les méthodes bi-objectif mais le troisième objectif est une composante de complémentarité géométrique. C'est la seule modélisation multi-critère qui fait intervenir cette composante dans la fonction d'évaluation. En effet, cette composante est généralement utilisée comme filtre pour supprimer des conformations dans d'autres approches [136].

Une autre modélisation bi-objectif proposée dans [67] fait la différence entre les interactions énergétiques entre le ligand et le récepteur (Δ_{inter}) et celles existant uniquement entre les atomes du ligand(Δ_{intra}). Chaque terme utilise donc la même fonction d'évaluation mais sur des systèmes différents. Cette fonction d'évaluation est celle d'Autodock.

3.2.3 Logiciels existants

Dans [16], une étude comparative permet de distinguer les algorithmes existants selon 2 catégories : ceux à base de population (majoritairement des algorithmes génétiques) et ceux dits constructifs. La première catégorie est représentée par la suite d'outils Autodock [97] mais aussi par le logiciel GOLD [69]. La seconde catégorie est représentée quand à elle par des logiciels tels que Flex [108] et DOCK [41]. Il est important de remarquer que derrière chacun de ces logiciels se cachent en fait un certain nombre de sous-logiciels qui jouent un rôle important durant chaque étape du docking (formatage des entrées/sortie, calcul distribué, calcul des scores, ...). De plus, pour quiconque développe aujourd'hui un logiciel de docking, il faut nécessairement être capable de faire du docking flexible (ligand/site ou au moins ligand).

Quelque soit le logiciel docking, l'analyse des résultats est en soit un exercice très difficile. C'est pourquoi des logiciels qui permettent de comparer et de visualiser des résultats de docking sont très intéressants. Par exemple, le logiciel PostDock [133] permet de visualiser en 3 dimensions les résultats de docking obtenus via Autodock ou MOE-Dock. Ce dernier est un logiciel industriel faisant partie de la suite MOE (Molecular Operating Environment). PostDock a été justement intégré à cette suite de logiciels.

Au vu des modèles et outils existants, nous allons proposer un certains nombre de modélisations multi-objectives basées sur des critères énergétiques mais aussi sur des critères géométriques.

3.3 Nouvelles modélisations multi-objectives

L'analyse des modèles existants et des critères possibles nous a amenés à proposer huit modèles multi-objectifs : quatre bi-objectif et quatre tri-objectif.

Les différences entre ces modèles dépendent des critères et du champ de force utilisés. La moitié se base sur le ICVFF et l'autre sur le champ de force d'Autodock 4. Le but de cette différenciation est de permettre de voir l'impact de l'utilisation des différents critères au sein d'un même champ de force sur la qualité des résultats. De plus, cela permet de connaître pour les mêmes critères mais un champ de force différent si la qualité des solutions est modifiée.

Chaque paire de modèles (ICVFF et Autodock) a été mise en place pour une raison particulière :

- Modèles M2 et M6 : il s'agit de modélisations bi-objectif basées sur le découpage de l'énergie entre intéractions entre atomes liés et atomes non liés. C'est un modèle bien connu notamment pour le problème de prédiction de structure de protéine. C'est une modélisation de référence qui est nécessaire pour évaluer nos autres modèles.
- Modèles M3 et M7 : ces modélisations correspondent à un modèle tri-objectif basé sur deux critères énergétiques et un critère de surface. Cela représente un premier modèle novateur pour le docking flexible. Par hypothèse, il est censé être le plus représentatif du fonctionnement du docking.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.1 – Différents modèles multi-objectifs utilisés. ICVFF signifie « inspired by the consistent valence force field », E_{total} énergie totale (sans séparation des termes énergétiques), E_{al} énergie atomes liés e E_{anl} énergie atomes non liés.

	Champ de force							
	ICVFF			Autodock 4.0				
Modèle	E_{total}	E_{al}	E_{anl}	E_{total}	E_{al}	E_{anl}	**Surface**	**Robustesse**
M1	X						X	
M2		X	X					
M3	X						X	X
M4		X	X				X	
M5				X			X	
M6					X	X		
M7				X			X	X
M8					X	X	X	

– Modèles M1 et M5 : ces modèles correspondent au modèle M3 et M7 sans la notion de robustesse. L'intérêt de ces modèles est double : d'une part, ils sont moins coûteux car le critère de robustesse engendre des calculs d'énergie supplémentaires et d'autre part, ils permettent d'évaluer l'impact du la notion de robustesse dans le problème du docking.

3.4 Etude du paysage énergétique autour des complexes optimaux

Dans cette section, nous allons générer des paysages énergétiques pour savoir si les hypothèses émises concernant le complexe considéré comme optimal sont avérées. Pour rappel, les hypothèses étaient les suivantes :

1. Le complexe ligand/récepteur optimal est robuste et donc ne doit pas se trouver dans un puit énergétique étroit.
2. Le meilleur complexe ligand/récepteur n'est pas forcément celui de plus basse énergie.

3.4.1 Protocole de génération

Le critère de robustesse proposé parmi les différents modèles se base sur l'hypothèse que le vrai complexe ligand/récepteur se situerait dans une vallée d'énergie plutôt qu'un puit. Dans une telle configuration, le complexe serait donc robuste à de petits changements. Pour valider notre hypothèse, des analyses de paysage ont été faites en utilisant des complexes ligand/récepteur considérés comme optimaux. La première analyse se base sur des modifications de positionnement du ligand par des rotations de celui-ci selon deux axes à chaque fois (X/Y, X/Z et Y/Z respectivement). La seconde analyse se base sur le même principe mais avec des translations.

Nous allons présenter les résultats obtenus sur un des complexes ligand/récepteur que nous utiliserons lors de notre phase de tests. Il s'agit du complexe Ribonuclease A / Uridine-2',3'-Vanadate et son identifiant dans la Protein Data Bank (PDB) est 6rsa. Par la suite, nous évoquerons les instances par leur identifiant PDB. Seules les paysages de l'instance 6rsa sont présentés ici, d'autres paysages sont cependant visibles en annexe C.4.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.4.2 Paysages énergétiques avec le ICVFF

Les figures 3.13, 3.14 et 3.15 montre les trois paysages obtenus à partir de l'instance 6rsa par rotation.

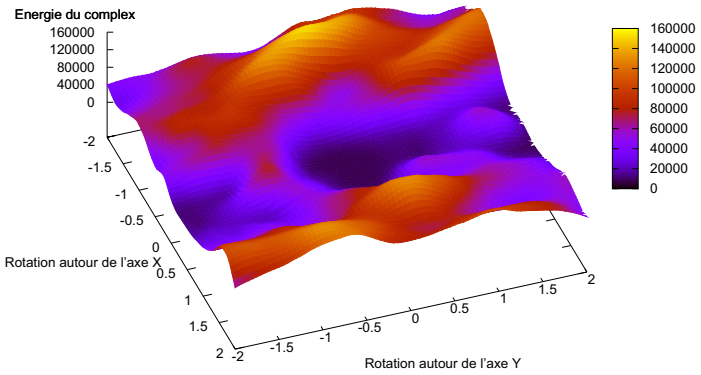


FIG. 3.13 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes X et Y.

Sur chaque figure, l'énergie du complexe dans sa configuration optimale est située au centre du paysage. On remarque que sur chacun des paysages, le complexe n'est jamais bloqué au milieu de très hautes barrières énergétiques. Il existe toujours un chemin pour accéder à la vallée où se situe le complexe.

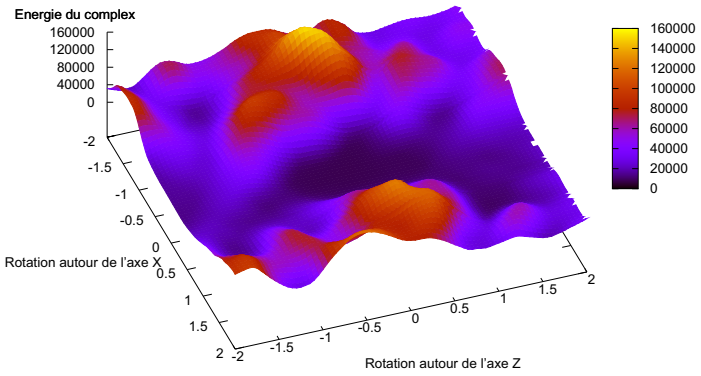


FIG. 3.14 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

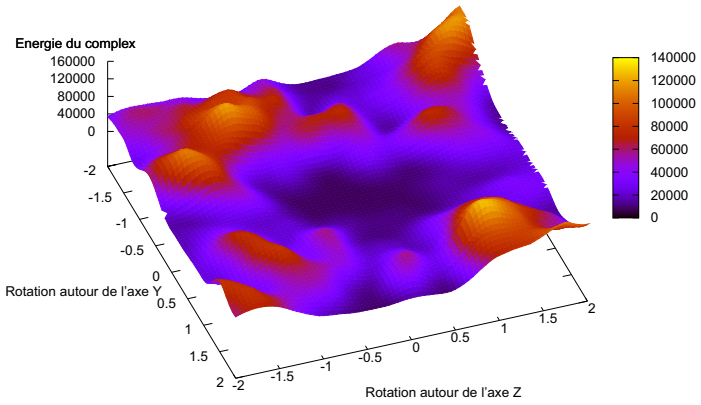


FIG. 3.15 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes Y et Z.

Les figures 3.16, 3.17 et 3.18 montre le paysage engendré par des translations du ligand sur la même instance.

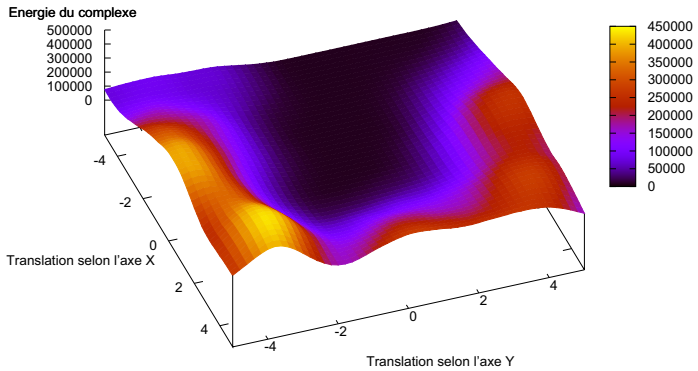


FIG. 3.16 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes X et Y.

On peut voir que les translations ont un impact moindre sur l'énergie du complexe que les rotations. On obtient ainsi de large vallée dans laquelle il n'est pas aisé de distinguer où se trouve le complexe optimal.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

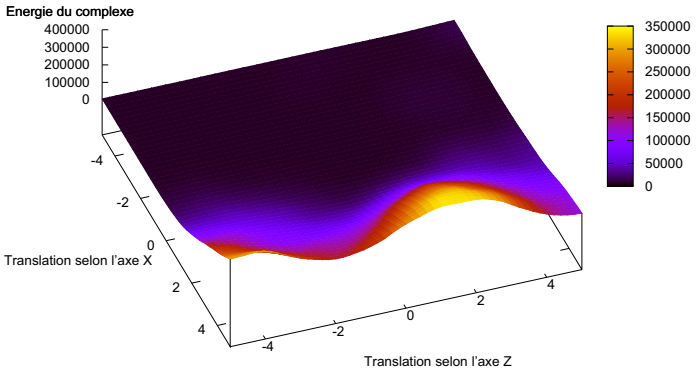


FIG. 3.17 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes X et Z.

Par conséquent, un paysage basé sur les rotations parait plus discriminant pour déterminer où se trouve le complexe.

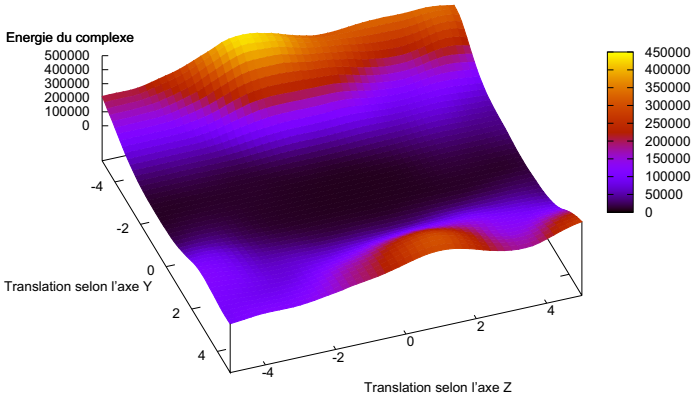


FIG. 3.18 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes Y et Z.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.4.3 Paysages énergétiques selon la fonction d'énergie d'Autodock 4.0

Les figures 3.19, 3.20 et 3.21 montre les trois paysages énergétiques obtenus à partir de l'instance 6rsa par rotation du ligand avec la fonction d'énergie d'Autodock.

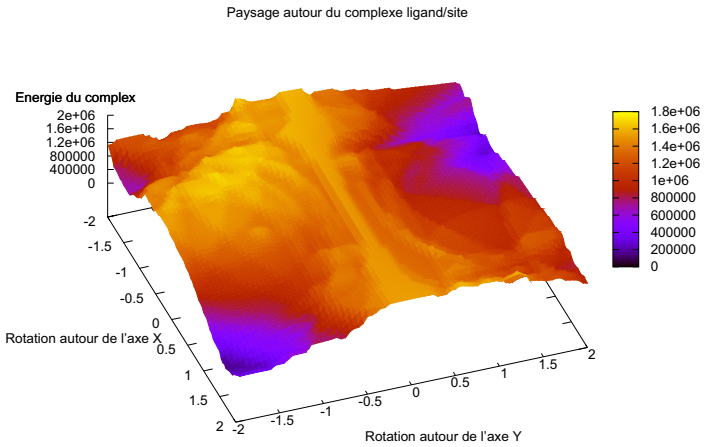


FIG. 3.19 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes X et Y.

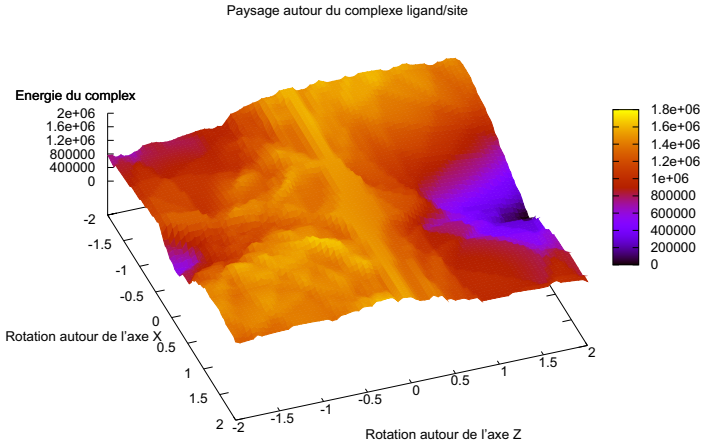


FIG. 3.20 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

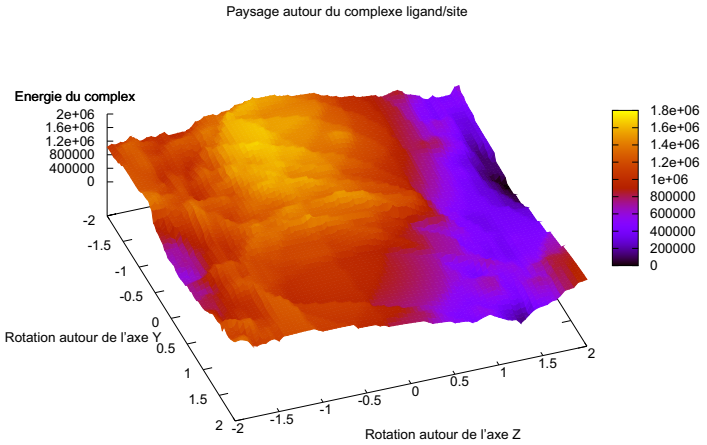


FIG. 3.21 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des rotations selon les axes Y et Z.

A la vue de ces trois paysages, on peut remarquer que l'optimum situé au centre de chaque paysage ne possède pas la plus basse énergie. Cela tend à prouver que pour le problème de docking, le meilleur complexe n'a pas la plus basse énergie.

Les figures 3.22, 3.23 et 3.24 montrent le paysage engendré par des translations du ligand sur la même instance.

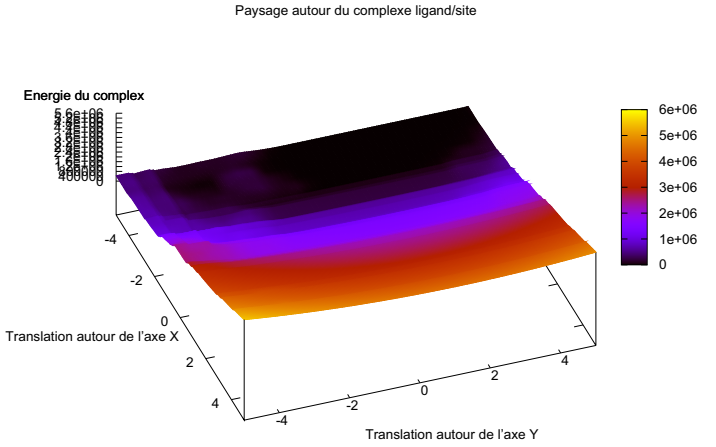


FIG. 3.22 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes X et Y.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

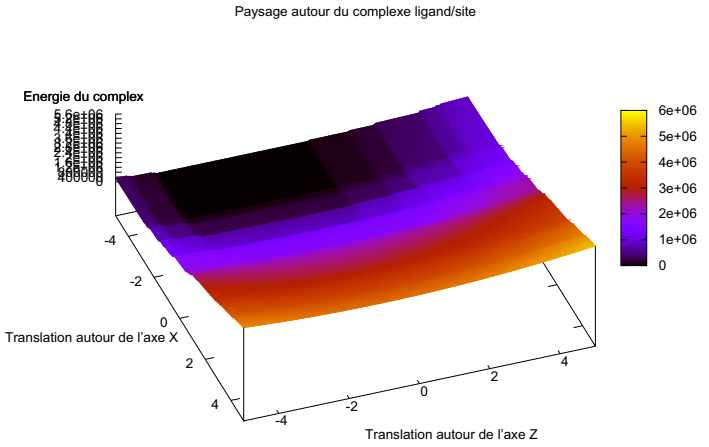


FIG. 3.23 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes X et Z.

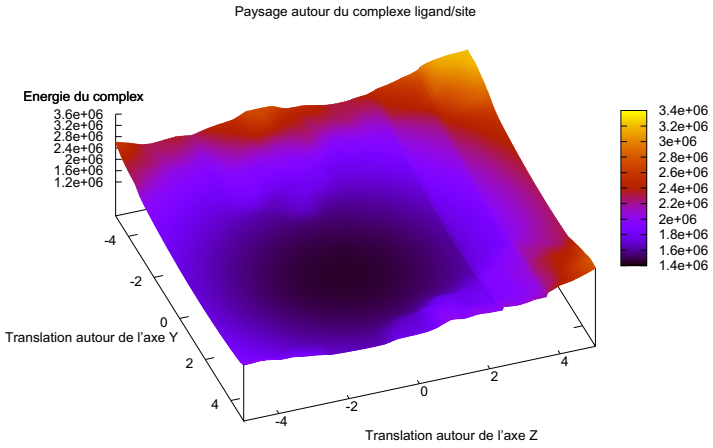


FIG. 3.24 – Paysage énergétique autour du complexe ligand/récepteur pour l'instance 6rsa. Le ligand a subi des translations selon les axes Y et Z.

Ces trois paysages à base de translations montre de larges vallées sans grande barrière énergétique. Seul celui de la figure 3.24 montre la solution optimale dans une zone de plus basse énergie (dans les limites de la zone générée).

3.4.4 Comparaison

A la vue des paysages générés selon le ICVFF ou la fonction d'évaluation d'Autodock. On peut remarquer que la fonction d'Autodock est plus sensible. Les paysages sont plus structurés alors que ceux avec le ICVFF sont plus lisses. La notion de robustesse pour un complexe parait valide selon les paysages obtenus avec le ICVFF. De plus, l'hypothèse selon laquelle le meilleur complexe n'est pas forcément celui de plus basse énergie paraît se vérifier sur les paysages obtenus selon la fonction

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

d'énergie d'Autodock. Concernant les deux types de paysage générés, par rotation ou translation, on peut affirmer que ce sont les rotations qui ont le plus d'impact d'un point de vue énergétique.

3.5 Résolution par coopération co-évolutionnaire de bas niveau : AG multi-objectif parallèle hybride

Nous allons détailler toutes les parties nécessaires à la mise en place d'un algorithme génétique multi-objectif. Nous insisterons tout particulièrement sur nos choix liés à la spécificité du problème.

3.5.1 Représentation d'une solution

3.5.1.1 Représentations existantes

Les représentations de solution pour les problèmes liés aux structures de protéines peuvent être classées selon deux grandes familles :

1. Modèles simples (ou simplifiés) : on retrouve principalement dans ce type de modèle les représentations par lattice [37] dont le plus connu est le HP-model. La figure 3.25 montre un exemple de représentation de protéine en HP-model. La protéine choisit une conformation 2D qui maximise le nombre de ponts hydrogènes. Ce type de modélisation était principalement utilisé pour faire de la prédiction de repliement de protein (« folding ») 2D mais aujourd'hui, il est aussi bien utilisé pour faire du folding 2D que 3D. Les modèles à base de lattice se différencient surtout par la forme de la lattice choisie car l'évaluation de ce type de modèle est généralement le même, il s'agit de calcul du nombre de pont hydrogène entre acides aminés hydrophobes.

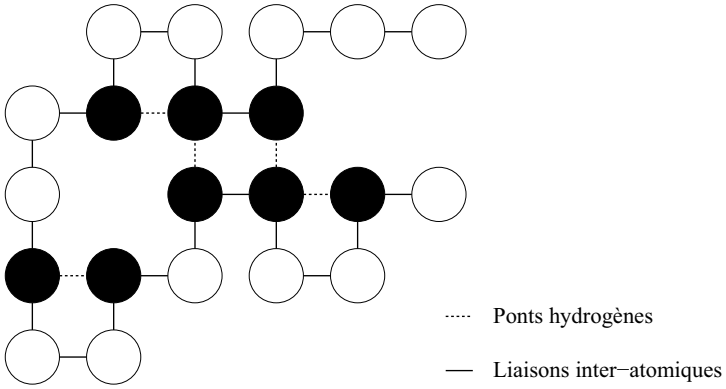


Fig. 3.25 – Exemple de HP-model pour le problème de structure de protéine 2D.

2. Modèles (plus) réalistes : à la différence des modèles à base de lattice qui se situent au niveau de l'acide aminé comme élément de base (qu'il soit hydrophobe ou hydrophile), les modèles plus complexes comme par exemple la modélisation par angle de torsion se situe au niveau atomique afin de prendre en compte toutes les énergies qui rentrent en jeu dans une molécule donnée. De par même son niveau de précision, les modélisations de ce type sont extrêment coûteuses ce qui fait que seule l'évaluation de protéines de petite taille est possible. Cependant, seuls ces types de modèle peuvent mettre en avant un respect fort des intéractions énergétiques réelles.

Comme nous travaillons sur des modèles énergétiques et géométriques basés sur des connaissances atomiques des molécules à évaluer, nous nous intéresserons aux représentations de la seconde famille.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Pour représenter des molécules, il est possible d'utiliser la représentation des positions des atomes dans l'espace. Cette représentation a l'avantage d'être simple et de ne pas être dédiée à un type de modèle donné. Avec celle-ci, aucune autre information extérieure n'est utile lorsque l'on veut appliquer des modifications simples comme des rotations ou translations. Par contre, cette représentation prend de la place en mémoire. Une autre représentation, plus économe en espace mémoire, est la représentation par angle de torsion. En effet, en ne sauvegardant que les angles de torsion d'une molécule, il est tout à fait possible d'en déduire la position des atomes. Cependant, ce type de représentation nécessite l'existence d'une conformation de référence à laquelle il faut appliquer les modifications d'angle de torsion pour déterminer les positions des atomes. Mais des modifications de rotation ou translation peuvent être ajoutées dans la représentation sous forme d'angle correspondant.

3.5.1.2 Représentation choisie

Dans notre cas, une solution, appelée un « Docking Complex » sera représentée selon deux vecteurs de flottants correspondant aux coordonnées atomiques du récepteur et du ligand. La figure 3.26 montre un exemple de ce codage.

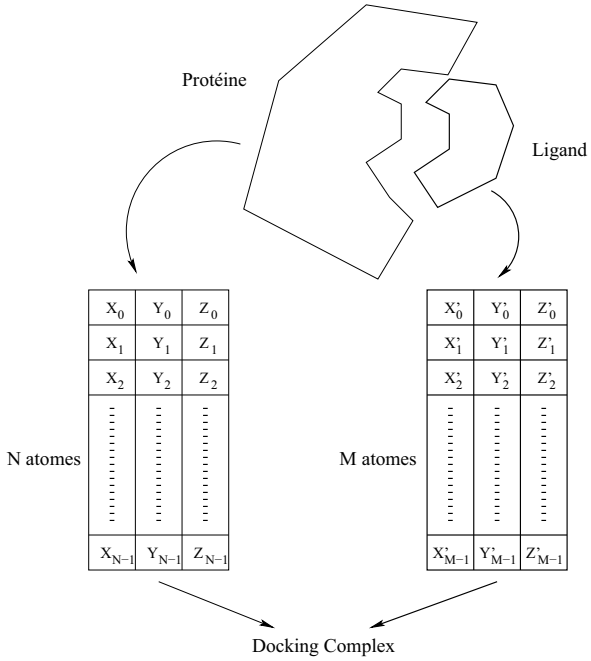


Fig. 3.26 – Représentation d'une solution dans l'algorithme génétique. N et M correspondent respectivement au nombre d'atomes composant le récepteur et le ligand.

Nous avons choisi cette représentation car même si elle prend de la place en mémoire, $3 * (N + M)*$ taille d'un nombre flottant avec N et M respectivement le nombre d'atomes du récepteur et du ligand, elle permet de ne pas être liée à une donnée extérieure pour être interprétée. Ainsi, la plupart des mécanismes de mutation et l'évaluation s'en trouvent ainsi facilités.

3.5.2 Opérateurs

Nous avons créé, pour notre algorithme génétique (AG), des opérateurs de recombinaison et de mutation adaptés à la représentation de notre problème.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.5.2.1 Recombinaison

Pour cet opérateur, nous avons privilégié un mécanisme ne nécessitant par de vérification de la viabilité des solutions obtenues. Ainsi, nous avons mis en place une recombinaison basée sur un échange de ligand entre deux récepteurs. Cela est représenté par la figure 3.27.

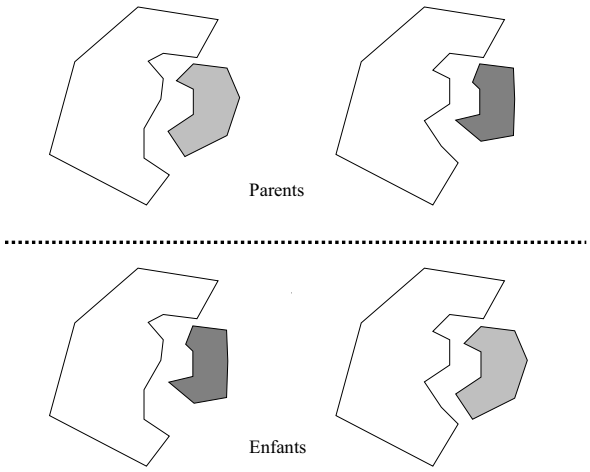


FIG. 3.27 – Opérateur de recombinaison de notre algorithme génétique.

Si à un moment donné, un ligand a subi une mutation qui a grandement amélioré la valeur du complexe associé, l'opérateur de recombinaison permettra de partager ce gain. Ce type de recombinaison assure d'obtenir des individus viables puisqu'on ne touche pas à leur conformation et ne nécessite pas par conséquent de mécanisme de réparation. Les risques de collisions sont aussi évités à cause du principe de Van der Waals qui assure une explosion de l'énergie si des atomes sont beaucoup trop proches.

3.5.2.2 Mutation

Nous avons mis en place différents types de mutation, certaines sont classiques et d'autres ont été spécifiquement mises en place pour ce problème.

Mutations standards Les mutations standards pour le problème de docking flexible sont au nombre de trois :

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

Ces trois premières mutations sont à la base d'un docking moléculaire flexible. Nous allons maintenant introduire de nouvelles mutations que nous avons mises en place pour des buts bien particuliers.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Mutations avancées Nous proposons trois nouvelles mutations qui ont chacune un but :

1. mutation « reverse » : cette mutation consiste à appliquer une rotation à 180°sur une deux conformations. Dans notre cas, il va s'agir du ligand. Le but de cette mutation est de permettre un redémarrage de la recherche soit dans le cas où une solution reste dans un optimum local et n'arrive plus à être améliorée, soit lorsque la solution est formée d'un ligand qui serait « rentré » dans le mauvais sens à l'intérieur du site ce qui par conséquent a un impact sur le RMSD final.
2. Rotations « big » and « small » : cette mutation correspond à une extension de la rotation de conformation. Le but est d'avoir des rotations qui font de grands mouvements et d'autres de plus petits mouvements. Dans la réalité, cela correspond à avoir des rotations de ligand dans l'intervalle]0° ;10°] pour les petites rotation et dans [30° ;60°] pour les grandes rotation. Cela permet d'avoir deux types de mutation qui n'auront pas la même utilité au même moment.
3. la mutation SMO : SMO signifie « **S**everal **M**utation in **O**ne time ». Il s'agit d'une mutation qui permet de faire plusieurs mutations en une seule fois. Elle est basée sur les limites de chaque mutation. Elle peut faire penser à la notion de voisinage variable. L'idée est de changer la forme du voisinage en permettant l'application de plusieurs mutations en une seule fois. L'intérêt est de tendre plus rapidement vers des solutions de meilleur compromis ou de s'extraire de puits énergétiques importants. Cependant, ce type de mutation, non contrôlée, peut aussi amener à des solutions non réalisables ou de très mauvaise qualité.

Hybridation Nous proposons deux mutations qui sont des recherches par descente. Le but est de faire tendre le complexe courant vers un complexe de plus basse énergie par rotation d'une torsion en particulier ou par rotation du ligand :

– Rotation du ligand : dans des fenêtres données, de nouveaux complexes sont générés par rotation du ligand. La taille de ces fenêtres (intervalles de valeurs pour les angles de rotation selon les axes x, y et z) a un impact sur le nombre de complexes générés. Cela représente un volume de taille donnée où toutes les rotations possibles du ligand dans un espace donné sont faites. Le but est de tendre vers le complexe de plus basse énergie le plus proche. Il est clair que pour des raisons de coût, l'intervalle entre les valeurs minimales et maximales d'angle de rotation est relativement restreint (quelques degrés avec un pas de l'ordre du dixième de degré).

– Rotation d'une torsion : le but est de trouver pour une torsion donnée, son meilleur angle pour minimiser l'énergie de la conformation courante. Cette mutation correspond à une intensification d'une partie de la conformation. Comme pour la rotation de ligand, cette mutation se base la génération d'un voisinage de taille donné selon l'intervalle de valeurs pour l'angle de rotation de la torsion. L'impact sur la conformation en son entier dépend de l'endroit où la torsion se trouve. Cela est schématisé par la figure 3.28 dans laquelle un exemple fictif de molécule à trois torsions est proposé. Quelle que soit la torsion choisie, c'est le plus petit morceau de la molécule qui est transformé durant la rotation de torsion ou la partie gauche (choix purement arbitraire ici pour l'exemple) quand les différents morceaux sont équivalents. Comme on peut le remarquer, la rotation des torsions T1 et T3 va avoir un impact moindre sur la conformation que la torsion T2. La modification d'énergie sera proportionnelle.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

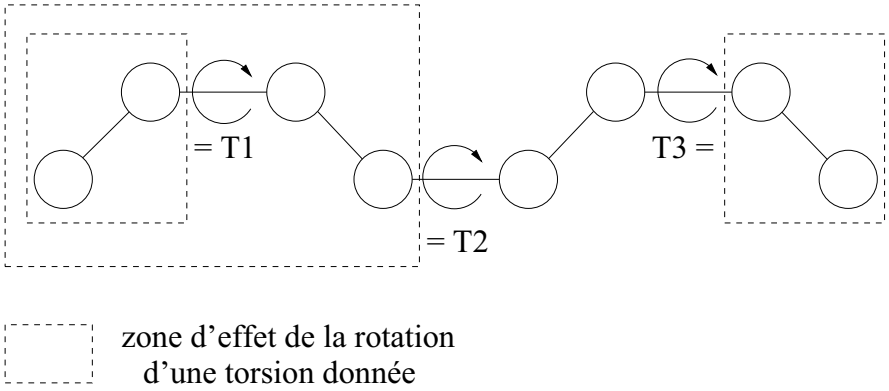


FIG. 3.28 – Impact de la rotation d'une torsion dans la modification de conformation d'une molécule donnée. Dans notre exemple, chaque rotation de torsion a un impact sur le plus petit morceau de la molécule (à gauche pour T1 et à droite pour T3) ou sur la partie gauche (T2)

L'utilisation de recherches par descente dans un algorithme génétique correspond à une coopération co-évolutionnaire de bas niveau comme cela était présenté dans la section 1.4.4.2. L'utilisation de recherches par descente va avoir un impact sur le principe de mutation de l'algorithme génétique. Ces recherches locales sont concentrées sur la minimisation de l'énergie de complexe quitte à dégrader les autres objectifs. Le but de cette mutation est d'essayer de pousser le complexe vers une conformation stable du point de vue énergétique. Ainsi dès qu'une grosse modification entraine une montée de l'énergie, ce type de mutation permet d'essayer de trouver le minimum local énergétique associé.

3.5.3 Evaluation, sélection et remplacement

Les stratégies d'évaluation, de sélection et de remplacement des solutions vont dépendre de l'algorithme génétique multi-objectif (AGMO) utilisé. Nos choix se portent sur des méthodes déjà éprouvées sur d'autres problèmes. Ainsi nous nous proposons d'utiliser NSGA-II et IBEA. Le premier est un des AGMO les plus connus et les plus utilisés, le second est considéré comme un des meilleurs AGMO à l'heure actuelle.

3.5.3.1 NSGA-II

Dans NSGA-II [36], les solutions de la population sont triées dans différentes classes à chaque génération. Les individus du premier front, c'est à dire les solutions non-dominées, appartiennent tous à la première classe. Les individus de second front, c'est à dire les solutions non-dominées lorsque les individus du premier front sont enlevés de la population, appartiennent à la seconde classe. Une fois que tous les individus appartiennent à une classe, deux valeurs sont calculées pour chacun d'entre eux. La première valeur correspond à la classe à laquelle l'individu appartient et par conséquent traduit la qualité de la solution en terme de convergence. La seconde évalue la densité de solutions entourant un point particulier dans l'espace des objectifs. Cela traduit la qualité d'une solution en terme de diversité. Ainsi, une solution est meilleure qu'une autre si elle appartient à une classe d'indice inférieure ou, dans le cas de deux solutions appartenant à la même classe, si elle a un meilleur indice de diversité. La stratégie de sélection consiste en un tournoi déterministe entre deux solutions choisies aléatoirement. A l'étape de remplacement, seuls les meilleurs individus survivent en accord avec la taille de la population. Cependant, une population extérieure, aussi appelée archive, contient à tout moment toutes les meilleures solutions découvertes durant l'évolution de l'algorithme.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Algorithme 8 : NSGA-II.

