République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Larbi Ben M'hidi Oum El Bouaghi
Faculté Sciences Exacte et des Sciences de la nature et de la vie
Département Mathématique et Informatique

Thèse de Doctorat en Science

-2021-

Présentée pour l'obtention du grade de Docteur en Informatique

Par : **Mahdi Samir**

Titre

Optimisation multiobjectif et son application aux problèmes bioinformatique

Soutenue devant le jury composé de :

Président du jury	Pr. Boutekkouk Fateh	Université de Larbi Ben M'hidi
Rapporteur	Pr. Nini Brahim	Université de Larbi Ben M'hidi
Examinateur	Pr. Chikhi Salim	Université de Constantine 2
Examinateur	Dr. Ghanem Khaddoudja	Université de Constantine 2

Remerciements et Dédicace

Je tiens tout d'abord à remercier mon encadrant, Brahim NINI, pour son support et son aide précieux.

Je remercie le Président et les membres de jury pour s'être déplacés et participer à l'évaluation de ma thèse et pour le temps qu'ils ont passé à le lire.

Je dédie cette humble thèse

A la mémoire de mon père

A la mémoire de ma mère

A mes frères et sœurs

A ma fille et mes fils

ملخص

يندرج النشاط البحثي لهذه الأطروحة تحت موضوعين رئيسيين: التحسين متعدد الأهداف والمعلوماتية الحيوية. يمكن صياغة العديد من المشكلات التي تمت مواجهتها في المعلوماتية الحيوية على أنها مشكلات تتعلق بالتحسين يجلب النهج متعدد الأهداف العديد من المزايا لعلماء الأحياء من حيث الجودة واختيار الحل الذي سيتم اعتماده. يتضمن ذلك، من ناحية، الاستفادة من الجانب المتناقض للأهداف للوصول إلى حل وسط جيد (توازن) تحسين الجودة (البيولوجية) للحلول، ومن ناحية أخرى، إمكانية الحصول على عدة حلول واحدة. وبالتالي، فإن الحلول تعطي مزيدًا من الخيارات لصانع القرار للحصول على حلول ذات مغزى بيولوجيًا. في هذه الأطروحة، سننشئ حالة من الفن تصف ميزة التطبيق المحتمل للتحسين متعدد الأهداف في مشاكل المعلوماتية الحيوية المختلفة. كدراسة حالة، اخترنا المشكلة الأكثر مناقشة في المعلوماتية الحيوية: محاذاة التسلسل المتعدد (MSA). الهدف هو تحديد القضايا المختلفة، واقتراح مخططات حل جديدة والتحقق من صحة الأساليب التي تم تطويرها على أساس المعايير. الدافع الرئيسي لهذا العمل البحثي هو المساهمة في تطوير خوارزميات أمثلية متعددة الأغراض لجعل تطبيقها على مشاكل المعلوماتية الحيوية بأكبر قدر ممكن من الكفاءة من حيث الوقت الحسابي (المعقول)، كحلول منتجة وقادرة على التعامل مع المشاكل الكبيرة. الأحجام. نحن مهتمون بتطوير الطريقة méta-exacte الهجينة. لقد صممنا طريقة بحث محلية لـ GPLS تعمل على مواضع الفجوات لتحسين جميع الأحفاد التي تنتجها NSGA-II: خوارزمية التحسين المعيارية متعددة الأهداف. لزيادة دقة خوارزمية منتج M-NSGA-II، سنطبق الخوارزمية الدقيقة لـ Needleman و Wunsch على المحاذاة الفرعية لبعض حلول Pareto front.

Summary

The research activity of this project consists of two main themes: multiobjective optimization and bioinformatics. Many problems encountered in bioinformatics can be formulated as optimization problems. The multiobjective approach brings many advantages to biologists in terms of quality and choice of the solution to be adopted. This involves, on the one hand, taking advantage of the trade-off between the different objectives to reach a good compromise (a balance), improving the (biological) quality of the solutions, on the other hand, the possibility of including a very large number of solutions, thus, giving more choice to the decision maker for biologically meaningful solutions. In this thesis, we will establish a state of the art describing the advantage of the potential application of multiobjective optimization in different bioinformatics problems. As a case study, we have chosen the most discussed problem in bioinformatics: Multiple Sequence Alignment (MSA). The goal is to specify the different issues, to propose new resolution schemes and to validate the methods developed on benchmarks. The main motivation of this research work is to contribute to the development of multiobjective optimization algorithms to make their application to bioinformatics problems as efficient as possible in terms of computational time (reasonable), the quality of solutions produced and able to deal with large-scale problems. We are interested in the development of hybrid meta-exact method. We have designed a GPLS local search method that works on the positions of gaps to improve all descendants produced by NSGA-II: the benchmark multiobjective optimization algorithm. To increase the precision of the M-NSGA-II product algorithm, we will apply the exact algorithm of Needleman and Wunsch on sub-alignments of some solutions of the Pareto front.

Résumé

L'activité de recherche de cette thèse s'inscrit dans deux thèmes principaux : l'optimisation multiobjectif et la bioinformatique. Nombreux problèmes rencontrés en bioinformatique peuvent être formulés comme problèmes d'optimisation. L'approche multiobjectif apporte de nombreux avantages aux biologistes en termes de qualité et de choix de la solution à retenir. Il s'agit d'une part, de tirer parti de l'aspect contradictoire des objectifs pour parvenir à un bon compromis (un équilibre) améliorant la qualité (biologique) des solutions et, d'autre part, la possibilité d'obtenir plusieurs solutions en une seule exécution donnant ainsi plus de choix au décideur pour des solutions biologiquement significatif. Dans cette thèse, nous allons établir un état de l'art décrivant l'avantage de l'application potentielle de l'optimisation multiobjectif dans différents problemes de bioinformatique. Comme étude de cas, nous avons choisi le problème le plus traité en bioinformatique : L'alignement multiple de séquences (MSA). Le but est de préciser les différentes problématiques, de proposer de nouveaux schémas de résolution et de valider les méthodes développées sur des benchmarks. La motivation principale de ce travail de recherche est de contribuer au développement des algorithmes d'optimisation multiobjectif de façon à rendre leur application aux problèmes bioinformatique la plus efficace possible en temps de calculs (raisonnable), en qualité de solutions produites et pouvant traiter des problèmes de grande tailles. Nous nous sommes intéressés au développement de méthode hybride méta-exacte. Nous avons conçu une méthode de recherche locale GPLS qui fonctionne sur les positions des gaps pour améliorer tous les descendants produits par NSGA-II : l'algorithme de référence d'optimisation multiobjectif. Pour augmenter la précision de l'algorithme produit M-NSGA-II, nous allons appliquer l'algorithme exact de Needleman et Wunsch sur des sous-alignements de quelques solutions du front de Pareto.

Table des matières

Chapitre VII. Optimisation multiobjectif application à l'alignement multiple de séquences

Introduction générale

Ces dernières années les avancées technologiques (le séquençage haut débit, la spectrométrie de masse, …) ont permis de produire des quantités de données biologiques phénoménales. A titre d'exemple, le séquençage du génome de la bactérie Escherichia coli produit un texte de 4,6 millions de caractères codant environ 4200 protéines (Blattner et al., 1997). Le projet américain TCGA (The Cancer Genome Atlas) qui fut lancé en 2005 pour cataloguer les variations génomiques de 10000 cancers, génère 10 téraoctets de données chaque mois (Tomczak et al. 2015). En raison de la grande quantité d'informations produites (qui ne peuvent pas être analysées manuellement), les ordinateurs sont devenus indispensables en biologie.

L'énorme quantité de données qui résultent des expérimentations biologiques nécessite le développement des bases de données spécifiques pour l'acquisition, l'organisation, le stockage et la classification de ces données. Mais il faudra aussi développer des algorithmes spécifiques et performants pour l'analyse de ces données pour prédire de nouvelles propriétés biologiques. Bref, il sera nécessaire de développer des méthodes informatiques spécifiques à la biologie (la bioinformatique), pour intégrer toutes ces informations et essayer de dériver les relations complexes qu'elles contiennent pour produire des connaissances significatives.

La bioinformatique est l'approche "in silico" de la biologie, qui consiste à effectuer des recherches et des essais au moyen des algorithmes sur des modèles informatiques qui représentent les éléments et les processus biologiques. Ainsi, cette discipline n'est pas un simple traitement automatique d'informations biologiques, mais plutôt une nouvelle manière de faire de la biologie en menant des expériences virtuelles très peu couteuses. Elle est devenue une branche à part entière de la biologie moderne, qui fait appel aux compétences de nombreuses disciplines (la biologie, l'informatique, les mathématiques, la chimie et la physique). Le terme "in silico" a été inventé par (Miramontes, 1989) par analogie aux termes "in vivo, in vitro et in situ" qui sont couramment utilisés en biologie. Il fait référence au silicium qui est l'élément principal utilisé pour la fabrication des puces d'ordinateurs. Les expériences biologiques effectuées entièrement dans un ordinateur est en train d'émerger comme un pilier du développement de la biologie, aux côtés du "in vitro" (des expériences en laboratoire), du "in vivo" (des expériences au sein de l'organisme vivant) et du "in situ" (des expériences dans le milieu naturel).

La recherche très active en bioinformatique a conduit à un développement accéléré dans de nombreux secteurs vitaux, en particulier dans les secteurs pharmaceutique, agricole, agro-alimentaire et médical. Bien qu'au départ, elle était censée concerner la recherche en biologie fondamentale. Aujourd'hui, cette discipline émergente vient de s'imposer avec de nouvelles idées et méthodes, révolutionnant ainsi la biologie moderne et tous les domaines qui l'affectent de près ou de loin.

Les essais cliniques "in silico" sur des modèles informatiques simulant l'effet d'une molécule sur une pathologie, représentent une révolution dans le domaine pharmaceutique. Alors que le développement de nouveaux médicaments qui repose sur des expériences "in vitro", puis "in vivo" est long, risqué (effets secondaire) et coûteux, avec un taux d'échec très important. Cette technique bioinformatique est une alternative pour accélérer le développement de médicaments et réduire les essais chez l'homme. Les expériences in silico ont permis de prédire la toxicité d'un médicament avant même son expérimentation in vivo et ainsi diminuer le nombre d'animaux de laboratoire utilisés. Elles permettent en outre de comprendre certains mécanismes de toxicité et d'éliminer très en amont des molécules à toxicité inadmissible (Claude et al., 2009). Le repositionnement de médicaments est une nouvelle stratégie in silico consistant à identifier de nouvelles utilisations thérapeutiques pour des médicaments existants. Cette approche a l'avantage d'être rapide puisque le médicament est déjà opérationnel et offre un niveau élevé de la sûreté car il y a déjà une richesse de

données accessibles décrivant la pharmacocinétique, toxicités, disponibilité biologique, et dosage (Li et al., 2014). L'argument le plus admis en faveur de cette méthode est la réduction du financement, ce qui permet à de plus petits groupes de recherche à participer dans l'industrie pharmaceutique. Une motivation derrière le repositionnement de médicament est le traitement des maladies rares, généralement difficiles à traiter pour des raisons financières. Pourtant quelques molécules sûres et déjà actives développées pour d'autres symptômes pourraient exister et jugés aptes pour certaines de ces maladies rares. Les technologies in silico ouvrent la voie à la médecine personnalisée (le traitement approprié pour un patient selon son génotype) en mettant à la disposition des médecins de nouveaux outils pour faire des diagnostics et de prescrire le traitement le plus approprié pour un patient selon son génotype (profil biologique) et en fonction des caractéristiques moléculaires de sa maladie. Cette nouvelle médecine en plein essor vise une objectivation accrue du recueil des données biologiques des patients et des données moléculaires des maladies.

Les retombées économiques et sociales engendrées par les applications bioinformatique sont considérables. Mais les enjeux sont également énormes, cette discipline est confrontée à des problèmes originaux qui sont dus à l'énorme richesse de données et à l'évolution des concepts à modéliser (séquences, structures, expressions, réseaux biochimiques, …). L'accroissement considérable des flux de données biologiques qui se produit a motivé le développement de bases de données spécifiques pour les conserver. La croissance en nombre de ces bases de données hétérogène et distribuées au niveau mondial pose deux questions majeures : celle de leur entretien et celle de leur intégration. Le défi est de pouvoir interpréter ces quantités massives de données. L'extraction de connaissances significative à partir de ces données implique le développement de nouvelles méthodes performantes, rapides et de qualité : fouille de données, algorithmique, modélisation, simulation, apprentissage, ***optimisation***, etc.).

Nombreux problèmes rencontrés en bioinformatique peuvent être formulés comme problèmes d'optimisation. L'optimisation multiobjectif constitue une approche essentielle pour aborder pratiquement toutes les grandes questions de la bioinformatique (Handel et al., 2007). Les principales motivations pour l'utilisation de cette approche est de capturer simultanément différents aspects de la qualité des solutions issues de l'aspect contradictoire des objectifs et la possibilité d'obtenir plusieurs solutions en une seule exécution, donnant ainsi plus de choix au décideur pour des solutions biologiquement significatif.

La résolution des problèmes d'optimisation multiobjectif comportent un très grand nombre de solutions, dont chacune représente le compromis entre les différents objectifs. Le but final est de trouver l'ensemble de solutions de compromis optimal (le front de Pareto dans l'espace objectif). Cependant, le temps de calcul pour trouver cet ensemble par des méthodes exactes augmente de façon exponentielle avec la taille du problème. Les problèmes de grande taille, qui ne peuvent pas être résolus exactement dans un délai de temps raisonnable, utilisent généralement des heuristiques ou des métaheuristiques. Elles ne garantissent pas des solutions optimales, mais essaient de trouver un ensemble de solutions aussi diversifiées et proches que possible du front Pareto-optimal. Parmi les nombreuses métaheuristiques actuellement utilisées, les algorithmes évolutionnaires multiobjectif (AEMO) sont clairement les plus populaires dans la littérature spécialisée et ont encore plusieurs opportunités de recherche à offrir aux nouveaux arrivants (Coello Coello, 2017). Dans la littérature on trouve des applications des AEMO dans pratiquement toutes les disciplines, y compris la biologie (Coello Coello & Lamont, 2004).

Dans cette thèse, nous mettons en évidence, à travers le problème le plus fondamental en bioinformatique (l'alignement multiple de séquences) l'intérêt de l'utilisation d'approche d'optimisation multiobjectif. La motivation principale de ce travail de recherche est de contribuer au développement des algorithmes d'optimisation multiobjectif de façon à rendre leur application aux

problèmes bioinformatique la plus efficace possible en temps de calculs (raisonnable), en qualité de solutions produites et pouvant traiter des problèmes de grande tailles. Malheureusement, d'après le théorème du « No Free Lunch » (Wolpert & Macready, 1997), il n'existe pas de méthode d'optimisation individuelle qui sera meilleure que toutes les autres sur tous les problèmes ou toutes les instances possibles d'un problème donné. L'hybridation et les hyper-heuristiques sont deux approches prometteuses basées sur l'idée de s'attendre à ce que plusieurs méthodes combinées de manière appropriée puissent produire de meilleurs résultats que si elles sont appliquées séparément. L'intérêt de l'approche coopérative est de permettre à différentes méthodes d'optimisation d'allier leurs avantages et de compenser leurs faiblesses, dans le but d'obtenir de bons résultats par rapport aux méthodes qui les composent. Une hyper-heuristique est une méthode de haut niveau qui manipule un ensemble de méthodes de résolution appelées heuristiques de bas niveau afin de trouver de bons compromis (entre les méthodes individuelles). Le principe général est d'essayer à un moment donné de sélectionner la bonne méthode (parmi un groupe de méthodes) durant le processus de recherche.

En général, pour résoudre un problème donné, on choisit une méthode qui paraît bien adaptée parmi celles existantes. Ensuite on essaie de l'améliorer afin d'obtenir la méthode la plus efficace possible. NSGA-II (elitist non-Sorted Genetic Algorithm) est un algorithme évolutionnaire multiobjectif basé sur l'approche Pareto qui semble être l'un des algorithmes les plus efficaces pour l'optimisation multiobjectif, en raison des caractéristiques qu'il présente, notamment la vitesse de tri et l'élitisme. Cependant, il présente certaines lacunes, telles qu'une faible précision de convergence et une distribution de front de Pareto inégale. Le concept d'algorithmes mémétiques (MA) introduit par Moscato (1989), permettant de combiner la capacité de recherche globale d'un algorithme évolutionnaire et une méthode de recherche locale (LS). Cette technique peut accélérer la convergence et obtenir un front de Pareto approximatif de haute performance (Gong et al., 2016). Un certain nombre d'articles utilisant la technique mémétique pour améliorer NSGA-II ont été publiés: mémétique NSGA-II (M-NSGA-II) (Wang et al., 2017), un algorithme évolutif mémétique qui combine NSGA-II avec une stratégie de recherche locale basée sur la stratégie d'évolution d'adaptation de la matrice de covariance (NSGA-CMA) (Zhang & Ma, 2015), un algorithme mémétique multiobjectif basé sur NSGA-II et le recuit simulé (NSGA-II-SA) (Cobos et al., 2016).

L'objectif est toujours de chercher à améliorer les résultats. Dans cette perspective, nous avons jugé nécessaire d'améliorer M-NSGA-II en augmentant sa précision de résolution. L'algorithme proposé pour améliorer M-NSGA-II est un nouveau schéma d'hybridation collaboratif, combinant M-NSGA-II comme méthode Pareto de recherche globale et l'algorithme exact Branch-and-Bound comme méthode Pareto de recherche locale (B&B-PLS). B&B-PLS qui est basé sur le critère de dominance et le concept de voisinage applique une «recherche profonde» sur certaines solutions non dominées (du front Pareto actuel) pendant le processus d'optimisation, contribue alors à améliorer la qualité des solutions. La méthode proposée nommée (Deep Memetic Non-Dominated Sorting Genetic Algorithm) (DM-NSGA-II) a été testée et validée sur des benchmarks standard du problème de sac à dos multiobjectif (Mahdi & Nini, 2021). Nous allons adapter la méthode DM-NSGA-II pour résoudre le problème MSA en utilisant l'algorithme exact de Needleman-Wunch (Needleman & Wunsch, 1970). Le détail de cette méthode sera présenté dans la partie 3 consacrée à notre contribution.

La motivation derrière le choix du problème d'alignement multiple de séquence est qu'il s'agit d'un problème central en bioinformatique, car il peut être utilisé pour résoudre de nombreux problèmes en biologie moléculaire, tels que la prédiction des fonctions et de la structure des protéines, l'analyse phylogénétique, identification des motifs et domaines conservés, classification des protéines, etc. Le problème d'alignement multiple de séquences a été considérablement traité comme un problème d'optimisation mono-objectif en utilisant la fonction objectif SP (sum of pairs) pour évaluer la qualité de l'alignement. Dans ce cas, rien ne garantit que le seul alignement optimal (le

meilleur) obtenu possède un sens biologique. Il est constaté en pratique qu'un alignement avec le meilleur score n'est bien souvent pas le meilleur d'un point de vue biologique et qu'un alignement avec un moins bon score peut être considéré par un biologiste comme étant le meilleur. Pour cela, il est préférable de lister plusieurs solutions intéressantes. L'approche multiobjectif qui a la possibilité d'obtenir plusieurs solutions en une seule exécution permet d'atteindre cet objectif. En plus, les solutions du front de Pareto obtenues sont incomparables entre eux (i.e. tous les individus sont candidats pour une solution biologiquement significative). L'aspect contradictoire des objectifs à optimiser permet de parvenir à un compromis (un équilibre) qui peut améliorer la qualité (biologique) des solutions.

Ce manuscrit se divise en trois parties : la première partie est consacrée à l'étude des méthodes d'optimisation multiobjectif. La deuxième partie est consacrée aux problèmes posés par la bioinformatique. Enfin, la dernière partie présente nos contributions qui correspondent aux applications de l'optimisation multiobjectif au problème d'alignement multiple de séquences.

I) Introduction

L'optimisation joue un rôle clé dans divers domaines de l'industrie, de l'ingénierie, de l'environnement, de l'économie, de la santé et de la bioinformatique. Historiquement, les premières méthodes d'optimisation sont issues des travaux de Fermat, Lagrange, Hamilton, Newton et Gauss. Les premières méthodes de la programmation linéaire (dédiées à la planification de programmes militaires) ont été introduites par (Kantorovich, 1960) puis améliorées par (Dantzig, 1963). Au fil du temps, une énorme croissance de la taille et de la complexité des problèmes d'optimisation s'est produite ; ce qui a motivé le développement de nouvelles méthodes performantes, rapides et de qualité (les métaheuristiques). En 1975, John Holland a développé l'algorithme génétique pour résoudre des problèmes énormes et complexes (Holland, 1975). Par conséquent, de nombreuses métaheuristiques ont été développées, la plupart été inspirées par la nature ou des processus artificiels, tels que le recuit simulé (Kirkpatrick et al., 1983), optimisation par des colonies de fourmis (Dorigo et al., 1996), optimisation par des essaims de particules (Kennedy & Eberhart, 1995), etc.

Dans ce chapitre nous allons décrire les principaux concepts en optimisation. Nous rappellerons brièvement quelques notions de la complexité d'un algorithme et les principales classes de complexité des problèmes, dont la résolution optimale est fortement liée. Nous allons définir les classes des problèmes d'optimisation pour savoir situer le problème posé, afin de choisir la méthode appropriée pour le résoudre. Les trois étapes de la résolution d'un problème d'optimisation :

- L'élaboration d'un modèle mathématique du problème : Nous présenterons deux modèles académiques (sac à dos et voyageur de commerce).
- Le développement d'un algorithme de résolution : Nous exposerons un petit état de l'art sur les algorithmes et les métaheuristiques de la littérature.
- L'évaluation de la qualité des solutions produites.