Données : Une taille de population « popMax » et un nombre maximum de « maxGen ».
Résultat : Une archive correspondant à l'approximation du front Pareto.

```
1  Population P de taille popMax;
2  Population A de taille 0;
3  Entier nbGenerations = 0;
4  tant que nbGenerations < maxGen faire
5      Calcul des différents fronts de solutions;
6      Assignement de son numéro de front à chaque individu;
7      Calcul de la diversité pour chaque individu;
8      Selection par tournoi binaire entre solutions choisies aléatoirement;
9      Application des opérateurs de recombinaison et de mutation;
10     Remplacement de la population avec les popMax meilleurs individus;
11     Mise à jour de l'archive A;
12     nbGeneration = nbGeneration + 1;
13 fin
14 retourner A;
```

L'algorithme 8 donne le pseudo code correspondant et la figure 3.29 montre un exemplev de la manière dont les solutions sont triées pour leur assigner leur numéro de classe.

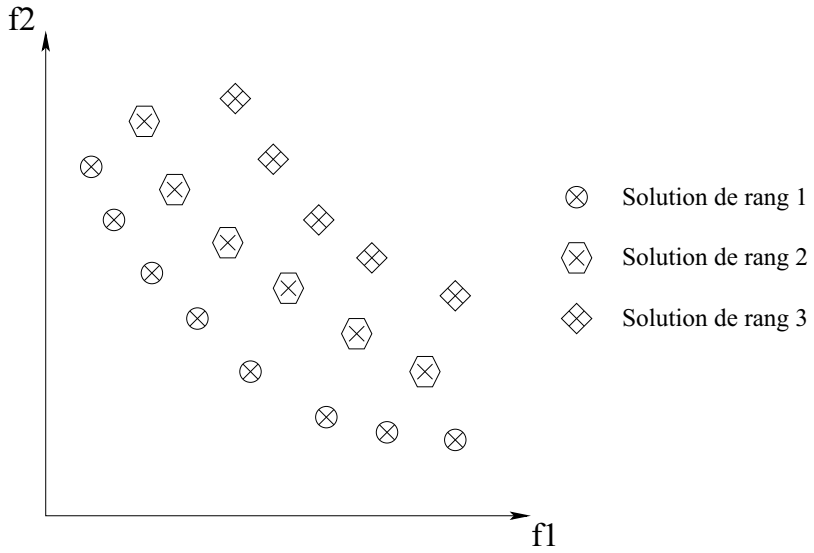


Fig. 3.29 – Exemple de classement des fronts comme fait dans NSGA-II en fonction de f1 et f2 deux fonctions objectif à minimiser.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.5.3.2 IBEA

Pour IBEA [140], le schéma d'affectation de la qualité d'une solution est basée sur une comparaison des solutions d'une population en utilisant un indicateur binaire de qualité. Selon l'indicateur utilisé, il n'y a pas de mécanisme de préservation de la diversité. La stratégie de sélection consiste en la suppression, un par un, des pires individus avec mise à jour de la qualité des solutions après chaque délétion. Cela est réitéré jusqu'à ce que la taille voulue pour la population soit atteinte. Une archive conserve les solutions non-dominées déjà trouvées afin de ne pas les perdre durant le processus stochastique de recherche. L'algorithme 9 donne le pseudo code correspondant.

Algorithme 9 : IBEA.

Données : Une taille de population « popMax », un nombre maximum de « maxGen » et un indicateur unaire de qualité « I_q ».
Résultat : Une archive correspondant à l'approximation du front Pareto.

```
1  Population P de taille popMax;
2  Population A de taille 0;
3  Entier nbGenerations = 0;
4  tant que nbGenerations < maxGen faire
5  |  Application de I_q sur chaque individu Selection par tournoi binaire entre solutions choisies
   |  aléatoirement;
6  |  Application des opérateurs de recombinaison et de mutation;
7  |  Remplacement de la population avec les popMax meilleurs individus;
8  |  Mise à jour de l'archive A;
9  |  nbGeneration = nbGeneration + 1;
10 fin
11 retourner A;
```

3.5.4 Critère d'arrêt

Le problème de docking possède un espace de recherche qui dépend d'un très grand nombre de critères : taille des instances, nombre de degrés de liberté, flexibilité du ligand et/ou du récepteur, ... Dans cette situation, il est difficile de fixer définitivement le nombre de générations maximum que l'algorithme génétique doit faire avant de s'arrêter. En optimisation mono-objective, un critère d'arrêt souvent utilisé est le nombre de générations sans amélioration de la meilleure solution de la population. Malheureusement, en optimisation multi-objective, la notion de meilleure solution n'est pas triviale. En se basant sur la notion de dominance, telle que définie dans la section 1.1.4, le critère d'arrêt correspond au nombre de générations sans obtention de nouvelles solutions non-dominées. Pour éviter une convergence trop précoce, un minimum de générations est imposé avant d'utiliser ce critère d'arrêt. Finalement, le critère d'arrêt de l'algorithme se résume à faire X générations et ensuite arrêter l'algorithme après avoir fait Y générations sans amélioration du front Pareto.

3.5.5 Parallélisation

Au vu du nombre exponentiel de conformations possibles pour les complexes ligand/récepteur et connaissant le coût des fonctions d'évaluation utilisée, nous avons décidé de mettre en place une parallélisation de notre algorithme. Grâce à ParadisEO-PEO, il est aisé de paralléliser un algorithme implémenté

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

sous ParadisEO-MOEO. Comme nous n'avons qu'un seul algorithme à lancer, nous allons utiliser un modèle maître/esclave c'est à dire qu'un processeur gère le fonctionnement global de l'algorithme génétique et demande au besoin de l'aide à d'autres processeurs pour les tâches couteuses. Dans notre cas, c'est l'évaluation des individus que nous confions aux autres processeurs. Cependant, vu que certains profils utilisent des recherches locales, nous avons aussi parallélisé l'application des opérateurs. Ainsi si des recherches locales sont utilisées, elles pourront être lancées en parallèle.

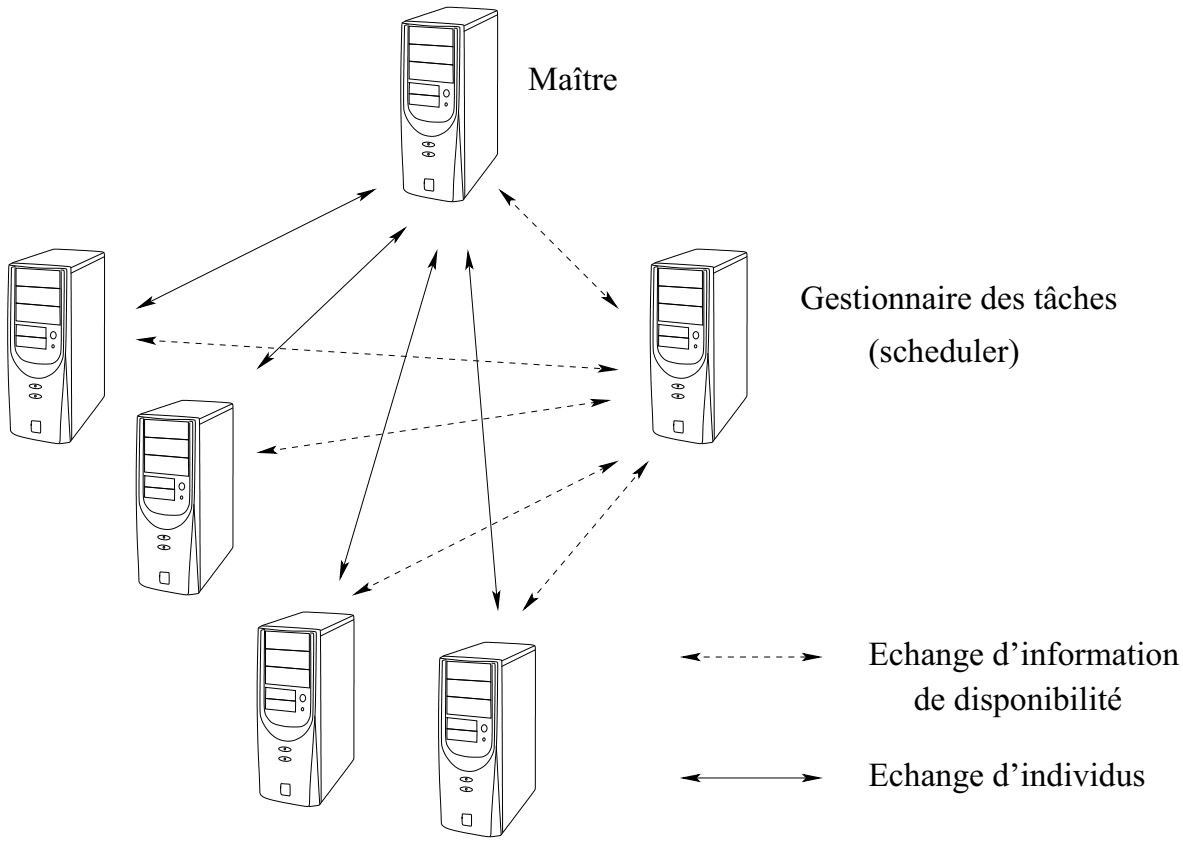


Fig. 3.30 – Paradigme maître/esclave sous ParadisEO. Exemple pour 4 esclaves (nécessite 6 machines).

Dans ParadisEO, utiliser le paradigme maître/esclave avec N esclaves nécessite en vérité $N + 2$ machine. En effet, une machine est utilisée comme maître et l'autre s'occupe de faire l'affectation des tâches sur les esclaves (figure 3.30)

3.5.6 Validation de l'approche

Après avoir présenté les données utilisées et avoir détaillé leur préparation, nous présenterons d'abord une comparaison entre NSGA-II et IBEA avant de d'analyser les résultats obtenus par les différents modèles multi-objectif que nous avons proposés.

3.5.6.1 Données

Afin de tester notre approche, nous avons utilisé des instances du benchmark CCDC/Astex. La version originale de ce benchmark a été présentée dans [59]. Il a été utilisé à l'origine pour évaluer le logiciel

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

GOLD. Les instances choisies proviennent de la liste dite « clean ». Cela correspond à 224 instances diversifiées considérées comme représentatives pour évaluer les méthodes de docking. Le tableau 3.2 décrit les instances utilisées. On retrouve notamment l'instance 6rsa déjà utilisée dans l'étude de paysage des complexes ligand/récepteur optimaux.

On pourra remarquer le faible nombre d'instances testées par rapport au nombre total d'instances disponible. Ce choix vient du nombre de modèles et de configurations possibles à tester pour chaque instance. De plus, chaque exécutation sera effectuée 10 fois.

Tab. 3.2 – Complexes ligand/protéines utilisés pour la validation. PDB signifie « Protein Data Bank » et identifie les instances, NbAtom correspond au nombre d'atomes et DegLib au nombre de degré de liberté. Pour le site, il s'agit du nombre de degrés de liberté lié à la partie considérée comme flexible.

Complexes ligand/récepteur	**PDB**	**NbAtom**		**DegLib**	
		Ligand	**Site**	**Ligand**	**Site**
Ribonuclease A / Uridine-2',3'-Vanadate	6rsa	31	1610	3	19
HIV-1 Protease / G26	1mbi	9	2539	0	22
Thymidilate / CB3	2tsc	56	2914	11	0
HIV-1 Protease / G26	1htf	79	3127	15	4
Glucoamylase-471 / Alpha-d-mannose	1dog	24	3362	5	6
IGA-kappa MCPC603 FAB / Phosphocholine	2mcp	25	1225	8	5

Pour faciliter la suite de la lecture, les complexes seront cités via leur identificateur PDB.

3.5.6.2 Protocole expérimental

Pour chaque instance utilisée, nous partons du complexe ligand/récepteur considéré comme optimal. A partir de celui-ci et grâce au logiciel Chimera [20], nous avons extrait le ligand de sa position optimale tout en le gardant face au site à une distance raisonnable pour éviter de biaiser le docking. Le tableau 3.3 indique pour chaque instance à quelle distance le ligand obtenu est situé par rapport à sa position optimale. La raison d'être de ce tableau est double : premièrement il permet d'évaluer la distance que le ligand a du parcourir avant d'arriver dans sa position finale dans le récetpeur. Deuxièment, il permet de comparer aussi les approches sur la question de la distance de départ entre le ligand et sa position cible. En effet, suivant les approches actuelles, le ligand peut partir d'une position éloignée (comme dans notre cas) ou d'une position proche (entre 3 et 5 Å).

Tab. 3.3 – RMSD entre le ligand « graine » et le ligand dans sa position cristallographique (selon la base CCDC-Astex). Les instances sont citées via leur identifiant PDB.

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

Ce ligand extrait est appelé ligand « graine ». En effet, lors de l'initialisation de la population de

[20] disponible sur www.cgl.ucsf.edu/chimera/

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

notre algorithme génétique, il sert de base pour générer un ensemble de complexes ligand/récepteur diversifiés. Un nombre fixé de mutations est appliqué sur le ligand « graine » pour obtenir chaque individu. Ces mutations sont de deux types : rotation de conformation dans lesquelles le ligand subit une rotation quelconque et rotation d'une torsion pour lesquelles une torsion choisie aléatoirement subit une rotation d'une amplitude aléatoire (dans une fenêtre donnée). Il n'y a pas de mutation de type translation car nous voulons que le ligand reste en face du site et qu'il ne puisse pas se retrouver dans une autre zone de l'espace à cause de mouvements non évalués. Grâce au ligand « graine » on obtient une population initiale de solutions pour notre approche.

A cause de leur taille, les récepteurs ont très souvent un très grand nombre de degrés de liberté et il n'est pas utile que l'ensemble du récepteur soit flexible. C'est pourquoi, d'après l'information de la conformation optimale, nous en avons déduit la zone du récepteur que l'on considère comme pouvant être flexible. Le reste des atomes est considéré comme fixe. La zone du récepteur considérée comme flexible correspond aux atomes du récepteur se situant à moins de 3 Å (par défaut) des atomes du ligand quand il est dans sa position flexible.

Selon que l'on utilise le ICVFF ou le champ de force d'Autodock 4.0, les données à fournir à l'algorithme se sont pas exactement les mêmes. Ainsi pour pouvoir utiliser la fonction d'énergie basée sur le ICVFF, il n'est pas nécessaire d'avoir d'autre information que le ligand et le site. Par contre, l'utilisation de la fonction d'énergie d'Autodock 4.0 demande des préparatifs supplémentaires et la génération d'un ensemble de fichiers essentiels à son fonctionnement. En effet, il faut calculer la grille dont la fonction d'énergie se sert pour faire son évaluation. Cette grille permet de faire tout un ensemble de pré-calcul qui permet des évaluations rapides des complexes ligand/récepteur. Pour nous aider à préparer les instances, nous avons utilisé les « Autodock Tools » (ADT) qui permettent via une interface graphique de préparer un docking avec Autodock. Cette outil graphique fait parti d'un paquetage nommé MGLTools [21] et a permis de générer tous les fichiers utiles à la fonction d'évaluation et notamment le site et le ligand sous le format utilisé par Autodock (pdbqt).

Ainsi, dans notre approche, toute instance X est décomposée de la manière suivante :

- un répertoire appelé « site » contenant le récepteur au format .mol2 ainsi que le fichier indiquant quels sont les atomes considérés comme fixes.
- un répertoire appelé « ligand » contenant le ligand « graine » au format .mol2.
- un répertoire appelé « optimal » contenant le ligand au format .mol2 dans sa position optimal au sein du récepteur. Il est clair que ce fichier n'est pas utilisé durant le docking en lui-même mais sert seulement à l'évaluation de la qualité des solutions finales obtenues en terme de RMSD.
- un répertoire appelé « autodock » contenant tout le nécessaire pour la fonction d'énergie d'Autodock : un fichier de configuration , des fichiers utilisés lors du calcul de la grille dans laquelle sont considérés être les molécules, le site et le ligand au format pdbqt et d'autres fichiers utilitaires concernant les types d'atomes présents.

Afin de facilement utiliser la fonction d'évaluation d'Autodock 4.0, j'ai transformé le fichier d'exécution en une bibliothèque proposant les méthodes standards d'une fonction d'évaluation (initialisation, évaluation, finalisation). L'arbre d'inclusion d'Autodock est visible en annexe C.5.

Remarque 3. *Pour des raisons purement techniques, dès que la fonction d'évaluation d'Autodock 4.0 sera utilisée, le docking sera semi-flexible et seul le ligand pourra voir sa conformation modifiée.*

[21] http ://mgltools.scripps.edu/

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.5.6.3 Configurations

Afin de mesurer l'impact des différents modèles et des différents opérateurs, nous avons mis en place 6 configurations différentes qui seront utilisées dans le cas rigide et le cas flexible. Par conséquent, 12 configurations ont été testées. Le tableau 3.4 résume ces différentes configurations. Il est à noter que la configuration P6_R existe même si dans un docking rigide, une mutation modifiant la conformation d'une des molécules n'est a priori pas possible. Le but de cette configuration est de tester si une modification intensive de conformation (mutation avancée avec recherche locale) a un plus grand impact qu'une simple rotation de torsion faite de temps de temps (mutation standard). Par conséquent, une comparaison entre P6 et P6_R permettra de répondre à cette question.

Tab. 3.4 – Différents profils de configurations utilisés. RT signifie rotation et translation standards, RotTors rotation de torsion, SBRot petite et grande rotation, SMO plusieurs mutations en une seule fois (parmi les précédentes), MDRL méthode par descente avec rotation du ligand, MDRT méthode par descente par rotation de torsion. A noter que P6_R existe à des fins comparatives.

Profil	**Docking**		**RT**	**RotTors**	**Reverse**	**SBRot**	**SMO**	**MDRL**	**MDRT**
	rigide	flexible							
P1_R	X		X						
P1		X	X	X					
P2_R	X		X		X				
P2		X	X	X	X				
P3_R	X		X			X			
P3		X	X	X		X			
P4_R	X		X				X		
P4		X	X	X			X		
P5_R	X		X					X	
P5		X	X	X				X	
P6_R	*X*		*X*						*X*
P6		X	X	X					X

3.5.6.4 Résultats

Les algorithmes génétiques multi-objectif ont été développés sur ParadisEO-MOEO [80] qui propose les squelettes d'AGs multi-objectif standards. Nos tests ont été réalisés sur un cluster de 64 processeurs sous forme de 8 noeuds de 4 bi-coeur Intel Xeon 3Ghz. Grâce à lui, tous les tests ont pu être réalisés en un peu moins de 2 mois d'exécution. En séquentiel, cela aurait pris plus de 9 mois de calcul.

Mesure de l'efficacité pour l'architecture maître/esclave Les résultats sont résumé dans le tableau 3.5. Le « speed-up » et l'efficacité sont calculés selon les formules suivantes :

$$S_N = \frac{Ts}{Tp}$$

$$Eff = \frac{S_n}{N}$$

Avec Ts le temps séquentiel d'exécution et Tp le temps d'exécution en parallèle avec N esclaves.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Nb Machines	ICVFF			Autodock 4.0		
	Temps	S_N	Eff	Temps	S_N	Eff
1	347	1	1	288	1	1
2	234	1,48	0,74	210	1,37	0,69
4	135	2,57	0,64	128	2,25	0,56
8	84	4,13	0,52	83	3,47	0,43
16	66	5,26	0,33	75	3,84	0,21
32	67	5,18	0,16	77	3,74	0,12
48	72	4,82	0,1	83	3,47	0,07
54	69	5,03	0,09	86	3,35	0,06

Tab. 3.5 – Temps d'exécution (en secondes), « speed-up » (S_N) et efficacité (Eff) sur un cluster pour pour les modèles M1 et M5 avec le profil P1_R selon nombre d'esclaves utilisés (Nb machines).

Comme on peut le constater, augmenter le nombre d'esclaves diminue le temps de calcul. Cependant, l'analyse de l'efficacité montre qu'un grande nombre d'esclave n'est pas forcément nécessaire. En effet, à partir de huit esclaves et plus, les temps de calcul sont a peu près les mêmes et par conséquent, l'efficacité de la parallélisation diminue. De part cette analyse, on se rend compte qu'il n'est pas nécessaire d'avoir autant d'esclave que d'individus dans notre population pour avoir un bon taux de parallélisme. Dans notre étude, un ratio de un esclave pour 10 individus environ paraît largement suffisant.

Comparaison de NSGA-II et de IBEA NSGA-II et IBEA ont été comparés en utilisant le modèle bi-objectif énergie moléculaire/surface avec le ICVFF (modèle M1) en utilisant les cinq premières instances (6rsa, 1mbi, 2tsc, 1htf et 1dog) car au moment de ces tests, l'instance 2mcp n'avait pas encore été préparée.

Pour chaque instance et chaque métaheuristique, un ensemble de dix expérimentations, avec différentes populations initiales, ont été menées. Afin d'évaluer la qualité des approximations des fronts non-dominés obtenus pour chaque instance, nous avons utilisé le protocole proposé dans [75]. Premièrement, un ensemble de référence $Z_N^\star$ des points non-dominés est calculé à partir de l'union de tous les fronts. Ensuite, on définit $z^{max} = (z_1^{max}, z_2^{max})$, où z_1^{max} (respectivement z_2^{max}) représente la borne supérieure du premier (respectivement du second) objectif dans l'ensemble des approximations des fronts non-dominés. Suit l'utilisation de la métrique hypervolume [141] pour évaluer la qualité d'un ensemble A comparé à $Z_N^\star$. La différence de l'indicateur hypervolume calcule la partie de l'espace des objectifs qui est faiblement dominé par $Z_N^\star$ et pas par A. L'indicateur ϵ ($I_{\epsilon+}^1$) est aussi utilisé pour donner la valeur minimal pour laquelle l'approximation A doit être décalée dans l'espace des objectifs pour faiblement dominer l'ensemble de référence $Z_N^\star$. Ainsi on obtient pour chaque instance dix différences d'hypervolume et dix mesures epsilon pour chaque algorithme. Knowles et al. [75] suggèrent d'utiliser au moins deux indicateurs, par conséquent nous avons utilisé l'hypervolume et ϵ défini précédemment. Une fois ces valeurs calculées, une analyse statistiques des méthodes d'optimisation est réalisée par pair sur chaque instance, c'est le test de Mann-Whitney avec une p-valeur de 5%. Il faut noter que cette analyse a été réalisée grâce à l'outil PISA [5] téléchargeable à l'adresse suivante : `http://www.tik.ee.ethz.ch/pisa/assessment.html`

Le tableau 3.6 montre que pour trois instances (6rsa, 1mbi et 1dog) IBEA est meilleur que NSGA-II sur les deux métriques. Pour une instance, 2tsc, IBEA est meilleur sur l'hypervolume mais pour l'ϵ on ne peut désigner lequel est le meilleur. Pour la dernière instance, 1htf, aucune déduction ne peut être faite pour comparer les deux algorithmes selon les deux critères utilisés. Au vu de ces résultats, on constate

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.6 – Comparaison de différentes métaheuristiques pour les métriques $I^1_{\epsilon+}$ et I^-_H métriques en utilisant le test statistique de Mann-Whitney statistical avec une p-valeur de 5%. Les valeurs $\succ$, $\prec$ et $\equiv$ indique si, selon les métriques utilisées, les résultats de l'algorithme d'une ligne sont meilleur, moins bons ou équivalenst à ceux de l'algorithme de la colonne correspondante.

		$I^1_{\epsilon+}$		I^-_h	
Instance	***algorithmes***	**IBEA**	**NSGA II**	**IBEA**	**NSGA II**
6rsc	IBEA	-	$\succ$	-	$\succ$
	NSGA II	$\prec$	-	$\prec$	-
1mbi	IBEA	-	$\succ$	-	$\succ$
	NSGAII	$\prec$	-	$\prec$	-
2tsc	IBEA	-	$\equiv$	-	$\succ$
	NSGAII	$\equiv$	-	$\prec$	-
1htf	IBEA	-	$\equiv$	-	$\equiv$
	NSGAII	$\equiv$	-	$\equiv$	-
1dog	IBEA	-	$\succ$	-	$\succ$
	NSGA II	$\prec$	-	$\prec$	-

que IBEA est meilleur que NSGA-II dans 70% des cas et que dans les 30% restant, on ne peut les départager. A aucun moment, NSGA-II a été prouvé meilleur, par conséquent, IBEA est meilleur que NSGA-II pour les instances du problème utilisées d'un point de vue multi-objectif. Reste maintenant à savoir si la qualité des solutions obtenues, dans notre cas leur RMSD, est aussi meilleure.

Tab. 3.7 – Meilleurs résultats pour chaque instance avec les algorithmes NSGA-II et IBEA. Pour chaque algorithme, la meilleure RMSD ainsi que la déviation standard associée sont données.

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

Le tableau 3.7 confirme les résultats selon lesquels IBEA serait meilleur que NSGA-II sur le problème du docking moléculaire pour les instances considérées. Pour chaque instance, IBEA amène à des RMSD meilleures ou au pires égales à celles de NSGA-II.

Au vu de ses résultats, nous avons décidé que IBEA qui serait utilisé par la suite pour tester nos différents modèles et profils de configuration.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Comparaison des modélisations Nous avons comparé les résultats des différentes instances selon les modèles et les profils utilisés.

Remarque 4. *Le modèle M2 n'apparaît pas dans les résultats car son comportement l'a rendu inutilisable pour faire du docking. N'ayant que des objectifs énergétiques et n'ayant pas de processus pour rapprocher les molécules ou, comme pour Autodock, les empêcher de trop s'éloigner ; ce modèle a entrainé une sortie du ligand du site quelque soit l'instance. Par conséquent, les RMSD associées étaient forcément extrêmement mauvaises. Nous avons donc par conséquent pris la décision de ne pas détailler les résultats associés à ce modèle. Cependant, dans un future proche, une délimitation de l'espace entourant les deux molécules sera mis en place afin d'éviter ce type de problème.*

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

TAB. 3.8 – Synthèse des résultats selon chaque modèle pour l'instance 6rsa.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	2,4	2,4	**1,5**	3,7	3,2	3,5	2,9
1er quartile	3,6	3	**2,6**	4	5,5	3,7	3,8
Médiane	3,9	4,6	**3,7**	4	6	4	4
Moyenne	4,008	4,754	**3,246**	4,046	5,846	4,192	4,146
3ème quartile	4,4	6,2	**3,8**	4,2	6,8	4,1	4,5
Maximum	6,1	7,4	4,5	**4,4**	7,4	5,8	5,9
Déviation standard	0,864	1,77	0,859	**0,203**	1,342	0,734	0,85

Par modèle Le tableau 3.8 présente, pour l'instance 6rsa, la distribution des RMSD selon chaque modèle. Les mêmes données sont aussi proposées sous forme de « boxplots » dans la figure 3.31

On remarque que le modèle M4 amène aux meilleures valeurs de RMSD comparé aux autres. Cependant, l'examen des « boxplot » tendrait à prouver que ce modèle n'est pas très robuste. Au contraire, les modèles M5 et M7 d'une part et M1 et M8 d'autre part semblent avoir un meilleur comportement pour cette instance même si les meilleures RMSD trouvées sont inférieures au modèle M4. Les modèles M3 et M6 quand à eux ne donnent pas de résultats satisfaisants et peuvent être classés derniers sur cette instance.

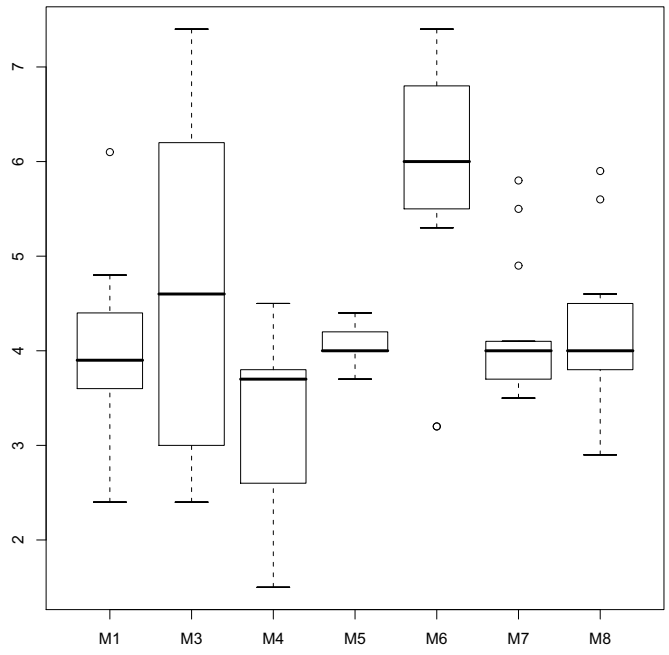


FIG. 3.31 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 6rsa.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

TAB. 3.9 – Synthèse des résultats selon chaque modèle pour l'instance 1mbi.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	3,7	2,4	2,5	2	1,9	**1,5**	1,9
1er quartile	3,8	3,725	3,525	**2,5**	2,125	2,425	2,225
Médiane	3,90	3,95	3,75	2,9	**2,25**	2,75	2,35
Moyenne	4,3	4,05	3,62	2,79	2,91	2,92	**2,3**
3ème quartile	4,25	4,4	3,9	3	2,85	3,125	**2,4**
Maximum	6,7	6	4	3,4	6,9	5,7	**2,5**
Déviation standard	0,927	0,916	0,454	0,458	1,536	1,095	**0,189**

La seconde instance analysée par modèle est celle de PDB 1mbi. Le tableau 3.9 et les « boxplots » de la figure 3.32 résume les résultats obtenus. D'après ces résultats, c'est le modèle M7 qui amène à la meilleure RMSD suivie par les modèles M6 et M8 ainsi que de M5. A la vue des « boxplots », on remarque qu'il y a bien une césure entre les modèles M1, M3 et M4 relativement stable mais de RMSD moins bonne et les modèles M5, M6, M7 et M8 de meilleure RMSD. On remarquera notamment la bonne stabilité du modèle M8.

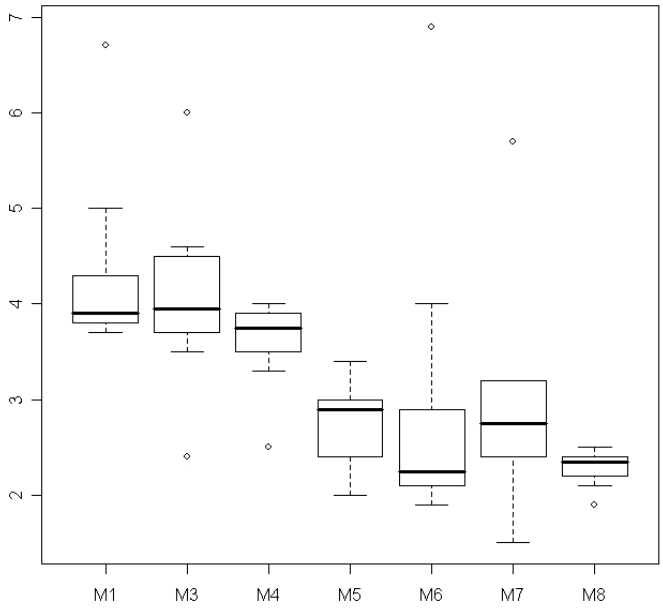


FIG. 3.32 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 1mbi.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.10 – Synthèse des résultats selon chaque modèle pour l'instance 2tsc.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	5,7	6	5,8	4	4,4	**3,6**	**3.6**
1er quartile	6,1	6	5,85	6,85	9,85	**4,4**	4,65
Médiane	7	6,3	6	7	10	**4,8**	6,8
Moyenne	6,882	6,536	6,082	6,591	9,927	**5,673**	5,964
3ème quartile	7,55	7,05	**6,2**	7,15	11,55	6,9	7,25
Maximum	8,6	8	**7**	7,6	12,6	10,5	7,7
Déviation standard	0.899	0.686	**0.343**	1.163	2.415	2.003	1.499

Le tableau 3.10 et la figure 3.33 nous donnent les résultats de l'instance 2tsc. La meilleure RMSD a été trouvée avec les modèles M7 et M8. Cependant, à l'examen des « boxplots », ils ne paraissent pas être les modèles les plus stables sur cette instance (ce serait même le contraire). Les modèles M4 et M5 sont les modèles les plus stables même si les RMSD associées sont moins bonnes. Par conséquent, on peut classer les modèles de la manière suivante : M7 et M8 de par les valeurs de RMSD, M4 et M5 de par leur stabilité suivi de M1 et M3 qui sont moins stable mais de meilleure RMSD équivalent. Enfin M6, comme pour l'instance 6rsa, se retrouve être le dernier modèle.

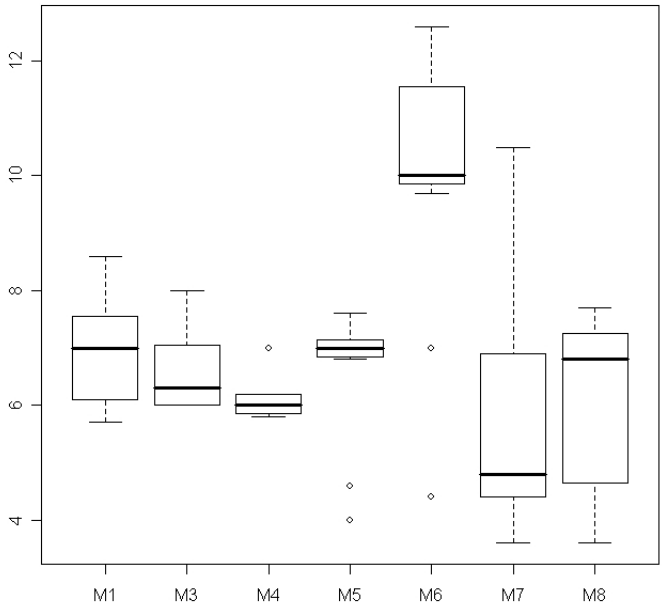


Fig. 3.33 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 2tsc.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

TAB. 3.11 – Synthèse des résultats selon chaque modèle pour l'instance 1htf.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	6,7	6,1	6,2	**3,1**	4,6	4	4,1
1er quartile	6,80	6,85	7	4,325	7,8	**4,425**	4,525
Médiane	7	7,85	7	**4,5**	10,8	4,85	4,9
Moyenne	7,84	7,92	7,07	4,610	9,62	5,850	**5,490**
3ème quartile	9	8,8	7,25	4,675	11,75	5,95	**5,750**
Maximum	10,5	10,5	7,9	**6**	13,1	13,8	8,5
Déviation standard	1,423	1,37	**0,452**	0,839	3,008	2,889	1,553

Les résultats par modèle concernant l'instance 1htf sont présentés dans le tableau 3.11 et la figure 3.34. Le modèle M5 permet d'obtenir la meilleure RMSD mais globalement, les modèles M7 et M8 sont aussi de bonne qualité même s'ils sont moins stable. Les modèles M1, M3 et M4 sont, de par la valeur de leur RMSD, en retrait par rapport aux modèles M7 et M8. Enfin, le modèle M6, comme pour les instances 6rsa et 2tsc, est le modèle le moins robuste même s'il amène une meilleure RMSD comparable aux modèles M5, M7 et M8.

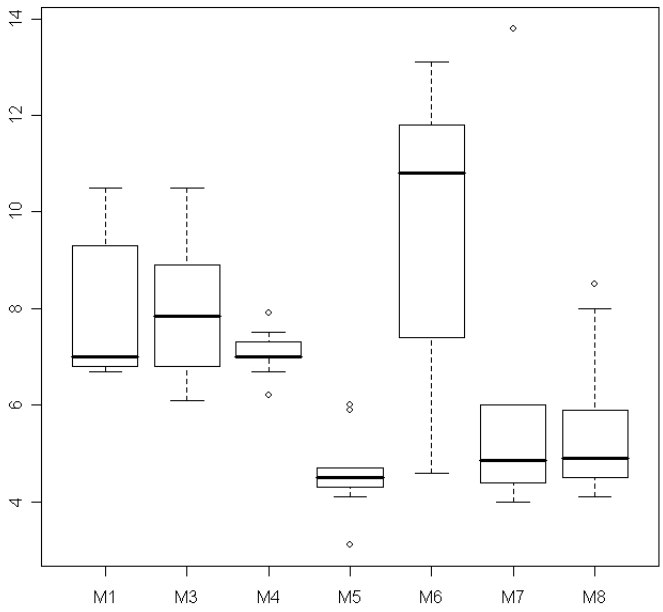


FIG. 3.34 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 1htf.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.12 – Synthèse des résultats selon chaque modèle pour l'instance 1dog.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	2,7	**1,9**	2,5	3,1	2,6	3,2	3,3
1er quartile	2,85	**2,5**	2,55	3,55	4,55	3,5	3,65
Médiane	3	3,8	**2,6**	4	7	3,9	3,9
Moyenne	3,827	3,891	**2,645**	4,1	6,555	4,318	3,827
3ème quartile	4,55	5,15	**2,7**	4,55	8,25	4,25	4,05
Maximum	6,2	6,1	**2,9**	5,4	11,2	9,3	4,2
Déviation standard	1,348	1,6	**0,144**	0,744	2,593	1,714	0,287

Les résultats de l'instance 1dog donnés par le tableau 3.12 et la figure 3.35 amènent le modèle M3 en tête selon la meilleure RMSD. Sur cette instance, le modèle le plus stable s'avère être le modèle M4 suivi du modèle M8. Ensuite, les modèles M5 et M7 sont de robustesse équivalente même si leur RMSD est moins bonne que celle des modèles M1 et M3 qui eux sont moins stables. Le modèle M6 se révèle être le plus instable, capable du meilleur comme du pire.

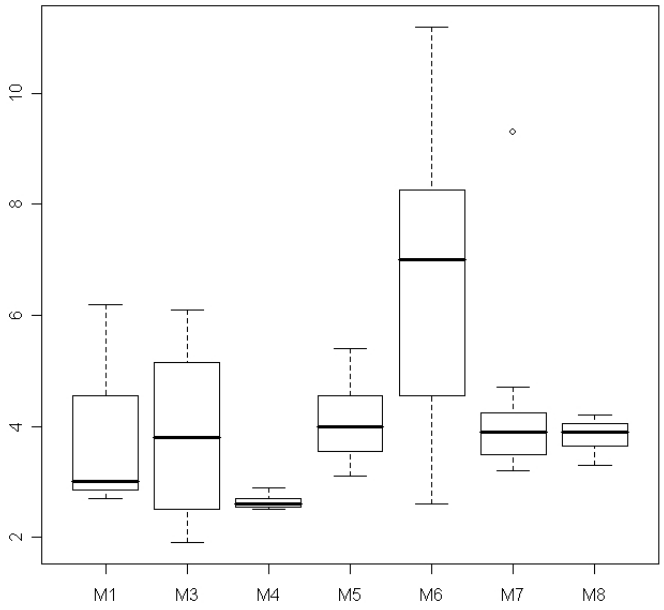


Fig. 3.35 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 1dog.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.13 – Synthèse des résultats selon chaque modèle pour l'instance 2mcp.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	2,7	**2**	3	2,5	2,2	2,6	2,1
1er quartile	3,825	4,175	3,45	2,6	2,7	2,775	**2,175**
Médiane	4,45	4,35	3,6	2,75	2,85	2,95	**2,4**
Moyenne	4,312	4,275	3,6	2,938	3,475	2,95	**2,475**
3ème quartile	4,9	4,65	3,75	3,075	3,375	3,050	**2,7**
Maximum	5,6	5,6	4,2	4	7	3,4	**3**
Déviation standard	0,963	1,05	0,363	0,501	1,573	**0,262**	0,37

Par l'examen des résultats de l'instance 2mcp à l'aide du tableau 3.13 et de la figure 3.36, on remarque que même si le modèle M3 a la meilleure RMSD, c'est le modèle M8 qui se trouve être le meilleur modèle sur cette instance. Suivent ensuite les modèles M5, M6 et M7 relativement robuste et enfin les modèles M1, M2 et M3.

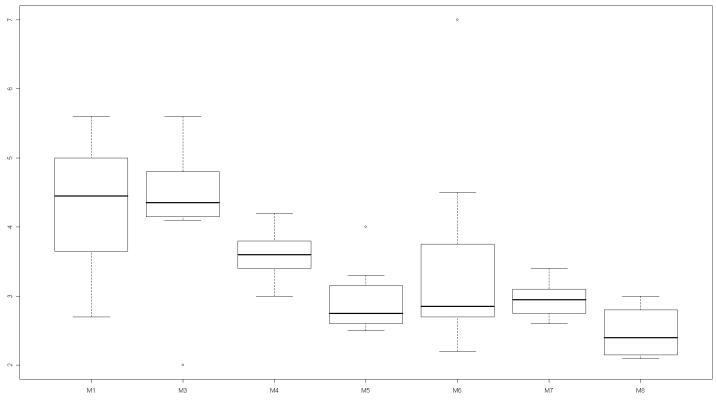


Fig. 3.36 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle pour l'instance 2mcp.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.14 – Synthèse des résultats selon chaque modèle sur l'ensemble des instances.

	M1	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Minimum	2,41	1,9	1,5	2,2	1,9	**1,5**	1,9
1er quartile	3,475	3,8	3,3	3,275	3,65	3,15	**2,445**
Médiane	5	5,1	4,75	4	6,35	3,95	**3,7**
Moyenne	5,315	5,335	5,022	4,051	6,748	4,255	**3,706**
3ème quartile	6,8	6,525	6,2	**4,325**	10	4,625	4,45
Maximum	11,01	11	11	**7,3**	14	13,8	7,5
Déviation standard	2,291	2,112	2,129	1,326	3,494	1,925	**1,267**

Le tableau 3.14 et la figure 3.37 proposent les résultats selon les modèles toutes instances confondues. Selon ce tableau, les modèle M4 et M7 obtiennent la meilleure RMSD mais l'examen des « boxplots » montre que les meilleurs modèles seraient les modèles M5, M7 et M8. Ils possèdent en effet de bonnes valeurs de RMSD et une robustesse relativement bonne. Les modèles M1, M2 et M3 sont globalement moins bons. Enfin, le modèle M6, comme pour quatre des six instances, se révèle être le moins bon modèle notamment à cause de sa mauvaise stabilité.

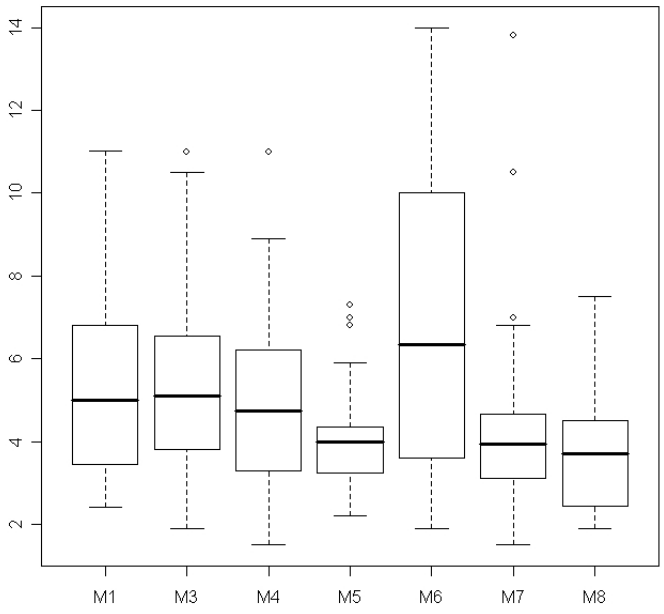


Fig. 3.37 – Synthèse graphique des résultats (« boxplot ») selon chaque modèle sur l'ensemble des instances.

Nous allons maintenant faire les mêmes analyses mais selon les configurations d'algorithme (profils) pour voir celles qui se détachent des autres.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

TAB. 3.15 – Synthèse des résultats selon chaque profil pour l'instance 6rsa.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	2,7	2,4	2,3	3,7	3,4	2,6
1er quartile	**3**,125	3,425	3,6	3,975	3,875	3,675
Médiane	**3,75**	3,9	4,1	4,85	4,4	4
Moyenne	3,950	**3,712**	4,125	5,062	4,612	4,438
3ème quartile	4,525	4,05	4,65	6,125	5,5	5,725
Maximum	6	**4,6**	5,9	6,7	6,2	6,1
Déviation standard	1,11	**0,677**	1,258	1,1987345	1,0412047	1,2659694
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	3,7	3	3,7	3,9	**1,5**	2,4
1er quartile	3,7	**3,075**	3,775	3,975	4,6	3,925
Médiane	3,9	**3,75**	4	4,2	4,95	4,1
Moyenne	4,388	3,725	4,675	4,912	4,85	4,1
3ème quartile	4,75	**3,925**	5,55	6	5,7	4,25
Maximum	6	5,3	7	6,7	6,8	5,7
Déviation standard	0,964	0,759	1,237	1,239	1,57	0,924

Par profil Le tableau 3.15 et la figure 3.38 montre les résultats par profil pour l'instance 6rsa. D'après le tableau, c'est le profil P6 qui donne la meilleure RMSD. Cependant quand on examine les « boxplots » associés, on se rend compte que les profils ayant le meilleur couple RMSD/robustesse sont les profils P1_R et P4_R. Le profil le plus robuste est P6_R.

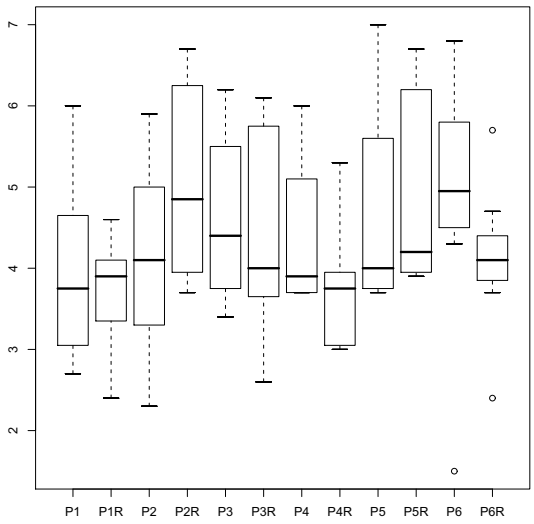


FIG. 3.38 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 6rsa.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.16 – Synthèse des résultats selon chaque profil pour l'instance 1mbi.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	2,1	2,1	2,3	**1,5**	2,2	2
1er quartile	2,375	2,35	2,725	3	2,45	2,925
Médiane	2,65	2,75	3,4	4	2,75	3,65
Moyenne	2,7	3,075	3,275	3,975	2,862	3,337
3ème quartile	**2,825**	3,850	3,625	4,875	3,325	3,825
Maximum	3,8	4,6	4,5	6,2	3,7	4
Déviation standard	0,518	0,942	0,719	1,627	0,5629958	0,7089983
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	2	1,9	2	2,2	2,1	2,6
1er quartile	2,275	**2,075**	2,3	2,725	2,675	3,1
Médiane	2,5	**2,3**	2,45	3,35	3	3,85
Moyenne	3,2	**2,513**	2,562	3,975	2,975	4,325
3ème quartile	3,325	**2,825**	2,85	4,725	3,425	5,1
Maximum	7	3,8	**3,2**	6,9	3,5	6,9
Déviation standard	1,662	0,633	**0,403**	1,918	0,506	1,738

Pour l'instance 1mbi, le tableau 3.16 et la figure 3.39 résume les résultats par profils. La meilleure RMSD est obtenue avec le profil P2_R mais sa mauvaise stabilité a pour conséquence que les meilleurs profils sont P1 et P5 de par leur RMSD et leur robustesse. Il est intéressant de remarquer que pour la plupart des profils pour cette instance, la prise en compte de la flexibilité qui amène des résultats plus stables.

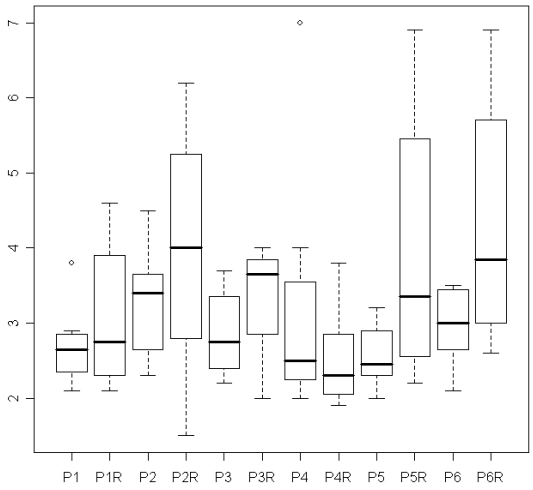


Fig. 3.39 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 1mbi.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.17 – Synthèse des résultats selon chaque profil pour l'instance 2tsc.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	**3,5**	4,3	4,5	3,5	5,7	6
1er quartile	**4,7**	5,3	6	5,7	5,9	6,4
Médiane	**5,9**	7	6,3	7,2	6	7
Moyenne	**5,371**	6,30	7,286	6,9	6,829	7,414
3ème quartile	**6**	7,25	8,8	8,1	7,15	7,55
Maximum	**6,8**	7,7	10,6	10	10	11
Déviation standard	**1,12**	1,414	2,359	2,234	1,528	1,7054
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	5,8	6,2	7	7,5	4,6	4
1er quartile	6,5	6,75	7,1	7,7	5,05	8,6
Médiane	7	7	7,7	8,3	6	9
Moyenne	7,871	7,386	8,471	9,286	5,843	9,029
3ème quartile	8,35	7,5	8,85	10,3	6,1	10,3
Maximum	12,6	10	12.700	13,2	8	12,4
Déviation standard	1,705	1,242	2,135	2,347	1,137	2,632

D'après le tableau 3.17 et la figure 3.40 qui résument les résultats pour l'instance 1mbi, c'est le profil P1 qui amène le meilleur RMSD. Avec le profil P6, ce sont les profils de meilleure qualité. Une nouvelle fois, on peut remarquer que les profils flexibles sont plus stables que les modèles rigides dans la plupart des cas (quatre sur six).

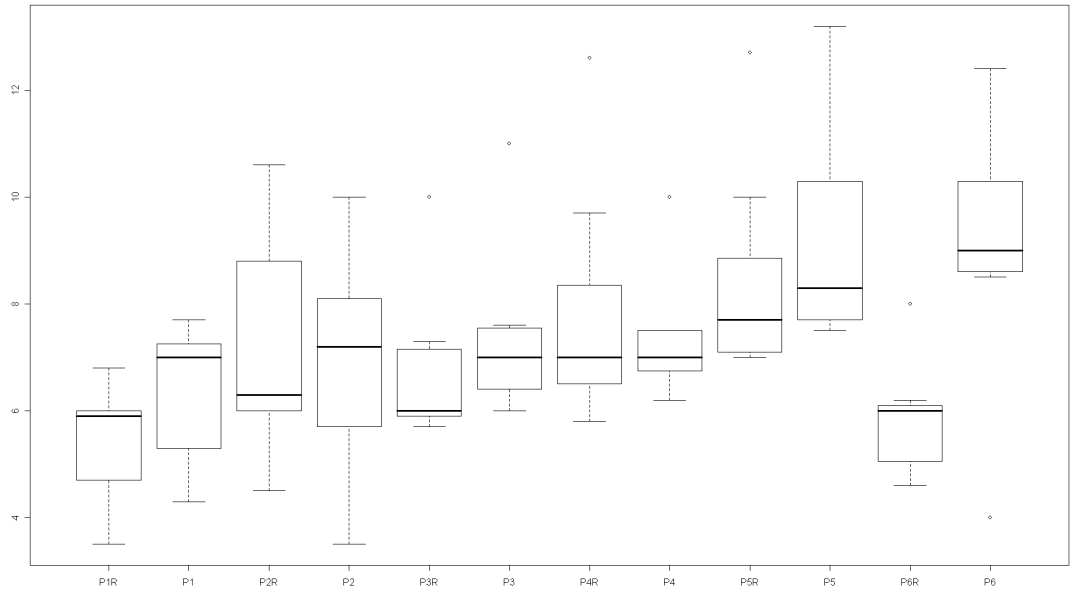


Fig. 3.40 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 2tsc.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

TAB. 3.18 – Synthèse des résultats selon chaque profil pour l'instance 1htf.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	6	**3,1**	5,8	4,6	4,2	4,1
1er quartile	6,85	**4,25**	6,3	6,5	6,95	5,6
Médiane	7,1	6,1	6,9	7,1	7,4	6,7
Moyenne	7,40	6,357	7,7	8,557	7,329	6,614
3ème quartile	7,80	8,150	8,5	10,700	7,900	7,650
Maximum	9,4	10,5	11,6	13,8	10	9
Déviation standard	**1,112**	2,766	2,04	3,243	1,733	1,802
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	5,3	4,3	4,1	4,4	4,6	5,1
1er quartile	5,9	4,5	5,1	4,5	6,35	5,85
Médiane	6	4,7	5,3	**4,6**	7	5,9
Moyenne	6,729	6,216	**6,186**	5,814	8,643	6,6
3ème quartile	**6**	6,2	5,95	5,35	9,05	7,3
Maximum	12	13,11	11,8	12	11,1	**8,9**
Déviation standard	2.338	3.18	2.554	2.762	2.309	1.473

Les résultats de l'instance 1htf sont présentés au travers du tableau 3.18 et de la figure 3.41. Le profil avec le meilleur RMSD est P1_R mais sa mauvaise stabilité impose le choix de P5_R comme meilleur profil pour cette instance. A la différence des autres instances, les profils rigides sont ici plus table que les profils flexibles.

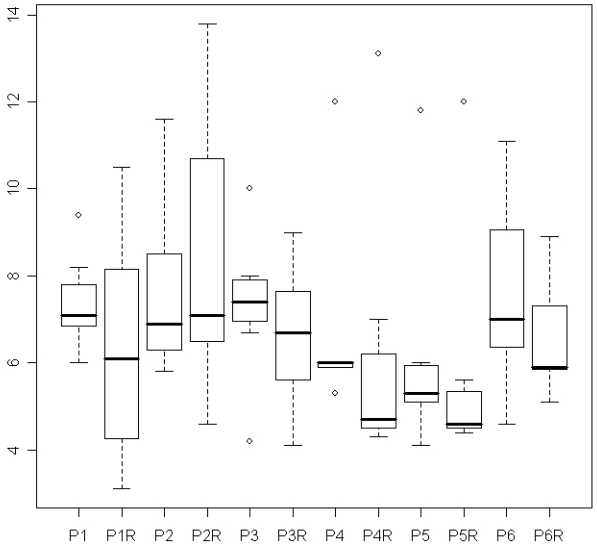


FIG. 3.41 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 1htf.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.19 – Synthèse des résultats selon chaque profil pour l'instance 1dog.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	3,5	2,6	2,7	2,5	2,8	2,6
1er quartile	3,6	2,9	3,375	**2,6**	3,175	2,975
Médiane	3,8	**3,4**	3,7	3,4	4,65	4,05
Moyenne	3,737	3,638	4,7	4,287	5,325	4,025
3ème quartile	3,825	3,95	5,025	4,625	6,675	4,775
Maximum	**3,9**	6,1	8,7	9,3	9,7	5,9
Déviation standard	**0.151**	1.146	2.32	2.422	2.669	1.151
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	3,5	4	3,6	4	**1,9**	2,4
1er quartile	3,675	4,325	3,7	4,15	2,85	3,025
Médiane	3,75	4,8	3,8	4,75	4,45	**3,4**
Moyenne	4,287	4,812	4,7	5,287	4,875	**3,5**
3ème quartile	3,8	5,325	4,775	5,5	6,25	**3,5**
Maximum	8,4	5,7	7,7	9,2	10	6,1
Déviation standard	1,665	0,624	1,763	1,734	2,739	1,139

Pour l'instance 1dog dont les résultats sont donnés dans le tableau 3.19 et la figure 3.42. Le profil amenant la meilleure RMSD est P6. On peut noter qu'il a une robustesse correcte même si les profils P1 et P4 se revèlent plus stables.

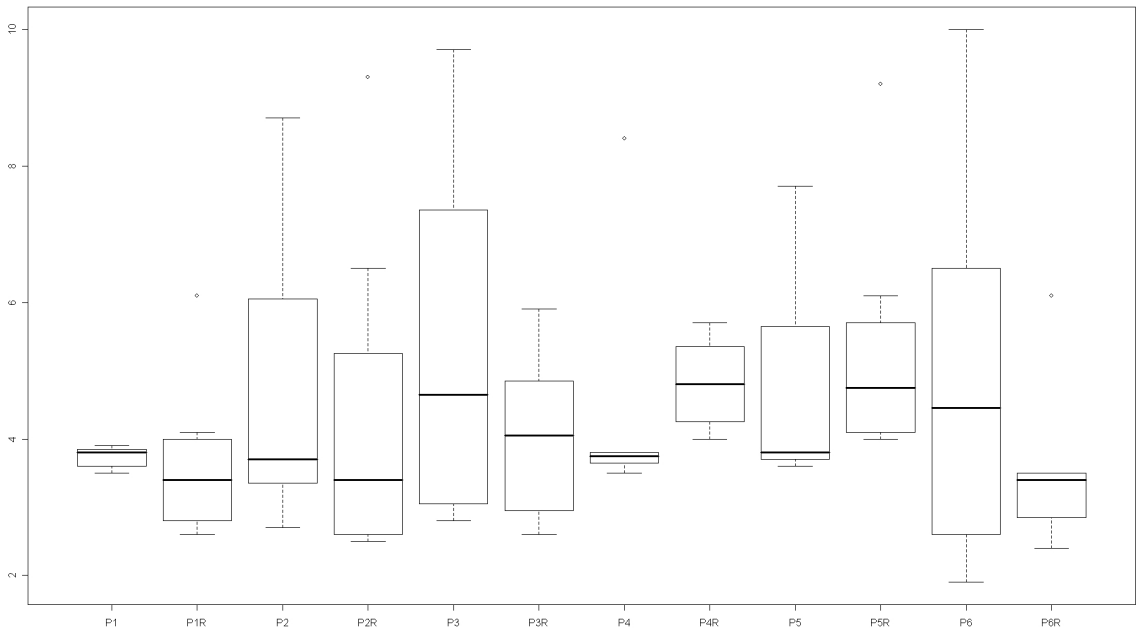


Fig. 3.42 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 1dog.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.20 – Synthèse des résultats selon chaque profil pour l'instance 2mcp.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	**2**	2,6	3,0	2,9	2,7	2,1
1er quartile	**2,1**	3,8	3	3	2,8	2,3
Médiane	4,2	4,1	3,1	3	3,2	2,8
Moyenne	3,64	4,08	3,1	3,12	3,2	**3,1**
3ème quartile	4,8	4,3	3,2	3,3	3,3	3,1
Maximum	5,1	5.60	3.2	3.40	4.0	5.2
Déviation standard	1,488	1,076	0,1	**0,217**	0,515	1,239
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	2,6	2,2	2,6	2,2	2,9	2,7
1er quartile	2,7	2,2	2,7	2,3	2,9	3,1
Médiane	2,7	2,5	3,3	**2,4**	3,4	3,4
Moyenne	3,18	2,94	3,4	2,58	3,7	3,5
3ème quartile	2,9	3,5	3,6	**3**	3,7	3,6
Maximum	5	4,3	4,8	**3**	5,6	4,7
Déviation standard	1,023	0,929	0,886	0,39	1,116	0,752

Le tableau 3.20 et la figure 3.43 détaillent les résultats pour l'instance 2mcp. Le profil avec la meilleure RMSD et le P1_R mais sa stabilité est vraiment très mauvaise, c'est le profil P4_R qui propose le meilleure ratio RMSD/stabilité.

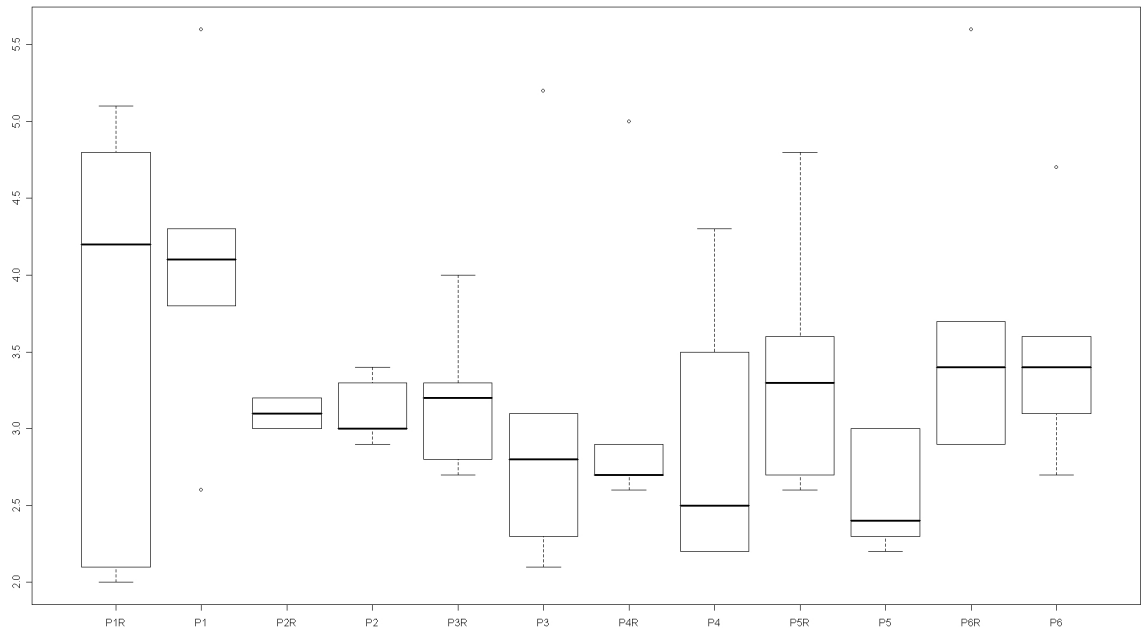


Fig. 3.43 – Synthèse graphique des résultats (« boxplot ») selon chaque profil pour l'instance 2mcp.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Tab. 3.21 – Synthèse des résultats pour chaque profil sur l'ensemble des instances.

	P1	**P1_R**	**P2**	**P2_R**	**P3**	**P3_R**
Minimum	2	2,1	2,4	2,3	2	2,1
1er quartile	3,2	3,4	3,3	3,425	3,4	3,225
Médiane	**4,05**	4	4,5	5,75	**4,1**	**4**
Moyenne	**4,343**	4,4	4,729	5,293	5,126	4,686
3ème quartile	5,075	4,875	6	6,75	6,3	5,875
Maximum	10,5	**7,7**	10,8	10	11	11
Déviation standard	1,651	**1,556**	1,937	2,078	2,524	2,183
	P4	**P4_R**	**P5**	**P5_R**	**P6**	**P6_R**
Minimum	1,9	2	2,2	1,9	2,4	**1,5**
1er quartile	3,3	3,5	3,825	3,6	3,25	**3,1**
Médiane	4,35	**4,1**	**4,1**	4,05	4,6	4,15
Moyenne	5,169	5,129	4,733	4,964	4,776	4,686
3ème quartile	6,325	6,15	**4,775**	7	6	6,8
Maximum	12,6	12	14	11,8	8,5	11,1
Déviation standard	2,865	2,576	2,29	2,42	1,794	2,524

La synthèse de toutes les instances par profil est donnée par le tableau 3.21 et la figure 3.44. Selon eux, c'est le modèle P6_R qui donne la meilleure RMSD. De ce fait, on remarquera que la comparaison P6/P6_R tend en faveur du second. Cette constatation renforce le principe qu'une simple modification de conformation faite régulièrement a moins d'impact qu'une vraie intensification pour amener aux meilleurs conformations rapidement (en dépit du coût associé).

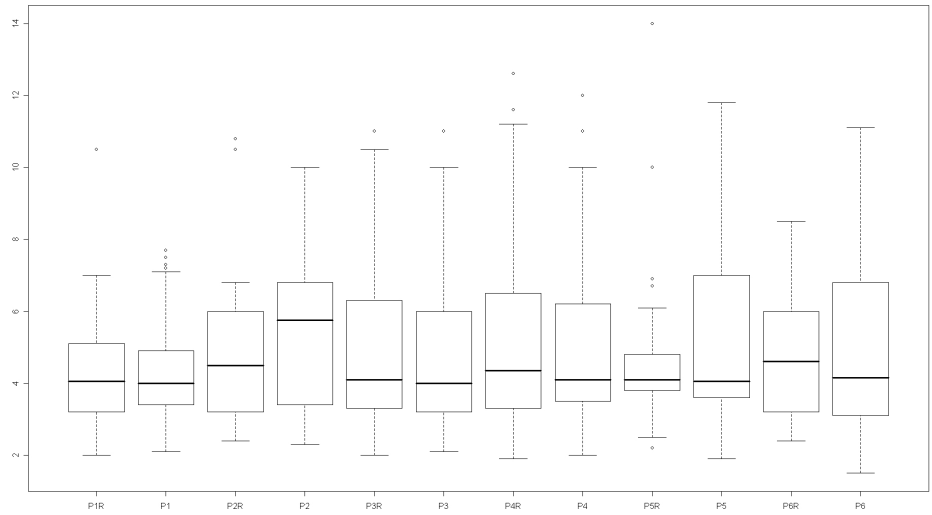


Fig. 3.44 – Synthèse graphique des résultats (« boxplot ») selon chaque profil sur l'ensemble des instances.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Synthèse L'examen des résultats suivant les modèles, les profils et les instances a mis en évidence quatre modèles : M4, M5, M7 et M8. Ce résultat prouve que le critère de surface a un impact bénéfique sur les résultats. De la même manière, le critère de robustesse a été retenu ce qui valide une nouvelle fois l'hypothèse que le complexe ligand/récepteur de meilleure qualité se trouve dans une vallée énergétique et non forcément dans un puit comme cela peut être le cas dans le problème de prédiction de structure de protéines. De plus, trois de ces modèles sépare l'énergie entre intéractions entre atomes liés et atomes non liés. Cela tend à prouver que la séparation de ces intéractions est peut-être plus efficace que la prise en compte de l'énergie sous forme d'une seule aggrégation de termes.

Concernant les profils, ce sont les profils P1, P5 et P6 qui proposent les meilleurs couple RMSD/stabilité. Par conséquent, la gestion de la flexibilité mais aussi l'utilisation de recherche locale amène à de meilleures résultats que les configurations classiques rigides.

Enfin, les figures 3.45 et 3.46 d'une part et les figures 3.47 et 3.48 d'autre part montrent des exemples comparatifs complexes optimal/meilleur complexe pour les instances 6rsa et 2mcp.

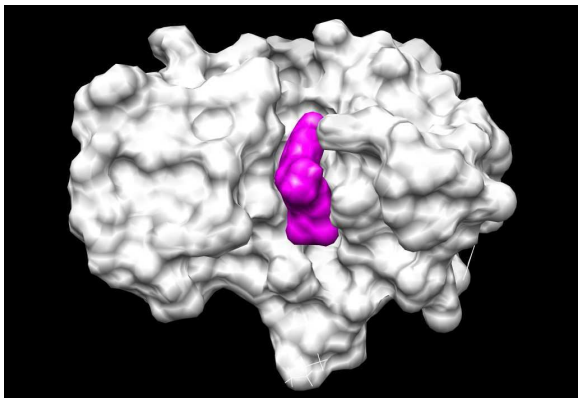


Fig. 3.45 – Complexe optimal (6rsa).

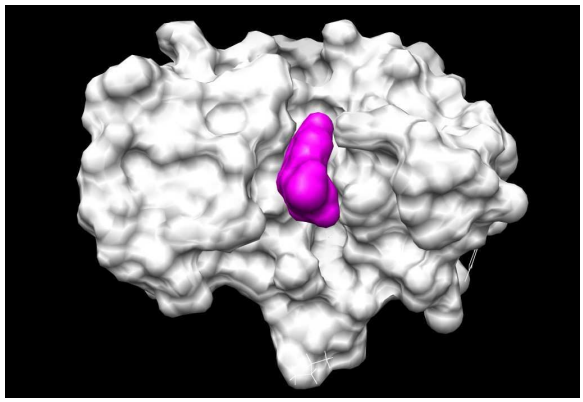


Fig. 3.46 – Meilleur complexe trouvé (6rsa).

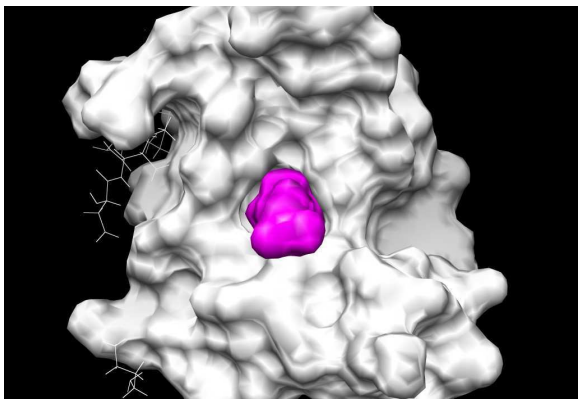


Fig. 3.47 – Complexe optimal (2mcp).

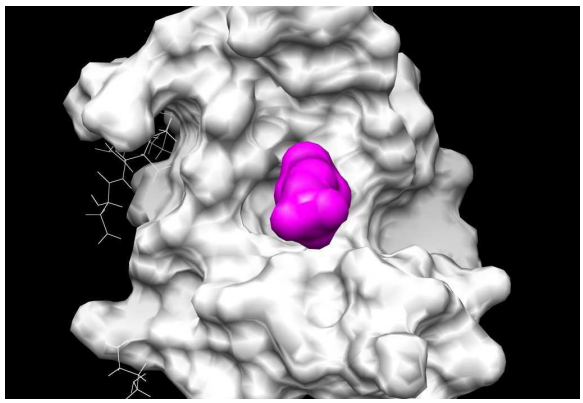


Fig. 3.48 – Meilleur complexe trouvé (2mcp).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

3.6 Docking@GRID

Dans cette section, nous présentons la plate-forme Docking@GRID dans laquelle le travail a été intégré dans le cadre de l'ANR Dock.