II) Rappels sur la théorie de la complexité

Pour bien traiter un problème, il est important d'étudier la complexité des algorithmes qui permettent de le résoudre, voire de découvrir des bornes qui montrent qu'il est impossible de le résoudre avec des ressources trop limitées. La théorie de la complexité est le domaine des mathématiques qui étudie formellement la quantité de ressources (temps, espace mémoire) nécessaire pour résoudre un problème donné au moyen de l'exécution d'un algorithme. Le but est de savoir si un problème est calculable efficacement par un ordinateur ou non. Dans la plupart des cas, c'est le temps de calcul excessif qui limite l'utilisation de certains algorithmes. Ce temps d'exécution dépend de la taille du problème (les données) et de l'efficacité de l'algorithme. Certains problèmes peuvent être résolus très vite, d'autres ont une complexité qui explose avec la taille des données. Les définition et notations utilisées dans cette section sont pour l'essentiel largement inspirées de l'ouvrage (Papadimitriou, 1994). Pour plus d'information le lecteur pourra se référer au livre de (Gary and Johnson, 1979).

II.1) La complexité des algorithmes

La complexité temporelle d'un algorithme est mesurée par le nombre de ces opérations élémentaires (affectations, comparaisons, opérations arithmétiques), indépendamment de l'ordinateur et du langage de programmation. Elle est exprimée en fonction de la taille de la donnée à traiter (prévoir comment ce temps de calcul augmente quand la donnée augmente). On distingue trois manières d'exprimer la complexité : dans le meilleur des cas (Ω), dans le pire des cas (O) et en moyenne (Θ). Formellement, soit $g : \mathbb{N} \rightarrow R^+$ une fonction positive.

- $O(g)$ est l'ensemble des fonctions positives $f \mid \exists c>0, \exists n_0 \in \mathbb{N}$ tels que $\forall n \geq n_0$, $f(n) \leq cg(n)$.

- $\Omega(g)$ est l'ensemble des fonctions positives f | $\exists c>0$, $\exists n_0 \in \mathbb{N}$ tels que $\forall n \geq n_0$, $f(n) \geq cg(n)$.
- $\Theta(g)$ est l'ensemble des fonctions positives f pour lesquelles $\exists c_1>0$ et $c_2>0$, $\exists n_0 \in \mathbb{N}$ tels que $\forall n \geqslant n_0$, $c_1\, g(n) \leqslant f(n) \leqslant c_2\, g(n)$.

C'est généralement la complexité au pire qui est utilisée. Dans le pire des cas, la complexité d'un algorithme sera exprimée en fonction de la taille des données n par la notation O(g(n)), on dira qu'il est O de g(n) ou encore de l'ordre de g(n). Lorsque g(n) est une constante, la complexité de l'algorithme est O(c) i.e. indépendamment de la taille des données, l'algorithme réalisera sa tâche toujours en un temps de calcul constant. La complexité d'un algorithme est d'habitude :

- polynômiale $O(n^k)$ pour un certain entier k,
- logarithmique O(log n) où la durée d'exécution croît légèrement avec n,
- exponentielles $O(2^n)$ un tel algorithme est en générale inefficace voir inutilisable dès que n dépasse des instances de taille modérée (Cormen et al. 2002).

On considère généralement qu'un algorithme est plus efficace qu'un autre si son temps d'exécution du cas le plus défavorable à un ordre de grandeur inférieur.

II.2) Les classes de complexité

La complexité des algorithmes a permis de proposer une classification des problèmes selon l'efficacité de l'algorithme de résolution à trouver une solution optimale. Certaines classes de problèmes sont faciles, ce qui signifie qu'elles peuvent être résolues dans un délai raisonnable avec une capacité mémoire acceptable. D'autres sont difficiles dans la mesure où la recherche d'une solution optimale nécessite une exploration exponentielle. Les deux principales familles de problèmes sont : la classe (P) des problèmes pour lesquels il existe une méthode de résolution rapide, et la classe des problèmes (NP) dont on ne connaît pas (jusqu'à présent) une méthode de résolution rapide.

- La classe P (problèmes déterministes Polynomiaux) représente l'ensemble des problèmes pouvant être résolus en temps polynomial sur une machine déterministe (effectuera toujours la même suite d'opérations, donc pour les mêmes données en entrée elle donnera toujours en sortie le même résultat). Les algorithmes des problèmes de P sont dits efficaces.

- La classe NP (problèmes Non-déterministes Polynomiaux) représente l'ensemble des problèmes qui peuvent être résolus en temps polynomial sur une machine non déterministe (il existe au moins une étape où elle a le choix entre plusieurs opérations à effectuer). La classe NP est équivalente à l'ensemble des problèmes pour lesquels il existe un algorithme de vérification de la solution en temps polynomial.

II.2.1) Réduction et complétude

Soient P_1 et P_2 deux problèmes. On dit qu'un algorithme est une réduction de P_1 vers P_2 s'il permet de transformer une solution de P_1 en une solution de P_2. Une des réductions de problèmes les plus utilisées est la réduction polynomiale : c'est-à-dire que le nombre d'opérations nécessaires pour réaliser la réduction peut être exprimé de façon polynomiale.

Soit x une classe de complexité. On dit qu'un problème P_1 est x-Complet s'il vérifie les deux critères :

– P_1 est dans la classe x,

– il existe un problème P_2 de x pouvant être réduit vers P_1.

II.2.2) NP-complet, NP-difficile

Un problème P_1 est NP-Complet s'il est dans la classe NP et s'il existe un problème P_2 de NP pouvant être réduit de façon polynomiale vers P_1. Le théorème de Cook (Cook, 1971) établit que le problème SAT (satisfiabilité) est NP-Complet. Soit $E = (x_1, x_2, \ldots, x_n)$ une expression logique de n variables booliennes. Le problème SAT consiste à trouver les valeurs des variables $x_{i=1..n}$ pour $E = vrai$.

Il devient alors possible par réduction polynomiale à SAT de montrer qu'un problème est NP-Complet. La classe NP-complet représente l'ensemble des problèmes NP qui concentrent en eux toute la difficulté des problèmes NP. En fait, ce sont les plus difficiles parmi NP et ils sont tous équivalents entre eux. Autrement dit, trouver un algorithme polynomial déterministe pour résoudre un quelconque des problèmes NP-complets, alors du coup tous les problèmes de la classe NP peuvent être résolus en temps polynomial (P=NP). Inversement, si on réussissait à démontrer qu'un problème NP-complet ne peut pas être résolu en temps polynomial déterministe, cela signifierait qu'aucun problème NP-complet ne peut l'être (P≠NP).

La classe NP-difficile représente l'ensemble des problèmes vers lequel on peut ramener tout problème de la classe NP par une réduction polynomiale. S'il est également dans la classe NP, on dit que c'est un problème NP-complet. Un problème NP-difficile n'est pas forcement NP. Tous les NP-complet sont NP-difficile, mais l'inverse n'est pas vrai. Les NP-difficile pourraient être encore plus difficile que les NP-complet, trouver un algorithme polynomial pour résoudre un quelconque des problèmes NP-difficile, alors du coup tous les problèmes de la classe NP-complet peuvent être résolus en temps polynomial. Le schéma ci-dessous (Figure.1) montre la relation qui existe entre les différentes classes de problèmes

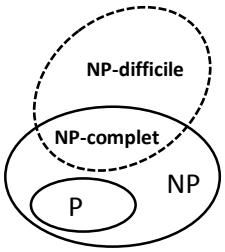


Figure.1 les différentes classes de problèmes

Trouver un algorithme polynomial pour un problème NP-complet (et prouver que P = NP) ou démontrer qu'il n'en existe pas (et prouver que P ≠ NP). C'est une question ouverte qui fait partie des problèmes non résolus en mathématiques les plus importants à ce jour. Cette conjecture émise en 1971 est sélectionnée parmi les problèmes du millénaire par l'institut Clay qui propose un million de dollars pour sa résolution.

En optimisation, on est souvent confrontés à des problèmes NP-complets. Il ne faut pas croire que dès qu'on a pu démontrer qu'un problème est NP-complet, alors, on peut cesser de s'y intéresser, car on ne connait pas une méthode qui traitera efficacement toutes les instances du problème en temps raisonnable. Cependant, savoir que le problème sur lequel on travaille est NP-complet est une indication du fait que le problème est difficile à résoudre, donc qu'il vaut mieux chercher des solutions approchées en utilisant des algorithmes d'approximation ou des heuristiques.

Les problèmes biologiques que nous allons étudier ont été démontés comme étant NP-Complet, ce qui implique qu'il est possible de calculer des solutions pour de petites instances au moyen d'algorithmes exacts, toutefois d'autres méthodes doivent être envisagées pour les instances de plus grande taille. Devant l'importance de ces problèmes, il est nécessaire de pouvoir obtenir des résultats de bonne qualité pour qu'ils puissent être utilisés par les biologistes. L'objectif est donc de contribuer à l'amélioration des méthodes de résolution approchées.

III) Principaux concepts en optimisation

On parle de problème de recherche lorsqu'il s'agit de trouver une solution quelconque parmi d'autres. Si on établit un ordre de préférence sur les solutions potentielles, exprimé sous la forme d'une fonction mathématique permettant d'attribuer un score à une solution, le problème de recherche peut être étendu au problème d'optimisation. L'optimisation est une branche des mathématiques qui

s'intéresse à la résolution analytique ou numérique des problèmes qui consiste à chercher la meilleure solution (l'optimum) parmi un espace de solution potentiel, selon un ou plusieurs objectifs à minimiser et/ou à maximiser. Tous les domaines sont concernés par l'optimisation, En effet, que l'on s'intéresse à l'optimisation d'un système de production, à la conception de systèmes, au design de réseaux de télécommunication, au traitement d'images, à la santé publique ou à la bioinformatique nous pouvons être confrontés à des problèmes d'optimisation. La plupart des problèmes d'optimisation appartiennent à la classe des problèmes NP-difficiles, classe où on ne connait pas (à ce moment) un algorithme qui fournit la solution optimale en temps polynomial en fonction de la taille du problème. Ce qui implique qu'il est possible de calculer des solutions optimales pour de petites instances au moyen d'algorithmes exacts, néanmoins des méthodes approchées doivent être envisagé pour les instances de plus grande taille.

III.1) Définition d'un problème d'optimisation

Un problème d'optimisation est défini par un espace de recherche, une ou plusieurs fonction(s) objectif(s), un ensemble de contraintes à respecter et un espace objectif.

III.1.1) espace de recherche

Appelé aussi espace de décision, il représente l'ensemble de toutes les solutions ou les configurations constituant les différentes valeurs prises par les variables du problème (variables de décision). Les variables décisionnelles peuvent être réelle, entier, booléenne, etc) et expriment des données qualitatives ou quantitatives. Le vecteur des variables de décision est noté $X(x_1, x_2,..., x_n)$ avec n le nombre de variables (la taille du problème). Les différentes valeurs possibles prises par les variables de décision x_i constituent l'ensemble des solutions potentielles définissant l'espace décisionnel de dimension n qui sera noté par δ^n.

III.1.2) fonction objectif

Les fonctions objectif $F(f_1, f_2,\ldots, f_p)$,modélisent le but à atteindre, il s'agit des critères qui doivent être optimisé (minimiser ou maximiser). Concrètement, une fonction objectif associe une valeur (un score) à une instance (une solution) d'un problème pour déterminer la meilleure solution.

III.1.3) ensemble de contraintes

L'ensemble de contrainte ς à respecter définit des conditions sur l'espace de recherche que les variables de décisions doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité ou d'égalité permettant de limiter l'espace de recherche en espace réalisable, représentant l'ensemble des valeurs des variables décisionnelles satisfaisant les contraintes.

III.1.4) espace objectif

L'espace objectif $\Re^p$ est l'image de l'espace de recherche, déterminé par toutes les valeurs possibles de la/des fonction(s) objectif ($\Re^p = F(\delta^n)$). La valeur dans l'espace objectif d'une solution est appelée coût, ou fitness.

III.2) Classification des problèmes d'optimisation

Les problèmes d'optimisation sont classés selon plusieurs caractéristiques : la nature et le domaine de définitions des variables de décision, le nombre de critères à optimiser, la taille du problème à résoudre, etc.) Il est donc important de bien identifier à quelle catégorie le problème appartient, afin de choisir la méthode appropriée pour le résoudre.

III.2.1) Problème continu ou combinatoires

Si les variables de décision sont discrètes (entiers ou binaires), le problème est dit discret ou combinatoire. Dans les problèmes d'optimisation continue, les variables sont des réelle. Les

problèmes combinatoires sont généralement plus difficiles à résoudre car ils se heurtent à l'explosion du nombre de combinaisons à explorer. En plus la nature discrète des variables forme un espace non dérivable qui rend inutiles les techniques basées sur le gradient.

III.2.2) Problème linéaire ou non-linéaire

Basée sur la nature des expressions pour la fonction objectif et les contraintes. Selon cette classification, les problèmes d'optimisation peuvent être classés comme des problèmes linéaire et non-linéaire.

III.2.3) Problème mono-objectif ou multiobjectif

Lorsqu'un seul objectif (critère) est considéré, le problème d'optimisation est mono-objectif. Dans ce cas la solution optimale est clairement définie, c'est celle qui a le coût optimal (minimal, maximal). De manière formelle, à chaque instance d'un tel problème est associé un ensemble δ des solutions potentielles respectant certaines contraintes et une fonction objectif $F : \delta \to \mathcal{R}$ qui associe à chaque solution admissible $x \in \delta$ une valeur $F(x)$. Résoudre l'instance (δ, F) du problème d'optimisation consiste à trouver la solution optimale $x^* \in \delta$ qui optimise (minimise ou maximise) la valeur de la fonction objectif F. en définissant une relation d'ordre total permettant de comparer différentes solutions. Une solution x est meilleure qu'une solution y si $F(x) < F(y)$ dans le cas de minimisation ou $F(x) > F(y)$ dans le cas de maximisation. Pour le cas de la minimisation : le but est de trouver $x^* \in \delta$ tel que $F(x^*) \leq F(x)$ pour tout élément $x \in \delta$. Un problème de maximisation peut être défini de manière similaire ($F(x^*) \geq F(x)$, $\forall x \in \delta$). Notons que la maximisation d'une fonction $F(x)$ peut facilement être transformée en un problème de minimisation : $max\ F(x) = -\ min\ (-F(x))$ est vise versa.

Un problème multiobjectif est défini par la recherche d'un compromis entre plusieurs fonctions objectifs contradictoires. Les problèmes d'optimisation multiobjectif sont plus difficiles à résoudre. Les problèmes d'optimisation multiobjectifs comportant plus de trois objectifs sont appelés problèmes d'optimisation "Many-objective". L'optimisation Many-objective apporte avec elle un certain nombre de défis qui doivent être abordés, ce qui souligne la nécessité de nouveaux algorithmes qui peuvent gérer efficacement le nombre croissant d'objectifs.

III.2.4) Problème avec contrainte ou sans contrainte

Selon la présence ou non des contraintes sur les variables de décision, on parle de problème sans contrainte ou avec contrainte. Ces contraintes peuvent être simplement des bornes et aller jusqu'à un ensemble d'équations de type égalité et de type inégalité. Les problèmes avec contraintes sont plus compliqués à résoudre.

III.2.5) Problème de petite ou de grande taille

La difficulté d'instance dépend de la complexité du problème à résoudre. Pour les problèmes NP-difficile, le nombre de solutions réalisables croît exponentiellement avec la taille de l'instance. Ce qui implique qu'il est possible de traiter efficacement des instances de petite taille par une méthode exacte. Malheureusement, les méthodes exactes par natures énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles (dès que n dépasse des instances de taille modérée). Dans ce cas, il est nécessaire de faire appel à des méthodes approchées.

III.2.6) Problème statique ou dynamique

Un problème d'optimisation dynamique est un problème dont la/les fonction(s) objectif change(nt) au cours du temps. Ce changement implique une autre problématique pour les méthodes d'optimisation (par rapport aux problèmes statiques). La robustesse de ces méthodes sera évaluée par rapport à leur capacité à détecter un changement et de fournir une réponse appropriée.

III.2.7) problème convexe ou non convexe

Selon la nature de la fonction objectif et de l'ensemble contraint, on distingue deux classes de problèmes : les problèmes convexes, et les problèmes non convexes. Un ensemble C est dit convexe si tout segment joignant deux points quelconques de C est inclus dans C (figure.2).

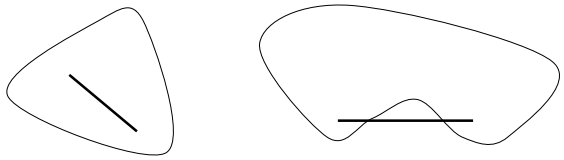


Figure.2 espace convexe (à gauche) et non convexe (à droite)

Un problème est convexe si la fonction objectif et les contraintes sont toutes convexes. La plupart des problèmes d'optimisation de la vie courante sont de nature non convexe. La convexité est le premier indicateur de la difficulté du problème. En effet, certaines méthodes sont dans l'incapacité de résoudre des problèmes non convexes de manière optimale. Pour plus de détail sur la convexité, on pourra se référer à l'ouvrage de (Hiriart-Urruty, 2013).

III.2.8) Problème multidisciplinaire

L'optimisation multidisciplinaire fait référence à la conception et l'optimisation de problèmes d'ingénierie complexes (avions, bateaux, …), nécessitant l'intervention simultanée d'au moins deux disciplines, chacune pouvant également avoir plus d'un objectif à optimiser. Par exemple, la conception d'une aile d'avion nécessite la collaboration de deux disciplines qui sont l'aérodynamique et le calcul de structure, l'aérodynamique voulant à la fois maximiser la portance et minimiser la trainée, tandis que la structure veut minimiser le poids et la déflexion (Guedas et al., 2010).

III.2.9) Problème déterministe ou stochastique

Les problèmes rencontrés seront à caractère stochastique ou déterministe. Les problèmes d'optimisation déterministe considèrent que les données sont connues parfaitement, alors que dans les problèmes d'optimisation stochastique (à ne pas confondre avec les méthodes de résolutions stochastiques) des variables aléatoires apparaissent dans la formulation du problème lui-même, qui implique des fonctions objectifs aléatoires ou des contraintes aléatoires (maxima de vraisemblance, processus de décision markovien).

Différents problèmes d'optimisation issus du monde réel, pour lesquels les données ou les fonctions objectifs sont souvent incertaines, et où il s'avère parfois nécessaire de trouver des solutions robustes à des modifications ultérieures susceptibles de survenir sur les variables de décision. C'est la prise en compte de ces incertitudes dans le processus d'optimisation qui a donné naissance à l'optimisation stochastique.

IV) Résolution d'un problème d'optimisation

La résolution optimale d'un problème est liée à sa complexité, elle consiste à trouver la solution ou un ensemble de solutions de l'espace de recherche qui satisfait au mieux la ou les fonction(s) objectif(s). La plupart des problèmes d'optimisations appartiennent à la classe des problèmes NP-difficile, classe où on ne connait pas d'algorithme qui fournit la solution optimale en temps polynomial en fonction de la taille du problème. Néanmoins, en raison de la taille des problèmes réels, la résolution optimale s'est souvent montrée impossible dans un temps raisonnable. Cette impossibilité technique impose la résolution approchée du problème, qui consiste à trouver une solution ou un ensemble de solutions de bonne qualité (la plus proche possible de l'optimum). Il est nécessaire pour déterminer si une solution est meilleure qu'une autre, que le problème introduise un critère de comparaison défini sur l'espace des objectifs.

Face à un problème d'optimisation :

- Elaborer un modèle mathématique : l'expression des objectifs à optimiser et les contraintes à respecter.
- Développer un algorithme de résolution.
- Evaluer la qualité des solutions produites.

IV.1) modèles mathématiques de problèmes d'optimisation

Un problème d'optimisation peut être représenté par le modèle mathématique suivant :

$$\begin{cases} optimiser\ f_i(x), \\ \quad x \in \delta^n \\ \text{sans/sous contraintes} \\ \quad g(x) \leq ou\ \geq 0, \\ \quad h(x) = 0 \end{cases}$$

Où les fonctions *f, g* et *h* sont linéaires et/ou non-linéaires, le problème peut être avec ou sans contraintes, si δ est l'ensemble des entiers *N* ou l'espace boolien {0,1} le problème est combinatoire, si δ est l'ensemble des valeurs réelles $\Re$ le problème est dit continu, *n* est la taille du problème. Si *f* est scalaire, le problème est mono-objectif sinon (vectorielle) il est multiobjectif. Si *x* est une variable aléatoire le problème est stochastique sinon déterministe. Dans la littérature il existe des problèmes d'optimisation académiques utilisés comme des benchmarks et des problèmes réels.

IV.2) Les méthodes d'optimisation

Lorsqu'on veut résoudre un problème d'optimisation, soit on recherche la meilleure solution possible dans tout l'espace de recherche, c'est-à-dire l'optimum global. Soit on recherche la meilleure solution possible, mais seulement pour un sous-espace restreint de l'espace de recherche, c'est-à-dire un optimum local. Les méthodes d'optimisation peuvent donc être partagées en deux catégories : les méthodes d'optimisation globale et les méthodes d'optimisation locale (à ne pas confondre aux méthodes de recherche locale). En effet, le principe des méthodes de recherche locale est de parcourir l'espace de recherche par voisinages successifs d'une ou plusieurs solutions initiales afin de les améliorer. Ainsi, une méthode de recherche locale peut être une méthode d'optimisation globale, si elle possède une technique permettant d'éviter le processus de rester bloquées dans les optimums locaux pour essayer de trouver l'optimum global de la fonction objectif. Ces méthodes locales et globales ne s'excluent pas mutuellement, comme nous le verrons plus loin, de nombreux algorithmes d'optimisation globale utilisent des méthodes de recherche locale pour améliorer leur efficacité.