3.6.1 Description de la plate-forme

Docking@GRID est une plate-forme web dédiée au problème d'échantillonnage conformationnel et au docking moléculaire. Des captures d'écran sont visibles dans l'annexe C.6. Cette plate-forme permet à l'utilisateur d'exécuter des calculs de conformations de molécules et d'effectuer des simulations de docking sur des couples protéine-ligand ou protéine-protéine. Grâce à cette plate-forme l'utilisateur a la possibilité de manipuler directement ses molécules mais aussi de gérer ses expérimentations. Un ensemble de modules de visualisation est également disponible grâce à de nombreux outils intégrés tels que les bibliothèques de visualisation Chemaxon ou encore l'application Java Ligand-Info. Docking@GRID est conçu pour être déployé sur des clusters ou des grilles de calcul sachant que tous les algorithmes qui y sont intégrés ont été pensés pour une utilisation parallèle.

Docking@GRID s'articule autour de trois fonctionnalités :

1. Le calcul de conformations de petites molécules.
2. Le calcul de conformations de molécule plus importantes.
3. Le docking moléculaire d'un couple de molécules.

Le calcul de conformation de petites molécules se fait grâce à un algorithme dédié aux molécules de petite taille distribué par Chemaxon [22], société hongroise distribuant tout un ensemble de bibliothèques JAVA gratuitement pour les académiques. Docking@GRID intègre un nombre non négligeable d'outils Chemaxon. Le calcul de conformation de molécules de taille plus importante est rendu possible par l'utilisation d'un algorithme génétique bi-objecif hybride mis en place pour la résolution du problème de prédiction de structure de protéine [127] (« protein structure prediction » ou PSP).

La partie docking propose d'utiliser plusieurs logiciels de docking dont notre approche multi-objectif.

3.6.2 Intégration de l'algorithme de docking

Comme nous avons développé notre algorithme sous ParadisEO, nous avons pu mettre en place aisément un système de fichier de configuration très complet. Celui permet de configurer complètement le docking ou au contraire de lui fournir un minimum de données pour qu'il puisse fonctionner. De la même manière, Docking@GRID propose un mode simple ou expert pour l'utilisation de cet algorithme. Ainsi, en mode simple, l'utilisateur effectue un docking de type ligand/récepteur pour lequel il n'aura besoin de fournir que deux fichiers : un pour le ligand et l'autre pour le récepteur. En mode expert, il pourra tout configurer et choisir :

- Un docking protéine/protéine ou ligand/récepteur.
- Le degré de flexibilité du docking : rigide, semi-flexible ou flexible.
- La fonction d'énergie utilisée, celle basée sur le ICVFF ou celle d'Autodock 4.0.
- Le modèle multi-objectif qu'il veut utiliser en activant ou désactivant les différents objectifs.
- Tous les paramètres liés à l'évolution de l'algorithme génétique.

C'est l'algorithme qui ensuite s'assurera de la cohérence de la configuration avant de s'exécuter. Docking@GRID est aussi en train d'intégrer un ensemble d'outils que j'ai pu mettre en place durant mon travail sur le problème : calcul de RMSD entre molécules, convertisseur Mol2 vers CAR, générateur de

[22] www.chemaxon.com

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

paysage énergétique, ….

Même si la plateforme Docking@GRID se base en priorité sur les travaux effectués au sein de l'équipe projet INRIA DOLPHIN, elle reste ouverte à l'intégration d'autres outils de calcul de conformation et de docking moléculaire. Le but est de fournir une plate-forme fonctionnelle aussi bien pour l'expert biologiste que pour les personnes n'étant pas du domaine.

3.7 Conclusions

Dans ce chapitre, par l'intermédiaire du projet ANR Dock, nous avons proposé et testé de nouvelles modélisations multi-objectives pour le problème du docking moléculaire flexible. A partir de l'existant et d'hypothèses concernant la forme du paysage énergétique autour des complexes ligand/récepteur optimaux, nous avons proposé huit modèles combinant des critères énergétiques et géométriques. Nous avons utilisés deux champs de force différents, un basé sur le Consistent Valence Force Field (ICVFF) et celui utilisé par Autodock 4.0. Nous avons pu ainsi constater que si ces champs de force avaient des comportement différents d'un point de vue paysage, leur comportement global vis à vis des meilleurs résultats tend à être identifique. En plus des opérateurs standards du docking moléculaire flexible (rotation, translation et rotation de torsion), nous avons spécialement mis en place cinq mutations dont deux qui sont des métaheuristiques à base de solution unique mono-objectif. Ainsi, nous avons pu mettre en place des algorithmes génétiques hybrides dont le comportement a été comparé à travers une douzaine de combinaison d'opérateurs. Avant de procéder aux tests, deux schémas d'algorithme génétique multi-objectif ont été testé (NSGA-II et IBEA) et seul le meilleur a été gardé (IBEA).

Les résultats de notre étude ont prouvé que les nouveaux critères proposés sont valable (surface et robustesse) et que la coopération métaheuristique à base de population de solutions et de métaheuristiques à base de solution unique est efficace. Grâce à cette étude de modélisation, il est maintenant possible de se concentrer sur les quelques modèles et configurations qui sont apparues comme les meilleurs afin d'améliorer les résultats. Il faut nécessairement continuer de tester d'autres instances de la base CCDC/Astex afin d'améliorer encore le comportement des modèles sélectionnés. Des instances en provenance d'autres bases de données seront aussi testées pour valider la robustesse de notre approche. Le but est d'obtenir des résultats proches des outils actuels de docking moléculaire (comme Autodock) qui non seulement obtiennent des RMSD inférieure à deux mais arrive aussi généralement à des résultats inférieure à un.

http://doc.univ-lille1.fr

Conclusions et perspectives

Dans ce mémoire de thèse, nous avons présenté les travaux effectués en collaboration avec des laboratoires de chimie et de biologie de Lille, à savoir le Laboratoire de Chimie Organique et Macromoléculaire (LCOM, UMR 8009, UFR de Chimie) qui fait partie intégrante de Centre Commun de Spectrométrie de Masse pour le chapitre 2 ; et le Laboratoire de Glycobiologie Structurale et Fonctionnelle (LGSF, UMR 8576, UFR de Biologie) pour le chapitre 3.

Ces deux laboratoires sont situés sur le campus universitaire de Lille1, cela a donc permis de faciliter les échanges. J'ai pu découvrir la protéomique avec le LCOM notamment en participant aux manipulations standards de l'étude des protéines (de l'extraction à l'identification). J'ai pu aussi comprendre de quelle manière les techniciens du LCOM appréhendent l'identification de protéines et comment se positionnent leurs travaux actuels par rapport à l'existant. De là j'ai pu me former à ce domaine que je ne connaissais pas avant mon stage de Master II Recherche [6]. J'ai pu ainsi prendre du recul et mettre en évidence les axes de travail possibles, c'est à ce moment que le problème de « de novo **protein** sequencing » a été mis en évidence. De la même manière, la collaboration avec le LGSF m'a permis de me rendre compte des buts et des enjeux actuels de l'analyse de structures des protéines et plus spécifiquement ceux liés au problème de prédiction de structure des protéines dans le cas du docking moléculaire flexible. J'ai pu ainsi apprendre progressivement les notions de champ de force et d'intéractions au sein des molécules. Ces deux collaborations m'ont permis de pouvoir travailler aussi bien au niveau de la protéine (LCOM) qu'au niveau de l'atome (LGSF) ce qui est tout à fait complémentaire.

Ces échanges avec les spécialistes de chaque domaine ont permis de réaliser un gros travail de modélisation des problèmes étudiés sous forme de problèmes d'optimisation combinatoire que ce soit pour l'identification de protéines ou pour le docking moléculaire flexible. Chaque modélisation fut validée par les experts du domaine, ainsi le travail réalisé a été collaboratif de la modélisation à la résolution. Pour l'identification de protéines, cela a consisté à valider un modèle de comparaison spectrale par l'intermédiaire du moteur d'identification ASCQ_ME. En effet, c'est avec les techniciens du laboratoire que le protocole de test a été mis en place et c'est ensemble que les tests ont été effectués. ASCQ_ME est à l'heure actuelle utilisé au sein du LCOM comme approche complémentaire pour l'identification de protéines mais est aussi testé par d'autres laboratoires. Concernant le « de novo **protein** sequencing », la mise en place du modèle SSO a nécessité un travail important qui a été également validé par les experts. Ce modèle en trois étapes, « Sequence », « Shape » et « Order », permet de séquencer une protéine expérimentale à partir de données spectrales sans utilisation des bases de données. L'étape « Sequence » réalise un « de novo **peptide** sequencing » rapide à partir de spectres MS/MS. Le résultat de cette étape sert de base à l'étape « Shape » qui permet une complémentation des séquences de peptides obtenus grâce à un spectre MS. Enfin l'étape « Order » permet de retrouver la séquence de la protéine originelle par réordonnancement des peptides complétés par l'étape précédente. C'est seulement à partir des objectifs fixés et de l'étude des paysages des fonctions d'évaluation de chaque étape que les méthodes de résolution ont été choisies. Il a fallu également travailler sur une représentation des solutions

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

du problème ainsi que sur les opérateurs associés afin d'obtenir une cohérence avec le modèle SSO. Cela montre à quel point le choix d'une méthode n'est pas une chose triviale mais une étude structurée. De cette étude, il a naturellement découlé l'utilisation d'une coopération séquentielle de trois méthodes différentes fonctionnant sur des données et selon des fonctions d'évaluation différentes. L'approche SSO est un modèle complet pour le « de novo **protein** sequencing » qui est tout à fait modulaire. Chaque étape de l'approche a été validée et peut être utilisée indépendemment. En effet, il est tout à fait possible de configurer l'approche pour n'utiliser qu'une partie de celle-ci : ne faire que du « de novo **peptide** sequencing » via l'étape « Sequence » seule, faire une amélioration d'un « de novo **peptide** sequencing » existant (venant de « Sequence » ou non) via l'étape « Shape » ou tout simplement de séquencer une protéine à partir de ces seuls peptides via l'étape « Order ». Les expérimentations ont montré que grâce à cette modélisation du problème de « de novo **protein** sequencing » il est possible d'obtenir des portions importantes des séquences expérimentales correspondant à des ensembles de peptides contigüs, ce qui n'avait jamais été proposé.

Sur le problème du docking moléculaire flexible, par l'intermédiaire du projet ANR Dock, le travail de modélisation a d'abord débuté par une étude de l'existant pour chercher qu'elles étaient les modélisations multi-objectives possibles pour ce problème. Comme pour l'identification de protéines, le choix de la méthode utilisée, ici les algorithmes génétiques (AGs), n'est pas le fruit du hasard mais se basent sur leur efficacité reconnue sur les problèmes multi-objectifs. En effet, les AGs font partis des quelques algorithmes pour lesquels l'utilisation du multi-objectif a bien été formalisée et donnent de très bons résultats sur des problèmes académiques et réels. Dans la volonté de trouver un nouveau modèle multi-objectif performant pour le docking moléculaire flexible, nous avons mis en place et testé huit modèles différents selon douze configurations d'opérateurs et cela pour six instances. De ces tests, nous avons mis en évidence l'intérêt du critère de surface ainsi que celui de robustesse au sein du docking. En effet, que ce soit la surface seule ou la combinaison de la surface et de la robustesse en plus de l'énergie, leur utilisation, valide par hypothèse, s'est révélé efficace expérimentalement. D'un point de vue énergétique, nous avons aussi pu valider d'une part la fonction d'énergie du ICVFF qui avait déjà été testée sur le problème de prédiction de structures de protéines mais qui a nécessité une adaptation pour être applicable au problème du docking ; et d'autre part la modélisation bi-objective de l'énergie au lieu d'une simple aggrégation de termes ce qui rejoint la littérature actuelle sur ce point. Enfin, la prise en compte de la flexibilité a amené des résultats plus stables pour une grande partie des instances étudiées.

Pour chacun de ces problèmes, où des très grands espaces de recherche sont parcourus, notre utilisation du parallélisme à petite ou grande échelle (clusters, grilles) reste une valeur sûre pour l'obtention de résultats en temps raisonnable. Il est vrai que les métaheuristiques sont des approches qui donnent des résultats de bonne qualité rapidement. Cependant, la taille des instances utilisées dans les problèmes réels ainsi que la complexité des fonctions d'évaluation utilisées entraine une explosion combinatoire de l'espace de recherche. Dans ce cas, même l'utilisation de métaheuristiques ne permet pas d'avoir des solutions de bonne qualité en un temps raisonnable. Seul le parallélisme peut, dans la mesure du possible, repousser les limites imposées par les caractéristiques du problème étudié. Il est important de rappeler que tous les résultats du docking moléculare ont été obtenu en moins de deux mois en parallèle alors qu'ils en auraient nécessité plus de neuf en séquentiel. De plus, outre l'aspect accélération de la résolution d'un problème, le parallélisme permet aussi, à travers la co-évolution, d'obtenir des meilleurs résultats. Ainsi il serait intéressant de tester plusieurs algorithmes génétiques coopératifs aussi bien pour l'approche SSO que pour le problème du docking pour encore améliorer la qualité des résultats.

A travers tous ces travaux, c'est une méthodologie complète pour la mise en place de modèles et leur validation qui a été proposée. On a ainsi montré que tout modèle, même s'il paraît cohérent par

http://doc.univ-lille1.fr

hypothèse, se doit d'être validé par un ensemble de processus qui vont chacun apporter une information différente mais complémentaire et permettre ensemble de guider la démarche de mise en place d'une méthode de résolution. Par exemple pour le docking, avant même de tester différents modèles, la comparaison d'algorithmes génétiques multi-objectifs que nous avons effectuée (NSGA-II VS IBEA) a montré que, même si une méthode multi-objective est choisie, il est nécessaire de valider son comportement à travers des analyses statistiques. Cela concerne aussi bien la qualité des fronts obtenus que le critère d'évaluation des solutions du problème étudié, dans notre cas le RMSD des solutions finales. Une telle démarche a notamment été récompensée par l'attribution d'un « Best Student Paper » durant la conférence CIBCB 2008 à Sun Valley, Idaho, USA [10]. De plus, chaque étude de problème a permis de se rendre compte à quel point la coopération de métaheuristiques est un outil essentiel dans la résolution de problèmes réels. La mise en place de ces coopérations ne se fait pas en partant du principe qu'il faut faire des métaheuristiques coopératives mais sur la base que pour bon nombre de problèmes réels, et notamment ceux étudiés dans ce mémoire, il s'est révélé que la coopération de métaheuristiques représentait soit l'unique moyen d'obtenir un modèle complet de résolution pour un problème où aucune approche à l'heure actuelle n'existait (« de novo **protein** sequencing »), soit le meilleur modèle pour résoudre un problème donné (docking moléculaire flexible). Nous avons aussi évoqué l'utilité de l'auto-adaptativité dès qu'il est question d'utiliser un ensemble de mécanisme sans connaître à priori pour quelles instances et à quel moment chacun d'entre eux sera le plus utile. C'est un mécanisme essentiel qui voit son application aussi bien dans les métaheuristiques à base de population unique (Recherche Tabou) que dans les algorithmes à base de population de solutions (algorithmes génétiques). Tout cela permet d'avoir une vue globale de la modélisation de méthodes d'optimisation ce qui est un atout non négligeable lorsqu'il est nécessaire d'ajouter de nouveaux modèles de métaheuristiques à une plateforme comme ParadisEO-MO.

Que ce soit pour le « de novo protein sequencing » ou le docking moléculaire flexible, les résultats obtenus ne doivent pas faire oublier qu'il est toujours nécessaire de continuer à valider les approches à travers l'utilisation de nouvelles instances avec l'aide des différents acteurs du domaine. Il est aussi important de penser à la viabilité à long terme des méthodes, c'est notamment une question importante pour l'identification de protéine par « de novo **protein** sequencing » car la protéomique repose énormément sur la qualité des données fournies par notamment la spectromètrie de masse. Par conséquent, à la question « est-ce que mon modèle deviendra obsolète avec l'évolution de la technologie » la réponse est « non, elle n'en sera que meilleure ». En effet, si les spectromètres tendent à devenir de plus en plus précis, les résultats de « de novo **peptide** sequencing » s'en trouveront améliorés. Par contre, il sera toujours nécessaire de procéder à du réordonnancement de peptides pour retrouver la séquence expérimentale. Concernant le docking, l'amélioration des techniques de crystallographie permettra d'obtenir des données de référence de meilleure qualité ce qui permettra d'une part d'améliorer la paramétrisation des champs de force existants et d'autre part de configurer au mieux les meilleures modélisations actuelles.

Dans les travaux présentés, l'aspect valorisation n'a pas été été oublié : tous les modèles présentés sont inclus généralement dans des plateformes web où il est possible de faire des requêtes d'identification de protéines (ASCQ_ME) ou de docking moléculaire (Docking@GRID). Les autres travaux, comme le modèle SSO ou toutes les méthodes annexes pouvant être utilisées indépendamment (ADP, ATP, SpecS, DNS) sont fournis sous formes de programmes open-source sous licence CeCILL téléchargeables sur le web. Par conséquent, il est tout à fait possible que d'autres laboratoires ou des entreprises veuillent réutiliser ce travail pour pouvoir l'utiliser tel quel ou l'améliorer. Cela peut être aussi à la base de nouvelles collaborations. Ainsi des laboratoires locaux et étrangers ont déjà montré leur intérêt pour le modèle SSO. En effet, pouvoir faire du « de novo **protein** sequencing » permet d'étudier le protéome d'êtres vivants sans connaissance à priori de leur génome. On pourra notamment penser aux plantes où le nombre de génomes déjà séquencés est limité. Enfin, mon application dans le dévelopement de la plateforme

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

ParadisEO-MO est motivé par ma volonté de valoriser, de rendre accessible et de faciliter l'utilisation des métaheuristiques à base de solution unique ainsi que la coopération de métaheuristiques.

Concernant la suite à apporter à mes travaux, il y a encore de nombreux points à évoquer aussi bien au niveau modélisation que résolution. Ainsi, pour l'identification de protéines, il serait intéressant de savoir si des modèles multi-objectifs pourraient être mis en place en combinant par exemple en un seul modèle des critères d'évaluation utilisant des données différentes. La combinaison de la fonction de comparaison spectrale avec d'autres fonctions de la littérature mériterait également une étude approfondie. Concernant l'étape « Order », il serait utile de tester une recherche locale à voisinage variable (disponible sous ParadisEO-MO). Cependant, cela nécessite au moins deux voisinages pour être utilisé, il faudrait par conséquent en développer au moins un nouveau. Une piste sur ce point correspondrait à faire une homologie avec les problèmes de type TSP où plusieurs voisinages sont possibles. Ainsi, notre « peptide swap » correspondrait au « city swap », par conséquent on pourrait faire des voisinages de type K-opt adapté à l'étape « Order ». De plus, comme je l'ai évoqué précédemment, l'utilisation du parallélisme à travers une coopération de type co-évolutionnaire de haut niveau (section 1.4.4.4) permettrait une meilleure exploration de l'espace de recherche.

Concernant le docking moléculaire flexible, il y a encore beaucoup à faire. Par principe, il serait utile de tester l'impact d'une coopération avec une méthode à solution unique multi-objective plutôt que mono-objective. En effet, à priori, une recherche locale multi-objective serait plus à même de donner de meilleurs résultats dans le cadre de cette coopération ; encore faudrait-il mettre en place de telles recherches locales ce qui n'est pas aisé. De plus, nous avons proposé l'utilisation des algorithmes génétiques mais il existe d'autres métaheuristiques à base de population qui sont bien adaptées aux problèmes continus. Je pense notamment à l'optimisation par essaim de particules (PSO [73]) ou aux colonies de fourmis (Ant Colonies [38]). Cependant, à la différence de la partie de l'identification de protéines où la modélisation est mono-objectif, des schémas multi-objectifs pour ces deux types d'algorithmes ne sont pas encore aujourd'hui aussi répandus que pour les AGs. Par conséquent, il faudrait déjà mettre en place une version multi-objectif de ces méthodes avant de pouvoir les tester.

Au sein de ce manuscrit, il a beaucoup été question de coopération que ce soit de métaheuristiques ou de personnes i.e. collaboration. J'ai voulu montrer que la notion de coopération, à tous les niveaux, est pour moi le moyen le plus efficace de répondre aux problèmes combinatoires actuels. Ainsi l'interdisciplinarité inhérente à la plupart des problèmes doit amener à des coopérations aussi bien de personnes (biologiste, chimiste, physicien, informaticien, …) que de moyens (méthodes exactes, métaheuristiques, …).

http://doc.univ-lille1.fr

Bibliographie

[1] J. Atkins and W.E. Hart. On the intractability of the protein folding with finite alphabet of amino-acids. *Algorithmica*, 25(2-3) :279–294, 1999.

[2] V. Bafna and N. Edwards. Scope : A probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, 1(1) :1–9, 2001.

[3] M. Basseur and E. K. Burke. Indicator-based multi-objective local search. In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3100–3107, Singapore, 2007. IEEE Press.

[4] M. Basseur, F. Seynhaeve, and E-G. Talbi. Design of Multi-objective Evolutionnary Algorithms : Application to the Flow-shop Scheduling Problem. In *Congress on Evolutionary Computation (CEC'02)*, pages 1151–1156, May 2002. Honolulu, Hawaii, USA.

[5] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — a platform and programming language independent interface for search algorithms. In *EMO'2003*, volume 2632 of *LNCS*, pages 494–508, Faro, Portugal, 2003. Springer-Verlag.

[6] J-C. Boisson. Méthodes d'optimisation pour l'identification automatique de protéines. Master's thesis, Université des Sciences et Technologies de Lille, 2005.

[7] J-C. Boisson and E-G.Talbi. Software Framework for Metaheuristic. In 22^{nd} *European Conference on Operational Research (EURO XXII)*, page 45, July 8-11 2007. Prague, Czech Republic.

[8] J-C. Boisson, O. Guerre, L. Jourdan, E-G. Talbi, and C. Rolando. ASCQ_ME : a New Engine for Protein and Post-Translational Modification Identification from MS and MS/MS Raw Spectrum without Mass List Extraction. In 55^{th} *ASMS Conference on Mass Spectrometry*, June 3-7 2007. Indianapolis, Indiana, USA.

[9] J-C. Boisson, L. Jourdan, E-G.Talbi, C. Cren-Olive, and C. Rolando. ASCQ_ME : A New Engine For Peptide Mass Fingerprinting Without Mass List Extraction. In *Molecular Cellular Proteomics*, volume 4, page S18. HUPO 4th Annual World Congress, August 28-September 1st 2005. Munich, Germany, poster.

[10] J-C. Boisson, L. Jourdan, E-G.Talbi, and D. Horvath. Parallel multi-objective algorithms for the molecular docking problem. In *IEEE 2008 Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8, September 15-17 2008. Best student Paper. Sun Valley Resort, Idaho, USA.

[11] J-C. Boisson, L. Jourdan, E-G.Talbi, and C. Rolando. A Preliminary Work on Evolutionary Identification of Protein Variants and New Proteins on Grids. In *20th International Conference on Advance Information Networking and Application (AINA)*, volume 2, pages 583–587, April 18-20 2006. Vienna, Austria.

[12] J-C. Boisson, L. Jourdan, E-G.Talbi, and C. Rolando. Protein Sequencing with an Adaptive Genetic Algorithm from Tandem Mass Spectrometry. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1412–1419, July 16-21 2006. Vancouver, BC, Canada, ISBN : 0-7803-9489-5.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

[13] J-C. Boisson and E-G. Talbi. ParadisEO-MO : a Framework to Design Single Solution Metaheuristics. Technical report, INRIA Lille Nord Europe, 2008.

[14] D.E. Brown, C.L. Huntley, and A.R. Spillane. A parallel genetic heuristic for the quadratic assignment problem. In *3rd International Conference on Genetic Algorithms*, pages 406–415, 1989.

[15] R. Bruni, G. Gianfranceschi, and G. Koch. On peptide de novo sequencing : a new approach. *Journal of Peptide Science*, 11 :225–234, 2005.

[16] B.D. Bursulaya, M. Totrov, R. Abagyan, and C.L. Brooks. Comparative study of several algorithms for flexible ligand docking. *Journal of Computer-Aided Molecular Design*, 17(11) :755–763, november 2003.

[17] S. Cahon. *ParadisEO : Une plateforme pour la conception et le déploiement de métaheuristiques parallèles hybrides sur clusters et grilles*. PhD thesis, Université des sciences et technologie de Lille, 2005.

[18] S. Cahon, N. Melab, and E-G. Talbi. ParadisEO : A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. *Journal of Heuristics*, 10(3) :357–380, 2004.

[19] P-Y. Calland. On the structural complexity of a protein. *Protein Engineering*, 16(2) :76–86, 2003.

[20] P.A. Castillo, J.J. Merelo, V. Rivas, G. Romero, and A. Prieto. Evolving Multilayers Perceptrons. *Neural Processing Letters*, 12(2) :115–127, 2000.

[21] V. Cerny. A thermodynamical approch to the traveling salesman problem : An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45 :41–51, 1985.

[22] I. Charon and O. Hudry. The noising method : a new combinatorial optimization method. *Operations Research Letters*, 14 :133–137, 1993.

[23] T. Chen, M-Y. Kao, M. Tepel, J. Rush, and G.M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8(3) :325–337, 2001.

[24] J. Colinge, D. Chiappe, S. Lagache, M. Moniatte, and L. Bougueleret. Differential proteomics via probabilistic peptide identification scores. *Analytical Chemistry*, 77 :596–606, 2005.

[25] J. Colinge, A. Masselot, I. Cusin, E. Mahé, A. Niknejad, G. Argoud-Puy, S. Reffas, N. Bederr, A. Gleizes, P.A. Rey, and L. Bougueleret. High-performance peptide identification by tandem mass spectrometry allows reliable automatic data processing in proteomics. *Proteomics*, 4 :1977–1984, 2004.

[26] J. Colinge, A. Masselot, M. Giron, T. Dessingy, and J. Magnin. OLAV : Towards high-throughput tandem mass spectrometry data identification. *Proteomics*, 3 :1454–1463, 2003.

[27] P. Crescenzi, D. Goldman, C.H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3) :423–466, 1998.

[28] V. Cutello, G. Narzisi, and G. Nocisia. A Class of Pareto Archived Evolution Strategy Algorithms Using Immune Inspired Operators for Ab-Initio Protein Structure Prediction. *Proceedings of EvoWorkshops 2005*, LNCS 3449 :54–63, 2005.

[29] V. Cutello, G. Narzisi, and G. Nocisia. A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of the Royal Society Interface*, 2005. Published online.

[30] M. Damsbo, B.S. Kinnear, M.R. Hartings, P.T. Ruhoff, M.F. Jarrold, and M.A. Ratner. Application of evolutionary algorithm methods to polypeptide folding : comparison with experimental results for unsolvated ac-(ala-gly-gly)5-lys+. In *PNAS*, pages 7215–7222, 2004.

http://doc.univ-lille1.fr

[31] V. Dancik, T.A. Addon, and K.R. Clauser. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6(3/4) :327–342, 1999.

[32] C. Darwin. *On the Origin of Species*. John Murray, 1859.

[33] L. Davis. Adapting operator probabilities in genetic algorithms. In J. D. Schaffer, editor, *Third International Conference on Genetic Algorithms*, pages 61–69. Morgan Kaufmann, 1989. San Mateo, CA.

[34] I. de Falco, R. del Balio, and E. Tarantino. An analysis of parallel heuristics for task allocation in multicomputers. *Computing*, 3(59) :259–275, 1995.

[35] I. de Falco, R. del Balio, E. Tarantino, and R. Vaccaro. Improving search by incorporating evolution principles in parallel tabu search. In *International Conference on Machine Learning*, pages 823–828, 1994.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm : NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2) :181–197, 2002.

[37] K.A. Dill. Theory for the Folding and Stability of Globular Proteins. *Biochemistry*, 24 :1501–1509, 1985.

[38] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system : an autocatalytic optimizing process. Technical report, Politecnico di Milano, 1991. Technical Report 91-016.

[39] G. Dueck. New optimization heuristics : The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1) :86–92, 1993.

[40] G. Duesh and T. Scheuer. Threshold accepting : A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90 :161–175, 1990.

[41] T.J.A. Ewing and I.D. Kuntz. Critical evaluation of search algorithms for automated molecular docking and database screening. *Journal of Computational Chemistry*, 18(9) :1175–1189, 1997.

[42] P.W. Finn and L.E. Kavraki. Computational Approaches to Drug Design. *Algorithmica*, 25 :347–371, 1999.

[43] C. Fleurent and A. Ferland. Genetic hybrids for the quadratic assignment problem. *DIMACS Series in discrete Mathematics and Theoretical Computer Science*, 16 :173–188, 1994.

[44] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization : Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, USA, 1993. Morgan Kaufmann publishers.

[45] R. Fraczkiewicz and W. Braun. Exact and Efficient Analytical Calculation of the Accessible Surface Areas and Their Gradients for Macromolecules. *Journal of Computational Chemistry*, 19(3) :319–333, 1998.

[46] A. Frank and P. Pevzner. Pepnovo : de novo peptide sequencing via probabilistic network. *Analytical Chemistry*, 77 :964–973, 2005.

[47] N. Futamura, S. Aluru, D. Ranjan, and K. Mehrotra. Efficient Algorithms for Protein Solvent Accessible Surface Area. *In Proceedings of the The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region*, 2 :586–592, 2000.

[48] M.R. Garey and D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[49] F. Glover. Heuristic for integer programming using surrogate constraints. *Decision Sciences*, 8 :156–166, 1977.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

[50] F. Glover. Tabu Search - part i. *ORSA Journal on Computing*, 1(3) :190–206, 1986.

[51] F. Glover. *Interfaces in Computer Science and Operation Research*, chapter Tabu search and adaptive memory programming advances, application and challenges, pages 1–75. R.S. Barr and R.V. Helgason and J.L. Kennington, 1996.

[52] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimisation. In 2^{nd} *International Conference on Genetic Algorithms*, pages 41–49, 1987.

[53] S.M. Le Grand and Jr. K.M. Merz. Rapid Approximation to Molecular Surface Area via the Use of Boolean Logic and Look-up Tables. *Journal of Computational Chemistry*, 14 :349–352, 1993.

[54] R. Gras, M. Müller, E. Gasteiger, P.A. Binz S. Gay, W. Bienvenut, C. Hoogland, J.C. Sanchez, A. Bairoch, D.F. Hochstrasser, and R.D. Appel. Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. *Electrophoresis*, 20 :3535–3550, 1999.

[55] Robin Gras. Structure des espaces de recherche, complexité des algorithmes d'optimisation combinatoire et applications à la bioinformatique. HDR, Institut Suisse de BioInformatique (ISB), Suisse, 2004.

[56] A.T. Hagler, E. Huler, and S. Lifson. Energy functions for peptides and proteins. I. derivation of a consistent force field including the hydrogen bond from amide crystals. *Journal of American Chemical Society*, 96(17) :5319–5327, 1974.

[57] A.T. Hagler and S. Lifson. Energy functions for peptides and proteins. II. the amine hydrogen bond and calculation ofamide crystal properties. *Journal of American Chemical Society*, 96(17) :5327–5335, 1974.

[58] B.D. Halligan, V. Ruotti, S.N. Twigger, and A.S. Greene. DeNovoID : a web-based tool for identifying peptides from sequence and mass tags deduced from *de novo* peptide sequencing by mass spectroscopy. *Nucleic Acids Research*, Vol. 33, 2005. Web Server issue.

[59] M.J. Hartshorn, M.L. Verdonk, G. Chessari, S.C. Brewerton, W.T.M. Mooij, P.N. Mortenson, and C. W. Murray. Diverse, High-Quality Test Set for the Validation of Protein-Ligand Docking Performance. *Journal of Medical Chemistry*, 50 :726–741, 2007.

[60] M. Havilio, Y. Haddad, and Z. Smilansky. Intensity-based statistical scorer for tandem mass spectrometry. *Analytical Chemistry*, 75 :435–444, 2003.

[61] A. Heredia-Langner, W.R. Cannon, K.D. Jarman, and K.H. Jarman. Sequence optimization as an alternative to de novo analysis of tandem mass spectrometry data. *Bioinformatics*, 20(14) :2296–2304, 2004.

[62] F. Herrera and M. Lozano. Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In F. Herrera and J. L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 95–125. Physica-Verlag, Heidelberg, 1996.

[63] D.S. Hochbaum. *Approximation algorithms*. International Thomsom Publishing, 1996.

[64] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[65] T.P. Hong, H.S. Wang, and W.C. Chen. Simultaneosly applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, 6 :439–455, 2000.

[66] R. Huey, G.M. Morris, A.J. Olson, and D.S. Goodsell. A semiempirical free energy force field with charge-based desolvation. *Journal of Computational Chemistry*, 28 :1145–1152, 2007.

[67] S. Janson, D. Merkle, and M. Middendorf. Molecular docking with a multi-objective Particle Swarm Optimization. *Applied Soft Computing*, 8(1) :666–675, 2008.

http://doc.univ-lille1.fr

[68] R.S. Johnson, S.A. Martin, K. Biemann, J.T. Stults, and J.T. Watson. Novel fragmentation process of peptides by collision-induced decomposition in a tandem mass spectrometer : differentiation of leucine and isoleucine. *Analytical Chemistry*, 59(21) :2621–2625, 1987.

[69] G. Jones, P. Willet, and R.C. Glen. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology*, 245(1) :43–53, 1995.

[70] L. Jourdan, J-C. Boisson, E-G.Talbi, and C. Rolando. An online engine for peptide mass fingerprinting from spectrum. In *Recomb 2006*, April 2-5 2006. Venise, Italia, Poster.

[71] Bryant A. Julstrom. What have you done for me lately ? adapting operator probabilities in a steady-state genetic algorithm. In L.J. Eshelman, editor, *Proceedings of the sixth International Conference on Genetic Algorithms*, pages 81–87. Morgan Kaufmann, 1995. San Francisco, CA.

[72] M. Keijzer, J. J. Merelo, G. Romero, and M. Schoenauer. Evolving objects : A general purpose evolutionary computation library. *Lecture notes in computer science*, 2310 :231–242, 2002.

[73] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[74] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, 1983.

[75] J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006. (revised version).

[76] J.D. Knowles and D.W. Corne. On metrics for comparing non-dominated sets. In *IEEE Congress of Evolutionary Computation*, pages 711–716, 2002.

[77] J. Kosa and D. Andre. Parallel genetic programming on a network of transputers. Technical report, Standford University, 1995. Technical Report CS-TR-95-1542.

[78] A. Leaver-Fay, G.L. Butterfoss, J. Snoeyink, and B. Kuhlman. Maintaining solvent accessible surface area under rotamer substitution for protein design. *Journal of Computational Chemistry*, 28(8) :1336–1341, 2007.

[79] B. Lee and F.M. Richard. The interpretation of protein structures : Estimation of static accessibility. *Journal of Molecular Biology*, 55 :379–400, 1971.

[80] A. Liefooghe, M. Basseur, L. Jourdan, and E-G. Talbi. ParadisEO-MOEO : A Framework for Multi-Objective Optimization. In Springer-Verlag, editor, *Proceedings of EMO'2007*, pages 457–471, 2007.

[81] P.G. Lokhov, O.V. Tikhonova, S.A. Moshkovskii, E.I. Goufman, M.V Serebriakova, B.I. Maksimov, I.Y. Toropyguine, V.G. Zgoda, V.M. Govorun, and A.I. Archakov. Database search post-processing by neural network : advanced facilities for identification of components in protein mixtures using mass spectrometric peptide mapping. *Proteomics*, 4 :p.633–642, 2004.

[82] H.R. Lourenco, O. Martin, and T. Stutzle. *Handbook of Metaheuristics*, volume 57 of Operations Research and Management Science, chapter Iterated local search, pages 321–353. Kluwer Academic Publishers, 2002.

[83] B. Lu and T. Chen. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *Drug Discovery Today : BIOSILICO*, 2(2) :85–90, 2004.

[84] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie. PEAKS : powerful software for peptide *de novo* sequencing by tandem mass spectrometry. *Rapid Communication in Mass spectrometry*, 17(20) :2337–2342, 2003.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

[85] M.J. MacCossa, C.C. Wu, and J.R. Yates. Probability-based validation of protein identification using a modified SEQUEST algorithm. *Analytical Chemistry*, 74 :5593–5599, 2002.

[86] J. Magnin, A. Masselot, C. Menzel, and J. Colinge. OLAV-PMF : a novel scoring scheme for high-throughput peptide mass fingerprinting. *Journal of Proteome Research*, 3 :55–60, 2004.

[87] J.M. Marlard, A. Heredia-langner, D.J. Baxter, K.H., Jarman, and W.R. Cannon. Constrained de novo peptide identification via multi-objective optimization. In *International Parallel and Distributed Processing Symposium*, page 191a, 2004.

[88] O. Martin, S. Otto, and E. Felten. Large-step markov chains for the tsp : Incorporating local search heuristics. *Operations Research Letters*, 11(4) :219–224, 1992.

[89] O. Martin, S.W. Otto, and E.W. Felten. Large-step markov chains for the traveling salesman problem. *Complex Systems*, 5(3) :299–326, 1991.

[90] R. Matthiesen, M.B. Trelle, P. Hjrup, J. Bunkenborg, and O.N. Jensen. VEMS 3.0 : Algorithms and Computational Tools for Tandem Mass Spectrometry Based Identification of Post-translational Modifications in Proteins. *Journal of Proteome Research*, 4 :2238–2347, 2005.

[91] H. Meunier, E-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimisation. In *Congress on Evolutionary Computation (CEC'00)*, volume 1, pages 317–324, July 2000. Piscataway, New Jersey, USA.

[92] D.H. Milone, J.J. Merelo, and H.L. Rufiner. Evolutionary algorithm for speech segmentation. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1115–1120, 2002.

[93] M. Mitchell, J.H. Holland, and S. Forrest. When will a genetic algorithm outperform hill climbing ? *Advances in Neural Information Processing Systems*, 6 :51–58, 1994.

[94] M. Mladenovic and P.Hansen. Variable neighborhood search. *Computers and Operations Research*, 24 :1097–1100, 1997.

[95] M.Mann and M. Wilm. Error-tolerant identification fo peptides in sequences databases by peptide sequence tags. *Analytical Chemistry*, 66 :4390–4399, 1994.

[96] F. Monigatti and P. Berndt. Algorithm for accurate similarity measurements of peptide mass fingerprints and its application. *American Society for Mass Spectrometry*, 2005. Switzerland.

[97] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Har, R.K. Belew, and A.J. Olson. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19 :1639–1662, 1998.

[98] A. Neumaier. Molecular modelling of proteins and mathematical prediction of protein structure. *SIAM Review*, 39 :407–460, 1997.

[99] A. Neumaier. *Acta Numerica 2004*, chapter Complete Search in Continuous Global Optimization and Constraint Satisfaction, pages 271–369. Cambridge University Press, 2004.

[100] A. Oduguwa, A. Tiwari, S. Fiorentino, and R. Roy. Multi-Objective Optimisation of the Protein-Ligand Docking Problem in Drug Discovery. In *GECCO'06*, July 8-12 2006. Seattle, Washington, USA.

[101] C.H. Papadimitriou. *The complexity of combinatorial optimization problems*. PhD thesis, Princeton, 1976.

[102] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Prentice-Hall, 1982.

[103] L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem : An experimental study. In X. Gandibleux, M. Sevaux, K. Sörensen, and

http://doc.univ-lille1.fr

V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer-Verlag, Berlin, Germany, 2004.

[104] B. Parent, A. Kökösy, and D. Horvath. Optimized evolutionary strategies in conformational sampling. *Soft Comput.*, 11(1) :63–79, 2006.

[105] K.C. Parker. Scoring methods in MALDI peptide mass fingerprinting : Chemscore and the chemapplex program. *American Society for Mass Spectrometry*, 13 :22–39, 2002.

[106] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, and J.S. Cottrel. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20 :3551–3567, 1999.

[107] G. Polya. *How to solve it*. Princeton University Press, 1945.

[108] M. Rarey, B. Kramer, T. Lengauer, and G.Klebe. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology*, 261(3) :470–489, 1994.

[109] F.M. Richard. Areas, volumes, packing and protein structure. *Annual Review of Biophysics and Bioengineering*, 6 :151–176, 1977.

[110] V.M. Rivas, J.J. Merelo, I. Rojas, G. Romero, P.A. Castillo, and J. Carpio. Evolving two-dimensional fuzzy systems. *Fuzzy Sets and Systems*, 138(2) :381–398, 2001.

[111] A.L. Rockwood and S.L. Van Orden. Ultra-Speed Calculation of Isotope Distribution. *Analytical Chemistry*, 68(13) :2027–2030, 1996.

[112] A.L. Rockwood, S.L. Van Orden, and R.D. Smith. Rapid Calculation of Isotope Distribtion. *Analytical Chemistry*, 67(15) :2698–2704, 1995.

[113] O. Roudenko, T. Bosio, R. Fontana, and M. Schoenauer. Optimization of car front crash members. In *EA'2001 - Selected papers*, pages 202–214, 2002. LNCS 2310.

[114] J. Ryu, R. Park, and DS. Kim. Connoly Surface on an Atomic Structure via Voronoi Diagram of Atoms. *Journal of Computer Science and Technology*, 21(2) :255–260, 2006.

[115] J.R. Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. PhD thesis, Massachusetts Institute of Technologie, Cambridge, Massachusetts, USA, 1995.

[116] S. Schulze-Kremer and U. Tiedmann. Parameterizing genetic algorithms for protein folding simulation. In *Twenty-Seventh Hawaii International Conference on System Sciences*, pages 345–354, 1994.

[117] B.C. Searle, S. Dasari, M. Turner, A.P. Reddy, D. Choi, P.A. Wilmarth, A.L. McCormack, L.L. David, and S.R. Nagalla. High-throughput identification of proteins and unanticipated sequence modifications using a mass-based alignment algorithm for MS/MS de novo sequencing results. *Analytical Chemistry*, 76 :2220–2230, 2004.

[118] A. Shrake and J.A. Rupley. Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of Molecular Biology*, 79(2) :351–364, 1973.

[119] A.V. Spirov, D.L. Timakin, J. Reinitz, and D. Kosman. Experimental Determination of Drosophila Embryonic Coordinates by Genetic Algorithms, the Simplex method and Their Hybrid. *Lecture Notes In Computer Science*, 1803 :97–106, 2000.

[120] S. Stutzle. *Local search algorithms for combinatorial problems - analysis, algorithms and new applications*. PhD thesis, Sankt augustin, 1999.

[121] J. Suh and D. van Gucht. Incorporating heuristic information into genetic search. In 2^{nd} *International Conference on Genetic Algorithms*, pages 100–107, 1987.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

[122] B. Sundqvist, I. Kamensky, P. Hakansson, J. Kjellberg, M. Salehpour, S. Widdiyasekera, J. Fohlman, P.A. Peterson, and P. Roepstorff. Californium-252 plasma desorption time of flight mass spectroscopy of proteins. *Biological Mass Spectrometry*, 11(5) :242–257, 1984.

[123] E-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8 :541–564, 2002.

[124] E-G. Talbi. *Parallel Combinatorial Optimization*. John Wiley & Sons, 2005.

[125] E-G. Talbi, T. Muntean, and I. Samarandache. Hybridation des algorithmes génétiques avec la recherche tabou. In *Evolution Artificielle (EA'94)*, 1994.

[126] R. Tanese. Parallel genetic algorithms for a hypercube. In 2^{nd} *International Conference on Genetic Algorithms*, pages 177–183, 1987.

[127] A-A. Tantar, N. Melab, E-G. Talbi, and B. Toursel. A Parallel Hybrid Genetic Algorithm for Protein Structure Prediction on the Computational Grid. *Elsevier Science, Future Generation Computer Systems*, 23(3) :398–409, 2007.

[128] R. Thomsen. *Computational Intelligence in Bioinformatics*, chapter Protein-Ligand Docking with Evolutionary Algorithms. 2008.

[129] R.M. Twyman. *Principles of proteomics*. Garland Science/BIOS scientific publishers, 2004.

[130] H-M. Voigt, J. Born, and I. Santibanez-Koref. Modelling and simulation of distributed evolutionary search processes for function optimization. *Parallel Problem Solving from Nature*, 496 :373–380, 1990. LNCS.

[131] Y.N. Vorobjev and J. Hermans. SIMS : Computation of a Smooth Invariant Molecular Surface. *Biophysical Journal*, 73 :722–732, 1997.

[132] C. Voudouris and E. Tsang. Guided local search. *European Journal of Operational Research*, 113(2) :469–499, 1999.

[133] E.A. Wiley and G. Deslongchamps. PostDock : A novel visualization tool for the analysis of molecular docking. *Computing and Visualization in Science*, 2005. Published online.

[134] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionnary Computation*, 1(1) :67–82, 1996.

[135] I. Wood and T. Downs. Demon algorithms and their application to optimization problems. In *IEEE World Congress on Computational Intelligence*, pages 1661–1666, 1998.

[136] M.E.B. Yamagishi, N.F. Martins, G. Neshich, W. Cai, X. Shao, A. Beautrait, and B. Maigret. A fast surface-matching procedure for protein-ligand docking. *Journal of Molecular Modeling*, 12 :965–972, 2006.

[137] N. Zhang, X-J Li, M. Ye, S. Pan, B. Schwikowski, and R. Aebersold. ProbIDtree : An automated software program capable of identifying multiple peptides from a single collision-induced dissociation spectrum collected by a tandem mass spectrometer. *Proteomics*, 5 :4096–4106, 2005.

[138] W. Zhang and B.T. Chait. Profound : an expert system for protein identification using mass spectrometric peptide mapping information. *Analytical Chemistry*, 72 :2482–2489, 2000.

[139] E. Zitzler. *Evolutionnary algorithms for multiobjective optimization : methods and applications*. PhD thesis, Swiss Federal Institute of Technology, 1999.

[140] E. Zitzler and S.Kunzli. Indicator-based selection in multiobjective search. *Parallel Problem Solving from Nature, PPSN VIII*, 3242 :832–842, 2004.

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

http://doc.univ-lille1.fr

Glossaire

« Fitness » (d'une solution) : Qualité d'une solution donnée. C'est une valeur unique en mono-objectif ou un ensemble de valeurs en multi-objectif.

ADN : Acide désoxyribonucléique, support de l'information génétique sur des fragments déterminés : les gènes. Il permet la transmission de l'information génétique sans modifications (réplication).

ADP : ***1***. Adénosine diphosphate. Résultat de la réaction d'hydrolyse de l'adénosine tryphosphate (ATP). La synthèse d'ATP à partir d'ADP est une réaction endergonique : $ADP + P_i \longrightarrow ATP + H_2O$.
2. « **A**utomatic **D**igestion **P**rocess », processus automatique de digestion. Désigne la méthode complète de digestion que que j'utilise mais aussi l'outil autonome pour faire de la digestion de banques de données au format FASTA.

ANR : Agence national de la recherche.

ARN : Acide ribonucléique, structure auxiliaire dans la transmission (transcription) et l'expression de l'information qui est traduite sous forme de protéine (traduction).

ATP : ***1***. Adénosine triphosphate. Source d'énergie de la celule de par son hydrolyse en adénosine diphosphate (ADP) qui est une réaction exergonique : $ATP + H_2O \longrightarrow ADP + P_i$.
2. « **A**utomatic fragmen**T**ation **P**rocess », processus automatique de fragmentation. Désigne la méthode complète de fragmentation que j'utilise mais aussi l'outil autonome pour faire de la fragmentation de banques de données au format FASTA. ATP inclus ADP.

Bioinformatique : Processus d'analyse et d'interprétation au moyen de méthodes informatiques de données biologiques.

Complexité : Indicateur du coût de l'application d'une méthode donnée. La valeur associée est liée à la taille de l'espace de recherche à parcourir.

Digestion : Action d'une enzyme sur une protéine qui produit des sous-ensembles d'acides aminés appelés peptides.

Diversification : ***1***. Principe visant l'exploration des différentes zones de l'espace de recherche dans l'objectif d'atteindre une majorité d'optima locaux.
2. Indicateur de qualité concernant un front (optimisation multi-objective). L'idéal est d'obtenir une répartition uniforme des solutions le long d'un front.

Dogme central : Principe du flux de l'information génétique allant des nucléotides aux polypeptides (et jamais dans le sens contraire).

Endergonique : Ce dit d'une réaction ayant une enthalpie libre standard positive, c'est-à-dire qui consomme de l'énergie. La synthèse l'ADP est un exemple de réaction endergonique.

Enthalpie : Variation entre un état initial et final d'une réaction exprimé en Joule. Son signe permet de qualifier la réaction d'endergonique ou d'exergonique.

Espace de décision (ou décisionel) : Ensemble des solutions réalisables d'un problème.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

Espace de recherche : Combinaison des toutes les valeurs possibles pouvant être prises par les paramètres d'un problème.
Eucaryotes : Ensemble des organismes composés d'une ou plusieurs cellules comportant un noyau.
Exergonique : Ce dit d'une réaction ayant une enthalpie libre standard négative, c'est-à-dire qui fournit de l'énergie. L'hydrolyse de l'ATP est un exemple de réaction exergonique.

FASTA : Format de données largement répandu pour les bases de données de protéines (et d'autres types de données). Pour une protéine, il consiste en une description de celle-ci précédée du caractère « > » suivi de la séquence correspondante sous forme de séries de 80 caractères maximum par ligne parmi ceux acceptés par le format (les acides aminés standards plus certains caractères spéciaux).

Grid5000 : Grille hétérogène distribuée sur 9 sites dédiée à la recherche, son nom vient de la volonté d'avoir à disposition une grille de 5000 processeurs sur le sol français. Elle sert principalement aux tests de passage à l'échelle de méthodes de résolution pour des problèmes réels. Ce n'est pas une grille de production.
Génome : Ensemble des gènes d'un organisme donné. Il est invariant.

Instance (d'un problème) : Un jeu de valeurs particulières donné à tous les paramètres d'un problème.
Intensification : Principe d'amélioration d'une solution ou d'un ensemble de solutions en restant dans une zone précise de l'espace de recherche.

LCOM : Laboratoire de chimie organique et macromoléculaire. Partenaire du travail effectué en identification de protéines.
LGSF : Laboratoire de Glycobiologie Structurale et Fonctionnelle. Partenaire du travail effectué sur le problème du docking moléculaire flexible.

Modification post-traductionnelle : Comme son nom l'indique, la modification post-traductionnelle a lieu en dehors du principe de traduction de la protéine. C'est un ajout (ou une de déletion) de molécule qui a généralement un impact sur l'activation, l'inhibition ou la modification de la fonction d'une protéine.
Mole : Unité de base du système international représentant la quantité de matière d'un système contenant autant d'entités élémentaires qu'il y a d'atomes dans 0,012 kilogramme de carbone 12.
MPI : « Message Passing Interface ». Norme définissant une bibliothèque de fonctions permettant d'exploiter des ordinateurs distants et/ou multiprocesseurs par passage de messages. C'est un standard de communication pour l'exécution de programmes parallèles sur des systèmes à mémoire distribuée.
MS^x : Abréviation de « mass spectrum » de niveau x. Ce type de notation est utile pour éviter les écritures de type MS/MS/MS pour MS^3 par exemple.
Métaheuristique : Méthode de résolution approchée générique c'est à dire non dédiée à un problème en particuliers. Comme tout méthode approchée, une métaheuristique ne garantie pas l'obtention d'une solution optimale mais amène, généralement rapidement, à des solutions de bonne qualité.

Nucléotide : Motif de base des acides nucléiques formé de l'assemblage de plusieurs molécules : un sucre, un acide phosphorique et une base azotée.

Optimum global : Equivalent à solution optimale.
Optimum local : Meilleure solution d'un sous-espace de l'espace de recherche, un optimum local peut être aussi un optimum global.

http://doc.univ-lille1.fr

PFF : Abréviation de « peptide fragment fingerprinting », sert à indiquer les méthodes d'identification de protéines par empreinte de fragment c'est-à-dire à partir de spectres MS/MS. Il est à noter que ce terme n'est pas forcément très répandu mais permet de faire une analogie avec les méthodes d'identification par PMF.

Pic monoisotopique : Correspond au premier pic du profil isotopique c'est-à-dire aux isotopes les plus stables.

PMF : Abréviation de « peptide mass fingerprinting », sert à indiquer les méthodes d'identification de protéines par empreinte de masse peptidique c'est-à-dire à partir de spectres MS.

PPF BioInformatique : Action structurante de l'Université des Sciences et Technologies de Lille pour le soutien de la recherche en bioinformatique et ses application.

Protéome : Ensemble des protéines d'un organisme donné. Il varie au court du temps.

PSO : « Particle Swarm Optimization », optimisation par essaim de particules. Algorithme à base de populations de solutions bien adapté aux problèmes continus.

Réplication : Synthèse de l'ADN conduisant à une multiplication conforme du matériel génétique et à la copie de l'information.

Solution optimale (d'un problème) : Meilleure solution d'un problème mono-objectif.

SpecS : « **S**pectrum **S**imulator », simulateur de spectres. Désigne la méthode complète de génération de spectre MS mais aussi un outils autonome pour faire de la génération de spectres à partir de banques de données au format FASTA. SpecS inclut ADP.

Spot (sur un gel) : Décrit une zone d'intérêt sur un gel que l'on va analyser.

Traduction : Transformation d'une séquence d'acides nucléiques (polynucléotides) en séquence d'acides aminés (polypeptides).

Transcription : Passage de l'information contenue dans des segments d'ADN sous forme d'ARN.

Transcriptome : Ensemble des ARNs messagers d'un organisme donné. Il varie au court du temps.

TSP : « Traveling Salesman Problem » ou problème du voyageur de commerce. C'est un problème classique en optimisation combinatoire qui peut s'énoncer de la manière suivante : étant donné un ensemble de villes et le coût du parcours entre chacune d'entre elles, trouver le chemin le moins coûteux pour visiter toutes les villes et retourner au point de départ.

USTL : Université des Sciences et Technologie de Lille.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

A

Données complémentaires dédiées à ParadisEO-MO

A.1 Les algorithmes implémentés dans les *moAlgo*

A.1.1 *moHC*

Algorithme 10 : la recherche par descente à travers un objet *moHC*. « > » doit être interprété comme « meilleur que ».

Données : Une solution initiale « Sol ».
Résultat : Une solution meilleure ou égale à « Sol ».

```
1  booléen continu = vrai;
2  Solution meilleurCourant;
3  Solution voisinCourant;
4  tant que continu est vrai faire
5      continu = faux;
6      si Sol.aUnVoisin() alors
7          meilleurCourant= Sol.voisinSuivant();
8          meilleurCourant.calculeObjectifEfficacement();
9      fin
10     tant que Sol.aUnVoisin() faire
11         voisinCourant = Sol.voisinSuivant();
12         voisinCourant.calculeObjectifEfficacement();
13         si voisinCourant.objectif() > meilleurCourant.objectif() alors
14             meilleurCourant = voisinCourant;
15         fin
16     fin
17     si meilleurCourant.objectif() > Sol.objectif() alors
18         Sol = meilleurCourant;
19         continu = vrai;
20     fin
21 fin
22 retourner Sol;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

A.1.2 *moTS*

Algorithme 11 : la recherche taboue à travers un objet *moTS*. « > » doit être interprété comme « meilleur que ».

```
   Données : Une solution initiale « Sol »
   Résultat : Une solution meilleure ou égale à « Sol »
 1 Solution meilleurCourant;
 2 Solution voisinCourant;
 3 Vecteur<Solution> meilleuresSolutionsTrouvées;
 4 liste<Solution> listeTaboue;
 5 booléan ajouteSolution;
 6 tant que critèreDarrêt est faux faire
 7     si Sol.aUnVoisin() alors
 8         meilleurCourant= Sol.voisinSuivant();
 9         meilleurCourant.calculeObjectifEfficacement();
10     fin
11     tant que Sol.aUnVoisin() faire
12         voisinCourant = Sol.voisinSuivant();
13         voisinCourant.calculeObjectifEfficacement();
14         if voisinCourant.objectif() > meilleurCourant.objectif() then
15             meilleurCourant = voisinCourant;
16         end
17     fin
18     ajouteSolution = faux;
19     si listeTaboue.contient(meilleurCourant) est faux alors
20         ajouteSolution = vrai;
21     sinon
22         si critèreDaspiration.validéAvec(meilleurCourant) alors
23             ajotueSolution = vrai;
24         fin
25     fin
26     si ajouteSolution = vrai alors
27         Sol = meilleurCourant;
28         meilleuresSolutionsTrouvées.metAJourAvec(meilleurCourant);
29         listeTaboue.ajoute(meilleurCourant);
30     fin
31 fin
32 retourner meilleuresSolutionsTrouvées.meilleur() ;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

A.1.3 *moSA*

Algorithme 12 : le recuit simulé à travers un objet *moSA*.

Données : Une solution initiale « Sol », une température initiale « T_{init} », une température minimale « T_{min} », un nombre maximum d'itérations « maxIteration » et un facteur de refroidissement « facteur ».

Résultat : Un solution meilleure ou égale à « Sol ».

```
1  Solution voisinCourant;
2  flottant température = T_init;
3  Objectif differenceDobjectif;
4  tant que température ≥ T_min faire
5      pour i de 1 à maxItération faire
6          voisinCourant = Sol.voisinAléatoireSuivant();
7          voisinCourant.calculeObjectifEfficacement();
8          différenceDobjectif = voisinCourant.objectif() - Sol.objectif();
9          si différenceObjectif < 0 alors
10             Sol = voisinCourant;
11         sinon
12             Sol = voisinCourant selon une probabilité de e^(-différenceDobjectif/température)
13         fin
14     fin
15     température = température * facteur;
16 fin
17 retourner Sol ;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

A.1.4 *moILS*

Algorithme 13 : la recherche locale itérative à travers un objet *moILS*. « > » doit être interprété comme « meilleur que ».

Données : Une solution initiale, un nombre maximum d'itérations « maxItération ».
Résultat : Une solution meilleure (ou au moins égale) à « Sol ».

```
1  Solution meilleureSolution = Sol;
2  entier nombreDitérations = 0;
3  tant que nombreDitérations < maxItération faire
4  |   Sol = résultat d'une recherche par descente sur Sol;
5  |   si Sol.objectif() > meilleureSolution.objectif() alors
6  |   |   meilleureSolution = Sol;
7  |   sinon
8  |   |   Sol = meilleureSolution;
9  |   fin
10 |   Sol = Sol ayant subi une perturbation;
11 |   nombreItérations = nombreItérations + 1;
12 fin
13 retourner meilleureSolution ;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

A.1.5 *moVNS*

Algorithme 14 : la recherche à voisinage variable à travers un objet *moILS*. > doit être interprété comme « meilleur que ».