Dans la littérature, il existe plusieurs manières de classifier ces méthodes d'optimisation :

- classification basée sur les origines d'inspiration de la méthode (méthodes inspirées des principes physiques, méthodes inspirées par des comportements biologiques, méthodes évolutionnaires, …)
- classification basée sur le nombre de solutions manipulées par la méthode (méthodes à solution unique versus méthodes à population de solutions).
- classification basée sur la garantie ou non de la solution optimal (méthodes exacte versus méthodes approchées)
- classification basée sur la présence ou non d'un processus stochastique dans la méthode (méthodes stochastiques versus méthodes déterministes) (Figure.3).

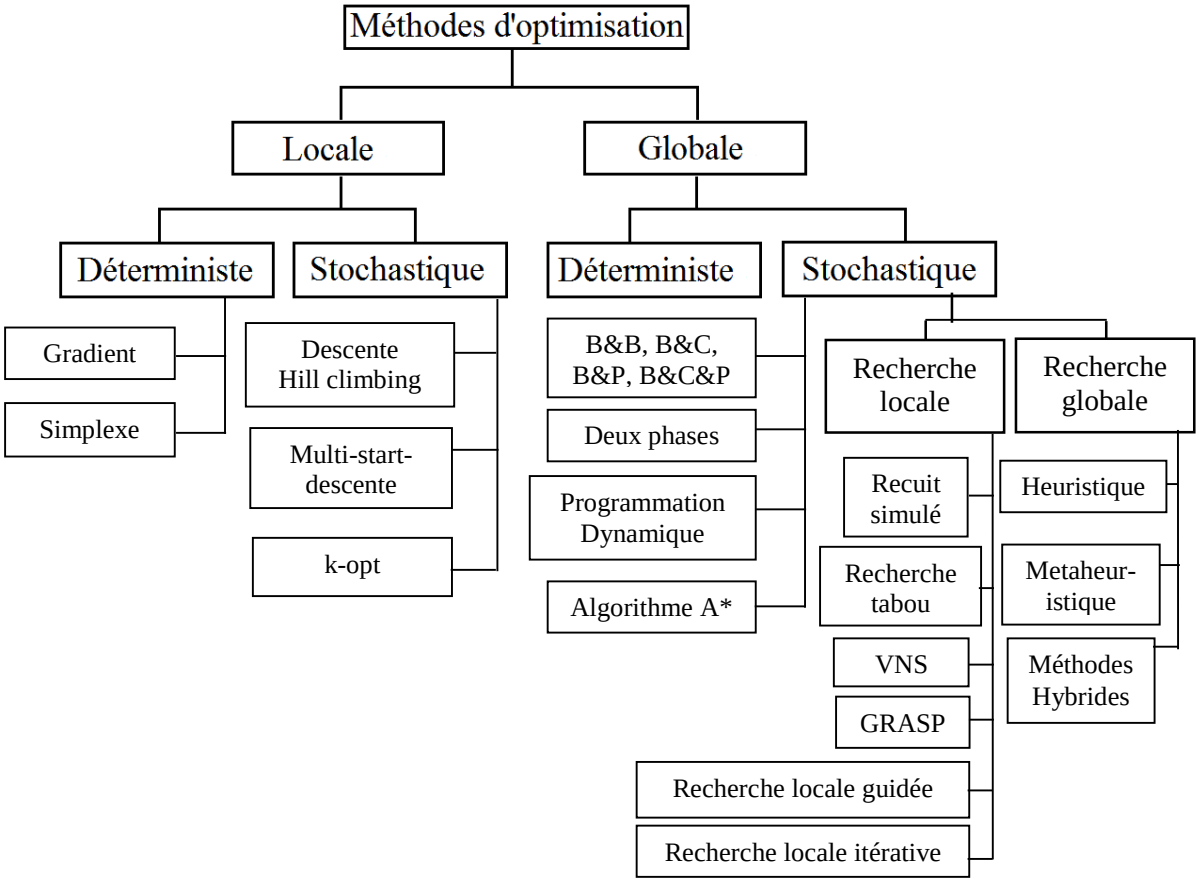


Figure.3 classification des méthodes d'optimisation

IV.2.1) les méthodes d'optimisation déterministe

Les méthodes déterministes se caractérisent par une exploration systématique (en ne laissant aucune place au hasard) de l'espace de recherche. Elles permettent de résoudre de manière exacte des problèmes de petites tailles ou des problèmes particuliers : généralement lorsque la fonction objectif est strictement convexe, continue et dérivable, comme par exemple l'algorithme du simplexe de (Dantzig, 1963) pour les problèmes continus et linéaires sous contraintes linéaires. L'aspect déterministe fait que les résultats sont les mêmes d'une exécution à l'autre. On distingue les méthodes locales et les méthodes globales.

IV.2.1.1) les méthodes déterministes en optimisation locale

Les méthodes d'optimisation locale assurent la convergence vers un optimum local en explorant le voisinage d'une solution de départ donnée (figure.4). D'une manière générale ces méthodes consistent à sélectionner une solution X réalisable, à trouver une solution Y dans le voisinage de X tel que F(Y) soit meilleure que F(X) ; alors Y devient la solution courante et cette étape est répéter jusqu'à ce qu'il n'y a plus d'amélioration de la solution courante. Parmi ces méthodes on trouve les méthodes de gradient, la méthode multistart et la méthode du simplexe.

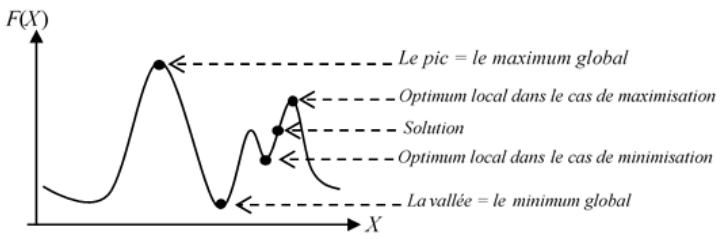


Figure.4 le plus proche optimum local de la solution courante

a) les méthodes de gradient

Cette famille de méthodes est destinée à optimiser une fonction objectif réelle continue et dérivable. Le principe consiste à choisir un point de départ x_0, on calcule le gradient $\nabla f(x_0)$ indiquant la direction de plus grande augmentation de f, on se déplace d'une quantité λ_0 dans le sens opposé au gradient et on définit le point x_1 : $x_1 = x_0 - \lambda_0 \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}$. Cette procédure est répétée pour engendrer les points x_k : $x_{k+1} = x_k - \lambda_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$

λ_k est le pas de déplacement à chaque itération ($\lambda_k > 0 \quad \forall k$ entier naturel), ainsi, pas à pas la distance entre le point d'indice k est l'optimum diminue.

b) la méthode du simplexe

C'est une technique directe et simple de recherche qui a été largement employée pour résoudre des problèmes d'optimisation dans de nombreux cas où les informations sur le gradient n'est pas toujours disponible. Cette méthode ne doit pas être confondue avec la méthode du simplexe de Dantzig pour la programmation linéaire, qui est complètement différente, car elle résout un problème linéaire sous des contraintes linéaires. La première méthode basée sur le principe de simplexe a été proposée par (Spendley et al. 1962). Des extensions de cette méthode ont été proposées par (Nelder & Mead, 1965) en incluant deux transformations supplémentaires (expansion et contraction).

IV.2.1.2) les méthodes déterministes en optimisation globale (méthodes exactes)

Les méthodes déterministes globales garantissent la convergence à la solution exacte du problème (l'optimum global de la fonction objectif). Cependant, il faut savoir que ces méthodes restent utilisables tant que la taille du problème d'optimisation considéré ne devient pas trop importante (au-delà d'un seuil modéré du nombre de variables, elles atteignent leurs limites). Ces méthodes génériques sont : Branch & Bound, Branch & Cut, Branch & Price, Branch, Cut & Price et la méthode à deux phases. D'autres méthodes sont moins générales, comme : La programmation dynamique et L'algorithme A*. D'autres méthodes sont spécifiques à un problème donné comme l'algorithme de Johnson pour l'ordonnancement (Johnson, 1977).

a) Méthodes par séparation-évaluation (Branch & Bound)

La méthode par séparation-évaluation B&B (Land & Doig, 1960) est une méthode générique de résolution exacte de problèmes d'optimisation, et plus particulièrement d'optimisation combinatoire. Elle consiste à diviser un problème en sous-problèmes indépendants (qui ne se chevauchent pas), et de résoudre chaque sous-problème, puis combiner les sous-solutions pour former une solution du problème initial. Elle réduit l'espace de recherche par l'utilisation du principe (diviser pour régner), qui divise récursivement l'ensemble faisable d'un problème en sous-ensembles disjoints, organisés en une structure arborescente, dans laquelle le nœud racine correspond à l'ensemble du problème, les autres nœud correspondent aux sous-problèmes et les arêtes sont les conditions de partition. Une borne est attribuée à chaque nœud afin d'éviter d'explorer certaines branches de l'arbre. B&B commence par le nœud racine (solution vide). La fonction évaluation du sous-problème est calculée et comparée à la meilleure solution actuelle. S'il peut être établi que le sous-problème exclut la possibilité que son sous-ensemble contienne la solution optimale, tout le sous-problème est séparé, sinon il sera partitionné. Pour chaque nœud, la borne correspond à une solution incomplète et pour les feuilles la borne est égale à la valeur de la solution correspondante.

Padberg et Rinaldi (Padberg & Rinaldi, 1991) ont amélioré l'idée du B&B basé sur la programmation linéaire en décrivant la méthode du Branch & Cut utilisant des inégalités renforçant la relaxation par programmation linéaire.

Les algorithmes de B&B résolvant des programmes linéaires en générant les variables dynamiquement quand cela est nécessaire sont appelés algorithmes de Branch & Price. Lorsque les variables et les plans de coupes sont générés dynamiquement durant l'algorithme de B&B, on appelle cette technique Branch, Cut & Price.

b) Méthode à deux phases

La méthode deux-phases a été proposée par Ulungu et Teghem pour la résolution d'un problème d'affectation bi-objectif (Ulungu & Teghem, 1995). Comme son nom l'indique, cette méthode est décomposée en deux étapes : la première consiste à trouver toutes les solutions dominées du front Pareto, puis la deuxième phase cherche de façon indépendante les solutions non dominées situées entre tous les couples de solutions dominées adjacentes. Cette méthode travaille donc essentiellement dans l'espace objectif.

c) La programmation dynamique

Le concept de la programmation dynamique a été introduit par Richard Bellman (Bellman, 1957), une solution optimale d'un problème s'obtient en combinant des solutions optimales à des sous-problèmes. Cette méthode consiste donc à résoudre un problème d'optimisation en le décomposant en sous-problèmes qui sont non indépendants (qui se chevauchent), puis à résoudre les sous-problèmes, des plus petits aux plus grands en conservant les valeurs de ces sous-problèmes dans une table de programmation dynamique, jusqu'à obtenir la solution de notre problème global.

d) L'algorithme A*

Cet algorithme a été proposé par (Hart et al., 1968). Il s'agit d'une extension de l'algorithme de Dijkstra (Dijkstra, 1959). C'est un algorithme de recherche de chemin dans un graphe. C'est l'un des plus efficaces en la matière. C'est un algorithme simple qui ne consomme que peu de mémoire.

Les méthodes déterministes globales vues ci-dessus sont très efficaces sur des problèmes de petite taille ou des problèmes particuliers. Pour les problèmes de grande taille, le décideur devra plutôt s'orienter vers des méthodes stochastiques (approchées).

IV.2.2) les méthodes stochastiques ou approchées

Les méthodes stochastiques se caractérisent par une exploration intelligente dirigée par une heuristique guidant la recherche vers des sous espaces sans parcourir l'ensemble des solutions possibles. Elles permettent de résoudre de manière approchée des problèmes d'optimisation de grande taille dans un temps raisonnable. L'aspect stochastique fait que les résultats varient d'une exécution à l'autre. Le terme heuristique vient du grec ancien « heuriskein », qui signifie trouver et qui désigne une méthode spécifique conçue pour résoudre un type de problème donné. Le terme métaheuristique (méta qui est un suffixe signifiant au-delà ou dans un niveau supérieur), a été inventé par Fred Glover (Glover, 1986) lors de la conception de la recherche tabou. Une métaheuristique désigne une méthode générale (ne demandant aucune hypothèse sur la fonction objectif), qui peut être utilisée et adaptée à plusieurs types de problèmes d'optimisation. On distingue les méthodes locales et les méthodes globales.

IV.2.2.1) les méthodes stochastiques en optimisation local

Ce sont des méthodes de recherche locale pour une optimisation locale. Le principe général de ces méthodes est le suivant : à partir d'une solution initiale x, dont on connaît la valeur de la fonction objectif $f(x)$, on cherche la meilleure solution x_0 dans le voisinage de x. Si l'on ne parvient pas à améliorer x, alors on s'arrête, sinon on remplace x par x_0, et on recommence la procédure. Parmi ces méthodes, on peut citer :

a) Algorithmes de descente (minimisation**) ou hill climbing (**maximisation**)**

Cette méthode d'optimisation locale est l'une des plus simples de la littérature (dans les problèmes de maximisation, elle est appelée hill climbing). Son principe consiste, à partir d'une solution initiale (courante), de chercher un voisin qui améliore la fonction objectif, il devient la solution courante et ce processus est itéré jusqu'à ce que le critère d'arrêt est atteint (c'est à dire quand il n'existe pas de meilleure solution dans le voisinage). Il existe plusieurs moyens de choisir ce voisin:

- first improvement : le choix du premier voisin parmi ceux qui améliorent la solution courante.
- best improvement : le choix du meilleur voisin qui améliore la solution courante.

Cette méthode converge assez rapidement vers l'optimum local le plus proche et ne peut plus en sortir (bloqué dans le premier optimum local). Une version d'optimisation globale de cet algorithme appelée algorithme de descente avec relance (random restart hill climbing) consiste à s'échapper de l'optimum local trouvé, en repartant d'une nouvelle solution générée aléatoirement. L'algorithme est toujours considéré comme une méthode de recherche locale, mais pour une optimisation globale.

b) Multi-start descent

Cette méthode utilise plusieurs fois une technique de recherche locale ou de voisinage à partir de plusieurs points de départ générés aléatoirement dans l'espace de recherche. C'est un moyen très simple de diversifier la recherche. Il est clair que si la fonction comporte beaucoup d'optima locaux, on est certain de rester bloqué sur l'un d'eux (on choisit le meilleur).

c) Algorithme k-opt

La méthode 2-opt proposé par (Croes, 1958) pour résoudre le problème du voyageur de commerce. Son principe est très simple : à chaque itération elle permute aléatoirement deux variables de la solution. Cette méthode a ensuite été généralisée en k-opt, c'est-à-dire en cherchant à permuter k variables à chaque itération. La méthode k-opt est généralement utilisée dans des problèmes discrets d'ordonnancement et de planification de trajectoire, comme les problèmes de tournées de véhicules.

IV.2.2.2) les méthodes stochastiques en optimisation global

On distingue des méthodes de recherche locale pour une optimisation globale et des méthodes de recherche globale pour une optimisation globale.

IV.2.2.2.1) Algorithmes de recherche locale pour une optimisation globale

Tout algorithme local par son mécanisme de fonctionnement et global par sa recherche d'optimum global utilisant des mécanismes pour s'échapper de ces minima locaux.

a) Recuit simulé (Simulated annealing)

Le recuit simulé a été Introduit par (Kirkpatrick et al., 1983), son principe de fonctionnement repose sur une analogie avec le processus de recuit en métallurgie qui se base sur les lois de thermodynamique énoncées par Boltzmann : Cette technique consiste à chauffer le matériau à haute température, puis d'abaisser cette température lentement. Lorsque le solide est à une forte température, chaque particule possède une très grande énergie et peut effectuer de grands déplacements aléatoires dans la matière. Au fur et à mesure que la température est abaissée, chaque particule perd de l'énergie et sa capacité de déplacement se réduit. Les différents états transitoires de refroidissement permettent d'obtenir des matériaux très homogènes et de bonne qualité. En optimisation, l'analogie permettant d'associer une solution à un état du métal, passer d'un état du métal à un autre correspond à passer d'une solution à une solution voisine. La fonction objectif est assimilée à l'énergie du système basée sur un paramètre appelé température *T*, que l'on fait décroître au fur et à mesure des itérations, pour atteindre un état de quasi-équilibre thermodynamique. Quand

l'algorithme atteint les très basses températures, les états les plus probables constituent en principe de bonnes solutions au problème d'optimisation.

b) Recherche tabou (Tabu search)

La méthode tabou a été Introduite par (Glover, 1986), le principe de base est inspiré de la mémoire adaptative pour éviter le risque de revenir à une configuration déjà visitée. Une liste taboue (qui a donné son nom à la méthode) est mise à jour permettant d'empêcher de revenir à des solutions déjà explorées. Cette méthode est la première à avoir porté le nom de métaheuristique. À chaque itération, on choisit le meilleur voisin non tabou (même si celui-ci dégrade la fonction-objectif).

c) Recherche à voisinage variable (Variable neighbourhood search VNS)

VNS (Maldenovic ,1995) applique une stratégie basée sur le changement dynamique des structures de voisinages. Le principe basé sur une idée simple : l'optimum global d'un problème peut être un optimum local sur un voisinage donné. Donc réussir à trouver un optimum local pour plusieurs voisinages augmente la probabilité de trouver l'optimum global. Ces méthodes sont basées sur l'exploration successive de plusieurs voisinages. Dès que l'optimum local d'un voisinage est obtenu et 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.

d) La méthode GRASP (Greedy Randomized Adaptative Search Procedure**)**

La procédure de recherche gloutonne aléatoire adaptative introduit par (Feo & Resende 1995) est une métaheuristique multi-départ en deux phases pour la résolution approchée de problèmes NP-difficile de l'optimisation combinatoire. Combine une heuristique gloutonne et une recherche locale aléatoire. A chaque itération, on construit une solution comme dans une heuristique gloutonne (en se servant d'une liste d'attributs comme liste de priorité). Cette solution est améliorée par l'intermédiaire d'une méthode de recherche local. En se basant sur la qualité générale de la solution ainsi obtenue, on met à jour l'ordre de la liste des attributs et le processus est itéré jusqu'à satisfaction d'un critère d'arrêt (un nombre maximum d'itérations).

e) recherche locale itérative (Iterated Local Search**)**

La recherche locale itérative est un modèle qui améliore le principe de recherche locale (multiple start) dans lequel des méthodes de descente sont lancées successivement sur des solutions initiales générées aléatoirement. Ce type de recherche locale a été la première fois utilisé dans (Martin et al., 1991). Le principe est simple, une recherche par descente est appliquée sur une solution initiale générée aléatoirement. 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.

f) recherche locale guidée (Guided local search)

La recherche locale guidée (Voudouris & Tsang, 1999) est une variante assez élaborée d'une méthode de descente classique. La méthode de base est simple, elle consiste à modifier la fonction objectif tout au long du processus en ajoutant des pénalités dans le but de s'échapper des optima locaux. La recherche locale est appliquée alors sur cette fonction modifiée. La solution trouvée (un optimum local) sert à calculer les nouvelles pénalités. Pour cela, on calcule l'utilité de chacun des attributs de la solution et on augmente les pénalités associées aux attributs de valeur maximale. Ces étapes successives sont répétées jusqu'à ce qu'un critère d'arrêt soit valide.

IV.2.2.2.2) Algorithmes de recherche globale pour une optimisation globale

Tout algorithme global par son mécanisme de fonctionnement et global par sa recherche d'optimum global.

a) Méthode de Monte-Carlo

La plus simple méthode globale est la méthode de Monte-Carlo qui a été développée par (Metropolis & Ulam, 1949). Elle consiste à faire simplement une exploration de l'espace de recherche, sans essayer d'améliorer la qualité des solutions trouvées. Elle commence par générer une solution aléatoire, puis évalue sa valeur par la fonction objectif. Ensuite, elle génère une autre solution dans tout l'espace de recherche et compare les deux résultats. Si le nouveau résultat est meilleur, elle le garde et ainsi de suite jusqu'à ce qu'un critère d'arrêt soit vérifié.

b) Les heuristiques

Une heuristique est une démarche permettant de guider le processus de recherche. Elles fournissent des schémas de résolution spécifiques permettant de les appliquer à des problèmes particuliers. Elles offrent rapidement des solutions de bonnes qualités pour des instances de problèmes de grande taille.

c) Les métaheuristiques

Le terme métaheuristique a été utilisé pour la première fois par Glover (Glover, 1986). Les métaheuristiques fournissent des schémas de résolution généraux permettant de les appliquer potentiellement à tous les problèmes. Il existe un grand nombre de métaheuristiques de recherche globale dans la littérature: ***Genetic Algorithm*** **(GA)** (Holland, 1975), ***Particale Swarm Optimization*** **(PSO)** (Kennedy & Eberhart, 1995), ***Ant Colonny Optimization*** **(ACO)** (Dorigo et al., 1996), ***Harmony Search*** **(HS)** (Geem et al. 2001), ***Bees Optimization*** **(BO)** (Nakarani & Tovey, 2004), ***Artificial Bee Colony Algorithm*** **(ABC)** (Karaboga, 2005), ***Firefly Algorithm*** **FA,** (Yang, 2008), ***Cuckoo Search*** **(CS)** (Yang & Deb, 2009), ***Gravitational Search Algorithm*** **(GSA)** (Rashedi et al. 2009), ***Galaxy based Search Algorithm*** **(GbSA)** (Shah-Hosseini, 2011), ***Teaching Learning Based Optimization*** **(TLBO)** (Rao et al. 2012), ***Swallow Swarm Optimization*** **SSO** (Neshat et al. 2012), ***Bat Algorithm*** **(BA)** (Yang, 2013), ***Water Wave Optimization*** **(WWO)** (Zheng, 2015), ***Duelist Algorithm*** **(DA)** (Biyanto et al. 2015), ***Virus Colony Search*** **(VCS)** (Li et al. 2016), ***Sperm Whale Algorithm*** **(SWA)** (Ebrahimi & Khamehchi, 2016), ***Killer Whale Algorithm*** **(KWA)** (Biyanto et al., 2017), ***Rain Water Algorithm*** **(RWA)** (Biyanto et al., 2017) et ***Monarchy Metaheuristic*** **(MM)** (Ahmia & Aider, 2019)...

d) Les méthodes hybrides

Toutes les méthodes d'optimisation n'avaient pas les mêmes propriétés et on a cherché à profiter de leurs avantages. L'intérêt de l'approche coopérative est de permettre à différentes méthodes d'optimisation d'allier leurs atouts dans le but d'améliorer les performances globales obtenues par chacune d'elles afin d'obtenir de bon résultats par rapport aux méthodes qui les composent. Actuellement les meilleurs résultats obtenus sont issus de ce type d'approche, en particulier sur les problèmes réels (Basseur, 2005).