Données : Une solution initiale « Sol », un ensemble de voisinage N_k pour $k \in [1, k_{max}]$.
Résultat : Une solution meilleure (ou au moins égale) à « Sol ».

```
1  tant que critèreDarrêt est faux faire
2      k=1;
3      tant que k ≤ k_max faire
4          Solution Sol' = une solution aléatoire dans N_k;
5          Solution Sol" = recherche locale sur (Sol');
6          si Sol".objectif() > Sol.objectif() alors
7              Sol = Sol";
8              k = 1 ;
9          sinon
10             k = k+1;
11         fin
12     fin
13 fin
14 retourner Sol ;
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

B

Données complémentaires pour l'identification de protéines

B.1 Preuve de la complétude de l'arbre de digestion

Soit c le nombre de points de coupures possibles selon l'enzyme utilisée pour la digestion et le nombre maximum possible de clivages manqués et $k \in [1, c]$ le nombre de clivages manqués voulu. MPT signifie modification post-traductionnelle et m le nombre de sites de fixation pour la MPT considérée sur un peptide donné.

- Cas trivial : afin d'obtenir tous les peptides avec aucun clivage manqué, il suffit de couper la protéine au niveau des c sites de clivage possibles. On récupère ainsi $c + 1$ peptides. L'arbre de digestion est donc complet.

- Cas général : par hypothèse, tous les peptides avec un nombre de clivages manqués dans $[0, k-1]$ sont déjà calculés et par conséquent l'arbre de digestion pour $k-1$ clivages manqués au maximum est déjà complet. Afin d'obtenir tous les peptides avec k clivages manqués, il faut générer tous les peptides formés de la fusion de $k+1$ peptides contigus sans clivage manqué. Par hypothèse, tous les peptides avec $k-1$ clivages manqués ont déjà été générés, par conséquent il suffit de former les peptides de k clivages manqués en fusionnant les peptides de $k-1$ clivages manqués avec des peptides sans clivage manqué. Cela permet de générer, les $c-k+1$ peptides avec k clivages manqués. On obtient donc ainsi tous les peptides avec un nombre de clivages manqués dans $[0, k]$. Par conséquent, l'arbre de digestion est complet pour k clivages manqués au maximum.

- Cas des MPTs fixes : une MPT fixe signifie que s'il existe un site possible pour son implentation, il est nécessaire de le mettre. Par conséquent, une fois l'arbre de digestion complet obtenu avec le nombre de clivages manqués maximum désirés, il suffit d'appliquer sur tous les peptides résultats de la digestion la ou les MPTs fixes là où elles peuvent être mises. Le nombre de peptides générés lors de la digestion reste alors inchangé et l'arbre de digestion reste complet.

- Cas des MPTS variable : pour chaque site de fixation possible pour une MPT variable, il y a deux possibilités : soit mettre la MPT, soit ne pas la mettre. Par conséquent, la gestion de MPTs variable va augmenter le nombre de peptides résultant de la digestion si ceux-ci possèdent des sites de fixation compatibles. Ainsi, pour chaque peptide résultant de la digestion (après prise en compte des clivages manqués et des MPTs fixes) on obtient la génération de $\left(\left(\sum_{i=0}^{m} 2^i\right) - 2^m\right)$ peptides lors du

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

calcul de toutes les configurations possibles de la fixation de MPTs variables. La soustraction de 2^m permet de gérer le cas où il n'y a pas de site de fixation possible et donc où $m = 0$. La figure B.1 montre un exemple de génération de nouveaux peptides lors de l'utilisation d'une MPT variable fictive s'appliquant sur l'acide aminé K. L'arbre de digestion étant complet après le calcul des peptides avec clivage manqué et MPT fixes, la génération des nouveaux peptides avec des MPTs variable permet d'obtenir l'arbre de digestion complet quelque soit le type de MPT utilisé.

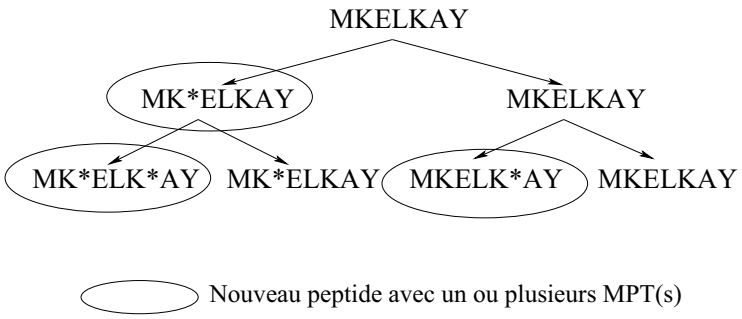


FIG. B.1 – Génération de nouveaux peptides dûe à l'utilisation d'une MPT variable fictive sur un peptide possédant deux sites de fixation possibles (acide aminé K). * corresponds à la fixation de la MPT. Seuls trois nouveaux peptides sont générés.

Finalement le nombre de peptides générés lors d'une digestion est de :

$$Nb_{peptides} = N + \sum_{i=1}^{N} \left(\left(\sum_{j=0}^{m_i} 2^j \right) - 2^{m_i} \right)$$

$$avec \;\; N = \frac{c + 1(c+2)}{2}$$

N est le nombre de peptides générés lors de la digestion avec prise en compte des clivages manqués (et des MPTs fixes). Le second terme rajoute les peptides générés lors de la prise en compte des MPTs variables où m_i correspond au nombre de sites de fixations possibles pour les MPTs variables activées sur le $i^{ième}$ peptide généré avec la seconde phase de digestion (prise en compte des MPTs fixes).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.2 Graphe de dépendance d'ADP

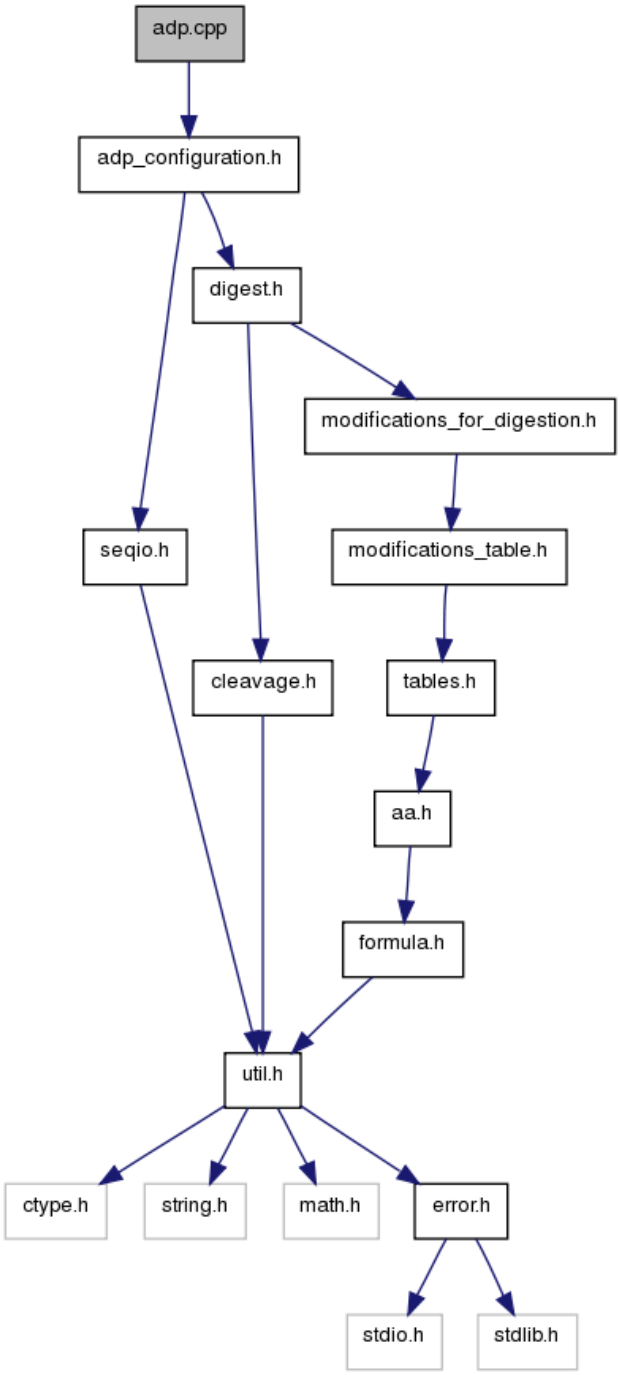


FIG. B.2 – Graphe des dépendances d'ADP permettant de voir le découpage modulaire de toutes les parties composant cette approche. Ce schéma a été généré automatiquement avec la documentation de l'outil grâce à doxygen. ADP représente environ 6100 lignes de code.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.3 Graphe de dépendance d'ATP

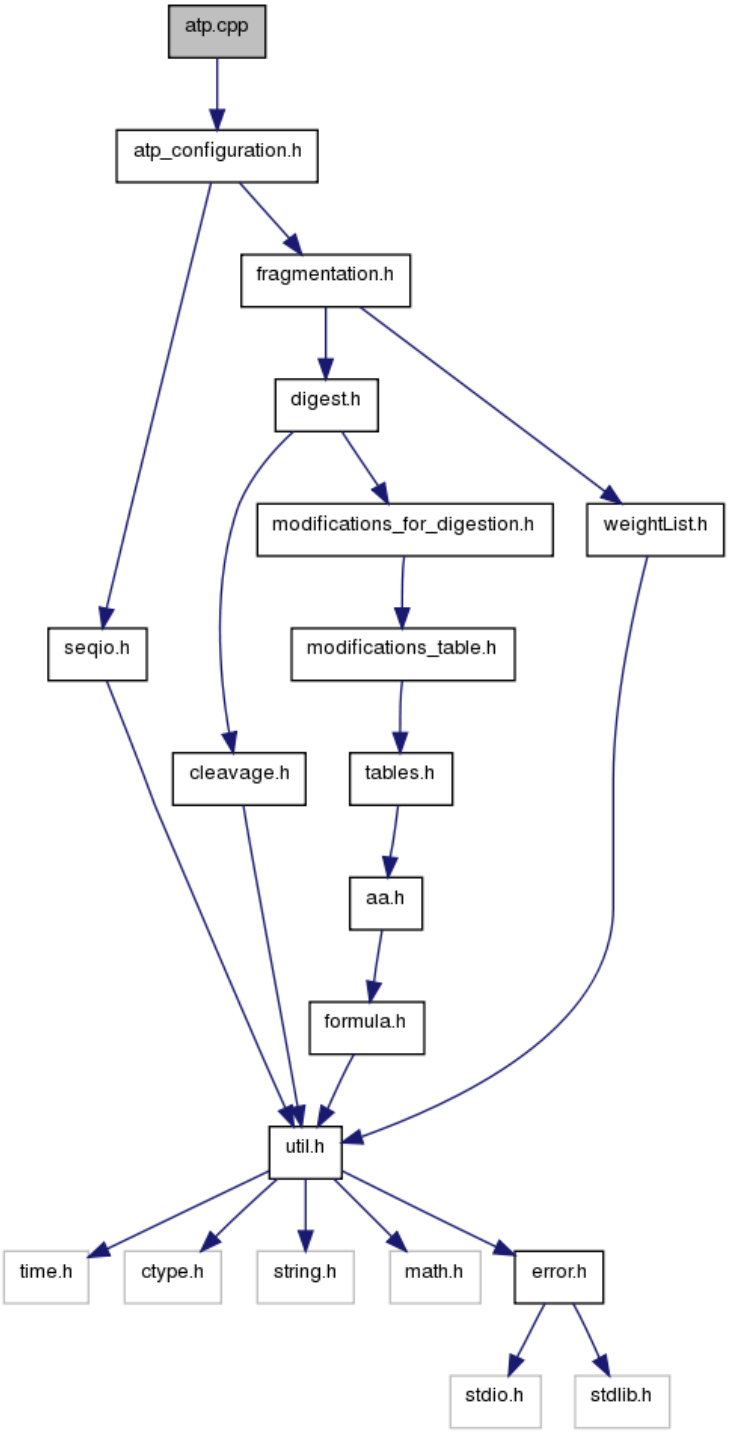


FIG. B.3 – Graphe des dépendances d'ATP qui permet de voir qu'il ne fait que compléter ce qu'ADP proposait. Ce schéma a été généré automatiquement avec la documentation de l'outil grâce à doxygen. ATP représente 8000 lignes de code (mais contient ADP).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.4 Graphe de dépendance de SpecS

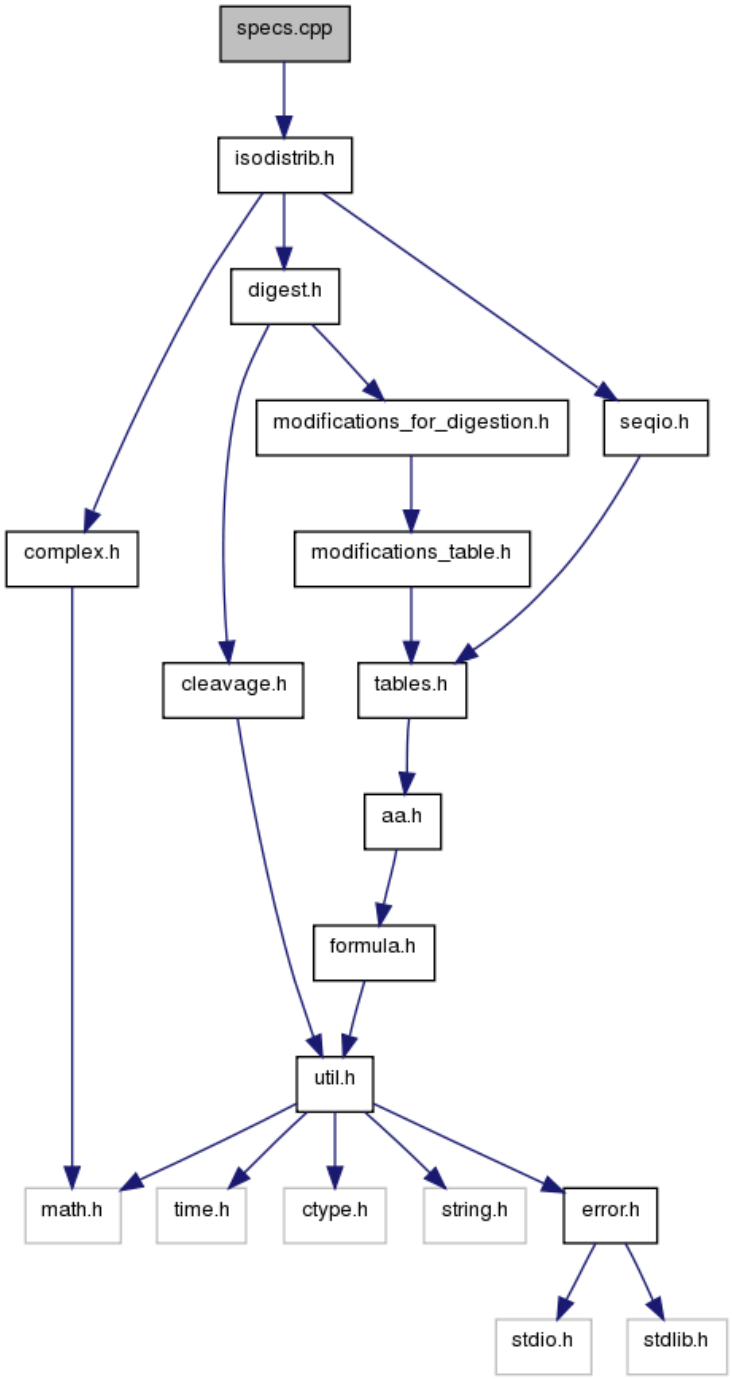


FIG. B.4 – Graphe des dépendances de Specs qui permet voir qu'il ne fait que compléter aussi ce qu'ADP proposait. A noter que SpecS est dans une version moins avancée qu'ADP ou ATP d'où l'absence de notion de configuration. Ce schéma a été généré automatiquement avec la documentation de l'outil grâce à doxygen. SpecS représente environ 8000 lignes de code (contient ADP).

http://doc.univ-lille1.fr

B.5 Graphe de dépendance de Ascq_me

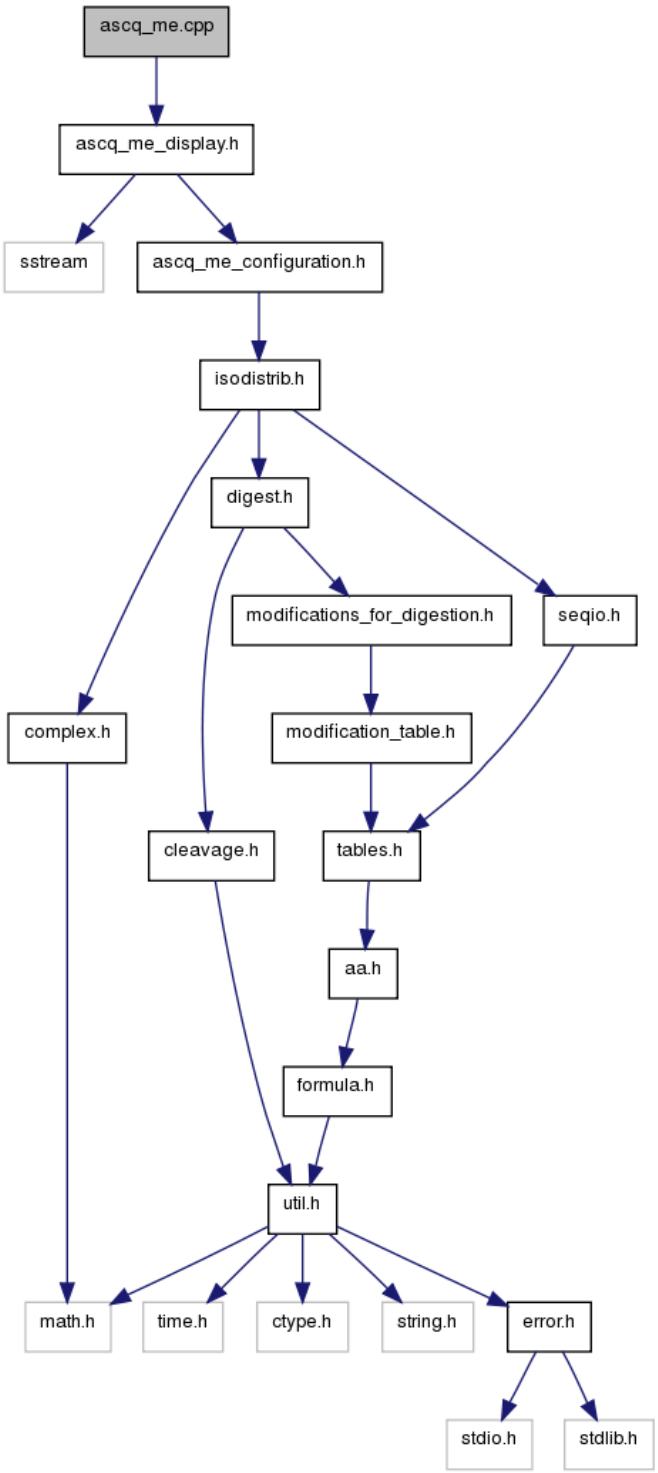


FIG. B.5 – Graphe des dépendances d'Ascq_me où l'on peut voir qu'aussi bien ADP et SpecS sont inclus. Ce schéma a été généré automatiquement avec la documentation de l'outil grâce à doxygen. Ascq_me représente environ 14 400 lignes de code (inclus ADP et SpecS).

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.6 Interface web d'Ascq_me

Comme la plupart des outils d'identification de protéines, Ascq_me propose la possibilité de soumettre des requêtes en ligne. Il reste cependant possible de télécharger l'application pour faire des requêtes en local. La version actuelle d'Ascq_me est la 1.0.4.. Pour le moment, Ascq_me compte plus d'une trentaine d'utilisateurs dans différents pays : France, Suisse, Espagne, Canada, Angleterre, Australie, Inde et USA.

B.6.1 Page de soumission de requêtes

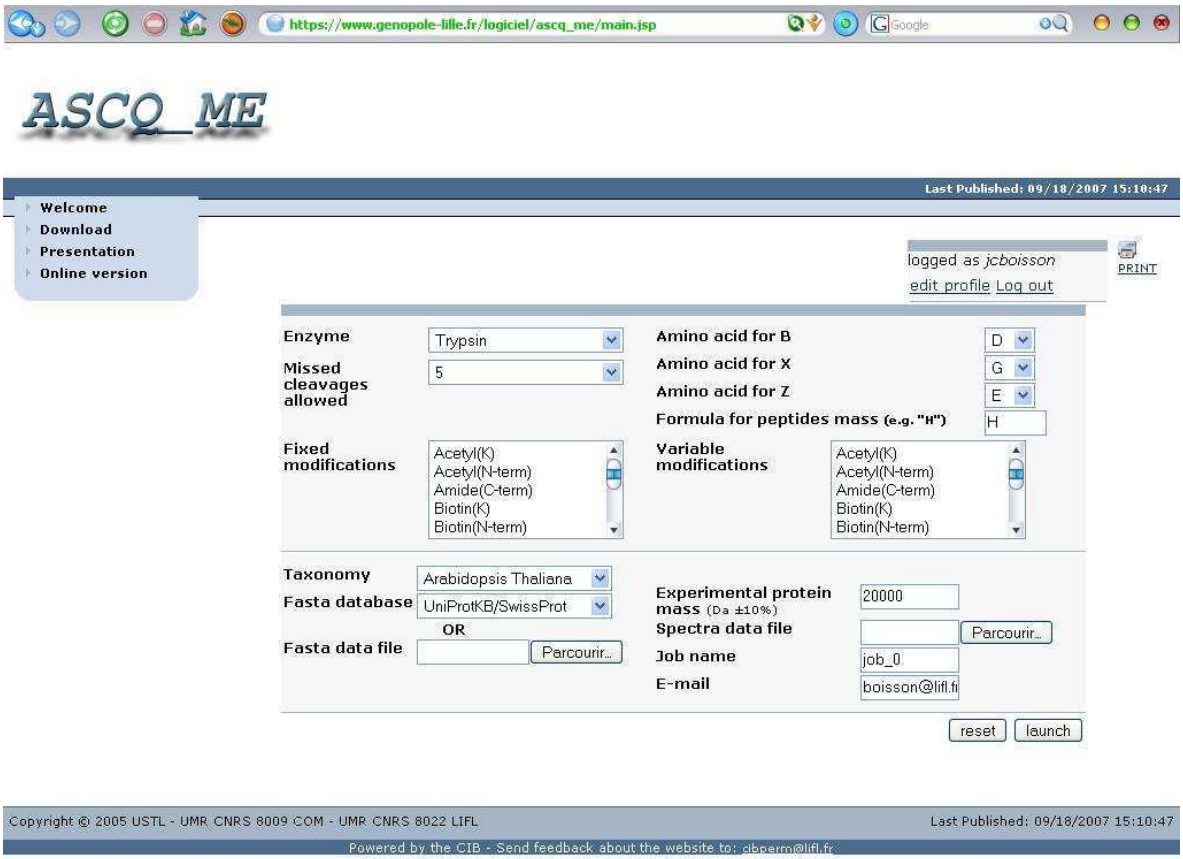


FIG. B.6 – Capture d'écran de la page de soumission d'une requête d'identification. Les acteurs de domaine reconnaîtront sans peine le style d'interface qui rappelle MASCOT.

http://doc.univ-lille1.fr

B.6.2 Page de résultats

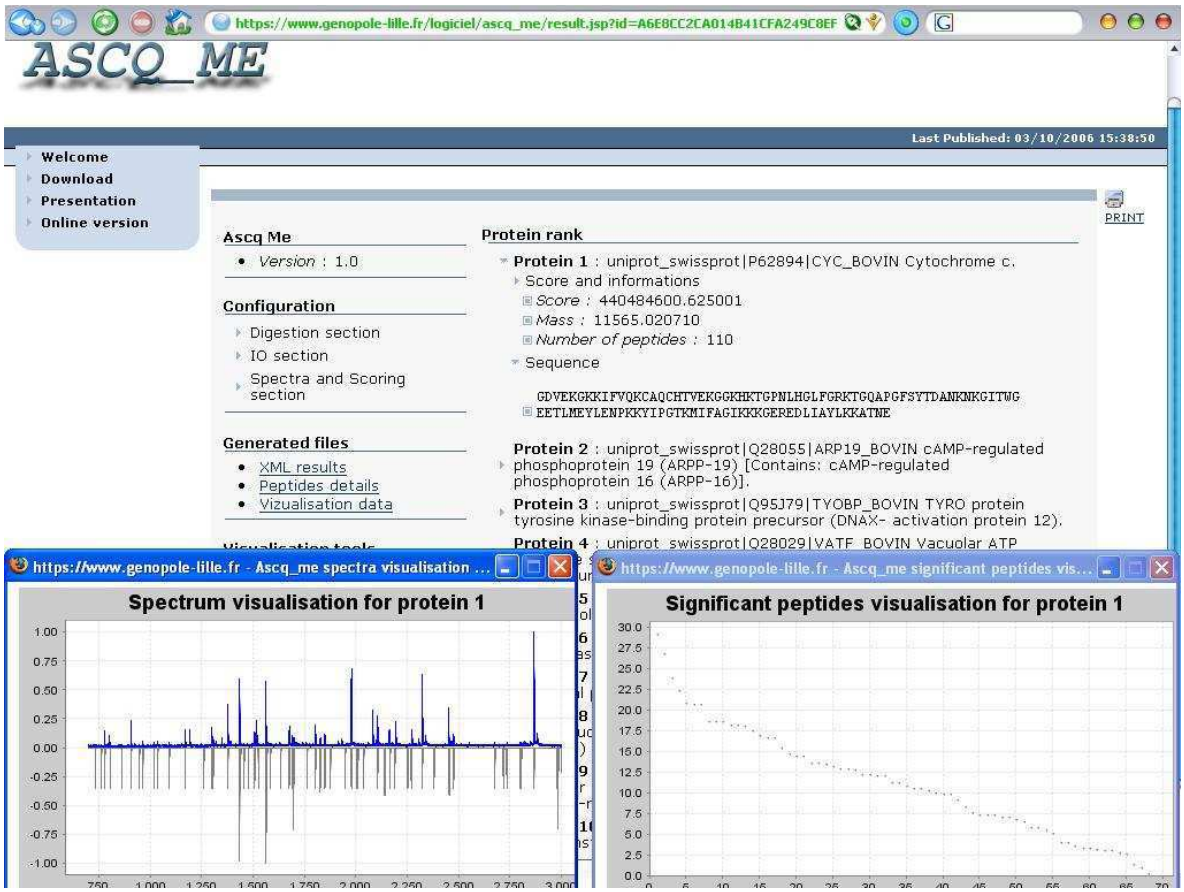


FIG. B.7 – Capture d'écran de la page des résultats avec toutes les fenêtres ouvertes : page de classement des protéines, dessin du spectre de corrélation et distribution des peptides ayant participé au score obtenu.

B.7 Exemple d'une très grosse protéine biaisant les scores d'identification

```
>sp|P62285|ASPM_BOVIN Abnormal spindle-like microcephaly-associated ...
PNEEAAAVWLSHGPAAERGFTIWPSAFVLQPKEKIVVSITWTPLKGGRIRETITFLVNDI
LKHQAILLGNAEEPKKKKRTLWDTINKKKASASSRHNKKASNIQNVNKTFNVSPKADRVR
SPLQACENLATNGNCSPPESNPLILEENKLPISPISPASQECHRGTCLPLPVRRSTTYTS
LPASENGGLVKADGANTAEDFHFNEKGITETSFDSIDNVNSQIEENGKLTLTPNYSSSLN
ITQSQGHFLSPDSFVNNSHASNNEPEFVKCLSPDMFVKGNTRPVILESKRVHEICRKILS
PDSFINDNYGLNEDLETESINPILSPNQFLKDNMAYICVSQQTCQLPLSTGHFQDSQPPQ
DERKNAAVPCISECQQLESPKATFEASKALEVMSNSYTFKKQNQPKFSAVQDISSHSRKK
PIKRRPILSATVTKRKPTCAVENQMETVKPKAKRCLNVVVGDCXKETDDQKEKDDFHPFL
PIRDLIXXRPKSSKNIVTPPCKVASVARKRKSEGHTGDENVRITVTECXEXQEVKRPHFS
PVESKTSTVKHTKKVVTSSLKRVSHREKLNLKKKTDSLGYRTPKTNRRTRPFVPVAQSNL
TFIKPLKGIPRHPMPFAAKNMFYDERWKEKQEQGFTWWLNFILTPDDFTVKTNISEVNAS
TLLLGVESQHKVSVPKAPTKDEVSLRAYTARCRLNRLRRAACRLFTSEKMVKAMKKLEIE
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

IEARRLIVRKDRHLWKDVGERQKVLNWLLSYNPLWLRIGLETIYGELVPLEDNSDVTGLA
MFILNRLLWNPDIAAEYRHPSVPHLYRDGHEEALSKFTLKKLLLLICFLDYAKISRLIDH
DPCLFCKDAEFKASKEILLAFSRDFLGGEGDLSRHLSLLGFPVTHVQMPFDEFDFAVKNL
AVDLQCGVRLVRTMELLTQNWNLSKKLRIPAISRLQKMHNVDIVLEILKSRGIQLNDEHG
NAILSKDIVDRHREKTLALLWKIALAFQVDISLNLDQLKEEIDFLKNTQSMKKTMSALSC
RPDAVISKKRDERHSGPFEQCSESVKLLMDWVNAVCGFYNKKVENFTVSFSDGRVLCYLI
HHYHPCYVPFDAICQRTTQTVECTHTGSVVLNSSSESDGSFLDFSLKPPDQENTSELYKE
LLENEKKNFQLVRSAARDLGGIPAMIHHSDMSNTIPDEKVVITYLSFLCARLLDLRKETR
AARLIQTTWRQYKLKKDLKHHQERDKAARIIQSAIINFLTKQRFKKKVSAALVIQKYWRR
ALAKRKLLMLKKEKLERVHSKSASIIQRHWRRYSTRKQFLKLKYYSIFLQSKIRMIIAVA
SYKRYHWATVTIQRRWRAHVRSKQDRQRYELLRSSTLVIQFAFRRWRRRKRQSQINAAIT
LQRAFRQWRVQKCAQEERAAVVIQSWYRMHRELRKYIHLRSCVIIIQARFRCFQAQKLYT
RTRESILTLQKHYRAYVKGKVERTGYLQKRAAAIRLQAAFRGRRARNLCRQIKAACVLQS
YWRMRQDRLRFLNLKKNIIRLQAHIRRRQQLHTYQKMKKAALIIQIHFRAYMSAKEVLAS
YQKTRSAVIVLQSACRRMQARKKFLHILTSIVKIQSYYRAYASRRKFLRLKKATVKLQSI
VRMKLARKQYLHLRAIAQQREEHRRASCIKLQAFLRGYLVRKQVRLQRKAAVSLQSYFRM
RKMRLDYLKVCHAAVVIQRYYRAHRAGAQQRKHFLQVRRAVTYLQATYRGYKVRRQLQQQ
SAAALKIQAAFRGYRQRTKYQSVLQSALKIQRWYRTHKTVSAIRSHFFKTRTAAISLQSA
YRGWKVRKQMRKEHEAAVKIQSAFRTARAQKEFRVLKTAASVIQQHLRARAAGRRQRTEY
TALRRAAVMLQSAWRGRAARRRIQKQQRCAIIIQAYYRRHVQQKRWEIMKKAAHLIQMHY
RAYRTGRKQHHLFLKTKXAAIILQSAFRGVRVRKKVKEMHQAAATIQSRYRAYQARKKYA
SYRAAAVIIQRWYRAAKLAGRQREEYLAVKKAALKIQAVYRGVRARRHIRRMHMAATLIK
AAFKMQQSRRRYQQMRTAAIIIQVRYRAYCQGRAQRAKYLMILKAVALLQAALRGARVRQ
SLRRMRTAATLIQAHYRGRRQQAYFNKLKKVTKTVQQKYRAARERHAQLRRYNQLRRSAI
CIQAAFRGMRARRRLKAMHSAAAVIQRRFRTLGMRRRFLSLRKTAVWVQRKYRAKVCTRH
HVQQLRLQKAAIKIQSWYRGWMVRKKIQEMRRAATVLQAAFRRHRTRARYQAWRCASQVI
QQRFRAGRAARLQRRQYLQQRHSALVLQAAFRGMRVRRRLKRMHASATLIQSRFRSIMMR
KRFLSLKKAAVFVQRKYRATICAKHHLHQFLELQKAIIIIQASYQRRMVKKQLQEMHRAA
ALIQASFRMHRARLAFQTWKHAAVLIQQRYRACRAAKLQRALYIRWRHSAVVIQAAYKGL
KARQLLREKHRAAVIIQSTYRMYRQHFFYQKLQWATKVIQERYRASKRKALQHDALKAAT
CARAGFQDMVVRRLIEERRHQAAITIQEHFRAFKTRKHYLHFRAKVVFVQRRYRELMAVR
TQAVICIQSCFRGFKARRGIQRMHLAATRIQSCYRRHRARADYQAKKRAVVVIQNHYRSY
IRVKMERKEFLAIQKSARTIQAAFRGMKVRQKLKTMPDKKMAAPATQPAFYCHRTESQHE
AGESPALVAQGLYKTSLVGPSQETEQHSQRKAAVTIQKAFRKMVTRRLEKQRRAAVRIQS
FLQMAVYRRRFLQQKRAALTLQRCFRTQQSRKQFLLYREAAVGLQNPHRTSLPAKHQREL
YLQIRSSVIIIQARVKGFIQKRKFRELKDSTIKIQAVWRRHKARKYLREVKAACRIQAWY
RCWKARREYLAVLRAVRIIQRCFCXQQQRRRFLNVRASAVIIQRRWRTVLSGRTTHEQSL
MTKRHQAACLIQANFRGYKARQAFLQQKSAALTIQRYIRARKAGKHQRMKYVELKKSTVV
LQALVRGWLVRKRISEQRAKIRLLHFAAAAFYHLSALRIQRAYRRHVALKHANNKQLNSA
ICIQRWFRARSQRKRFLQKYSIINIQREAREQARQHSRAASVIQRAVRRFLLRKKQENFN
KRIAKIQALWRGYSWRKKNDSTKTKAIRQRLQCVNREIREESKLYHRTAVALHHLLTYKY
LSTVLEALKHLEAVTRLSSICCEKMAQSGAISKIFVLIRSCNRSVPCMEVIRYAMQVLLN
VAKYEKTTSAIYDVENCVDTLLELMQMYQEKSGDKVADKSRSIFTKTCCLLAVLLKTTTR
ALDVQSRSKVVDRIYSLYKLTAHKHKVNTERILCKQKKNSSVSLSFFPETPVRTTMVSRL
KPDWVLRRDNV

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.8 Exemple de fichiers générés par Ascq_me

Dans cette section, nous donnons une partie ou la globalité de chaque fichier utilisé et généré par Ascq_me lors d'une identification. L'exemple utilisé concerne l'utilisation d'un spectre, simulé, contenant le cytochrome-C bovin que l'on compare avec une base de données contenant le cytochrome-C bovin mais aussi la très grosse protéine présentée dans la section précédente.

B.8.1 Fichier de configuration

B.8.1.1 Version longue

Ci-dessous est présenté un extrait d'un fichier type de configuration utilisée par Ascq_me. Il s'agit de la version complète et documentée d'un fichier de configuration. Il reste cependant tout à fait possible de donner une version simple où seuls les paramètres qui varient peuvent être donnés.

```
# Original author : Jean-Charles BOISSON (2005-2008)
#
# Default configuration file for ASCQ_ME
# All the line with a '#' for first character are considered as commentaries
#
# This file is splitted in different section
# For all the parameters, the style is :
#         parameter_name (space) = (space) value or the key word DEFAULT
#
# To change a value, delete the key word DEFAULT and put a value
# ex : You want to accord 1 missed cleavages so the line
# MISSED_CLEAVAGES = DEFAULT become MISSED_CLEAVAGES = 1
# to restore the old value, write MISSED_CLEAVAGES = DEFAULT
#
##################################
# Digestion section
##################################
#
# Give the number of allowed missed clevages
# DEFAULT = 0
MISSED_CLEAVAGES = 0
#
# Indicate if all the possible missed cleavages have to be used
# in this case the parameters MISSED_CLEAVAGES is not used
# DEFAULT = NO
FULL_DIGEST = DEFAULT
#
# Indicate the charge to add on peptide result of digestion
# i.e. give the chemical formula
# DEFAULT = H
PEPTIDE_CHARGE = DEFAULT
#
# gives the amino acid used when B is met
# DEFAULT = D
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008


```
AMINO_ACID_B = DEFAULT
#
# gives the amino acid used when X is met
# DEFAULT = G
AMINO_ACID_X = DEFAULT
#
# gives the amino acid used when Z is met
# DEFAULT = E
AMINO_ACID_Z = DEFAULT
#
# Give the used enzym
# look at the file internal/enztable.aprot ==> 0 = trypsin, 1 = Endoproteinase_Arg_C, ...
# add new enzym in this file if you want to use another enzyms
# DEFAULT = 0
ENZYME = DEFAULT
#
# Give the post-traductional modifications you want to activate (according to the file modif
# The syntax is :
# modif_index (according to the modification file) type : F (for fixed) or V (for variable)
# example : MODIFICATIONS = 1V3F
# DEFAULT = NONE
MODIFICATIONS = DEFAULT
#
# Give the maximum length for a peptide, if a peptide has a length greater than this value,
# the post translational modification computation (if there are activated modifications).
# If you do not need any limitation, put zero <=> MAX_PEPTIDE_LENGTH = 0
# DEFAULT = 30
MAX_PEPTIDE_LENGTH = DEFAULT
#
###################################
# I/O section
###################################
#
...
```

B.8.1.2 Version courte

La version équivalente du fichier de configuration dans laquelle seuls les paramètres dont la valeur n'est pas celle par défaut sont donnés est donnée ci-dessous.

```
FASTA_FILE = _fasta/cytc.fas
MIN_EXP_MASS = 500.0
MAX_EXP_MASS = 3500.0
EXPERIMENTAL_SPECTRUM = _spectrum_data/cytcdenature2.txt
```

B.8.2 Fichier « default_ascq_me_results.txt »

Ci-desous un extrait du fichier default_ascq_me_results.txt obtenu à partir du fichier de configuration précédent. On pourra remarquer que toutes les informations concernant la configuration de l'identifica-

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

tion sont rappelées avant de donner le classement des protéines.

```
ASCQ_ME VERSION 1.0.6
CURRENT CONFIGURATION :

Loaded from the configuration file : cytc.config

Digestion Section:
Number of missed cleavage(s) = 1
Activation of full digestion = NO
Activation of decoy mode = NO
Formula of the peptid charge = H1
Enzym used = Trypsin
Maximum peptide length = 30

I/O Section:
Does the fasta format have been chosen ? YES
Fasta file name = _fasta/cytc.fas
Wanted specie (filter) = None
Does an experimental mass have been given? NO
Number of proteins results displayed = 10

Spectra Scoring Section:
Fourier Precision = 65536
Spectrum maximum mass = 4096
Peptide Precision = 8
Peptide maximum mass = 16
Gaussian chosen = 0.300000
C quantity = 1000
H quantity = 4000
O quantity = 250
N quantity = 500
S quantity = 125
Se quantity = 100
Experimental file name = _spectrum_data/cytcdenature2.txt
Minimal experimental mass = 500.000000
Maximal experimental mass = 3500.000000
Sensitivity = 0.50
Threshold = 0.000000100
Minimum peptide number to consider = 5
Minimum matched peptide number to consider = 3

Protein 1 :
Description = CCBO cytochrome c - bovine
Sequence = GDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFSYTDANKNKGITWGEET...
Mass = 11565.020710
Score = 34.615385
Number of peptides = 41
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008


```
Score of "good" peptides = 24.292011
Peptide 1 = HKTGPNLHGLFGR of mass 1433.776611 has a score of 12.245394
Peptide 2 = TGQAPGFSYTDANKNK of mass 1698.808716 has a score of 2.694085
Peptide 3 = MIFAGIKK of mass 907.543945 has a score of 2.520749
Peptide 4 = TGPNLHGLFGR of mass 1168.622681 has a score of 2.106888
Peptide 5 = TGPNLHGLFGRK of mass 1296.717651 has a score of 1.919107
Peptide 6 = YIPGTKMIFAGIK of mass 1438.813232 has a score of 1.176171
Peptide 7 = GEREDLIAYLK of mass 1306.700684 has a score of 1.035457
Peptide 8 = KTGQAPGFSYTDANK of mass 1584.765869 has a score of 0.403884
Peptide 9 = MIFAGIK of mass 779.448975 has a score of 0.190276
Peptide 10 = EDLIAYLK of mass 964.535522 has a score of -0.554660
Peptide 11 = TGQAPGFSYTDANK of mass 1456.670898 has a score of -0.738128
Peptide 12 = EDLIAYLKK of mass 1092.630493 has a score of -0.849285
Peptide 13 = IFVQK of mass 634.392822 has a score of -1.126003
Peptide 14 = KYIPGTK of mass 806.477600 has a score of -2.453181
Peptide 15 = GITWGEETLMEYLENPKK of mass 2138.048096 has a score of -2.565499
Peptide 16 = GDVEK of mass 547.272766 has a score of -2.588467
Peptide 17 = GITWGEETLMEYLENPK of mass 2009.953003 has a score of -2.674078
Peptide 18 = GGKHK of mass 526.310181 has a score of -3.609556
Peptide 19 = KATNE of mass 562.283691 has a score of -3.618078
Peptide 20 = CAQCHTVEK of mass 1018.445007 has a score of -3.774608
Peptide 21 = GDVEKGK of mass 732.389221 has a score of -3.998237
Peptide 22 = IFVQKCAQCHTVEK of mass 1633.819458 has a score of -4.014167
Peptide 23 = YIPGTK of mass 678.382629 has a score of -4.024765
Peptide 24 = KIFVQK of mass 762.487793 has a score of -4.274108
Peptide 25 = CAQCHTVEKGGK of mass 1260.582886 has a score of -5.689468
Peptide 26 = NKGITWGEETLMEYLENPK of mass 2252.090820 has a score of -7.776789
Peptide 27 = GK of mass 204.134811 has a score of -48000.000000
Peptide 28 = K of mass 147.113358 has a score of -48000.000000
Peptide 29 = GGK of mass 261.156281 has a score of -48000.000000
Peptide 30 = HK of mass 284.172272 has a score of -48000.000000
Peptide 31 = K of mass 147.113358 has a score of -48000.000000
Peptide 32 = NK of mass 261.156281 has a score of -48000.000000
Peptide 33 = K of mass 147.113358 has a score of -48000.000000
Peptide 34 = K of mass 147.113358 has a score of -48000.000000
Peptide 35 = K of mass 147.113358 has a score of -48000.000000
Peptide 36 = GER of mass 361.183563 has a score of -48000.000000
Peptide 37 = K of mass 147.113358 has a score of -48000.000000
Peptide 38 = ATNE of mass 434.188690 has a score of -48000.000000
Peptide 39 = GKK of mass 332.229767 has a score of -48000.000000
Peptide 40 = KK of mass 275.208313 has a score of -48000.000000
Peptide 41 = KGER of mass 489.278534 has a score of -48000.000000


Protein 2 :
Description = ASPM_BOVIN (P62285) Abnormal spindle-like ...
Sequence = PNEEAAAVWLSHGPAAERGFTIWPSAFVLQPKEKIVVSITWTPLKGGRIRE.....
Mass = 395241.008195
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008


```
Score = 5.046948
Number of peptides = 1391
Score of "good" peptides = 127.545906
Peptide 1 = HSGPFEQCSESVK of mass 1434.632324 has a score of 10.380729
Peptide 2 = MHMAATLIKAAFK of mass 1432.780884 has a score of 9.533677
Peptide 3 = VSAALVIQKYWR of mass 1433.826904 has a score of 8.620093
Peptide 4 = IQSYYRAYASR of mass 1377.691528 has a score of 7.128503
                ...
        ...
```

B.8.3 Fichier « default_ascq_me_table_results.txt »

Un extrait du fichier default_ascq_me_table_results.txt obtenu à partir du fichier de configuration précédent est donné ci-dessous. La première colonne correspond à l'intervalle de masses sur lequel le spectre est défini et la seconde aux intensités du spectre expérimental. Les colonnes suivantes vont par paires et correspondent à chaque fois à un couple spectre simulé/spectre de corrélation pour une protéine donnée.