Afin d'améliorer les performances d'une recherche globale, de nombreux auteurs proposent d'utiliser une recherche globale pour bien explorer l'espace de recherche conjointement avec une recherche locale pour mieux exploiter les zones prometteuses (Scatter Search, Algorithme mémétique, Algorithme mémétique avec gestion de la population, …). La plupart des méthodes hybrides étaient réalisées entre différentes métaheuristiques. A peu près tous les types d'approches ont été proposés pour ce type de coopération, fait que les métaheuristiques hybrides sont devenues maintenant assez classiques dans le domaine de l'optimisation (Basseur, 2005). Cependant les méthodes exactes peuvent néanmoins être utiles lorsque des sous-problèmes peuvent être extraits du problème global. Leur résolution permet en effet de contribuer à la recherche de la solution globale, soit en combinant judicieusement différents sous-problèmes, soit en hybridant résolution exacte de

sous-problèmes et résolution heuristique du problème complet (Dhaenens, 2005). Un livre consacré entièrement à l'hybridation des métaheuristiques a été publié par (Blum et al. 2008).

- ***Recherche par dispersion (Scatter Search)*** **SS** (Glover, 1977)

La recherche par dispersion est une stratégie évolutionnaire, elle comporte : une méthode de génération de diversification, une méthode d'amélioration de solutions (recherche locale), une méthode de mise à jour de l'ensemble de références, Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combiné et une méthode de combinaison de solutions. Au départ une population de solutions est générée et chaque individu est amélioré par l'application d'une recherche locale. De cette population, on extrait un ensemble de référence (Refset) de petite taille, contenant les meilleures solutions. La taille de de la population est généralement au moins 10 fois la taille de RefSet. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité. On peut citer l'exemple de (Cotta, 2006) pour l'inférence phylogénétique.

- ***Algorithme mémétique (Memetic Algorithm)*** **MA** (Moscato, 1989)

L'idée principale de cette technique est de combiner la capacité de recherche globale d'un algorithme évolutionnaire et une méthode de recherche locale appliquée à tout nouvel individu obtenu au cours de la recherche. Par exemple, dans (Martin & Otto, 1996) la méthode de descente est insérée dans un algorithme de recuit simulé. Dans (Stutzle & Hoos, 1997) une fonction de recherche locale est incorporée dans un algorithme de colonie de fourmis. Martin (Martin, 1990) a inséré la méthode de descente dans un algorithme de recuit simulé. Il s'avère que la technique mémétique est un moyen efficace pour résoudre les problèmes multiobjectifs (Fang et al., 2018).

- ***Algorithme mémétique avec gestion de la population (Memetic Algorithm with population managment)*** **MA/PM** (Sevaux, 2005)

Cet algorithme combine les avantages de la recherche par dispersion et l'algorithme mémétique. Le fonctionnement est assez simple et est basé sur un algorithme génétique. Au départ, on génère une population initiale de petite taille et on choisit un paramètre Δ fixant le niveau de dissemblance des solutions entre elles. Ensuite, on procède comme dans un algorithme génétique, on choisit deux individus que l'on croise pour obtenir deux enfants. Pour chacun on applique une recherche locale de façon à obtenir des optima locaux. S'ils ne répondent pas au critère de diversité, on applique un opérateur de mutation sur ces individus jusqu'à satisfaction de ce critère. Sinon, on les insère dans la population à la place d'un autre individu. A chaque itération le paramètre Δ gérant la diversité est mise à jour. D'un côté elle applique une recherche locale à toutes les nouvelles solutions et de l'autre, elle maintient une population de petite taille et diversifiée. L'autre avantage, c'est l'évolution du paramètre de diversité Δ (comme dans les schémas de refroidissement de température du recuit simulé) qui permet à tout moment d'augmenter ou de réduire la diversité des individus dans la population.

- ***Hybridation entre différentes métaheuristiques***

(Krueger, 1990) optimise les paramètres d'un recuit simulé à l'aide d'un algorithme génétique. Dans (Azimi, 2005) une amélioration de la solution de colonie de fourmis pour chaque cycle en appliquant le recuit simulé a été proposée. Dans (Talbi et al,. 1998) les auteurs ont introduit la recherche Tabou à la fin d'un algorithme génétique pour améliorer les solutions obtenues.

- ***Hybridation entre méthodes exactes et métaheuristiques***

Un état de l'art de ce type de coopération a été proposé par (Dumitrescu & Stützle, 2003), (Puchinger & Raidl, 2005) et (Jourdan et al. 2009). Dans (Cotta t al., 1995) l'algorithme exact B&B est incorporé dansl'AG pour servir d'opérateur de recombinaison. (Portmann et al., 1998) propose une

coopération où une heuristique calcule des solutions initiales dans le but d'offrir de bonnes bornes pour le lancement de la méthode exacte. (Augerat et al., 1998) ont utilisé un algorithme de Branch and Cut et une recherche tabou pour résoudre le problème de tournée de véhicule avec contraintes de capacité. Woodrurff (Woodrurff, 1999) décrit une stratégie de la sélection permettant de décider à chaque nœud de l'arbre B&B si une recherche taboue est appelée pour trouver une meilleure solution titulaire. Dans (French et al,. 2001) les auteurs ont présenté une hybridation entre algorithme génétique et le B&B pour résoudre le problème de satisfiabilité. Dans (Jahuira, 2002), différentes coopérations entre AGs et méthodes exactes appliquées au problème du voyageur de commerce, ont été proposées. (Cotta & Troya, 2003) ont proposé une structure pour hybrider des algorithmes évolutionnaires avec le B&B. (Bent & Hentenrych, 2004) ont utilisé le principe de la recherche sur large voisinage faisant intervenir des méthodes exacte pour l'exploration du voisinage pour trouver la (les) meilleure(s) solution(s) dans un sous espace de l'espace de recherche global. Dans (Jozefowiez, 2004) une méthode coopérative pour la résolution d'un problème de tournées couvrante bi-objectif est proposée. Pour cela un algorithme génétique propose une approximation puis, un algorithme de B&C est utilisé pour résoudre de manière optimale des sous-problèmes suivant l'un des objectifs. Dans (Klau et al., 2004) les auteurs utilisent une approche combinant un algorithme mémétique avec la programmation linéaire en nombre entier pour résoudre approximativement le problème "prize-collecting Steiner tree". Dans (Basseur, 2005) trois méthodes réalisées entre un AG Adaptatif et la méthode exacte à deux phases sont présentées. Dans (Mahdi & Nini, 2021) l'algorithme proposé DM-NSGA-II améliore l'algorithme mémétique NSGA-II par l'ajout d'une méthode exacte B&B-PLS (branch and bound Pareto local search). Dans cette thèse cette méthode sera adaptée pour résoudre le problème le plus fondamentale et le plus utilisé en bioinformatique, l'alignement multiple de séquences.

IV.3) Evaluation de performances

Deux types de mesure sont considérés pour évaluer les performances d'un algorithme d'optimisation : le temps d'exécution de l'algorithme et la qualité de la solution obtenue. Pour les méthodes exactes, seule la mesure du temps de résolution a un sens. Pour les algorithmes approchés, on trouve les deux types de mesure. La comparaison d'algorithmes pour la résolution exacte des problèmes d'optimisation est triviale. Dans le cas des méthodes approximatives, la comparaison reste triviale pour l'optimisation mono-objectif, mais pas pour l'optimisation multiobjectif. En effet l'existence d'un ensemble de solutions de compromis et l'absence d'ordre total entre les solutions rendent la mesure de qualité d'un ensemble de solutions difficile. De nombreux indicateurs de performances ont été proposés dans la littérature (voir chapitre II section IV).

V) Conclusion

Tous les domaines sont concernés par l'optimisation, qui consiste à chercher la meilleure solution d'un problème donné. La plupart des problèmes d'optimisations appartiennent à la classe des problèmes NP-difficile. Dans cette classe de problèmes, on ne connait pas une méthode exacte qui traitera efficacement toutes les instances du problème en temps raisonnable. Ce qui impose la recherche des solutions approchées en utilisant des algorithmes d'approximation (des heuristiques ou des métaheuristiques). Les métaheuristiques sont des méthodes stochastiques qui visent à résoudre un large panel de problèmes, sans pour autant que l'utilisateur ait à modifier leur structure. Ces méthodes sont inspirées d'analogies avec des domaines aussi variés que la physique, la génétique ou encore le comportement social des animaux. Les métaheuristiques ont rencontré un succès grâce à leur simplicité d'emploi mais aussi à leur forte modularité. On a vu dans ce chapitre que les métaheuristiques sont facilement adaptables et hybridables en vue d'obtenir les meilleures performances possibles.

Parmi les nombreuses métaheuristiques actuellement utilisées, les algorithmes évolutionnaires EA sont clairement les plus populaires dans la littérature et ont encore plusieurs opportunités de recherche à offrir aux nouveaux arrivants. Les métaheuristiques comportent souvent plusieurs paramètres contrôlant les différents opérateurs influençant le processus de recherche. L'efficacité d'une métaheuristique dépend du choix de ses paramètres de contrôle. Ce réglage est complexe, surtout quand le nombre de paramètres est élevé.

Toutes les méthodes d'optimisation n'avaient pas les mêmes propriétés et on a cherché à profiter de leurs avantages. L'intérêt de l'approche coopérative est de permettre à différentes méthodes d'optimisation d'allier leurs atouts dans le but d'améliorer les performances globales obtenues par chacune d'elles afin d'obtenir de bon résultats par rapport aux méthodes qui les composent. Actuellement les meilleurs résultats obtenus sont issus de ce type d'approche, en particulier sur les problèmes réels

I) Introduction

Les décideurs veulent toujours trouver les meilleures solutions pour mieux atteindre les objectifs. L'optimisation combinatoire multiobjectif fait toujours l'objet de recherches intensives et s'applique à de nombreux domaines : économie, transport, biologie, etc. Les premiers travaux menés sur les problèmes d'optimisation multiobjectif furent réalisés au 19^{éme} siècle sur des études en économie par Edgeworth et généralisés par Pareto. Les métaheuristiques sont des approches d'optimisation pouvant s'attaquer à des problèmes complexes en essayant de manière itérative d'améliorer la ou les solutions candidates. Dans ce centre, nous travaillons sur la conception et l'application d'algorithmes hybrides métaheuristiques pour la résolution des problèmes de la bioinformatique.

Un problème multiobjectif est défini par la recherche d'un compromis entre plusieurs fonctions objectifs contradictoires, en considérant tous les objectifs comme importants. Les problèmes multiobjectif ont la particularité d'être beaucoup plus difficiles à traiter par rapport à leur équivalent mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions. Une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres. Donc il n'existe généralement pas une solution unique qui procure simultanément la solution optimale pour l'ensemble des objectifs. Dans ce cas la solution optimale ou de bonne qualité est un ensemble de solutions compromis entre les différents objectifs à optimiser. Il est vital pour identifier ces meilleurs compromis de définir une relation pour comparer les solutions entre eux. La plus utilisée est la relation de dominance au sens Pareto. L'ensemble des meilleurs compromis est appelé le front Pareto, la surface de compromis, les solutions non-dominées ou les solutions efficaces. Cet ensemble de solutions constitue un équilibre, dans le sens qu'aucune amélioration ne peut être faite sur un objectif sans dégradation d'au moins un autre objectif. La solution Pareto consiste à obtenir le front de Pareto *FP* ou d'approximer la frontière de Pareto *FP**.

La majorité des problèmes d'optimisation combinatoire multiobjectif, sont des problèmes NP-difficiles, et donc on ne possède pas à ce jour un algorithme de complexité polynomiale, valable pour toutes les tailles de problèmes. L'explosion combinatoire à explorer limite l'utilisation de méthodes exactes pour la résolution à des problèmes de petites tailles. Dans le cas de problèmes de grande taille, comme cela est souvent le cas dans les applications réelles, les méthodes approchées sont devenus une alternative inévitable. Dans ce chapitre, nous présenterons l'optimisation combinatoire multiobjectif qui sera le cadre de travail de cette thèse. Nous introduirons les concepts fondamentaux et les principales approches de résolution.

Cette partie a pour but d'introduire les prérequis nécessaires à la bonne compréhension de cette thèse. Ainsi, dans un premier temps, nous présenterons le contexte de l'optimisation multiobjectif, les principales définitions, et les problématiques essentielles liées à ce domaine de recherche, surtout vis à vis des approches de résolution pour lesquelles nous allons essayer d'apporter un regard critique sur chacune d'elles. Dans la suite, nous définissons tout d'abord la notion de dominance au sens de Pareto, le front de Pareto et l'équilibre souhaité entre l'intensification et la diversification.

II) Formulation de problèmes d'optimisation multiobjectif

Un problème d'optimisation combinatoire multiobjectif peut être représenté par le programme suivant :

$$\begin{cases} Optimiser\ F(X) = \left(f_1(X), f_2(X), \dots, f_p(X)\right)\ t.q.\ X \in\ \delta^n\ \ et\ p \geq 2 \\ Sans/Sous\ les\ contraintes \quad g(X) = 0, \qquad q(X) \leq 0, \qquad h(X) \geq 0 \end{cases}$$

X, étant un vecteur de solution $(x_1,..., x_n)$ défini dans un espace δ^n de dimension *n*, représentant des instances des variables de décision x_i. δ^n représente l'ensemble des solutions réalisables respectant un ensemble de contraintes ς (d'égalité et/ou d'inégalité représenté par les fonctions *g*, *h* et *q*). $F (f_1, f_2,..., f_p)$ est le vecteur des fonctions objectifs à optimiser, et *p* représente le nombre d'objectifs. $\Re^p = F (\delta^n)$ représente les points réalisables dans l'espace objectif de dimension *p*, $Y = (y_1,..., y_p)$ avec $y_i = f_i(X)$ représente un point de l'espace objectif.

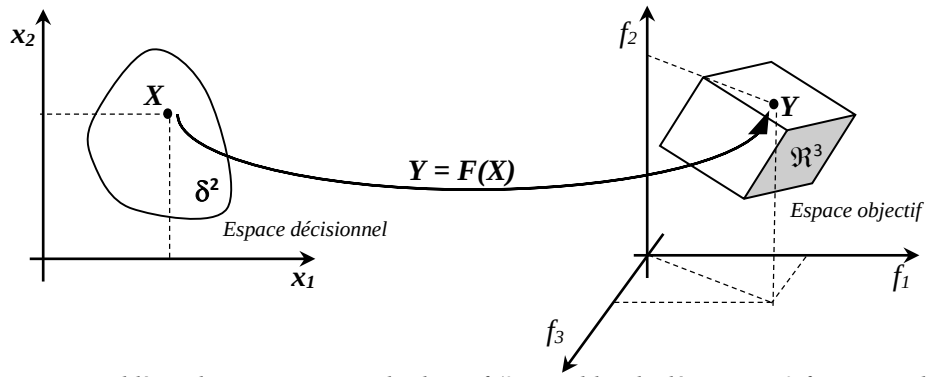


Figure.5 problème d'optimisation multiobjectif (2 variables de décision et 3 fonctions objectifs)

Contrairement à l'optimisation mono-objective, la solution d'un problème multiobjectif n'est pas unique, mais un ensemble de solutions Pareto-optimales (*PO**). La question qui se pose : laquelle de ces solutions optimales doit-on choisir ? Ainsi, dans une optimisation multiobjectif, l'effort doit être fait pour trouver l'ensemble de solutions optimales de compromis en considérant tous les objectifs comme importants. Après avoir trouvé un ensemble de ces solutions de compromis, le décideur peut alors utiliser des considérations qualitatives pour faire un choix. Les méthodes de résolution des problèmes multiobjectifs sont donc des méthodes d'aide à la décision car le choix final sera laissé au décideur. En effet, résoudre un problème multiobjectif peut être divisé en deux phases :

- L'optimisation multiobjectif : la recherche des solutions de meilleur compromis
- La prise de décision : le choix de la solution à retenir, c'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle qu'il utilisera (cela fait appel à la théorie de la décision).

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

III) Classifications des approches de résolution multiobjectif

Dans la littérature, nous rencontrons deux classifications différentes des approches de résolution de problèmes multiobjectifs. Le premier classement adopte un point de vue décideur, Le deuxième classement adopte un point de vue concepteur (figure.6).

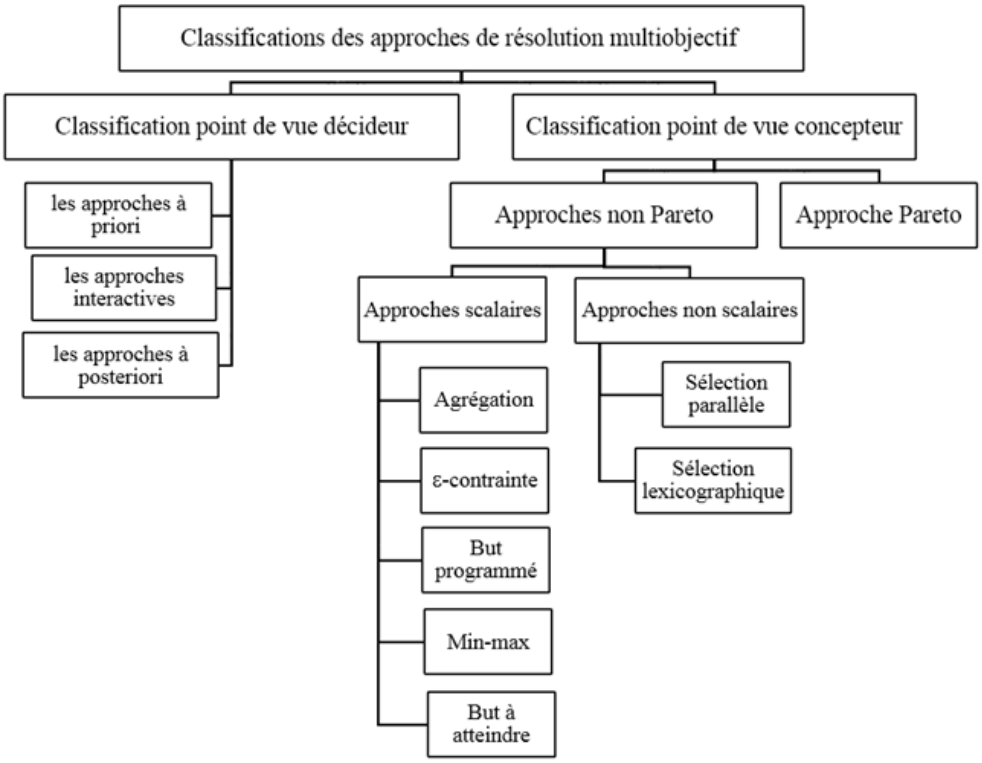


Figure.6 Classifications des approches de résolution multiobjectif

III.1) Classification point de vue décideur

Ce classement divise les approches de résolution multiobjectif en trois schémas possibles selon l'ordre d'intervention du décideur et la méthode d'optimisation :

III.1.1) les approches a priori

Le décideur intervient dès le début de la définition du problème (en amont de la méthode d'optimisation), en exprimant ses préférences, pour définir la fonction d'agrégation entre les différents objectifs. Dans ce cas le décideur est supposé connaître a priori le poids de chaque objectif afin de les mélanger dans une fonction unique et transformer ainsi le problème multiobjectif en un problème mono-objectif.

III.1.2) les approches interactives

Combinent de manière cyclique les processus de décision et d'optimisation. Le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

III.1.3) les approches a posteriori

Le décideur intervient en aval de la méthode d'optimisation, en lui fournissant un ensemble de bonnes solutions bien réparties. Il peut ensuite sélectionner celle qui lui semble la plus intéressante.

Dans le cadre de cette thèse, nous nous placerons dans le cadre de cette troisième famille de méthodes où la modélisation des préférences n'est pas requise et où le procédé d'optimisation doit être puissant afin de fournir une très bonne approximation de la frontière Pareto.

III.2) Classification point de vue concepteur

Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum Pareto. On distingue à cet égard les approches non Pareto et les approches Pareto.

III.2.1) Les approches non Pareto

Les approches non Pareto cherchent à ramener le problème initial à un ou plusieurs problèmes mono-objectif. Elles sont classées en deux catégories : les approches scalaires, qui transforment le problème multiobjectif en problème mono-objectif et les approches non scalaires, qui gardent l'approche multiobjectif, mais en traitant séparément chacun des objectifs.

III.2.1.1) Les approches scalaires (ces approches sont de type à priori)

Les approches scalaires ne traitent pas le problème comme un véritable problème multiobjectif (elles le transforment en un problème mono-objectif). Les approches les plus répandues de cette classe sont : les approches agrégées, programmation par but, et les approches ε-contraintes.

a) Approche d'agrégation

C'est l'une des premières approches utilisée pour résoudre les problèmes multiobjectifs (Ishibuchi, 1998). Elle consiste à transformer un problème multiobjectif en un problème mono-objectif, en définissant une fonction objectif unique F comme étant la somme pondérée des différentes fonctions objectifs du problème initial. En affectant à chaque objectif un coefficient de poids qui représente l'importance relative que le décideur attribue à l'objectif : $F(X) = \sum_{i=1}^{p} \lambda_i f_i(X)$

λ_i représente le poids affecté à la fonction objectif f_i, $\lambda_i \in [0..1]$ avec $\sum_{i=1}^{p} \lambda_i = 1$.

b) But programmé

Dans les approches de ce type (Charnes et al., 1955), le décideur doit fixer des buts T_i qu'il désire atteindre pour chaque fonction objectif f_i. Ces valeurs sont ensuite introduites dans la formulation du problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser les écarts entre les résultats et les buts à atteindre.

$$\min \sum_{i=1}^{p} \left| f_i(X) - T_i \right| \quad \text{Avec } X \in \delta^n$$

c) Min-max

Elle minimise le maximum de l'écart relatif entre un objectif f_i et son but associé T_i.

$$\min max_i \left(\frac{f_i(X) - T_i}{T_i} \right)$$

Avec $i = 1... p$, $X \in \delta^n$ et T_i représente le but à atteindre pour l'$i^{ème}$ objectif (Coello et al., 1995).

d) But à atteindre

Dans cette approche le décideur spécifie l'ensemble des buts T_i, qu'il souhaite atteindre et les poids associés λ_i. Cela revient à résoudre le problème suivant :

Minimiser α

Sous les contraintes

$T_i + \alpha \lambda_i \geq f_i(X)$

$\sum_{i=0}^{p} \lambda_i = 1$

Ces approches, bien que travaillant par agrégation des objectifs, permettent de générer des solutions non-dominées.

e) Approches ε-contraintes

Dans cette approche (Haimes et al., 1971), le problème consiste à optimiser une seule fonction objectif f_k sujette à des contraintes sur les autres fonctions objectifs (Convertir *p-1* des *p* objectifs du problème en contraintes). Cette méthode est basée sur la minimisation d'un objectif f_k en considérant que les autres objectifs f_j avec $j \neq k$ doivent être inférieurs à une valeur ε_j. En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité ; il peut ensuite réitérer ce processus sur un objectif différent jusqu'à ce qu'il trouve une solution satisfaisante.

Minimiser $f_k(X)$

Sous les contraintes

$$f_j(X) \leq \varepsilon_j, \ \forall j \neq k$$

f) Discussion sur les approches scalaires

Ces différentes approches transforment un problème d'optimisation multiobjectif en un ou plusieurs problèmes mono-objectif, que ce soit sous la forme d'une somme pondérée, ou sous la forme d'une distance à un but. Ces approches ont l'avantage de pouvoir utiliser facilement tous les algorithmes d'optimisation mono-objectif. Mais ces méthodes ont aussi des inconvénients, certaines ne peuvent traiter complètement des problèmes non convexes et sont donc très sensibles à la forme du front Pareto. Les autres, bien que pouvant traiter les problèmes non convexes, restent quand même sensibles à la forme du front Pareto. Un autre inconvénient important est qu'il faille relancer plusieurs fois les algorithmes de résolution avec des valeurs différentes pour certains paramètres (vecteur de poids par exemple) pour obtenir plusieurs points distincts de la surface de compromis. Ces méthodes nécessitent aussi souvent une bonne connaissance du problème a priori, notamment pour fixer les vecteurs de poids ou les points de référence. Nous présentons dans la section suivante des méthodes permettant de surmonter les difficultés présentées précédemment.

III.2.1.2) Les approches non scalaires non Pareto *(ces approches sont de type à posteriori)*

Ces approches ne transforment pas le problème multiobjectif en un problème mono-objectif, mais utilisent un processus de recherche qui traite séparément les différents objectifs (elles n'utilisent pas non plus la notion de dominance Pareto). On distingue deux approches : sélection parallèle, sélection lexicographique.

a) Sélection parallèle (méthode VEGA)

Cette approche a été la première proposant un algorithme génétique pour la résolution de problèmes d'optimisation multiobjectif (Schaffer, 1984). L'algorithme proposé, VEGA (Vector Evaluated Genetic Algorithm), sélectionne les individus selon chaque objectif de manière indépendante. L'idée est simple : Pour *k* objectifs et une population de *n* individus, une sélection de *n/k* meilleurs individus est effectuée pour chaque objectif. Ainsi *k* sous-populations vont être créées et ensuite mélangées afin d'obtenir une nouvelle population de taille *n*. le processus se termine par l'application des opérateurs génétiques (croisement et mutation). De nombreuses variations de cette technique ont été réalisées : utilisation d'un vecteur contenant les probabilités d'utiliser un certain objectif lors de la sélection (Kurwase, 1984), paramètre pour contrôler le taux de sélection (Ritzel, 1994) et utilisation de VEGA avec dominance de Pareto (Tanaki, 1995).

b) Sélection lexicographique

Cette approche, proposée par Fourman (Fourman, 1985), les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite l'optimum est obtenu en optimisant tout d'abord la fonction objectif la plus importante puis la deuxième en intégrant les valeurs obtenues comme contraintes pour la résolution sur des objectifs moins prioritaire et ainsi de suite. La solution

obtenue à l'étape *p* sera la solution du problème. Le risque essentiel de cette méthode est la grande importance attribuée aux objectifs classés en premier. La meilleure solution trouvée pour l'objectif le plus important va faire converger l'algorithme vers une zone restreinte de l'espace d'état et enfermer les points dans une niche.

c) Discussion sur Les approches non scalaires

Ces différentes approches surmontent les difficultés des approches scalaires. Elles sont souvent simples et faciles à mettre en œuvre. Une seule résolution du problème permet de trouver un ensemble de solutions Pareto optimales. L'inconvénient de ces approches est qu'elles tendent à générer des solutions qui sont largement optimisées pour certains objectifs et très peu pour les autres. Elles répartissent la population sur les extrémaux du front de Pareto, et les solutions de compromis sont négligées, et ainsi l'aspect multiobjectif du problème est contourné (Meunier, 2002).

Nous présentons dans les sections suivantes l'approche Pareto traitant les problèmes multiobjectifs sans transformation, sans favoriser un objectif par rapport à un autre et fournissant au décideur un ensemble compromis de solutions en une seule résolution du problème.

III.2.2) L'approche Pareto (cette approche est de type a posteriori)

Cette approche est la plus utilisée en optimisation multiobjectif. Elle a l'avantage de traiter les problèmes multiobjectif sans transformation, sans favoriser un objectif par rapport à un autre et peuvent générer des solutions compromis en un passage unique de l'algorithme. Elle est basée sur la notion de dominance au sens de Pareto définie comme une relation d'ordre partiel entre les solutions du problème. Au 19^{éme} Siècle, Vilfredo Pareto, un mathématicien Italien, formule le concept suivant :

Dans un problème multiobjectif, il existe un équilibre tel que l'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs. Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs contradictoires. Plusieurs techniques utilisant ce concept sont proposées dans la littérature : La méthode MOGA (Multi-Objective Genetic Algorithm) (Fonseca 1995), NPGA (Niched Pareto Genetic Algorithm) (Horn et al. 1994), SPEA (Strength Pareto Evolutionary Algorithm) (Zitzler & Thiele 1999), PAES (Pareto Archived Evolution Strategy) (Knowles et Corne 1999), PESA (Pareto-envelope based selection algorithm) (Knowles et al., 2000) et NSGA (Non dominated Sorting Genetic Algorithm) (Srinivas & Deb 1994). NSGA-II…

III.2.2.1) Notion de dominance et front de Pareto

On considère ici le cas de maximisation des objectifs. La minimisation est définie de manière analogue. Soient deux solutions X^1, $X^2 \in \delta^n$ et deux vecteurs objectifs Y^1, $Y^2 \in \mathfrak{R}^p$ | $Y^1=F(X^1)$ et $Y^2=F(X^2)$.

$X^1 = (x_1^1, x_2^1, \dots, x_n^1)$ et $X^2 = (x_1^2, x_2^2, \dots, x_n^2)$

$F = (f_1, f_2, \dots, f_p)$

$Y^1 = F(X^1) = (y_1^1, y_2^1, \dots, y_p^1)\ telque\ y_k^1 = f_k(x_1^1, x_2^1, \dots, x_n^1)$ pour $k = 1 \dots p$

$Y^2 = F(X^2) = (y_1^2, y_2^2, \dots, y_p^2)\ telque\ y_k^2 = f_k(x_1^2, x_2^2, \dots, x_n^2)$ pour $k = 1 \dots p$

Nous avons les équivalences suivantes :

$$Y^1 = Y^2 \Leftrightarrow \forall k = 1 \dots p, \quad y_k^1 = y_k^2$$

$$Y^1 \geq Y^2 \Leftrightarrow \forall k = 1 \dots p, \quad y_k^1 \geq y_k^2$$

$$Y^1 > Y^2 \Leftrightarrow \forall k = 1 \dots p, \quad y_k^1 > y_k^2$$

On dit que la solution X^1 domine X^2 (Y^1 domine Y^2), noté par $X^1 \succeq X^2$, si et seulement si:

$$\forall k \in \{1 \dots p\}, f_k(X^1) \geq f_k(X^2)\ et\ \exists k, f_k(X^1) > f_k(X^2)$$

Autrement dit $Y^1 \geq Y^2$ *et* $Y^1 \neq Y^2$ ($y_k^1 \geq y_k^2$ pour tout $k = 1..p$, et $y_k^1 > y_k^2$ pour au moins un k).
Si X^1 est meilleur que X^2 sur tous les objectifs ($y_k^1 > y_k^2$ pour tout $k = 1... p$) alors on dit que X^1 domine fortement X^2 ; On notera alors $X^1 \succ X^2$. Lorsque ni $X^1 \succeq X^2$, ni $X^2 \succeq X^1$, alors on dit qu'elles sont incomparables ou Pareto équivalentes, noté $X^1 \sim X^2$.

Notons que pour toute paire de solutions X^1 et X^2, une seule des situations suivantes peut se produire (figure.7) :

- X^1 domine X^2
- X^1 est dominée par X^2
- X^1 *et* X^2 sont équivalentes ou incomparables

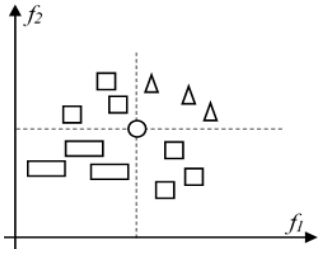


Figure.7 Relation de dominance (cas de deux objectifs à maximiser)

Le cercle est dominé par les triangles, domine les rectangles et incomparable aux carrés

La relation de dominance est la base des méthodes dites de Pareto. C'est une relation d'ordre partiel stricte transitive (si $X^1 \succeq X^2$et $X^2 \succeq X^3$alors $X^1 \succeq X^3$), non réflexive (une solution ne se domine pas elle-même) et n'est pas symétrique (on n'a jamais $X^1 \succeq X^2$et $X^2 \succeq X^1$).

Une solution $X \in \delta^n$ est dite Pareto optimale si elle n'est dominée par aucune autre solution réalisable. L'ensemble Pareto optimal $PO^* = \{X \in \delta^n \mid \nexists Y \in \delta^n$ tel que $Y \succeq X\}$. L'image de l'ensemble Pareto optimal $F(PO^*)$ dans l'espace objectif $\Re^p$ est appelée front Pareto, ou surface de compromis $PF = \{F(X) \in \Re^p \mid \forall X \in PO^*\}$.

L'ensemble Pareto optimal regroupe des solutions qui forment une frontière d'optimalité, que l'on nomme front de Pareto ou surface de compromis (figure.8). Les solutions placées sur le front de Pareto ne peuvent pas être comparées, aucune n'étant meilleure que les autres sur tous les objectifs. C'est le décideur qui aura pour rôle de choisir la solution à retenir.

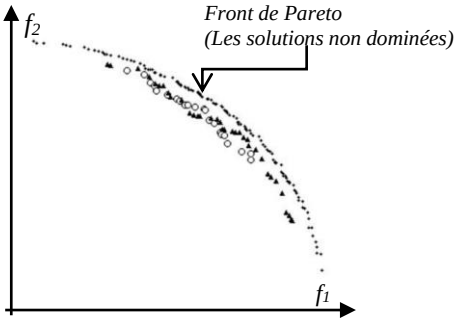


Figure.8 Front de Pareto (cas de deux objectifs à maximiser)

III.2.2.2) Point idéal et point nadir

Les coordonnées du point idéal $PI \in \Re^p$ correspondent aux meilleures valeurs de chaque objectif des points du front Pareto (figure.9). Les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément. 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 ramènerait le problème à un problème ayant une seule solution Pareto optimale.

Les coordonnées du point nadir PN$\in\Re^p$ correspondent aux mauvaises valeurs de chaque objectif (figure 9). Généralement ce vecteur n'appartient pas à l'espace objectif réalisable mais il est dans certains cas utile en tant que référence pour délimiter l'espace pour évaluer un ensemble de solution non-dominée.

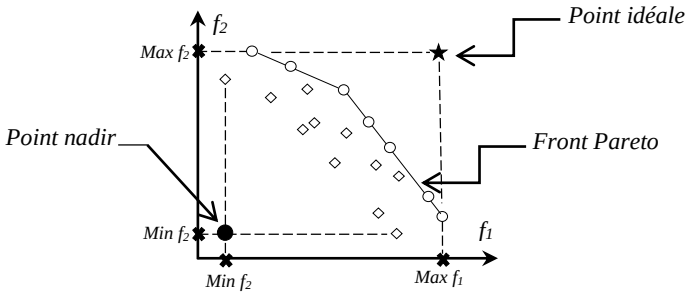


Figure.9 Point idéal et point nadir (cas de deux objectifs à maximiser)

III.2.2.3) Intensification et Diversification

Dans la résolution de problèmes multiobjectifs type Pareto, il est nécessaire que les solutions trouvées soient Pareto optimales, mais aussi qu'elles soient uniformément réparties dans le sous-espace des solutions Pareto optimales. Ces deux objectifs appelés intensification (exploitation) et diversification (exploration) doivent être pris en compte (figure.10).

- Intensification (exploitation) : la convergence vers la frontière Pareto.
- Diversification (exploration) : la répartition des solutions le long de la frontière Pareto.

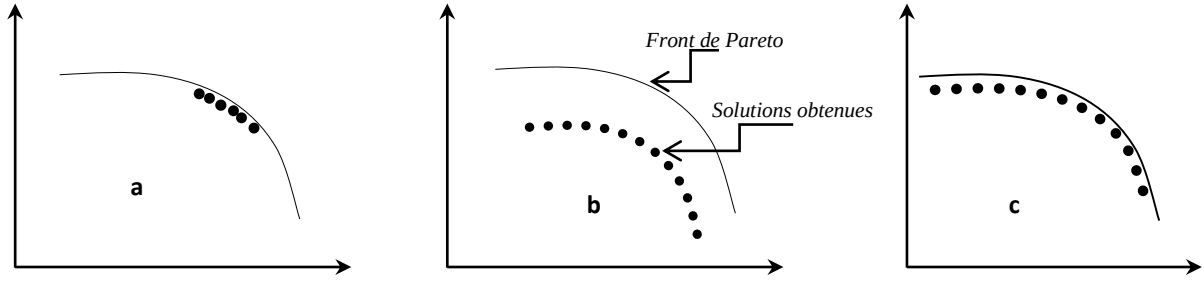


Figure.10 illustration de l'intensification et la diversification en multiobjectif

(a) présente une solution de bonne qualité en terme de convergence mais mauvaise pour la diversité. (b) une solution de bonne diversité mais de mauvaise qualité pour la convergence. (c) l'équilibre souhaité entre intensification et diversification.

Les notions d'intensification et de diversification sont nécessaires dans la résolution d'un problème d'optimisation multiobjectif. La diversification permet une exploration assez large de l'espace de recherche, par contre l'intensification permet une exploitation sur une zone précise. Il est important de bien alterner des phases d'intensification et de diversification durant la recherche, afin que l'exploration puisse rapidement identifier des régions de l'espace de recherche qui contiennent des points de bonne qualité, sans perdre trop de temps à exploiter des régions moins prometteuses. Le succès et l'efficacité d'une technique de résolution dépendent d'un équilibre délicat entre ces deux techniques de recherche.

En optimisation multiobjectif, le but est d'assurer une bonne répartition des solutions le long de la frontière Pareto et de maximiser les chances d'atteindre des zones les plus proches possible du front Pareto.

a) Mécanisme de convergence (intensification)

a.1) Ranking

Le ranking, consiste à classer les solutions en leur donnant un rang matérialisé par une valeur scalaire unique appelée fitness associée à chaque vecteur objectif. Cet ordre dépend de la notion de

dominance et donc directement de l'optimalité Pareto. La méthode de ranking permet ainsi de converger vers les solutions Pareto optimales.

- **Ranking NSGA de Goldberg**

Tous les individus non dominés de la population possèdent le rang 1. Ces individus sont ensuite enlevés de la population, et l'ensemble suivant d'individus non dominés est identifié et on leur attribue le rang 2. Ce processus est réitéré jusqu'à ce que tous les individus de la population aient un rang. Cette méthode de ranking (figure.11) a été utilisée dans les algorithmes génétiques pour la résolution de plusieurs problèmes.

$$Rang(X) < Rang(Y) \Rightarrow X \ est \ meilleur \ que \ Y$$

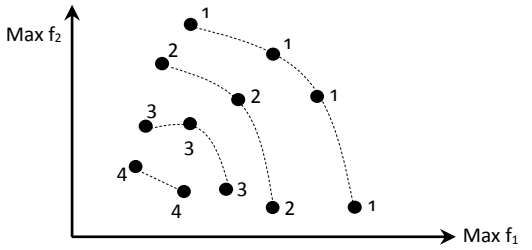


Figure. 11 Ranking NSGA (Classement des individus par fronts)

- **Ranking NDS de Fonseca et Fleming** Dans cette méthode, le rang d'un individu est le nombre de solutions dominant l'individu plus un. Considérons par exemple un individu *i* à la génération *t*, qui est dominé par p_i^t individus dans la population courante. Son rang dans la population est donné par : *rang (i, t)* = $1+p_i^t$.

Un individu non dominé de la population possède donc le rang 1. Les rangs associés à cette méthode sont toujours supérieurs à ceux de la méthode NSGA. Ce type de ranking induit donc une plus forte pression de sélection, et peut causer une convergence prématurée.

a.2) Elitisme

La notion d'élitisme fut introduite par (Zitzler et al., 2000) pour pallier au problème de la lenteur de convergence des algorithmes. Ce principe consiste à conserver les meilleures solutions non dominés trouvé durant la recherche. Cette méthode contribue à accélérer la vitesse de convergence et les performances générales des algorithmes. Actuellement, les algorithmes élitistes obtiennent de meilleurs résultats sur un grand nombre de problèmes multiobjectif.

Deux approches pour mettre en œuvre l'élitisme : la première consiste à préserver les meilleures solutions dans la prochaine génération. La second approche consiste à maintenir un ensemble externe d'individus appelés archives qui stockent les solutions non-dominées trouvées au cours du processus de recherche.

b) Méthodes de maintien de la diversité

Pour maintenir un équilibre souhaité dans le processus de recherche, les méthodes de convergence doivent être utilisées en conjonction avec les techniques de maintien de la diversité. Plusieurs approches visant à maintenir la diversité ont été proposées dans la littérature : crowding, restriction de voisinage, niches écologiques (sharing). Cependant, ces techniques ajoutent un coût calculatoire non négligeable pour l'algorithme, elles doivent donc être choisies avec soin.

- **Nichage séquentiel**

Dans le nichage séquentiel, la localisation de multiples niches se fait de manière séquentielle, à l'aide d'une exécution itérative de l'algorithme. Dans (Beasley, 1993) les auteurs ont décrit une méthode, basée sur le nichage séquentiel, pour l'optimisation de fonctions multimodales, qui évitent les inconvénients des méthodes d'exécutions itératives indépendantes. Leur stratégie est basée sur l'idée suivante : une fois qu'un optimum est trouvé, la fonction d'évaluation est modifiée dans le but

de pénaliser, dans le processus de recherche, l'optimum déjà trouvé. Les étapes principales de l'algorithme sont :

- Initialisation : affecter à la fonction coût modifiée la fonction coût originale.
- Exécuter l'algorithme en utilisant la fonction coût modifiée, et en sauvegardant la meilleure solution trouvée durant la recherche.
- Mettre à jour la fonction coût modifiée pour éviter la recherche dans les régions de la meilleure solution trouvée précédemment.
- Si toutes les solutions n'ont pas encore été trouvées, alors retour à l'étape 2.