```
Proteins Protein exp protein_1 correlation_1 protein_2 correlation_2
Description None CCBO cytochrome c - bovine none ASPM_BOVIN (P62285) Abnormal spindle-like microc
none
Mass unknown 11565.020710 none 395241.008195 none
Score None 34.615385 none 5.046948 none
500.000000 0.0119051 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.125000 0.0088166 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.250000 0.0122878 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.375000 0.0162861 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.500000 0.0107884 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.625000 0.0122565 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.750000 0.0116084 -0.0000000 -0.0000000 -0.0000000 -0.0000000
500.875000 0.0153061 -0.0000000 -0.0000000 -0.0000000 -0.0000000
501.000000 0.0104721 -0.0000000 -0.0000000 -0.0000000 -0.0000000
501.125000 0.0233261 -0.0000000 -0.0000000 -0.0000000 -0.0000000
501.250000 0.0120262 -0.0000000 -0.0000000 -0.1855063 0.0000000
501.375000 0.0123151 -0.0000000 -0.0000000 -0.4297248 0.0000000
501.500000 0.0145603 -0.0000000 -0.0000000 -0.1246509 0.0000000
501.625000 0.0101364 -0.0000000 -0.0000000 -0.0147982 0.0000000
501.750000 0.0106752 -0.0000000 -0.0000000 -0.0006976 0.0000000
501.875000 0.0177815 -0.0000000 -0.0000000 -0.0000129 0.0000000
502.000000 0.0117528 -0.0000000 -0.0000000 -0.0000001 0.0000000
502.125000 0.0085940 -0.0000000 -0.0000000 -0.0006622 0.0000000
...
```

B.8.4 Fichier « default_ascq_me_peptide_scoring.txt »

Un extrait du fichier default_ascq_me_peptide_scoring.txt obtenu à partir du fichier de configuration précédent est donné ci-dessous. Pour chaque protéine, on a l'ensemble des peptides qui a été analysé pour l'établissement de son score ainsi que toutes les informations qui y sont associées. Les peptides sont ordonnés par ordre décroissant de score (réel et normalisé).

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

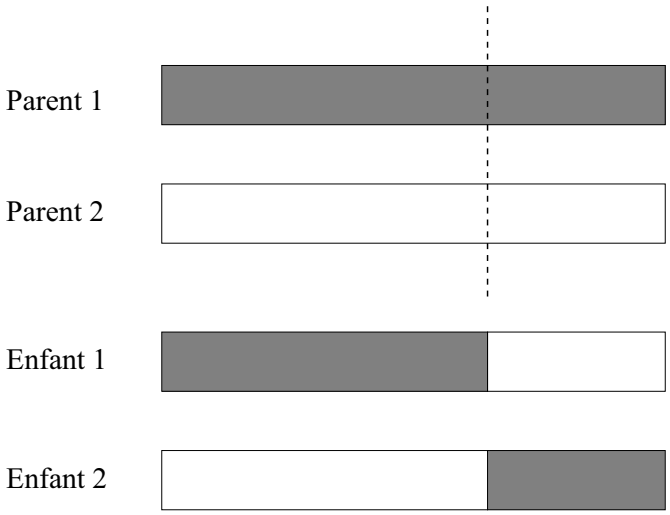

```
Rank score normalized score mass sequence

Protein 1: 41 peptides

1 20.022182 1.000000 1433.776611 HKTGPNLHGLFGR
2 10.470875 0.522964 1698.808716 TGQAPGFSYTDANKNK
3 10.297538 0.514306 907.543945 MIFAGIKK
4 9.883677 0.493636 1168.622681 TGPNLHGLFGR
5 9.695896 0.484258 1296.717651 TGPNLHGLFGRK
6 8.952960 0.447152 1438.813232 YIPGTKMIFAGIK
7 8.812246 0.440124 1306.700684 GEREDLIAYLK
8 8.180673 0.408580 1584.765869 KTGQAPGFSYTDANK
9 7.967065 0.397912 779.448975 MIFAGIK
10 7.222129 0.360706 964.535522 EDLIAYLK
11 7.038661 0.351543 1456.670898 TGQAPGFSYTDANK
12 6.927505 0.345991 1092.630493 EDLIAYLKK
13 6.650786 0.332171 634.392822 IFVQK
14 5.323608 0.265886 806.477600 KYIPGTK
15 5.211290 0.260276 2138.048096 GITWGEETLMEYLENPKK
16 5.188322 0.259129 547.272766 GDVEK
17 5.102711 0.254853 2009.953003 GITWGEETLMEYLENPK
18 4.167233 0.208131 526.310181 GGKHK
19 4.158710 0.207705 562.283691 KATNE
20 4.002181 0.199887 1018.445007 CAQCHTVEK
21 3.778552 0.188718 732.389221 GDVEKGK
22 3.762623 0.187923 1633.819458 IFVQKCAQCHTVEK
23 3.752024 0.187393 678.382629 YIPGTK
24 3.502681 0.174940 762.487793 KIFVQK
25 2.087321 0.104250 1260.582886 CAQCHTVEKGGK

16 peptides have not been used for the scoring


Protein 2: 1391 peptides

1 21.454025 1.000000 1434.632324 HSGPFEQCSESVK
2 20.606974 0.960518 1432.780884 MHMAATLIKAAFK
3 19.693390 0.917934 1433.826904 VSAALVIQKYWR
4 18.201799 0.848410 1377.691528 IQSYYRAYASR
5 17.186571 0.801088 1435.759888 IQALWRGYSWR
6 16.709929 0.778871 1563.720093 MAAPATQPAFYCHR
7 16.418129 0.765270 1430.765747 RYHWATVTIQR
8 15.889029 0.740608 1376.733398 VCTRHHVQQLR
9 15.767323 0.734935 1563.933472 VLKTAASVIQQHLR
10 15.232905 0.710025 1429.770386 HHLHQFLELQK
11 15.053904 0.701682 1375.806152 TKXAAIILQSAFR
12 15.034326 0.700769 1428.742188 AAVSLQSYFRMR
13 14.903566 0.694675 1517.801147 MHASATLIQSRFR
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008


```
14 14.897198 0.694378 1517.767334 ALEVMSNSYTFKK
15 14.811573 0.690387 1691.815063 KMAAPATQPAFYCHR
16 14.646008 0.682669 1688.846436 HLEAVTRLSSICCEK
17 14.599201 0.680488 1431.843628 AIIIIQASYQRR
18 14.265316 0.664925 1690.942627 YQQMRTAAIIIQVR
19 14.207728 0.662241 1802.914429 KPTCAVENQMETVKPK
20 14.201750 0.661962 1556.848389 HSALVLQAAFRGMR
...
```

B.9 Schéma classique d'un « crossover 1-point »



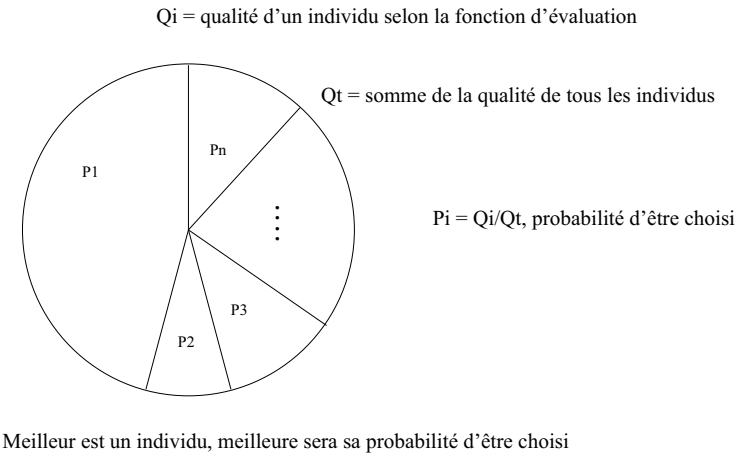
http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.10 Matrice de substitution BLOSUM62

```
SUBSTITUTION_MATRIX = BLOSUM62
BEGIN
  A  C  D  E  F  G  H  I  K  L  M  N  P  Q  R  S  T  V  W  Y
  A  0 -2 -1 -2  0 -2 -1 -1 -1 -1 -2 -1 -1 -1  1  0  0 -3 -2
     C -3 -4 -2 -3 -3 -1 -3 -1 -1 -3 -3 -3 -3 -1 -1 -1 -2 -2
        D  2 -3 -1 -1 -3 -1 -4 -3  1 -1  0 -2  0 -1 -3 -4 -3
           E -3 -2  0 -2  1 -3 -2  0 -1  2  0  0 -1 -2 -3 -2
              F -3 -1  0 -3  0  0 -3 -4 -3 -3 -2 -2 -1  1  3
                 G -2 -4 -2 -4 -3  0 -2 -2 -2  0 -2 -3 -2 -3
                    H -3 -1 -3 -2  1 -2  0  0 -1 -2 -3 -2  2
                       I -3  2  1 -3 -3 -3 -3 -2 -1  3 -3 -1
                          K -2 -1  0 -1  1  2  0 -1 -2 -3 -2
                             L  2 -3 -3 -2 -2 -2 -1  1 -2 -1
                                M -2 -2  0 -1 -1 -1  1 -1 -1
                                   N -2  0  0  1  0 -3 -4 -2
                                      P -1 -2 -1 -1 -2 -4 -3
                                         Q  1  0 -1 -2 -2 -1
                                            R -1 -1 -3 -3 -2
                                               S  1 -2 -2 -2
                                                  T  0 -2 -2
                                                     V -3 -1
                                                        W  2
                                                           Y
END
```

B.11 Sélection roulette



http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.12 Données expérimentales de validation de l'approche SSO

B.12.1 Séquences FASTA

B.12.1.1 Cytochrome-C bovin

```
>sp|P62894|CYC_BOVIN Cytochrome c OS=Bos taurus GN=CYCS PE=1 SV=2
MGDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFSYTDANKNKGITW
GEETLMEYLENPKKYIPGTKMIFAGIKKKGEREDLIAYLKKATNE
```

B.12.1.2 Sérum d'albumine bovine

```
>sp|P02769|ALBU_BOVIN Serum albumin OS=Bos taurus GN=ALB PE=1 SV=4
MKWVTFISLLLLFSSAYSRGVFRRDTHKSEIAHRFKDLGEEHFKGLVLIAFSQYLQQCPF
DEHVKLVNELTEFAKTCVADESHAGCEKSLHTLFGDELCKVASLRETYGDMADCCEKQEP
ERNECFLSHKDDSPDLPKLKPDPNTLCDEFKADEKKFWGKYLYEIARRHPYFYAPELLYY
ANKYNGVFQECCQAEDKGACLLPKIETMREKVLASSARQRLRCASIQKFGERALKAWSVA
RLSQKFPKAEFVEVTKLVTDLTKVHKECCHGDLLECADDRADLAKYICDNQDTISSKLKE
CCDKPLLEKSHCIAEVEKDAIPENLPPLTADFAEDKDVCKNYQEAKDAFLGSFLYEYSRR
HPEYAVSVLLRLAKEYEATLEECCAKDDPHACYSTVFDKLKHLVDEPQNLIKQNCDQFEK
LGEYGFQNALIVRYTRKVPQVSTPTLVEVSRSLGKVGTRCCTKPESERMPCTEDYLSLIL
NRLCVLHEKTPVSEKVTKCCTESLVNRRPCFSALTPDETYVPKAFDEKLFTFHADICTLP
DTEKQIKKQTALVELLKHKPKATEEQLKTVMENFVAFVDKCCAADDKEACFAVEGPKLVV
STQTALA
```

B.12.1.3 Ovalbumine de poulet

```
>sp|P01012|OVAL_CHICK Ovalbumin OS=Gallus gallus GN=SERPINB14 PE=1 SV=2
MGSIGAASMEFCFDVFKELKVHHANENIFYCPIAIMSALAMVYLGAKDSTRTQINKVVRF
DKLPGFGDSIEAQCGTSVNVHSSLRDILNQITKPNDVYSFSLASRLYAEERYPILPEYLQ
CVKELYRGGLEPINFQTAADQARELINSWVESQTNGIIRNVLQPSSVDSQTAMVLVNAIV
FKGLWEKAFKDEDTQAMPFRVTEQESKPVQMMYQIGLFRVASMASEKMKILELPFASGTM
SMLVLLPDEVSGLEQLESIINFEKLTEWTSSNVMEERKIKVYLPRMKMEEKYNLTSVLMA
MGITDVFSSSANLSGISSAESLKISQAVHAAHAEINEAGREVVGSAEAGVDAASVSEEFR
ADHPFLFCIKHIATNAVLFFGRCVSP
```


http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.12.2 Résultat d'identification de MASCOT (spectres MS/MS)

B.12.2.1 Cytochrome-C bovin

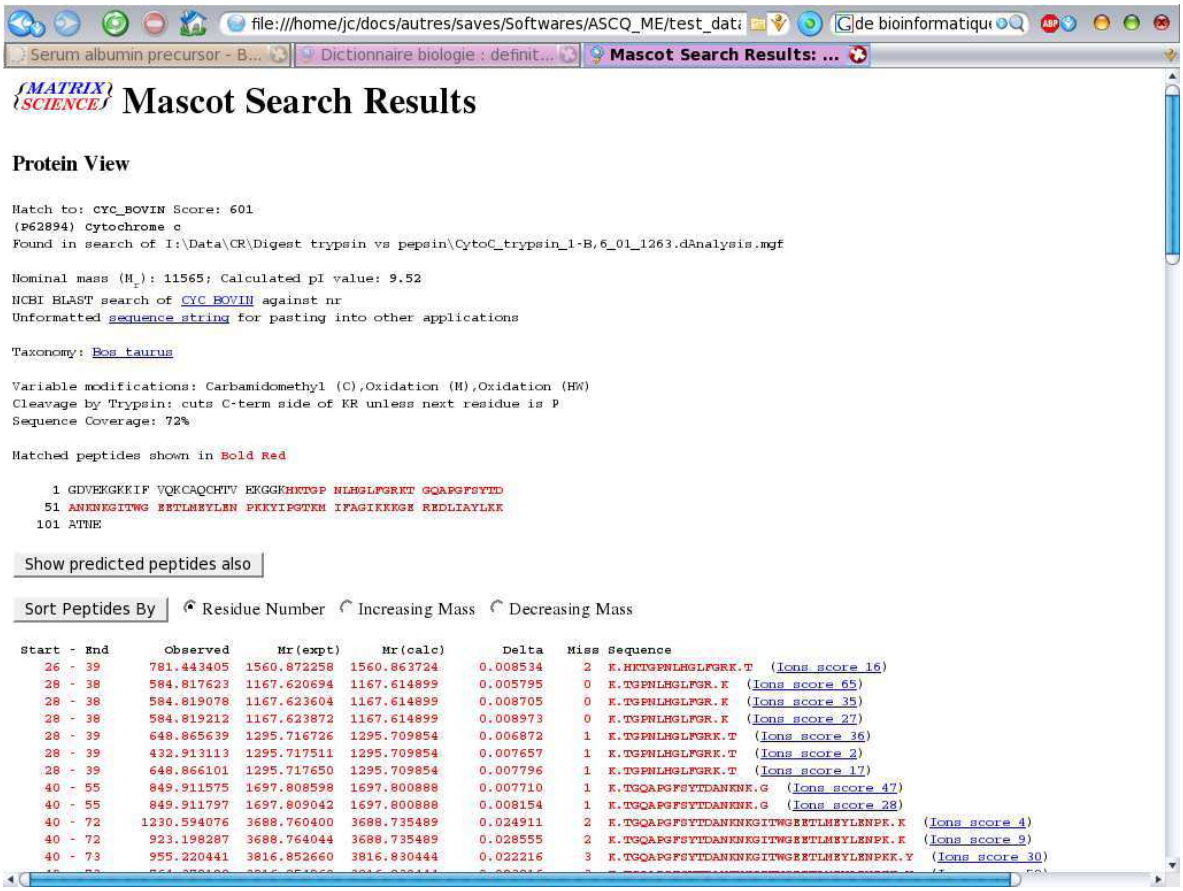


FIG. B.8 – Capture d'écran de la page des résultats d'identification du cytochrome-c bovin par MASCOT à partir de spectres MS/MS.

http://doc.univ-lille1.fr

B.12.2.2 Sérum d'albumine bovine

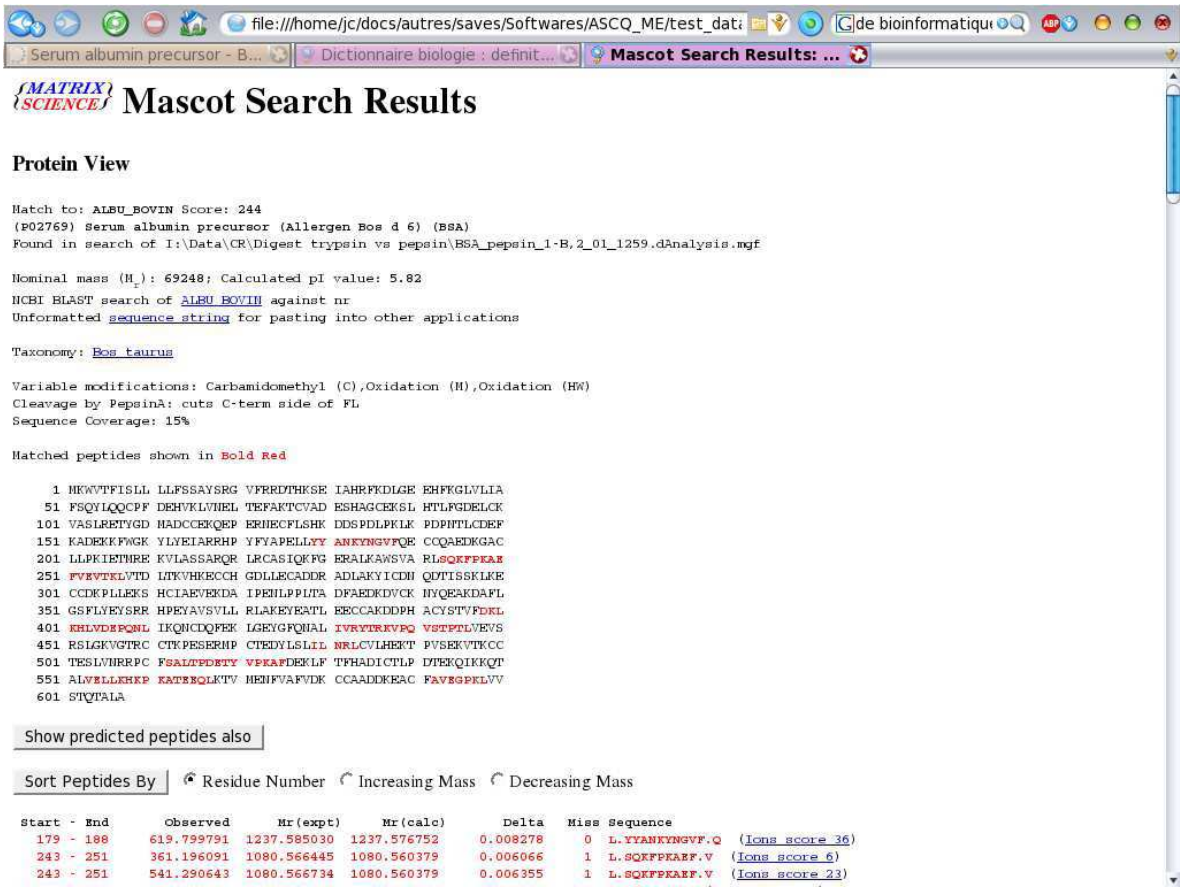


FIG. B.9 – Capture d'écran de la page des résultats d'identification du sérum d'albumine bovine par MASCOT à partir de spectres MS/MS.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

B.12.2.3 Ovalbumine de poulet

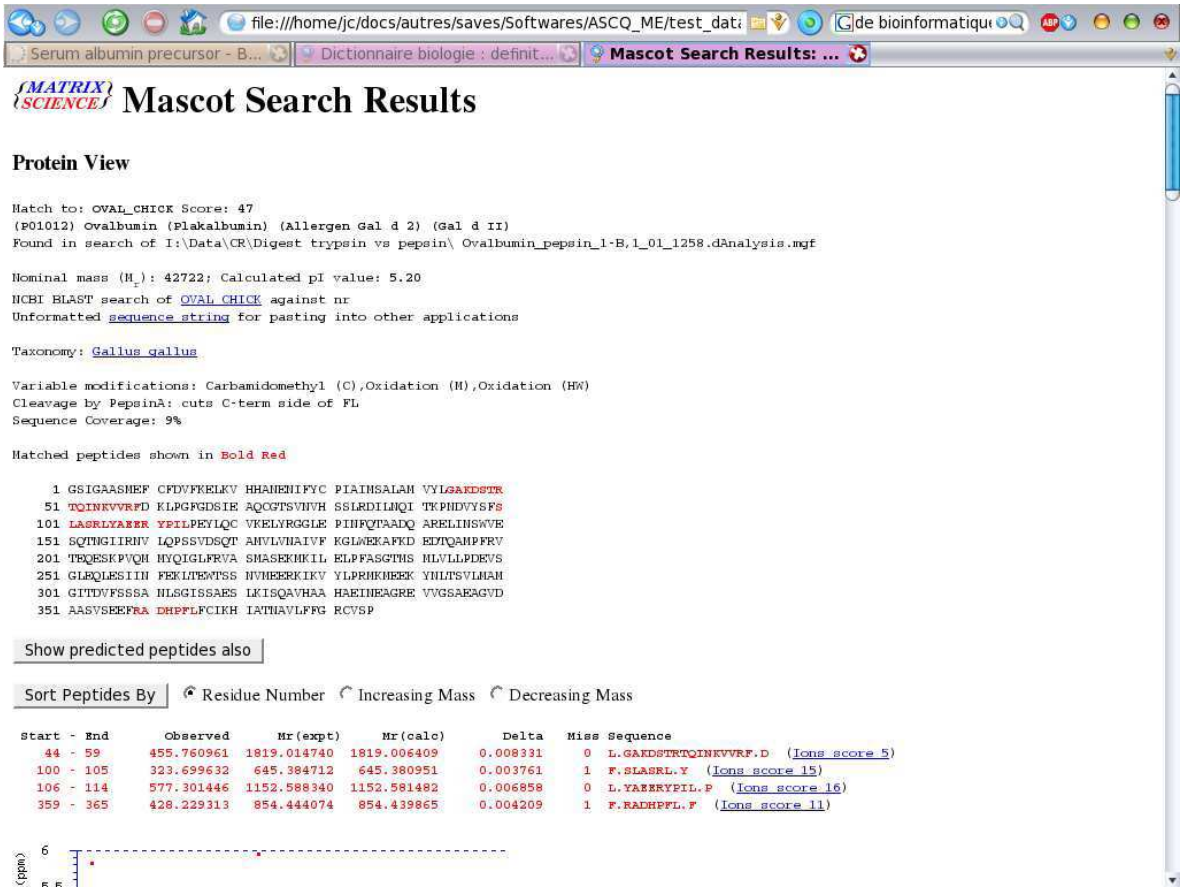


FIG. B.10 – Capture d'écran de la page des résultats d'identification de l'ovalbumine de poulet par MASCOT à partir de spectres MS/MS.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

C

Données complémentaires pour le docking moléculaire

C.1 Longueur et énergie d'un ensemble de liaisons

Liaison	Longueur (Å)	Energie (kJ/mol)	Liaison	Longueur (Å)	Energie (kJ/mol)
$H-H$	0,74	436	$C-C$	1,54	348
$H-N$	1,01	391	$C=C$	1,34	614
$H-F$	0,92	568	$C\equiv C$	1,20	839
$H-Br$	1,41	366	$C-N$	1,47	308
$H-C$	1,09	413	$C-O$	1,43	360
$H-O$	0,96	366	$C-F$	1,35	488
$H-Cl$	1,27	432	$C-Cl$	1,77	330
$N-H$	1,54	348	$C-Br$	1,94	288
$N-C$	1,47	308	$C-I$	2,14	216
$N-N$	1,45	170	$C-S$	1,82	272
$N\equiv N$	1,10	945	$O-H$	0,96	366
$F-H$	0,92	568	$O-C$	1,43	360
$F-F$	1,42	158	$O-O$	1,48	145
$F-C$	1,35	488	$O=O$	1,21	498
$Cl-H$	1,27	432	$Br-H$	1,41	366
$Cl-C$	1,77	330	$Br-C$	1,94	288
$Cl-Cl$	1,99	243	$Br-Br$	2,28	193

C.2 Les 4 règles de Gillespie

1. La géométrie dépend de l'ensemble des doublets électroniques de la couche de valence de chaque atome.
2. Les doublets de la couche de valence se disposent autour de l'atome de façon à s'éloigner le plus possible l'un de l'autre.
3. Les doublets non partagés exercent des répulsions plus grandes que les doublets de liaison et les répulsions exercées par ces derniers sont d'autant plus faibles que l'atome lié est plus électronéga-

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

tif.

4. Les liaisons multiples ne sont considérées que comme un seul centre de répulsion.

C.3 Rayon de Van der Waals d'un ensemble d'élément

Elément	Symbole	Rayon de Van der Waals (Å)
Hydrogène	H	1,2
Hélium	He	1,22
Bore	B	2,08
Carbone	C	1,85
Azote	N	1,54
Oxygène	O	1,40
Fluor	F	1,35
Néon	Ne	1,60
Sodium	Na	2,31
Aluminium	Al	2,05
Silicium	Si	2,00
Phosphore	P	1,90
Soufre	S	1,85
Chlore	Cl	1,81
Argon	Ar	1,91
Potassium	K	2,31
Arsenic	As	2,00
Sélénium	Se	2,00
Brome	Br	1,95
Krypton	Kr	1,98
Rubidium	Rb	2,44
Antimoine	Sb	2,20
Tellure	Te	2,20
Iode	I	2,15
Césium	Cs	2,62
Bismuth	Bi	2,40

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

C.4 Paysages d'instances

C.4.1 Instance 2tsc

Les figures suivantes présentent les paysages associés à l'instance 2tsc selon des rotations ou des translations du ligand à partir de sa position considérée comme optimale dans le site.

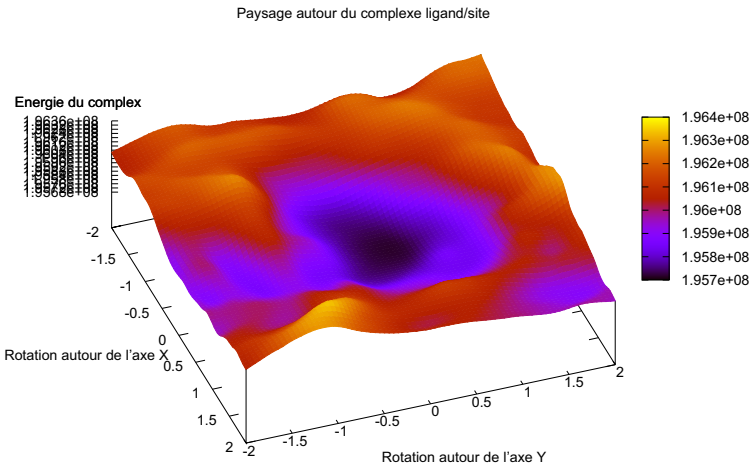


FIG. C.1 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des rotations selon les axes X et Y.

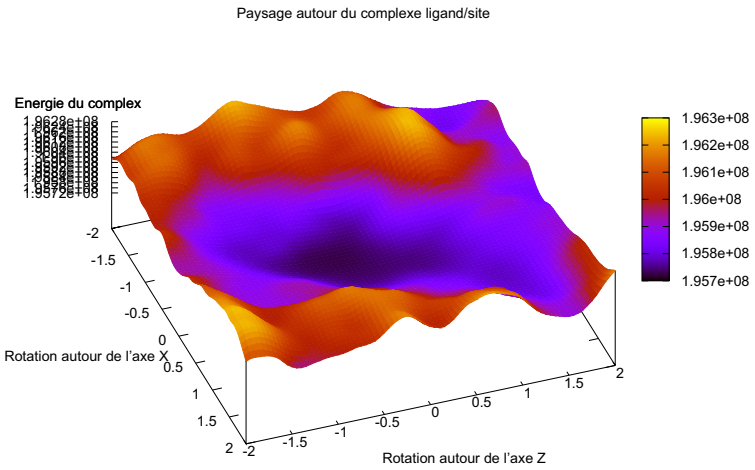


FIG. C.2 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

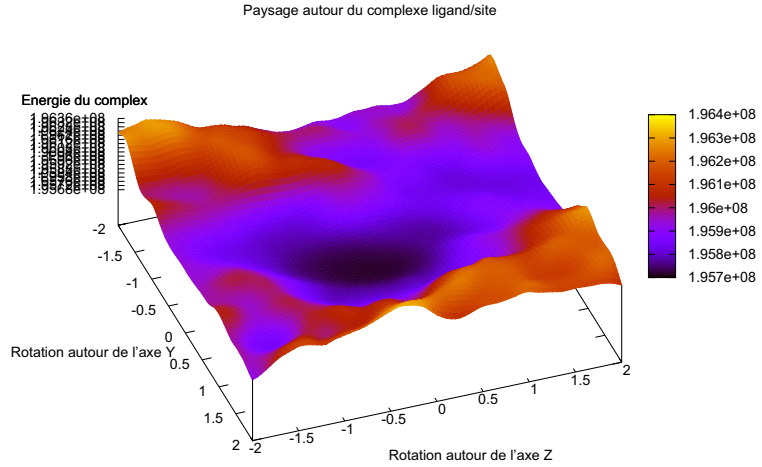


FIG. C.3 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des rotations selon les axes Y et Z.

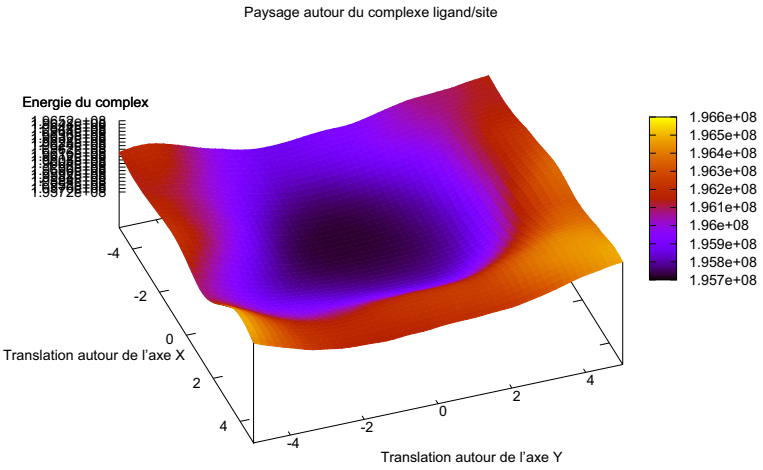


FIG. C.4 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des translations selon les axes X et Y.

http://doc.univ-lille1.fr

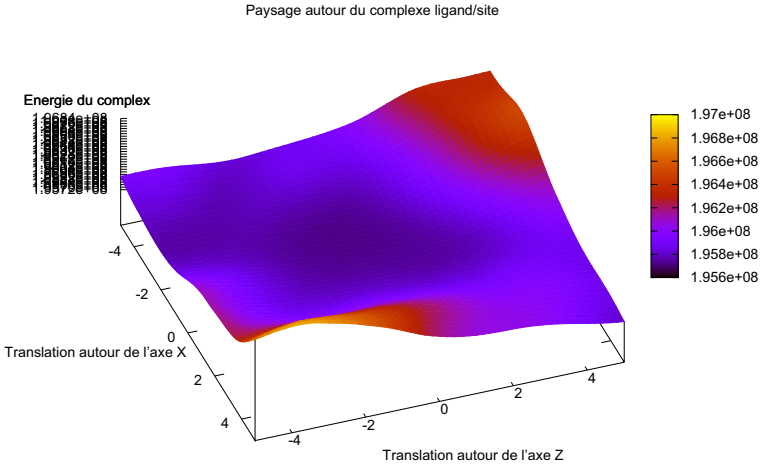


FIG. C.5 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des translations selon les axes X et Z.

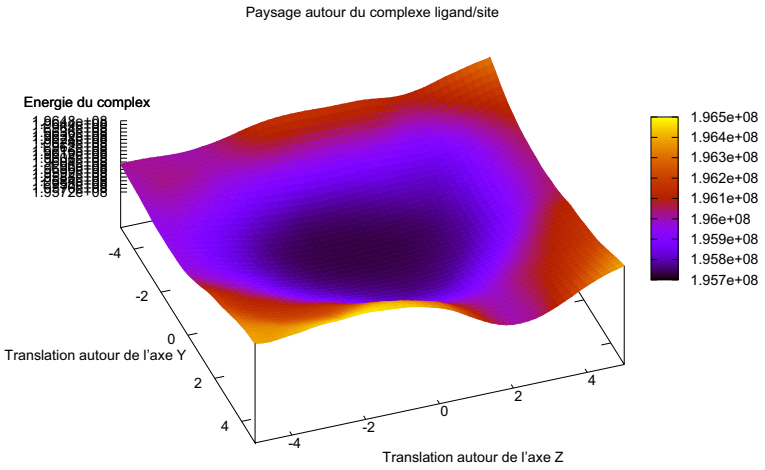


FIG. C.6 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec le ICVFF. Le ligand a subi des translations selon les axes Y et Z.

http://doc.univ-lille1.fr

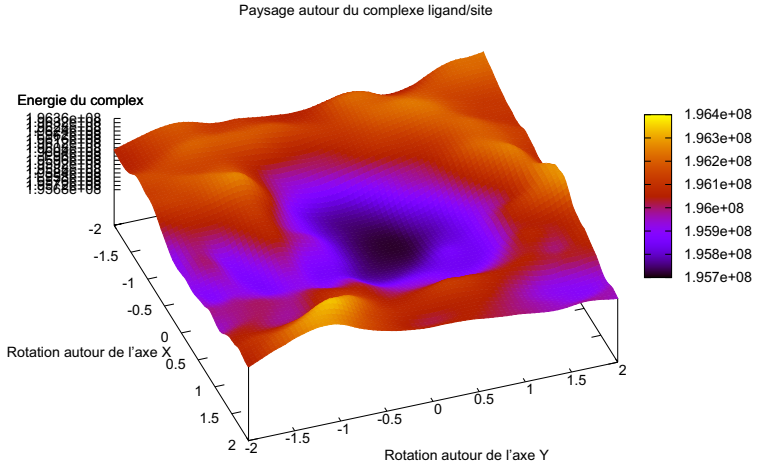


FIG. C.7 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des rotations selon les axes X et Y.

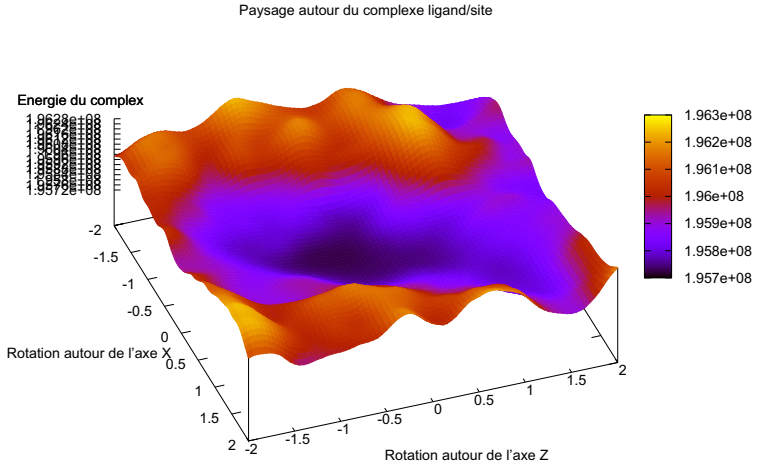


FIG. C.8 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

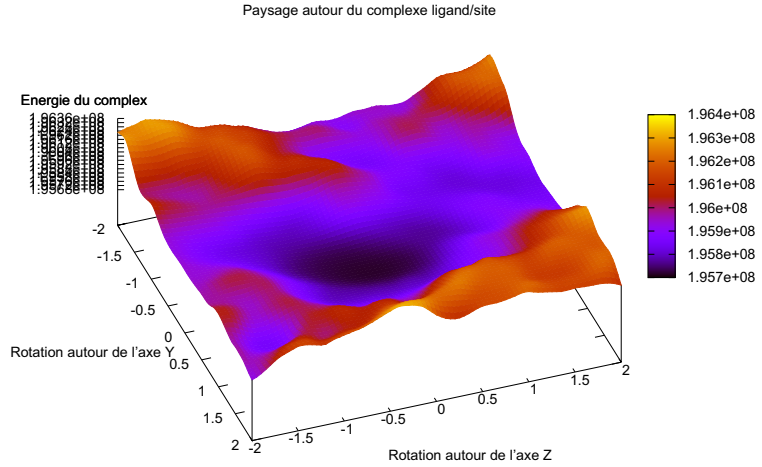


FIG. C.9 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des rotations selon les axes Y et Z.