L'inconvénient de cette approche est qu'elle modifie la structure du problème original. D'autres méthodes avancées ont donc été proposées pour favoriser la formation de niches dans les algorithmes génétiques. Ces méthodes de diversification sont basées sur le nichage parallèle, comme par exemple les fonctions de partage et le crowding (Meunier, 2002).

- **Fonction de partage (fitness sharing) et compteur de niche**

La fonction de partage a été introduite par Goldberg et Richardson (Goldberg & Richardson, 1987) et analysée en détail par Deb (1989). La principale difficulté dans un AG est que, un individu ayant une très bonne fonction d'adaptation, a tendance à se multiplier aux dépens des autres individus de la population. Le problème est que dans les fonctions multimodales, on essaie d'avoir plusieurs optimums (ou pics) et non pas un seul optima localisé. La fonction de partage a donc été introduite pour distribuer la population d'individus sur les différents pics de l'espace de recherche. Pour effectuer cette distribution, la fonction d'adaptation de chaque individu (*i*) est dégradée par un compteur de niche (m_i), calculé pour ce même individu. Le partage permet donc de modifier la valeur de la fonction d'adaptation d'un individu par rapport au nombre d'individus semblables dans la population. Cette nouvelle valeur sera utilisée comme valeur d'adaptation par l'opérateur de sélection. Pour éviter qu'un trop grand nombre d'individus ne se concentrent autour d'un même point, il faut pénaliser la valeur d'adaptation en fonction du nombre d'individus au voisinage du regroupement : plus les individus sont regroupés, plus leur valeur d'adaptation est faible, et des individus proches les uns des autres doivent partager leur valeur d'adaptation. Dans la pratique, on estime ce taux de concentration en ouvrant un domaine autour d'un individu, puis on calcule les distances entre les individus contenus dans ce domaine. Pour déterminer les bornes du domaine ouvert autour de l'individu choisi, on définit une distance maximale, appelée σ_{share}, au-delà de laquelle les individus ne seront plus considérés comme faisant partie du domaine ouvert. La distance séparant deux individus *i* et *j* est calculée grâce à la fonction $d(i,j)$.

La nouvelle fonction de partage $F(i)$ d'un individu $i \in P$ est obtenue en divisant la fonction d'adaptation de l'individu $F'(i)$ par le compteur de niche. Le compteur de niche m_i donne une estimation du nombre d'individus qui se trouvent dans le voisinage de l'individu *i*. Ce coefficient est calculé pour tous les individus *j* de la population courante P.

$$m_i = \sum_{j \in P} sh(d(i,j))$$

$$F(i) = \frac{F'(i)}{m_i}$$

Où la fonction *Sh* la plus communément utilisée, est la fonction triangulaire définie comme suit :

$$sh(d(i,j)) = \begin{cases} 1 - \dfrac{d(i,j)}{\sigma_{share}} & si\ d(i,j) < \sigma_{share} \\ 0 & sinon \end{cases}$$

Où σ_{share} est le rayon de niche, fixé dans la plupart des cas par l'utilisateur en fonction de la distance minimale de séparation voulue entre les différents pics. La fonction *d(i, j)* de calcule de distance peut être définie dans l'espace de recherche, par exemple à l'aide d'une distance de Hamming, ou dans l'espace objectif. Ce choix dépend souvent du problème, car le maintien de la diversité dans l'espace objectif, bien qu'il soit souvent plus simple à réaliser, n'assure pas forcément le maintien de la diversité dans l'espace de recherche.

- **Restriction de voisinage**

D'autres travaux proposés pour le maintien de la diversité sont basés sur la restriction de voisinage. L'idée est de permettre à deux individus de se reproduire s'ils sont similaires. Ceci induit la formation de différentes "espèces" (mating groups) dans la population. Dans (Loughlin & Ranjithan, 1997), les individus sont projetés sur les éléments d'une matrice de dimension *(p-1), p* : étant le nombre d'objectifs. Le voisinage est restreint aux individus se trouvant dans un rayon inférieur à *r*, *r* étant fixé avant l'exécution de l'algorithme.

- **Clustering**

L'objectif d'un algorithme de clustering est de partitionner un ensemble de points de telle sorte que chaque groupe contient des points très proches les uns des autres et les points d'un groupe sont très différents des points des autres groupes. Nous utilisons le clustering pour préserver la diversité dans l'archive et réduire sa taille. Ce processus est composé en trois étapes :

1. Partitionner l'archive en utilisant un algorithme de clustering.
2. Sélectionnez un individu représentatif de chaque groupe.
3. Enlever tous les autres individus dans le cluster.

- **Crowding**

Holland a été le premier à suggérer l'utilisation de l'opérateur de "crowding" dans la phase de remplacement des algorithmes génétiques (Holland, 1975), pour identifier les situations dans lesquelles de plus en plus d'individus dominent les niches écologiques. Dans la reproduction d'un nouvel individu, l'opérateur consiste à remplacer l'individu existant le plus semblable à l'individu généré, et non pas les parents comme dans les algorithmes génétiques standard.

La technique de crowding utilisée par le NSGA-II pour préserver la diversité des solutions sur le front Pareto, s'applique sur le dernier front pour compléter la taille de la population parent pour la génération suivante. Au lieu d'exclure arbitrairement certains membres du dernier front, les points qui réaliseront la diversité la plus élevée des points sélectionnés seront choisis.

IV) Mesures de performance

Dans tout problème d'optimisation après la modélisation et le développement d'un algorithme de résolution, vient la phase d'évaluation de la qualité des solutions produites. La comparaison d'algorithmes pour la résolution exacte des problèmes d'optimisation est triviale. Dans le cas des méthodes approximatives, la comparaison reste triviale pour l'optimisation mono-objectif, mais pas pour l'optimisation multiobjectif. En effet l'existence d'un ensemble de solutions de compromis et l'absence d'ordre total entre les solutions rendent la mesure de qualité d'un front difficile. Si la notion de dominance au sens de Pareto peut être utilisée pour comparer deux solutions, bien que ces deux solutions puissent être incomparables, la comparaison d'un ensemble de solutions est encore plus délicate. L'évaluation d'un algorithme en terme de qualité des solutions obtenues, nécessite soit de pouvoir évaluer un front (métriques absolues) soit de le comparer de façon quantitative avec les fronts produits par d'autres algorithmes (métriques relatives). De nombreux indicateurs de performances ont été proposés dans la littérature. Dans ce qui suit PO* représente l'ensemble des solutions potentiellement Pareto optimales trouvé par un algorithme.

IV.1) Ensemble *PO* connu

Lorsque l'ensemble *PO* est connu, les mesures calculent le plus souvent la distance entre l'approximation *PO** et *PO*.

a) Proportion d'erreur (Veldhuizen & Lamont, 2000)

Cette mesure compte le nombre de solutions *PO** qui n'appartiennent pas à *PO*, soit :

$ER = \frac{\sum_{i=1}^{|PO^*|} e_i}{|po^*|}$ où $e_i = 1$ si la $i^{ème}$ solution de $PO^* \in PO$, sinon $e_i = 0$.

Le désavantage de cette méthode est que si aucune solution de *PO** n'appartient à *PO*, elle n'apporte aucune information au sujet de la proximité relative de *PO** par rapport à *PO* puisque dans ce cas, quel que soit la distance séparant *PO** de *PO*, on a *ER = 0*.

b) Distance générationnelle (Veldhuizen & Lamont, 2000)

Cette mesure calcule la distance moyenne entre les solutions de *PO** et celles de *PO*. Elle se calcule selon la formule suivante : $GD = \frac{\left(\sum_{i=1}^{|PO^*|} d_i^{\,p}\right)^{\frac{1}{p}}}{|PO^*|}$

Pour $p = 2$, le paramètre d_i est la distance Euclidienne (dans l'espace des objectifs) entre la solution $i \in PO^*$ et le membre le plus proche de *PO*.

$d_i = \min_{k=1}^{|PO|} \sqrt{\sum_{j=1}^{p} \left(f_j^{\,i} - f_j^{\,k}\right)^2}$ où $f_j^{\,i}$ est la valeur de la $j^{ème}$ fonction objectif de *i*, et $f_j^{\,k}$ est la valeur de la $j^{ème}$ fonction objectif de la $k^{ème}$ solution de *PO*.

La difficulté avec cette méthode est que, s'il existe un ensemble *PO** pour lequel il y a une fluctuation importante dans la distance, la métrique peut ne pas retourner la véritable distance. Dans un tel cas, le calcul de l'écart type de la mesure est nécessaire.

c) Erreur maximale à la surface de compromis

Cette métrique permet de mesurer la distance entre *PO* et l'ensemble *PO**. Elle calcule en fait la plus grande distance minimale entre les solutions de *PO** et les solutions les plus proches de *PO*.

IV.2) Ensemble *PO* inconnu

Lorsque l'ensemble *PO* est inconnu, les mesures permettent le plus souvent de comparer de manière relative deux approximations. Il existe cependant des mesures qui renvoient une évaluation de la qualité d'une seule approximation. Les mesures évaluent la qualité des approximations soit par rapport à la convergence, soit par rapport à la diversification, ou par rapport aux deux buts en même temps.

IV.2.1) Mesures évaluant la convergence

a) La métrique *C*

Cette mesure, proposée par (Zitzler & Thiele, 1999), calcule la proportion de solutions d'un ensemble potentiellement Pareto optimal PO_B^* dominées par des solutions d'un ensemble PO_A^* :

$$C(A,B) = \frac{\left|\left\{\ b \in PO_B^* / \exists a \in PO_A^* : a \succ b\ \right\}\right|}{\left|PO_B^*\right|}$$

C(A,B) = 1 signifie que toutes les solutions trouvées par l'algorithme B sont dominées par celles trouvées par l'algorithme A. Tandis que *C(A,B)* = 0 indique qu'aucune solution générée par l'algorithme B n'est dominée par une solution trouvée par l'algorithme A. comme la relation de

dominance n'est symétrique, *C(B, A)* n'est pas forcément égal à 1- *C(A,B)*. Il est donc nécessaire de calculer *C(A,B)* et *C(B, A)*.

b) Mesure de contribution (Meunier, 2000)

Cette mesurer se base également sur la comparaison de deux ensembles Pareto PO_A^* et PO_B^*. Elle permet d'avoir rapidement une idée de l'apport d'un algorithme par rapport à un autre algorithme. Soit :

- $C = PO_A^* \cap PO_B^*$,
- W_A : l'ensemble des solutions de PO_A^* qui dominent des solutions dans PO_B^*,
- L_A : l'ensemble des solutions de PO_A^* dominées par des solutions dans PO_B^*,
- N_A : l'ensemble des solutions de PO_A^* non-comparables avec toutes solutions dans PO_B^*. Alors $N_A = PO_A^* \setminus (C \cup W_A \cup L_B)$.
- W_B, L_B, N_B : idem.
- *PO** l'ensemble des solutions potentiellement Pareto optimales trouvées par les deux algorithmes *A, B* : $PO^* = C \cup W_A \cup N_A \cup W_B \cup N_B$.

La contribution de l'algorithme *A* relativement à *B*, dénotée par *Contribut(A,B)*, est le ratio des solutions non dominées produites par *A* par rapport à *PO** :

$$Contribut(A,B) = \frac{\frac{|C|}{2} + |W_A| + |N_A|}{|C| + |W_A| + |N_A| + |W_B| + |N_B|}$$

Si les deux algorithmes produisent les mêmes solutions alors *Contribut(A,B)= Contribut(B,A)=1/2*
Si toutes les solutions produites par *B* sont dominées par ceux produites par *A* alors *contribut(B,A)=0*.
Dans le cas général on a *Contribut(A,B)+ Contribut(B,A)*= 1.

IV.2.2) Mesures évaluant la diversité

a) La métrique d'espacement

Cette métrique proposée par Schott (Schott, 1995), calcule la distance relative entre deux solutions consécutives de PO_A^* :

$$S = \sqrt{\frac{1}{|PO_A^*|}\sum_{i=1}^{|PO_A^*|}(d_i - \bar{d})^2} \text{ où } d_i = \min_{k \in PO_A^* \wedge k \neq i} \sum_{m=1}^{n} \left| f_m^i - f_m^k \right| \text{ et } \bar{d} = \frac{\sum_{i=1}^{|PO_A^*|} d_i}{|PO_A^*|}.$$

La distance d_i est la valeur minimale de la somme des différences absolues des valeurs des fonctions objectifs entre la $i^{ème}$ solution et toutes les autres solutions de l'ensemble généré. Il est noté que cette distance est différente de la distance Euclidienne minimale entre deux solutions. Cette métrique calcule les écarts type des différentes valeurs de d_i. Ainsi, si les solutions sont uniformément espacées, la distance correspondante sera faible. Donc, plus un algorithme trouve un ensemble de solutions pour lequel cette mesure est faible, meilleur il est.

b) Métrique maximum spread

Zitzler (Zitzler & Thiele 1999) défini une métrique mesurant la longueur de la diagonale d'une « hyper boîte » formée par les valeurs des fonctions objectifs extrêmes de l'ensemble potentiellement Pareto optimal généré : $D = \sqrt{\frac{1}{|PO_A^*|}\sum_{m=1}^{n}\left(\frac{\max_{i=1}^{|PO_A^*|} f_m^i - \min_{i=1}^{|PO_A^*|} f_m^i}{F_m^{\max} - F_m^{\min}}\right)^2}$

Où $F_m^{\max} et F_m^{\min}$ sont le maximum et le minimum pour le $m^{ème}$ objectif. Le problème de cette mesure est qu'elle ne fournit aucune information sur la distribution exacte des solutions de compromis.

c) Entropie

Cette mesure (Basseur et al., 2002) permet d'évaluer la diversité d'une approximation PO_A^* produite par un algorithme *A* par rapport à la diversité d'une approximation PO_B^* produite par un algorithme *B*. on notera PO^* l'ensemble des solutions non dominées de l'union de PO_A^* et PO_B^*.
Dans cette mesure, une niche est associée à chaque solution. Les solutions présentées dans chaque niche sont considérées comme voisines de la solution associée à la niche. L'entropie est alors donnée par la mesure : $E(A/B) = \frac{-1}{\log \gamma} \sum_{i=1}^{C} \frac{1}{N_i} \frac{n_i}{C} \log \frac{n_i}{C}$

Où N_i est le nombre de solutions de $PO_A^* \cup PO^*$ se trouvant dans la niche de la $i^{ème}$ solution de $PO_A^* \cup PO^* C$, est la cardinalité de $PO_A^* \cup PO^*$, n_i est le nombre de solutions de l'ensemble PO_A^* dans la niche de la $i^{ème}$ solution de $PO_A^* \cup PO^*$, et $\gamma = \sum_{i=1}^{C} \frac{1}{N_i}$ représente la somme des coefficients affectés à chaque solution.
Cette mesure permet donc une estimation de la diversité relative entre deux approximations. Toutefois l'interprétation des résultats n'est pas toujours évidente.

IV.2.3) Mesures évaluant la convergence et la diversité

La mesure *S*, proposée par Zitzler (Zitzler & Thiele 1999), calcule l'hypervolume de la région multidimensionnelle fermé par PO^* et un point de référence, c'est-à-dire la taille de l'espace des objectifs que PO^* domine. Un large hypervolume indique une meilleure qualité du front de Pareto. Par exemple, étant donné un point de référence *NP* (nadir point) et un ensemble de solutions non-dominées PO^*, l'image de PO^* dans l'espace objectif est un approximatif Pareto front $PF^* = \{F^j(X_i) \mid \forall X_i \in PO^*\}$. Pour chaque point $Y_i \in PF^*$, un volume $V(Y_i)$ délimité par *NP* and Y_i est construit (surface rectangulaire dans un problème biobjectif, figure 12), la mesure *S* est calculée comme l'union du volume des différents hypervolumes :

$$S(PF^*, NP) = \left\{ \bigcup_{i=1}^{i=|PF^*|} V(Y_i) \mid \forall Y_i \in PF^*, Y_i \succeq \mathrm{V}(Y_i) \succeq \mathrm{NP} \right\}.$$

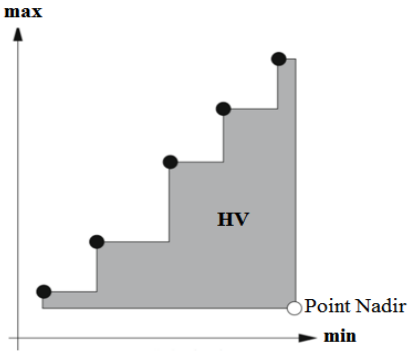


Figure. 12 l'hypervolume délimité par le point Nadir et le front de Pareto

V) Conclusion

Pendant longtemps, des approches classiques qui convertissent les problèmes multiobjectifs en un problème mono-objectif ont été utilisées. Elles comprennent les méthodes d'agrégation, la méthode de contrainte epsilon, la méthode de programmation par but, etc. Ces techniques présentent plusieurs limites, elles sont très sensibles à la forme ou à la continuité du front de Pareto et qu'ils ont tendance à générer un seul élément de l'ensemble optimal de Pareto par exécution (Coello Coello,

2017). Il devrait être préférable d'optimiser les fonctions objectifs en une seule fois. L'approche de Pareto traite les objectifs de manière équitable et peut fournir un ensemble de solutions de compromis appelé front de Pareto en une seule exécution de l'algorithme. Toutes les solutions non dominées (le front de Pareto) fournissent des informations utiles sur les relations de compromis entre des objectifs contradictoires et permettent à un décideur d'envisager plusieurs alternatives. Bien que la différence fondamentale entre l'optimisation mono-objectif et multiobjectif réside dans la cardinalité de l'ensemble de solutions. Le résultat de l'optimisation multiobjectif est un ensemble de solutions hautement performantes qui compromettent les objectifs contradictoires considérés dans l'étude. La convergence vers de bonnes approximations du front de Pareto (intensification), la préservation de la diversité des solutions (diversification) et le calcul du temps sont les trois propriétés importantes de tout optimiseur et déterminantes pour son succès.

Principalement, il y a trois aspects pour déterminer l'amélioration d'un ensemble de solutions (Okabe et al., 2003): la cardinalité (le nombre de solutions), la convergence (précision) et la distribution et l'étalement de la diversité. Un grand nombre de métriques a été proposé dans la littérature pour comparer les ensembles de solutions (fronts de Pareto). L'hypervolume (HV) (Zitzler & Thiele, 1999) est la métrique la plus utilisée dans la littérature, car HV (ou S-métrique) est la seule métrique unaire capable de mesurer les trois aspects (précision, diversité et cardinalité). L'inconvénient majeur de cette mesure est la consommation du temps.

I) Introduction

Tout comme pour l'optimisation mono-objectif, deux classes de méthodes de résolution pour traiter les problèmes multiobjectif : les méthodes exactes dédiées à résoudre de façon optimale les petites instances et les méthodes approchées : les heuristiques (spécifiques) et les métaheuristiques (génériques) permettant d'approximer les meilleures solutions sur les plus grandes instances. Enfin, le choix de la méthode de résolution à mettre en œuvre dépendra souvent de la complexité et la taille du problème. En effet, suivant sa complexité, le problème pourra ou non être résolu de façon optimale. Dans le cas de problèmes classés dans la classe P, Il suffit donc d'utiliser un algorithme polynomial. Dans le cas de problèmes NP-difficiles, deux possibilités sont offertes. Si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé (Branch & Bound, programmation dynamique...). Malheureusement, ces algorithmes souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles. Dans ce cas, il est nécessaire de faire appel à des heuristiques ou métaheuristiques permettant de trouver de bonnes solutions approchées.

Dans ce chapitre, nous nous intéressons à l'approche d'optimisation multiobjectif Pareto. Nous dressons un état de l'art non exhaustif des méthodes d'optimisation multiobjectif en accordant une importance particulière aux méthodes utilisées dans nos travaux de recherche ; notamment l'algorithme génétique, NSGA-II et les méthodes Pareto de recherche locale. D'autres méthodes bien connues seront brièvement décrites en faisant référence à la documentation pour tout besoin d'informations ou de détails supplémentaires.

II) Méthodes exactes Pareto

Dans la littérature, plusieurs méthodes exactes d'optimisation multiobjectif basées sur le B&B (Sen et al., 1988), sur l'algorithme A* (Stewart & White, 1991), la programmation dynamique (Caraway et al., 1990) et la méthode à deux phases (Ulungu & Teghem, 1995) ont été proposées pour résoudre de petits problèmes. Pour les problèmes de grandes tailles, il n'existe pas de procédures exactes efficaces, étant données les difficultés simultanées de la complexité NP-difficile, et le cadre multiobjectif des problèmes. Ainsi, on assiste ces dernières années à un accroissement de l'intérêt porté sur l'utilisation de métaheuristiques pour la résolution de Problèmes Multiobjectifs, et spécialement de problèmes réels.

a) Méthode à deux phases

Cette méthode est décomposée en deux étapes, la première consiste à trouver toutes les solutions dominées du front Pareto, puis la deuxième phase cherche de façon indépendante les solutions non dominées situées entre tous les couples de solutions dominées adjacentes. Durant la première phase de la méthode, les deux solutions extrêmes *r* et *s* (solutions optimisant chacune un des deux objectifs) sont calculées (figure 13.a). Puis, de façon récursive, la méthode recherche d'éventuelles autres solutions dominées entre *r* et *s*, à l'aide de combinaisons linéaires bien choisies des objectifs (figure.13.b et 13.c). A la fin de la première phase l'ensemble des solutions dominées est donc trouvé (figure 13.d). Cette première phase rappelle la méthode dichotomique.

La deuxième phase consiste alors en la recherche des solutions non dominées appartenant au front Pareto. Ces solutions ne peuvent être obtenues par combinaisons d'objectifs. Ulungu et Teghem proposent alors d'utiliser les solutions dominées trouvées pour réduire l'espace de recherche en argumentant que les solutions Pareto non dominées restantes sont forcément dans les triangles rectangles basés sur deux solutions dominées consécutives (figure13.e). Ainsi, une recherche de type deuxième phase est exécutée entre chaque couple de solutions dominées adjacentes (figure13.f et

13.g). La méthode de recherche au sein de ces triangles dépend du problème étudié. A la fin de la deuxième phase, toutes les solutions Pareto sont trouvées.

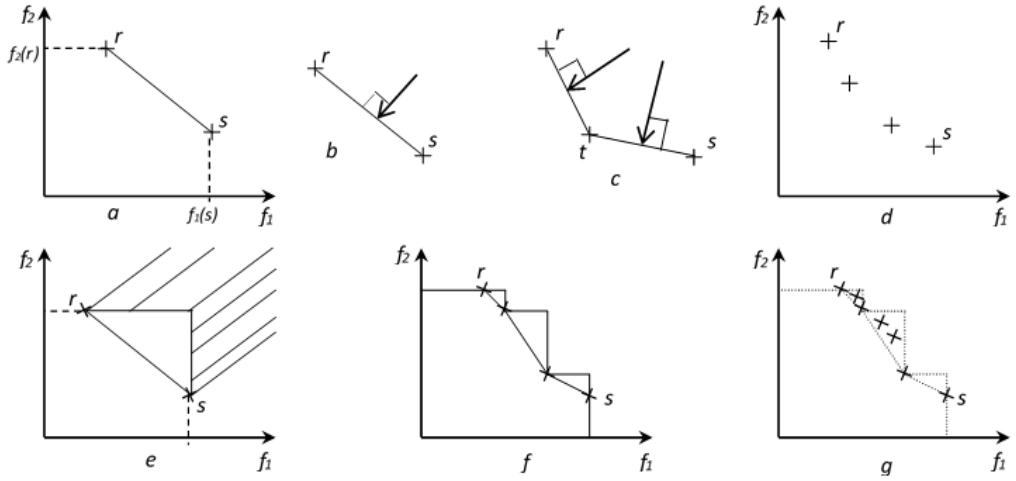


Figure.13 Illustration des différentes étapes de la méthode deux phases

b) B&B type Pareto

Le B&B adapté aux problèmes multiobjectifs utilise le concept de dominance Pareto (Mahdi & Nini, 2021). L'idée de base est de construire un arbre qui exprime implicitement toute les solutions du problème. Le problème initial est la racine (une solution vide), les feuilles correspondent à des solutions réalisables. Les autres nœuds correspondent à des solutions partielles. L'algorithme est basé sur deux concepts (la séparation et l'évaluation). La séparation consiste à diviser l'espace des solutions en sous-problèmes pour les optimiser chacun individuellement. L'évaluation consiste à calculer une borne à chaque nœud et de la comparer à une solution déjà trouvée, si la borne est dominée par la solution courante, on arrête d'explorer la branche à partir du nœud considéré. Si une solution domine la solution courante, elle la remplace et l'ensemble PO (Pareto Optimal) contenant les solutions non-dominées est mis à jour. Toutes les solutions non-comparables à la solution courante, sont stockées dans l'ensemble PO.

III) Méthodes approchées Pareto

L'approche Pareto a été introduite initialement dans les algorithmes génétiques par Goldberg (Goldberg, 1989). D'une manière générale, la plupart des approches Pareto trouvées dans la littérature sont incluses dans des algorithmes évolutionnaires, et la plupart de ces algorithmes évolutionnaires sont des algorithmes génétiques.

III.1) Recherche locale Pareto (Pareto Local Search)

Les techniques de recherche locale sont de plus en plus utilisées dans l'optimisation combinatoire multiobjectif en raison de leur capacité à améliorer les performances des métaheuristiques (Bezerra et al., 2013). En générale une recherche locale Pareto (PLS) fonctionne dans une structure de voisinage en se déplaçant d'une solution vers une autre. A partir d'une solution existante, chercher une solution qui n'est dominée par aucune solution de son voisinage ; qui devient la nouvelle solution. Et on recommence le processus jusqu'à ce qu'il n'y ait plus aucune solution améliorante.

III.2) Les principales métaheuristiques basées sur l'approche Pareto

Les algorithmes évolutionnaires sont très largement utilisés pour résoudre des problèmes multiobjectifs utilisant l'approche Pareto, on distingue deux familles d'algorithmes : les non-élitistes et les élitistes.

III.2.1) Les techniques Non-élitiste

Les approches Non-élitiste ne conservent pas les individus Pareto optimaux trouvés au cours du processus de recherche. Elles maintiennent difficilement la diversité sur la frontière Pareto et la convergence est lente (Berro, 2001).

a) Multiple Objective Genetic Algorithm (MOGA)

(Fonseca & Fleming 1993) ont proposé une méthode dans laquelle le rang d'un individu est proportionnel au nombre d'individus le dominant. Pour une solution *i*, un rang égal à $1 + n_i$ est attribué, n_i est le nombre de solutions qui dominent la solution *i*. Une fitness est ensuite attribuée à chaque solution en fonction de son rang, les individus avec les rangs les plus faibles ayant les meilleures fitness. Tous les individus non dominés sont de rang 1. La fitness de chaque individu est calculé par application d'une fonction de *scaling* sur la valeur de son rang. Cette fonction est en général linéaire.

Afin de maintenir la diversité entre les solutions non dominées, les auteurs utilisent une fonction de partage (Sharing), espérant ainsi répartir la population sur l'ensemble de la frontière de Pareto. Le sharing utilisé dans cette méthode agit sur l'espace des objectifs. Cette méthode obtient des solutions de bonne qualité et son implémentation est facile. Toutefois, les performances sont très dépendantes de la valeur du paramètre σ_{share} utilisé dans le sharing. Dans leur article les auteurs expliquent comment choisir au plus juste la valeur σ_{share}.

b) Non dominate Sorting Genetic Algorithm (NSGA)

Dans l'algorithme NSGA proposé par Srinivas et Deb (1993), le calcul de fitness s'effectue en divisant la population en plusieurs fronts en fonction du degré de dominance au sens de Pareto de chaque individu. Les individus non dominés de la population courante constituent le premier front de Pareto. On attribue alors à tous les individus de ce front la même valeur de fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité de la population, il est nécessaire d'appliquer une fonction de sharing sur cette valeur. Ensuite, ce premier groupe d'individus est temporairement supprimé de la population. On recommence cette procédure pour déterminer la seconde frontière de Pareto. La valeur factice de fitness attribuée à ce second groupe est inférieure à la plus petite fitness après application de la fonction de sharing sur le premier groupe. Ce mécanisme est répété jusqu'à ce qu'on ait traité tous les individus de la population. L'algorithme se déroule ensuite comme un algorithme génétique classique. Srinivas utilise une sélection basée sur le reste stochastique. Mais sa méthode peut être utilisée avec d'autres heuristiques de sélections (tournoi, roulette, etc). Cette méthode paraît moins efficace en temps de calcul que la méthode MOGA car le temps de calcul de la notation (trie de la population et sharing) est important. Mais cette technique semble plus appropriée à maintenir une grande diversité de la population et à répartir plus efficacement les solutions sur la frontière de Pareto.

Trois critiques ont été soulevées pour cette méthode (Berro, 2001) : Sa complexité de calcul de $O(k,N^3)$ avec k le nombre d'objectifs et N la taille de la population, essentiellement due au processus de trie de la population et d'application de l'heuristique de partage. Son approche non élitiste et la nécessité de spécifier un paramètre de sharing.

c) Niched Pareto Genetic Algorithm (NPGA)

Cette méthode proposée par Horn et al. (1994) utilise une sélection par tournoi en se basant sur la notion de dominance de Pareto. Le NPGA exécute les mêmes étapes que l'AG standard, la seule chose qui diffère étant la méthode de sélection. A chaque tournoi, elle compare deux individus pris au hasard avec une sous-population de taille t_{dom} également choisie au hasard. Les deux candidats sélectionnés sont comparés à chaque individu du sous-groupe. Si un seul de ces deux individus domine le sous-groupe, il est alors positionné dans la population suivante. Dans les autres cas une fonction de sharing est appliquée pour sélectionner l'individu. Le paramètre t_{dom} permet d'exercer une pression variable sur la population et ainsi d'augmenter ou de diminuer la convergence de l'algorithme. A travers leurs expérimentations Horn et al, (1994) établissent le constat suivant :

- Si $t_{dom} \approx 1\%$ de N, il y a trop de solutions dominées.
- Si $t_{dom} \approx 10\%$ de N, une bonne distribution des individus est obtenue.
- Si $t_{dom} >> 20\%$ de N, il y a une convergence prématurée, car la pression est trop importante lors de la sélection.

L'algorithme NPGA est considéré comme étant l'algorithme le plus rapide parmi les approches précédentes car à chaque génération la comparaison n'est appliquée que sur une portion de la population. Le principal inconvénient de cet algorithme est qu'il nécessite, en plus de spécifier le paramètre de sharing σ_{share} un autre paramètre supplémentaire qui est la taille du tournoi t_{dom}.

d) Niched Pareto Genetic Algorithm 2 (NPGA2)

(Erickson et al., 2001) ont proposé le NPGA2 qui est basé sur le degré de domination d'un individu. La sélection par tournoi est utilisée comme dans l'AG standard mais le critère de sélection de l'individu gagnant du tournoi, est basé sur le classement de Pareto (Pareto ranking). La variable qui contrôle la sélection des concurrents est la taille du tournoi. En premier, un groupe de k compétiteurs est sélectionné aléatoirement de la population. Puis, s'il existe un candidat qui a le plus petit rang (c'est-à-dire celui qui est le moins dominé), alors il sera sélectionné comme gagnant du tournoi. Si aucun des candidats n'est préférable aux autres alors la technique du partage permet d'identifier le gagnant et le candidat ayant le plus petit compteur de niche est sélectionné. Comme pour le NPGA, le compteur de niche est calculé en utilisant les individus de la population partiellement remplie, de la génération suivante, plutôt que d'utiliser la population de la génération courante. Erickson et al, (2001) ont suggéré l'ajustement de l'unité de mesure de fonctions objectives pour amener les valeurs des fonctions dans le même intervalle afin de déterminer la valeur du rayon de niche. Ils ont présenté comme exemple la relation suivante :

$$F_i' = \frac{F_i - F_{i,mim}}{F_{i,max} - F_{i,min}}$$

où : F'_i, $F_{i,min}$ $F_{i,max}$: valeurs de la fonction ajustée, le minimum et le maximum de l'objectif F_i, respectivement.

III.2.2) Les techniques élitistes

Dans le domaine de l'optimisation multiobjectif, la technique élitiste consiste, soit à préserver les meilleures solutions dans la prochaine génération, soit à maintenir une seconde population appelée archive, contenant les solutions non-dominées trouvées au cours des différentes générations. En outre des techniques de regroupements ou clustering peuvent être employés pour limiter la taille de cette archive. Cette technique est utilisée pour résoudre les difficultés des méthodes non-élitistes.

a) Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler & Thiele, 1999)

En 1999 Zitzler et Thiele proposent la méthode élitiste SPEA basée sur le concept Pareto. Pour réaliser cet élitisme, SPEA maintient une archive externe contenant le meilleur front de

compromis rencontré durant la recherche. La première étape consiste à créer une population initiale générée aléatoirement. L'archive externe, au départ initialisée à l'ensemble vide, est mise à jour régulièrement en fonction des individus non dominés de la population. À chaque itération de l'algorithme, on retrouve les étapes classiques sélection, croisement et mutation. Puis les nouveaux individus non dominés découverts viennent s'ajouter à l'archive, et les individus de l'archive dominés par le nouvel arrivant sont supprimés. Si l'archive vient à excéder une certaine taille (fixée au départ), alors une méthode de clustering est appliquée dans le but de garder les meilleurs représentants. Une nouvelle méthode de niche (n'exigeant pas de réglage de paramètre de sharing) basée sur Pareto, est utilisée afin de préserver la diversité.

b) Pareto Archived Evolution Strategy (PAES) (Knowles & Corne, 1999)

Cette méthode n'est pas basée sur une population mais, utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto optimales. Pour mesurer l'encombrement d'une zone, cette méthode utilise une technique de l'étalement (*crowding*) basée sur un découpage en hypercubes de l'espace des objectifs. Cette méthode est relativement simple à mettre en œuvre. De plus, n'étant pas basée sur un algorithme génétique, elle évite à l'utilisateur le réglage de tous les paramètres de celui-ci. Son efficacité va dépendre du choix du nouveau paramètre de discrétisation de l'espace des objectifs. La technique de crowding utilisée offre deux avantages par rapport aux méthodes de sharing classiques : le temps de calcul est moins important et le découpage étant adaptatif, cela ne nécessite pas de réglage de paramètre.

c) Pareto Envelope based Selection Algorithm (PESA)

Proposé également par Knowles et Corne (Knowles & Corne, 2000), cette méthode reprend approximativement le principe de crowding de PAES et définit un paramètre appelé (*squeeze_factor*) qui représente la mesure d'encombrement d'une zone de l'espace. Alors que PAES est basée sur une stratégie d'évolution, PESA est une méthode basée sur les algorithmes génétiques.

d) PESA II: Region-Based Selection

PESA II est une nouvelle technique de sélection basée sur l'utilisation d'hypercubes dans l'espace des objectifs (Corne et al., 2001). Au lieu d'effectuer une sélection en fonction de la fitness des individus comme dans PESA, cette méthode effectue une sélection par rapport aux hypercubes occupés par au moins un individu. Après avoir sélectionné l'hypercube, on choisit aléatoirement l'individu dans l'hypercube. Cette méthode se montre plus efficace à répartir les solutions sur la frontière Pareto. Cela est dû à sa capacité de choisir avec une plus grande probabilité que le tournoi classique, des individus situés dans des zones désertiques.

Par exemple dans la figure ci-dessous (figure.14), les 10 points sont répartis dans 6 cubes. Si l'on considère un tournoi binaire alors la probabilité de sélectionner la solution A est :

- Dans PESA : $1 - (9/10)^2 = 0{,}19$
- Dans PESA II : $1- (5/6)^2 = 0{,}31$

Si le découpage avait été de 4 cubes au lieu de 16 alors la probabilité de sélectionner la solution A serait passée de 0,31 à 0,89. Cela montre la très forte influence de la discrétisation de l'espace sur cette méthode

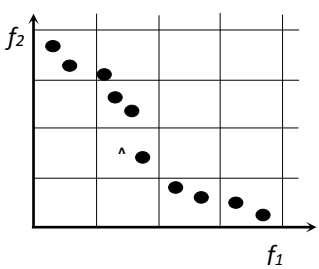


Figure.14 Exemple de sélection par tournoi hypercube

PESA II a permis de faire évoluer positivement la sélection de manière à privilégier les zones de l'espace les moins encombrées. Mais, même si cette technique de crowding basée sur un

découpage de l'espace est très supérieure en temps de calcul au sharing, elle possède ses propres difficultés.

e) Non dominate Sorting Genetic Algorithm (NSGA-II)

NSGA-II (Non-dominated Sorting Genetic Algorithm II) a été proposé par (Deb et al., 2002) et est classé comme l'un des algorithmes de référence dans le domaine de l'optimisation évolutionnaire multi-objectif. Il tient son appellation de l'algorithme NSGA qui a été proposé auparavant par (Srinvas & Deb, 1994). Dans cette deuxième version de NSGA, l'auteur tente de résoudre toutes les critiques faites sur NSGA : complexité, non élitiste et utilisation de sharing. NSGA-II intègre un opérateur de sélection, basé sur un calcul de la distance de crowding, très différent de celui de NSGA. NSGA-II est un algorithme élitiste n'utilisant pas d'archive externe pour stocker l'élite. Pour gérer l'élitisme, NSGA-II assure qu'à chaque nouvelle génération, les meilleurs individus rencontrés soient conservés. Cette nouvelle version de NSGA a permis de réduire la complexité de l'algorithme pour k objectifs et N individus à $O(kN^2)$, de créer une méthode élitiste et de supprimer les paramètres de sharing. Ce qui fait que cet algorithme est l'un des algorithmes les plus efficaces pour trouver l'ensemble optimal de Pareto avec une excellente variété des solutions.

Chaque individu i de la population a deux attributs : rang de non domination i_{rank} et distance crowding d_i. Un opérateur de comparaison défini en fonction de ces deux attributs permet de guider le processus de la sélection avec la répartition uniforme des solutions Pareto.

Soient deux individus i et j, on dit que i est meilleur que j si :

$$[(i_{rank} < j_{rank})] \quad \text{ou} \quad [(i_{rank} = j_{rank}) \text{ et } (d_i > d_j)].$$

Avec cette relation pour la comparaison de deux solutions non dominées appartenant à deux fronts Pareto, on préfère la solution appartenant au front Pareto d'ordre le plus faible. Sinon, dans le cas où les deux solutions appartenant au même front de Pareto (le dernier front pour compléter la taille de la population parent), on choisit la solution qui a la distance crowding la plus élevée. Le tri de Crowding des points de dernier front pour compléter la taille N est pris dans l'ordre décroissant de leurs valeurs de distance Crowding, et les points de la partie supérieure de la liste ordonnée sont choisis. La distance de Crowding d_i du point i est une mesure de l'espace objectif autour de i en estimant le périmètre du cuboïde ou hypercube formé en utilisant les voisins les plus proches dans l'espace objectif.

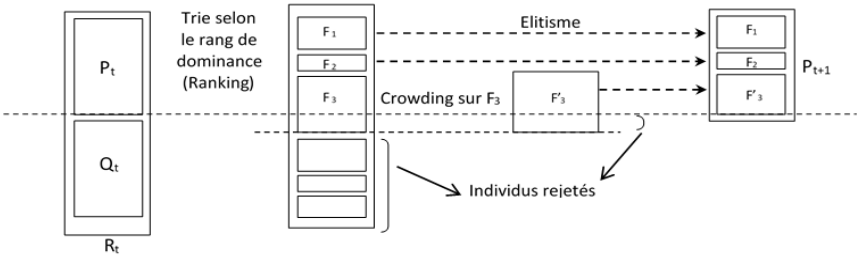


Figure.15 Schéma de fonctionnement de NSGA-II

Algorithme NSGA-II

- Initialiser les populations P_0 de taille N
- Tant que critère d'arrêt non rencontré faire
  - Création de $R_t = P_t \cup Q_t$
  - Calcul des différents fronts Fi de la population Rt par un algorithme de ranking
  - Mettre $P_{t+1} = \varnothing$ et i = 0,
  - Tant que $|P_{t+1}| + |F_i| < N$ faire
    $P_{t+1} = P_t \cup F_i$
    i = i + 1
    Fin Tant Que
  - Inclure dans P_{t+1} les $(N - |P_{t+1}|)$ individus de F_i les mieux répartis au sens de la distance de crowding
  - Sélection dans P_{t+1} et création de Q_{t+1} par application des opérateurs de croisement et mutation

Fin Tant Que

Le calcul de valeur d'adaptation pour NSGA-II ne sert pas uniquement pour la sélection des opérateurs de croisement et de mutation, mais intervient aussi dans la sélection des individus à inclure dans P_{t+1} (la population contenant les élites). C'est donc une phase importante pour laquelle les auteurs de NSGA-II ont développé une méthode particulière : la distance de crowding.

- Calcul de la distance de crowding

La distance de Crowding (surpeuplement**)** est un opérateur de sélection défini dans l'espace de recherche, utilisé pour estimer la densité au voisinage d'un individu *i*. Il calcule la distance moyenne sur chaque objectif, entre les deux points les plus proches situés de part et d'autre de la solution *i*. Cette distance notée d_i sert d'estimateur de taille du plus large hypercube incluant le point *i* sans inclure un autre point de la population et formé par les solutions du même front de Pareto les plus proches de *i* (Figure.16). La distance de crowding d_i pour un individu *i* se calcule en fonction du périmètre de l'hypercube ayant comme sommets les points les plus proches de *i* sur chaque objectif. Sur la (figure.16) est représenté l'hypercube en deux dimensions associé au point *i*. cet algorithme est de complexité *O (MN log(N))*, où *M* est le nombre d'objectifs du problème et *N* le nombre d'individus à traiter. Une fois tous les d_i calculés, il ne reste plus qu'à les trier par ordre décroissant et à sélectionner les individus possédant la plus grande valeur de crowding.

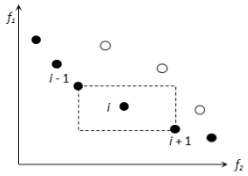


Figure.16 Calcul d'un représentant (crowding)

Algorithme Calcul de la distance de Crowding d'un individu *i* sur un front *F*

l : le nombre d'individus de front *F*.
$d_i = 0$: pour tout individu $i \in F$. (*initialisation des distances
Pour chaque objectif *m*
Trier *F* par ordre croissant selon le critère *m*
Pour *i*=2 jusqu'à *l* − 1 faire

$$\boldsymbol{d_i = d_i + \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{max} - f_m^{min}}}$$

Fin pour
Fin pour

f_m^{i+1}, f_m^{i-1}, représentent respectivement les valeurs de la $m^{\text{ième}}$ fonction objectif des solutions i+1 et i−1 alors que les paramètres f_m^{max}, f_m^{min} désignent les valeurs maximale et minimale de la $m^{\text{ième}}$ fonction objectif.