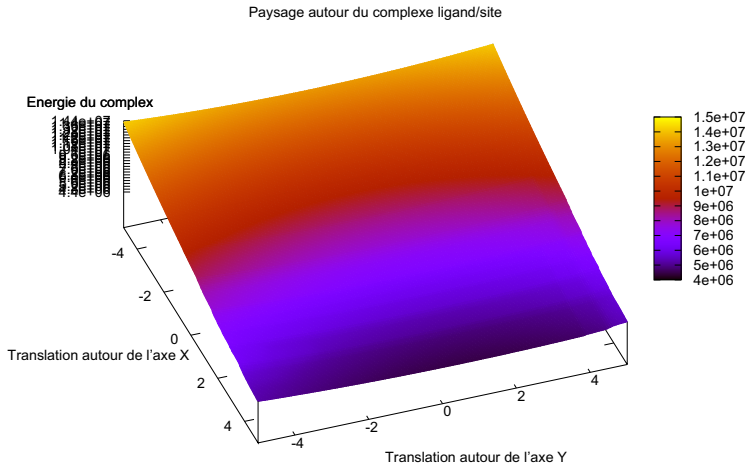


FIG. C.10 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des translations selon les axes X et Y.

http://doc.univ-lille1.fr

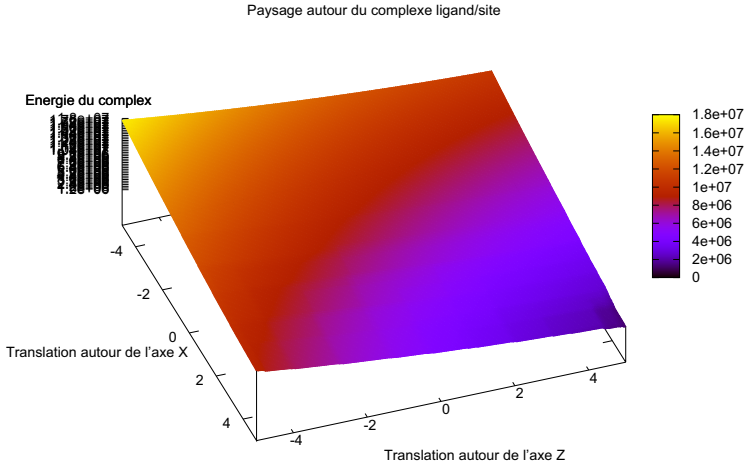


FIG. C.11 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des translations selon les axes X et Z.

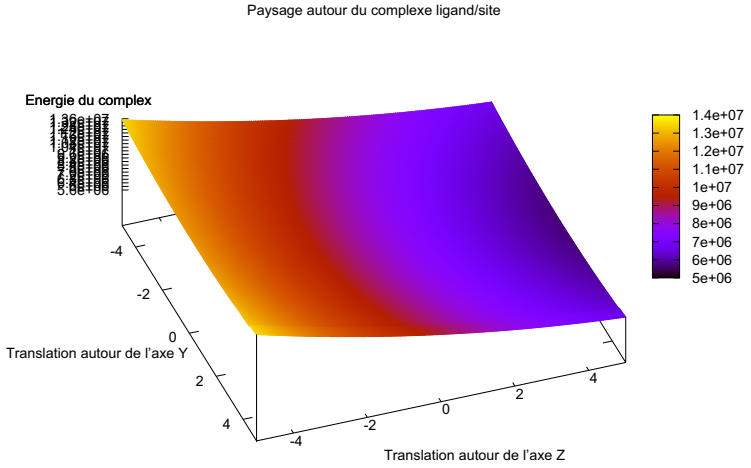


FIG. C.12 – Paysage énergétique autour du complexe ligand/site pour l'instance 2tsc avec Autodock 4.0. Le ligand a subi des translations selon les axes Y et Z.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

C.4.2 Instance 1htf

Les figures suivantes présentent les paysages associés à l'instance 1htf selon des rotations ou des translations du ligand à partir de sa position considérée comme optimale dans le site.

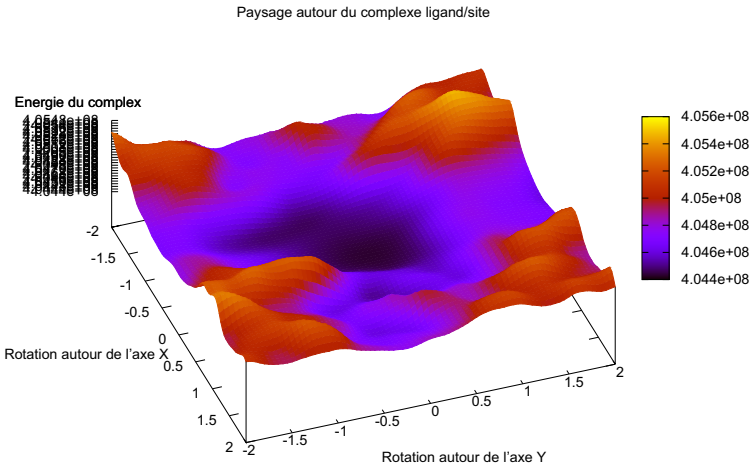


FIG. C.13 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des rotations selon les axes X et Y.

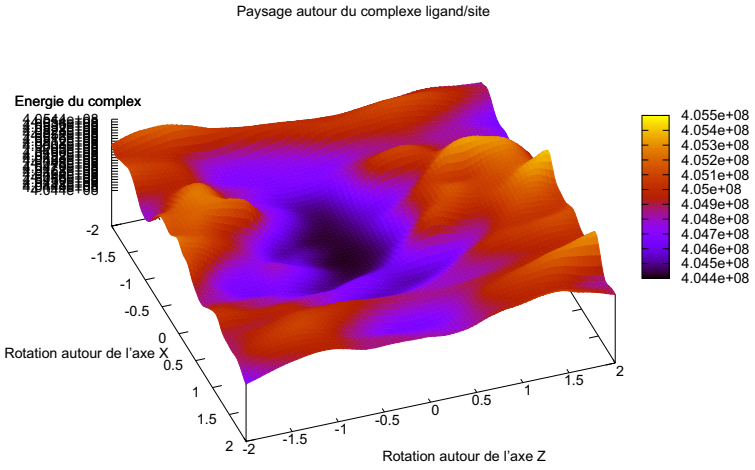


FIG. C.14 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

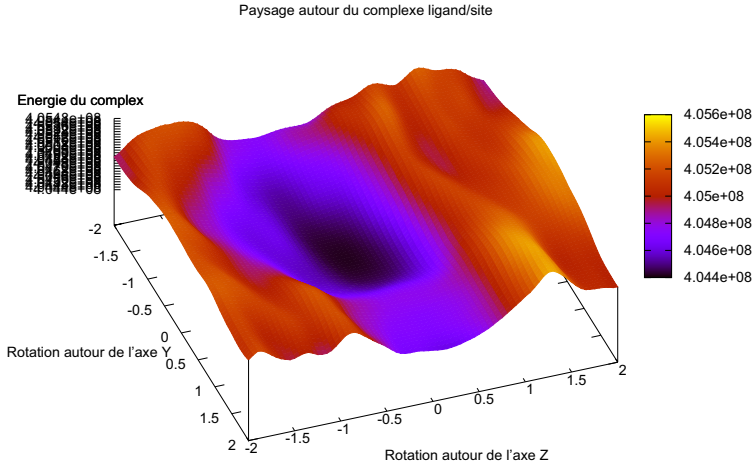


FIG. C.15 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des rotations selon les axes Y et Z.

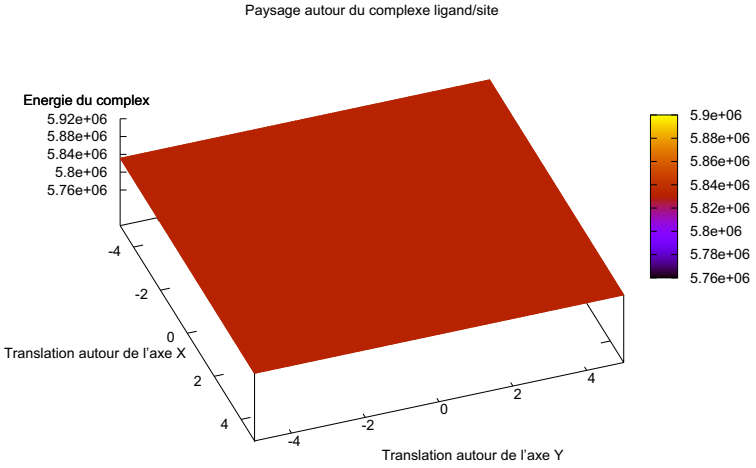


FIG. C.16 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des translations selon les axes X et Y.

http://doc.univ-lille1.fr

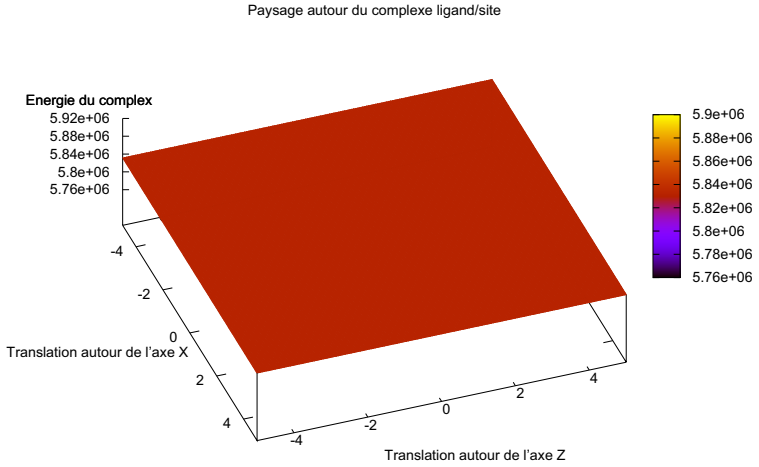


FIG. C.17 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des translations selon les axes X et Z.

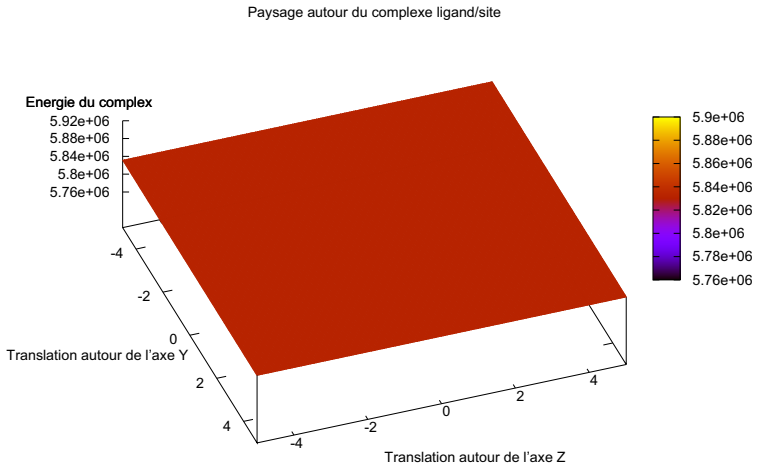


FIG. C.18 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec le ICVFF. Le ligand a subi des translations selon les axes Y et Z.

http://doc.univ-lille1.fr

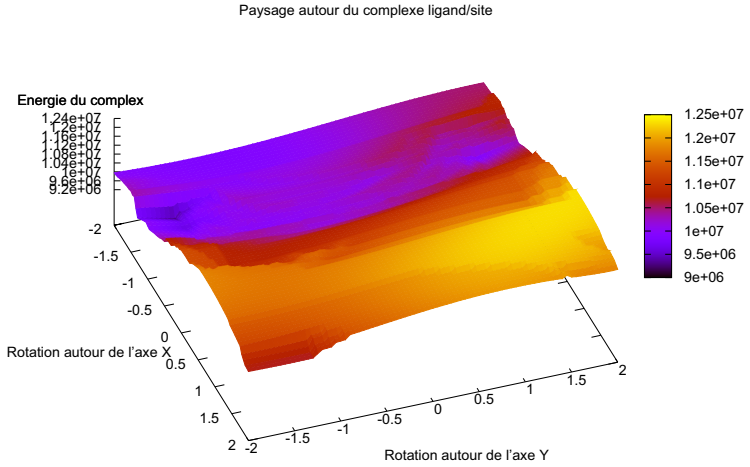


FIG. C.19 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des rotations selon les axes X et Y.

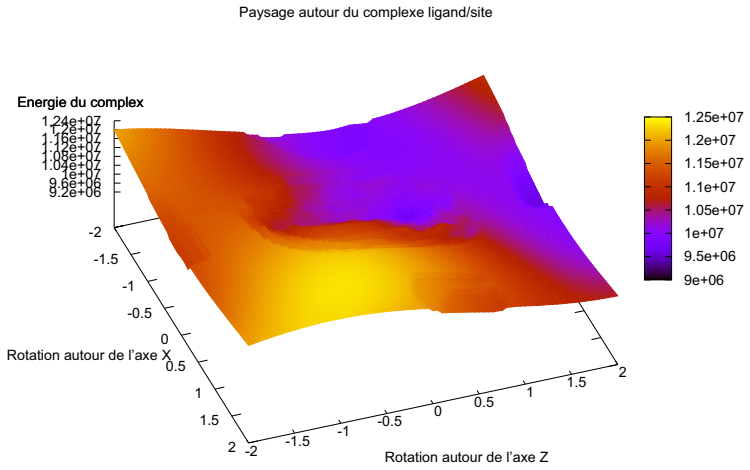


FIG. C.20 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des rotations selon les axes X et Z.

http://doc.univ-lille1.fr

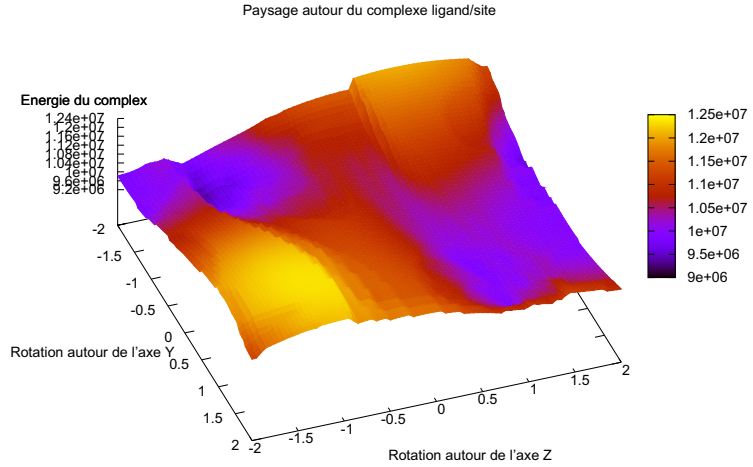


FIG. C.21 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des rotations selon les axes Y et Z.

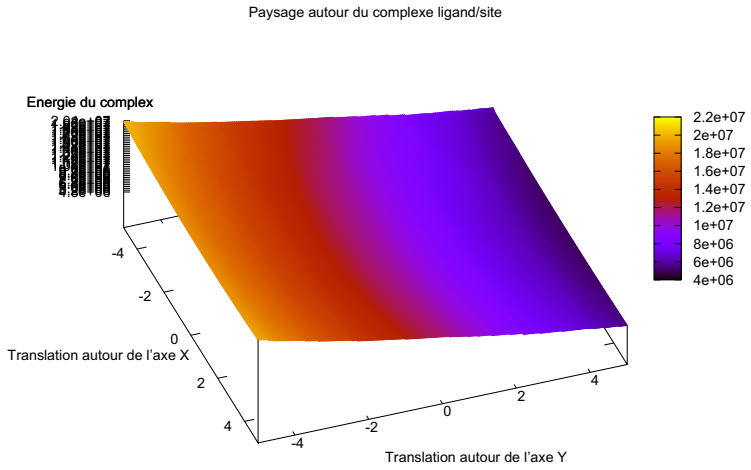


FIG. C.22 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des translations selon les axes X et Y.

http://doc.univ-lille1.fr

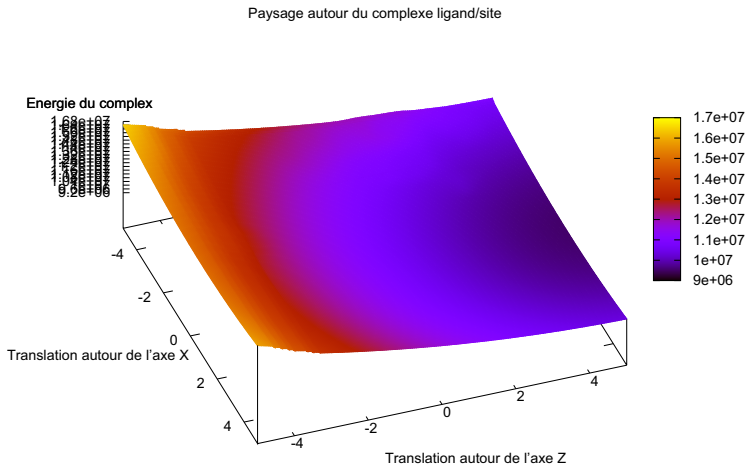


FIG. C.23 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des translations selon les axes X et Z.

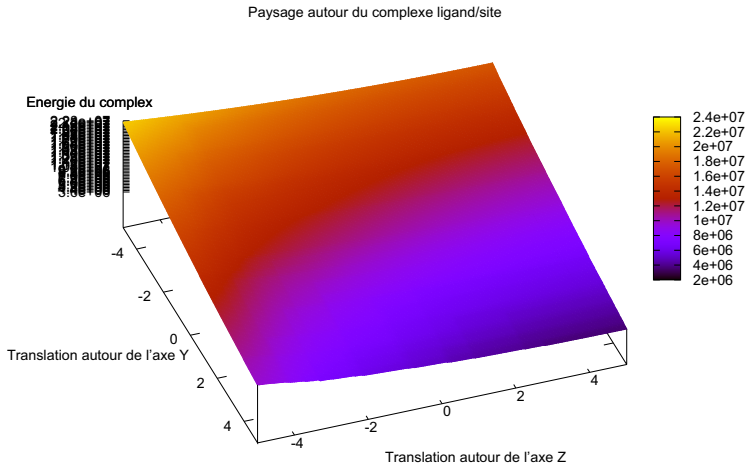


FIG. C.24 – Paysage énergétique autour du complexe ligand/site pour l'instance 1htf avec Autodock 4.0. Le ligand a subi des translations selon les axes Y et Z.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

C.5 Graphes des dépendances pour les différentes parties du docking

La figure C.25 présente le graphe des dépendances de la bibliothèque de manipulation de molécules utilisée par le docking. Cette bibliothèque n'est pas spécifique au docking et a déjà été utilisé pour le problème de prédiction de structure de protéine. Les figures C.26 et C.28 donne le graphe de dépendances d'Autodock 4.0. Pour ce faire, l'exécutable d'Autodock a été transformé en bibliothèque pouvant être incluse pour une utilisation en tant que fonction d'évaluation. Le graphe de dépendance de docking en lui-même est présenté via les figures C.27 et C.29. Dans les graphes d'Autodock et du docking, seuls les fichiers principaux sont donnés pour avoir des figures de taille raisonnable (pas ou peu de fichiers systèmes et/ou de fichiers de ParadisEO).

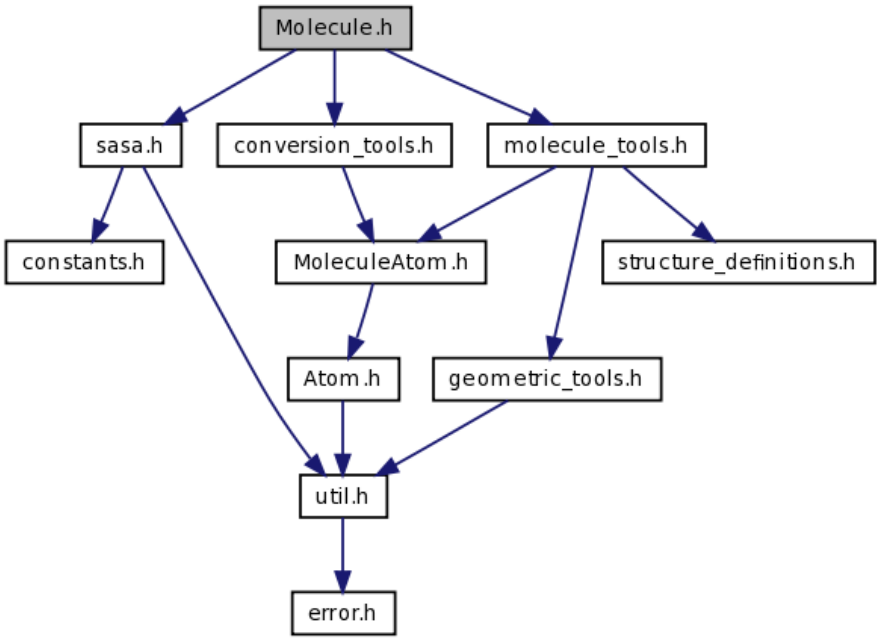


Fig. C.25 – Graphe des dépendances de la partie « core » du docking. Ce schéma a été généré automatiquement avec la documentation grâce à doxygen. Les inclusions des entêtes système ont été enlevées pour des soucis de lisibilité. Cette partie représente environ 14 800 lignes de code

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

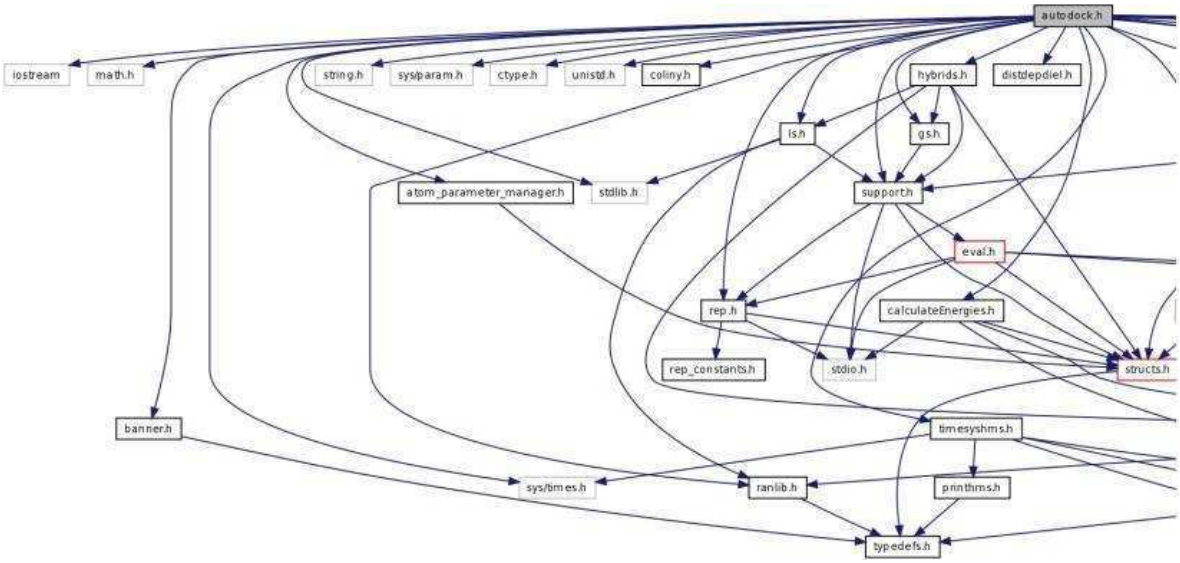


FIG. C.26 – Graphe des dépendances d'Autodock 4 (partie 1/2). Ce schéma a été généré automatiquement avec la documentation grâce à doxygen.

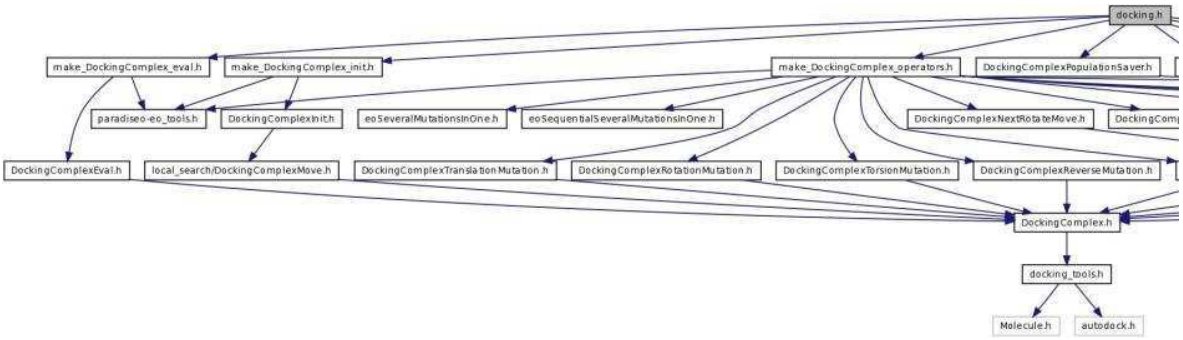


FIG. C.27 – Graphe des dépendances nécessaire au docking par l'intermédiare du fichier *docking.h* (partie 1/2). Ce schéma a été généré automatiquement avec la documentation grâce à doxygen. Les inclusions des entêtes système et des inclusions à ParadisEO ont été enlevées pour des soucis de lisibilité. La partie docking représente environ 9 800 lignes de code

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

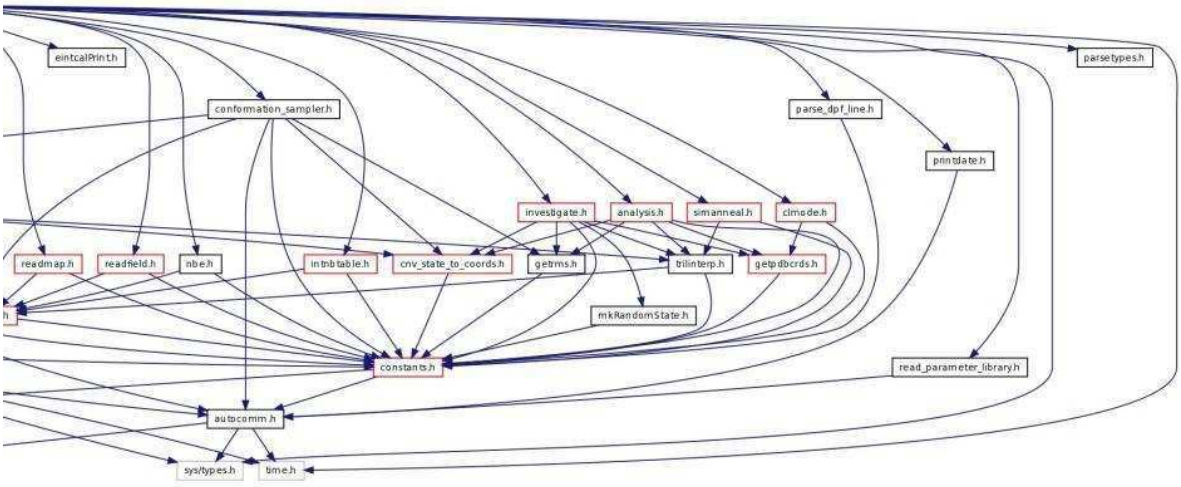


FIG. C.28 – Graphe des dépendances d'Autodock 4 (partie 2/2). Ce schéma a été généré automatiquement avec la documentation grâce à doxygen.

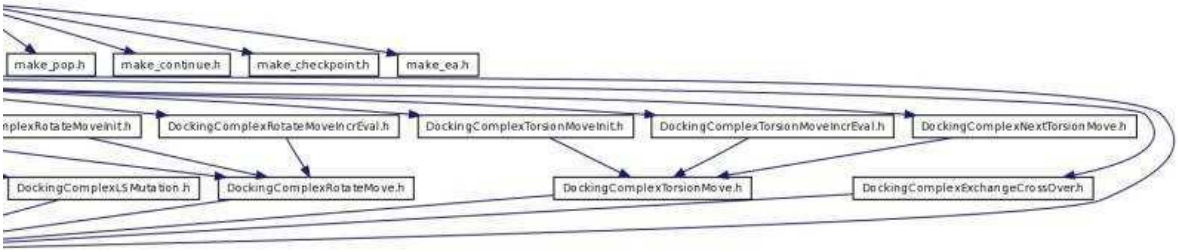


FIG. C.29 – Graphe des dépendances nécessaire au docking par l'intermédiare du fichier *docking.h* (partie 2/2). Ce schéma a été généré automatiquement avec la documentation grâce à doxygen. Les inclusions des entêtes système et des inclusions à ParadisEO ont été enlevées pour des soucis de lisibilité.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

C.6 Capture d'écran de Docking@GRID

Les figures C.30 et C.31 montrent des captures d'écran de la plate-forme Docking@GRID.

Fig. C.30 – Capture d'écran de la page de garde Docking@GRID. Il faut être un utilisateur enregistré pour pouvoir utiliser la plateforme.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

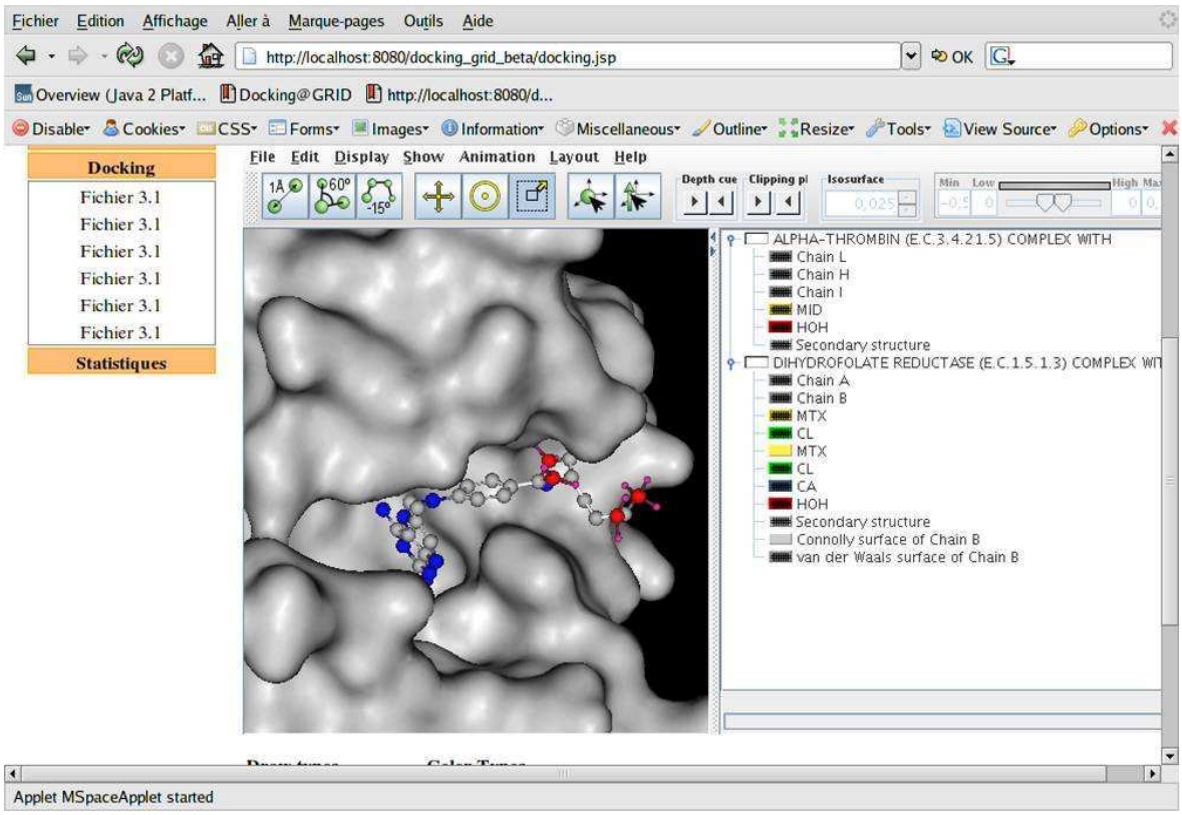


FIG. C.31 – Capture d'écran d'un résultat de docking visualisé grâce à un des outils de Chemaxon.

http://doc.univ-lille1.fr

Thèse de Jean-Charles Boisson, Lille 1, 2008

http://doc.univ-lille1.fr

http://doc.univ-lille1.fr

Résumé

A travers cette thèse, nous montrons l'importance de la modélisation et de la coopération de métaheuristiques pour la résolution de problèmes réels en bioinformatique. Pour ce faire, deux problèmes ont été étudiés : le premier dans le domaine de la protéomique pour l'identification de protéines à partir de données spectrales et le second dans le domaine de l'analyse structurale de molécules pour le problème du docking moléculaire flexible. Ainsi, pour le premier problème, un nouveau modèle basé sur une comparaison directe des bases de données protéiques avec les données expérimentales brutes a été mise en place. L'approche associée a été intégrée au sein d'un moteur d'identification par empreinte de masse peptide appelé ASCQ_ME. Ce modèle d'identification a permis ensuite de proposer et de valider une modélisation pour le problème de « de novo **protein** sequencing » qui consiste à retrouver la séquence d'une protéine à partir seulement des données expérimentales. Il s'agit d'un modèle en trois étapes appelé SSO pour « Sequence », « Shape » et « Order ». Après une étude de chacune de ces étapes, SSO a été implémenté et testé à travers trois métaheuristiques collaborant de manière séquentielle. Pour le second problème, une étude des nouvelles modélisations multi-objectives a été menée et a conduit à la définition d'un ensemble de huit modèles différents testés à l'aide d'algorithmes génétiques multi-objectifs parallèles. Une douzaine de configuration d'opérateurs génétiques ont été testé afin de mettre en évidence l'efficacité de l'hybridation des algorithmes génétiques avec des recherches locales. Pour chacune des parties, l'implémentation et la mise en place des collaborations fut possible grâce à la plateforme ParadisEO et notamment grâce à mes contributions à la partie ParadisEO-MO dédiée aux métaheuristiques à base de solution unique. L'ensemble de ces travaux a été soutenu par le PPF BioInformatique de l'Université des Sciences et Technologies de Lille et le projet ANR Dock.

Mots-clés: Modélisation, coopération, métaheuristique, protéomique, docking moléculaire flexible, ParadisEO.

Abstract

In this thesis, we show the importance of the modeling and the cooperation of metaheuristics for solving real problems in Bioinformatics. Two problems are studied : the first in the Proteomics domain for the protein identification from spectral data analysis and the second in the domain of the structural analysis of molecules for the flexible molecular docking problem. So, for the first problem, a new model has been designed based on a direct comparison of a raw experimental spectrum with protein from databases. This model has been included in an identification engine by peptide mass fingerprinting called ASCQ_ME. From this model, an approach for the « de novo **protein** sequencing » problem has been proposed and validated. In this problem, a protein sequence has to be found with only spectral information. Our model is a three step approach called SSO for « Sequence », « Shape » and « Order ». After a study of each step, SSO has been implemented and tested with three metaheuristics collaborating sequentially. For the second problem, a study of new multi-objective models has been made and has allowed to design eight different models tested with parallel multi-objective genetic algorithms. Twelve configurations of genetic operators has been tested in order to prove the efficiency of the hybridizing of genetic algorithms with local searches. For each part of this work, the ParadisEO platform has been used and more particularly the ParadisEO-MO part dedicated to single solution based metaheuristics for which we have substantially contributed. All this work has been funded by the « PPF BioInformatique » of the « Université des Sciences et Technologies de Lille » and by the ANR Dock project.

Keywords: Modeling, cooperation, metaheuristic, Proteomics, flexible molecular docking, ParadisEO.

http://doc.univ-lille1.fr