IV) Méthodes hybrides type Pareto

Les algorithmes évolutionnaires dédiés à l'optimisation multiobjectif ont bien évolué ces dernières années. L'objectif est toujours de chercher à améliorer les résultats. Dans cet objectif, une approche prometteuse concerne la coopération de différentes méthodes et en particulier la coopération entre méthodes exactes et méthodes heuristiques (Dhaenens, 2005).

- MO-GLS: Multi-Objective Genetic Local Search (Jaszkiewicz, 2001).
- MOGTS: Multi-Objective Genetic Tabu Search (Barichard & Hao, 2002).
- Ishibushi et Yoshida ont proposé pour le flowshop de faire coopérer des algorithmes évolutionnaires multiobjectifs (SPEA, NSGA II) en utilisant de la recherche locale en tant qu'opérateur de mutation.
- Les travaux de Basseur dans (Basseur, 2005) présentent trois méthodes réalisées entre un AG Adaptatif et la méthode à deux phases.

- Nicolas Jozefowiez dans (Jozefowiez, 2004) a proposé une méthode coopérative entre un algorithme génétique et un algorithme de séparation et coupes (Branch and Cut).
- Mahdi et Nini dans (Mahdi & Nini, 2021) ont développé un algorithme multiobjectif hybride DM-NSGA-II (Deep memetic NSGA-II) pour améliorer les résultats de l'algorithme mémétique NSGA-II en résolvant le problème académique du sac à dos multiobjectif. Le NSGA-II effectue une exploration globale de l'espace de recherche. L'algorithme de recherche locale HCFI-LS (Hill-Climbing First-Improvement Local Search) améliore chaque descendant produit par NSGA-II en explorant son voisinage. L'algorithme exact Branch & Bound de type Pareto (B&B-PLS) améliore les performances de l'algorithme mémétique NSGA-II, en appliquant une recherche locale approfondie sur le voisinage de certains points du front de Pareto actuel.

  La méthode B&B-PLS s'applique lorsque M-NSGA-II cesse d'améliorer les solutions, pendant un certain nombre de générations successives. Pour mesurer l'amélioration entre les solutions non dominées obtenues à la génération t avec celles obtenues à la génération t-1. Les auteurs ont utilisé La cardinalité et une métrique intuitive basée sur le point idéal (IP). Ils ne prennent pas beaucoup de temps et sont faciles à appliquer. La cardinalité est utilisée pour mesurer l'amélioration de la diversité tandis que (IP) est utilisée pour mesurer l'amélioration de la convergence.

V) Méthodes d'optimisation many-objectif

Les problèmes d'optimisation multiobjectifs comportant plus de trois objectifs sont appelés problèmes d'optimisation "many-objectif". Le principe reste le même mais, l'optimisation many-objectif apporte avec elle un certain nombre de défis qui doivent être abordés, ce qui souligne la nécessité de nouveaux algorithmes qui peuvent gérer efficacement le nombre croissant d'objectifs.

Le nombre croissant d'objectifs posent un grand défi aux algorithmes évolutionnaires multiobjectif (AEMO) classiques basés sur la dominance de Pareto, tels que NSGA-II. Ceci est principalement dû au fait que la pression de sélection basée sur la Pareto-dominance se dégrade fortement avec l'augmentation du nombre d'objectifs. Une augmentation du nombre d'objectifs entraîne qu'une grande partie de la population générée de façon aléatoire deviennent non-dominée. Avoir une population qui est en grande partie composée de solutions non-dominées ne donne pas place pour la création de nouvelles solutions à chaque génération et cela ralentit le processus de recherche.

Un algorithme basé sur des points de référence, appelée NSGA-III (Deb & Jain, 2014) est suggérée pour traiter des problèmes many-objectif, où l'opérateur de la distance crowding dans NSGA-II est Remplacé par un opérateur de clustering qui est aidé par un ensemble de points de référence bien distribués.

Les algorithmes évolutionnaires multiobjectif (AEMO) populaires basés sur la dominance de Pareto, tels que NSGA-II et SPEA2, ont r encontré de grandes difficultés dans l'optimisation à plusieurs objectifs, bien qu'ils aient montré d'excellentes performances sur des problèmes à deux ou trois objectifs.

VI) Conclusion

A travers ce chapitre, un grand nombre de méthodes de résolution de problèmes d'optimisation multiobjectif type Pareto a été présenté. Ces méthodes sont de nature exactes, métaheuristiques et méthodes hybrides. Suivant les contraintes imposées (temps de calcul, espace, . . .) et les spécificités du problème, le choix de la meilleure méthode n'est pas a priori aisé et demande souvent une bonne connaissance du problème. Les AEMO ont vite rencontré un vif succès grâce à leur simplicité d'emploi et à leur forte modularité. En plus ils sont facilement adaptables et/ou

hybridables en vue d'obtenir les meilleures performances possibles. La richesse des différentes méthodes d'optimisation nous mène à choisir le bon compromis de méthodes et algorithmes afin de résoudre un problème. Les chercheurs visent à surpasser les difficultés en proposant des techniques d'amélioration dont on cite l'hybridation des métaheuristiques.

I) Introduction

Le premier à avoir employé le mot biologie est le naturaliste Allemand (Treviranus, 1802). Dans son ouvrage il définit la biologie comme « la science qui étudie les différents phénomènes et formes de la vie, les conditions et les lois qui régissent son existence et les causes qui déterminent son activité ». La biologie s'intéresse donc à toutes les échelles d'observation du vivant, du niveau moléculaire jusqu'au niveau de la population et de l'écosystème en passant par la cellule et l'organisme. Ce spectre est immense montrant à quel point cette science est diversifiée. La biologie moléculaire est la discipline qui étudie les mécanismes de fonctionnement du vivant à l'échelle moléculaire (ADN, ARN et protéines). Il convient de noter que la grande partie de l'effort en bioinformatique est concentrée au niveau de la biologie moléculaire.

Le terme bioinformatique a été documenté pour la première fois dans une publication de (Hesper and Hogeweg, 1970), son utilisation ne s'est renforcée dans la littérature scientifique qu'au début des années 90. Cependant, les bases de ce qui deviendra par la suite la bioinformatique ont été élaborées bien avant. Durant les années 60 et 70, certains chercheurs travaillaient en biomathématiques (la modélisation formelle) lorsqu'il fallait étudier des séquences afin de rendre compte du degré de parenté de certaines molécules. Citons par exemple les deux articles fondateurs (Fitch & Margoliash, 1967) pour la reconstruction de phylogénie et (Needleman & Wunsch, 1970) pour l'alignement de séquences.

La bioinformatique est capable de prédire, par des moyens informatiques, de nouveaux concepts en biologie en combinant les données obtenues "in vivo", "in vitro" et "in silico". Ce domaine de recherche inclut le développement de méthodes pour le stockage et la récupération des données d'une part et l'analyse biologiques de ces données d'autre part. Ceci a mené à la fondation de deux groupes de chercheur spécialisés en bioinformatique, à savoir l'organisation des données dans des bases de données et l'analyse de ces données par des moyens informatiques. Le premier groupe a pour mission d'organiser l'énorme masse d'information biologique et de la rendre disponible à l'ensemble de la communauté des chercheurs. Cela a été rendu possible grâce à différentes bases de données, accessibles en lignes. Trois grandes institutions sont chargées de l'archivage de ces données : le NCBI aux États-Unis, l'EBI en Europe et le DDBJ au Japon. Ces institutions se coordonnent pour gérer les grandes bases de données de séquences nucléotidiques comme GenBank et l'EMBL, ainsi que les bases de données de séquences protéiques comme UniProt et TrEMBL. Le deuxième groupe a pour mission le développement des outils d'analyse afin de produire de nouvelles connaissances et de prédire de nouvelles propriétés biologiques.

Le développement de méthodes algorithmiques efficaces et puissantes pour l'analyse et l'interprétation des données biologiques constitue donc une partie importante en bioinformatique. Dans cette thèse, nous nous sommes intéressés à la conception et l'implémentation des algorithmes pour l'exploitation et l'analyse de ces données.

Ce chapitre va principalement rappeler les notions de base de la biologie qui seront utiles à la bonne compréhension des techniques dont le développement fait l'objet de cette thèse. Nous abordons aussi le point de vue informatique pour la représentation des données biologiques. Nous allons faire un tour rapide des principales bases de données et outils bioinformatique.

II) Quelques notions de biologie moléculaire

II.1) Cellule

La cellule constitue l'unité de base de tout être vivant. Deux organismes sont distingués selon le nombre de cellules qu'ils contiennent : les multicellulaires (eucaryotes) et les unicellulaires (procaryotes).

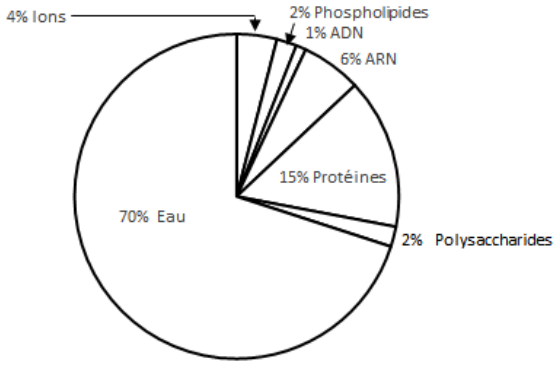


Figure.17 Les composants d'une cellule

II.2) ADN (Acide DésoxyriboNucléique)

L'ADN d'un organisme est une macromolécule biologique compactée dans les chromosomes, que les cellules abritent dans leur noyau (pour les eucaryotes). L'ADN est formé par deux chaînes (2 brins d'acides nucléiques) complémentaires qui s'emboîtent tout en s'enroulant l'une autour de l'autre pour former une double hélice. Chaque chaîne est composée d'une succession de quatre briques élémentaires (les nucléotides ou les bases azotées) symbolisées par les lettres {A, C, G, T} pour {Adénine, Cytosine, Guanine, Thymine}. A chaque base d'un brin est associée sur l'autre brin une base complémentaire : {A←→T} et {C←→G} Cette complémentarité permet donc de définir totalement l'ADN à partir d'un seul des deux brins. Les nucléotides peuvent être rangés en deux catégories : Les Purines {A, G} et les Pyrimidines {C, T}. La chaine ADN est constituée d'une alternance de zones codantes (les exons) et de zones non codantes (les introns). Les exons (environ 2% chez l'homme) sont des fragments qui contiennent le matériel génétique (les gènes). Chaque gène contient toutes les instructions nécessaires à la production d'une protéine (processus de traduction). Les introns (environ 90% chez l'homme) contiennent aussi de l'information indispensable, par exemple les promoteurs qui sont des segments de chaine situé avant le début d'un gène permettant d'initier la transcription.

II.3) ARN (Acide RiboNucléique)

L'ARN est une macromolécule biologique constitué d'une seule chaîne de nucléotides, obtenu à partir d'un des deux brins d'ADN par le processus de transcription (mécanisme biologique qui permet de dupliquer un brin d'ADN sous forme d'ARN). A chaque nucléotide de la séquence d'ADN va correspondre le nucléotide complémentaire pour former l'ARN, sauf pour l'Adénine. En effet, pour former l'ARN, le complémentaire de l'Adénine n'est pas la Thymine mais l'Uracile. La chaine ARN est composée de quatre bases azotées {A, C, G, U} pour {Adénine, Cytosine, Guanine, Uracile}, les bases complémentaires sont {A←→U} et {C←→G}.

L'ARN est subdivisé en trois catégories, chacune ayant un rôle différent :

- ARN messager (ARNm) est utilisé chez les eucaryotes pour véhiculer l'information génétique du noyau vers le cytoplasme où elle sera traduite en protéine par des ribosomes. L'épissage mécanisme consistant, sur l'ARN messager qui vient d'être transcrit, à éliminer les introns et réunir les exons entre eux. Le produit de l'épissage est un ARN messager mature, prêt à être traduit en protéine.

- ARN ribosomique (ARNr) forme une trame sur les ribosomes, afin de permettre aux protéines synthétisées par ces ribosomes de se fixer. Il constitue 80% de l'ARN total d'une cellule. Le ribosome constitue la tête de lecture de l'information génétique transcrite.

- ARN non-codant (ARNnc) est un ARN fonctionnel qui n'est pas traduit en protéine (tous les ARN autres que les ARNm). Les molécules d'ARN sont donc impliquées dans de nombreux processus biologiques au sein des cellules.

II.4) Protéines

Les protéines sont de très grosses molécules qui jouent le rôle des ouvrières du monde cellulaire. Dans le corps humain il y a plus d'un million de protéines différentes qui assurent ensemble des milliers de fonctions indispensables à notre vie (respirer, transporter, protéger, digérer, réparer, voir, contrôler, …). Tout le savoir-faire (le plan) pour fabriquer ces protéines est stocké dans des gènes de l'ADN et écrit dans un langage génétique composé de quatre lettres {A, C, G, T}. Au besoin, une cellule va déclencher la production d'une protéine donnée, à partir de l'ADN, elle va réaliser une copie (à un seul brin) des instructions du gène impliqué, c'est l'ARNm (dont la nature chimique est très proche de l'ADN) qui contient les instructions nécessaires à la construction de la protéine (les exons). Cette copie quitte le noyau (pour les eucaryotes) pour aller dans le cytoplasme qui contient une machinerie (les ribosomes) pour traduire la chaine de nucléotides en une chaine d'acides aminés en utilisant le code génétique. A chaque série de trois nucléotides (codons) correspond un des 20 acides aminés constitutifs des protéines, représentés par un alphabet de 20 lettres dont la liste est donnée dans la table. 2. Il existe donc $4^3 = 64$ possibilités pour former les codons. Trois des codons sont utilisés pour stopper la synthèse des protéines et les 61 restants servent à coder les acides aminés (il existe plusieurs codons possibles pour un même acide aminé).

1er nucléotide (en 5')	2e nucléotide U	2e nucléotide C	2e nucléotide A	2e nucléotide G	3e nucléotide (en 3')
U	Phe:F	Ser:S	Tyr:Y	Cys:C	U
	Phe:F	Ser:S	Tyr:Y	Cys:C	C
	Leu:L	Ser:S	STOP	STOP	A
	Leu:L	Ser:S	STOP	Trp:W	G
C	Leu:L	Pro:P	His:H	Arg:R	U
	Leu:L	Pro:P	His:H	Arg:R	C
	Leu:L	Pro:P	Gln:Q	Arg:R	A
	Leu:L	Pro:P	Gln:Q	Arg:R	G
A	Ile:I	Thr:T	Asn:N	Ser:S	U
	Ile:I	Thr:T	Asn:N	Ser:S	C
	Ile:I	Thr:T	Lys:K	Arg:R	A
	Met:M	Thr:T	Lys:K	Arg:R	G
G	Val:V	Ala:A	Asp:D	Gly:G	U
	Val:V	Ala:A	Asp:D	Gly:G	C
	Val:V	Ala:A	Glu:E	Gly:G	A
	Val:V	Ala:A	Glu:E	Gly:G	G

Table. 2 les 20 acides aminés constitutifs des protéines

Les protéines synthétisées adoptent ensuite une conformation tridimensionnelle spécifique, qui implique des proximités spatiales entre groupements chimiques distants dans la séquence. Cette conformation a des propriétés physiques et chimiques particulières qui sont le support de l'activité biologique de chaque protéine. Il s'établit donc une relation entre la séquence d'une protéine, sa structure et sa fonction. Les fonctions des protéines sont très variées et permettent de les classer :

- les protéines de structure sont comparables à des briques cellulaires (ex : le collagène) ;

- les protéines de transport sont chargées du transport d'autres molécules dans la cellule ou entre les cellules d'un organisme (ex : l'hémoglobine transporte l'oxygène) ;

- les enzymes permettent d'accélérer les réactions chimiques au sein de la cellule nécessaires à la vie ;

- les protéines de l'immunité (ou anticorps) contribuent à la défense de notre organisme.

La structure tridimensionnelle d'une protéine est l'un des éléments qui détermine sa fonction.

II.5) Gène et Allèle

Un gène est un fragment d'ADN qui mémorise tout le savoir nécessaire pour la fabrication d'une protéine. Ce fragment est formé par la séquence des codons (série de trois nucléotides). Un codon Stop marque la fin de la traduction d'un gène en protéine. Il n'est en général jamais traduit car il n'existe pas d'ARN de transfert correspondant (il existe 2 acides aminés supplémentaires, la sélénocystéine et la pyrrolysine qui sont insérés lorsqu'un codon STOP particulier est rencontré). Cette traduction débute au niveau du codon d'initiation formé le plus souvent des trois lettres AUG (et plus rarement CUG ou UUG), et se termine par un des trois codons de terminaison universels, UAA, UAG et UGA.

Les séquences génomiques abritent plusieurs types de gènes : les gènes codant pour des protéines, mais aussi des gènes codant pour des ARN structuraux, molécules indispensables au processus de la traduction des ARN messagers en protéines. Il s'agit des ARN ribosomiques, constituants essentiels des sous-unités ribosomiques impliquées dans le processus de la traduction, et des ARN de transfert qui permettent d'établir la correspondance entre les codons d'un gène en cours de traduction et les acides aminés qui composent la protéine finale.

L'ADN peut varier grâce à des mutations qui créent de nouvelles versions des gènes (allèles) qui peuvent avoir des séquences différentes mais qui définissent un même caractère.

II.6) Phénotype et Génotype

Le Phénotype est l'ensemble des caractères observables et mesurables d'un individu. Il dépend du génotype. Le génotype est l'ensemble des gènes et des allèles d'un individu. Les individus de la même espèce possèdent le même génome mais n'ont pas le même phénotype. Les mutations modifient l'ADN et conduisent notamment à de nouvelles versions des gènes (les allèles), et ainsi de la diversité du vivant. Exemple le groupe sanguin est déterminé par un gène, appelé ABO qui existe en trois versions : l'allèle A, l'allèle B et l'allèle O.

II.7) Génome, transcriptome et protéome

Le génome est la totalité du matériel génétique de la cellule d'une espèce. L'ensemble des transcrits (les ARN messagers) d'un organisme à un instant donné est nommé transcriptome. Le protéome désigne l'ensemble des protéines exprimées par le génome d'une cellule, d'un tissu ou d'un organe à un moment donné. La protéomique est la discipline qui a pour objectif d'aboutir à l'identification des protéines que contient un extrait de tissu organique. La Génomique est la discipline qui a pour objectif de séquencer l'ADN d'un organisme et de localiser sur celui-ci tous les gènes qu'il porte, puis de caractériser leurs fonctions.

II.8) Substitution, insertion et délétion (la mutation)

Une mutation est une modification ou un changement qui intervient dans la séquence du matériel génétique (ADN). Elle est produite suite à une erreur survenue au cours de la réplication de l'ADN ou des lésions diverses provoquées par des agents chimiques ou physiques (rayon X, les ultras violet …). Les mutations peuvent introduire une modification dans l'information génétique qui est transmissible à sa descendance, constituant ainsi la source de la diversité entre individus (le moteur de l'évolution). Mais elles sont aussi à l'origine des maladies génétiques et des prédispositions génétiques aux maladies (mutation pathogène).

Les mutations se produisent lorsqu'un nucléotide est perdu (délétion), remplacé par un autre (substitution) ou ajouté dans la séquence (insertion). Un gène muté peut alors coder une protéine différente ou ne pas modifier l'information et toujours coder la même protéine (c'est au moment de la traduction, que l'effet sur la protéine codée par le gène se manifeste). Parfois, la substitution d'un nucléotide n'entraîne pas de changement de la séquence peptidique (la structure dans l'espace de la

protéine reste inchangée, sa fonction de même). C'est dans les cas de délétion ou insertion que les conséquences sont le plus souvent très graves.

Les mutations sont la première source de variation du matériel génétique à la base de l'évolution des êtres vivants. La conséquence de toute mutation dépend de son effet fonctionnel, qui peut être neutre, conduire à l'amélioration d'une fonction (diversité, évolution) ou à l'altération d'une fonction (effet pathogène). Exemple, les SNPs (Single Nucleotide Polymorphisms) polymorphismes de substitution au niveau d'un nucléotide (variation de séquence ponctuelle) sont référencés dans la base de données dbSNP (http://www.ncbi.nlm.nih.gov/projects/SNP/).

II.9) Séquence, structure, fonction et réseau

En laboratoire, il est possible de déterminer la séquence (la structure primaire) à la fois des protéines, de l'ADN et de l'ARN. Cette expérience, appelé séquençage, consiste à identifier la chaine de caractères composée de la succession des nucléotides (pour l'ADN ou l'ARN) ou d'acides aminés (pour les protéines). Cet ordre est très important, il constitue l'identité qui pourrait expliquer leurs propriétés biologiques. La modification d'un acide aminé peut dans certains cas avoir des conséquences néfastes. Ainsi, la drépanocytose est une maladie caractérisée par l'altération de l'hémoglobine. Cette mutation entraîne de nombreux symptômes : anémies, infections et risques d'accidents vasculaires cérébraux. Cette maladie qui touche 50 millions d'individus dans le monde est causée par la modification d'un seul acide aminé (Ramstein, 2012).

On caractérise l'organisation dans l'espace des macromolécules biologiques (ADN, ARN ou protéines) par leurs structures secondaire et tertiaire. La représentation de l'ADN sous forme de double hélice est appelée la structure secondaire. Dans les protéines la structure secondaire décrit le repliement local de la chaîne principale. La structure tertiaire est la forme 3D d'une protéine (figure.18) correspondant au repliement de la chaîne polypeptidique dans l'espace. La structure tertiaire d'une protéine joue un grand rôle dans sa fonction, si celle-ci est modifiée, la protéine est alors dénaturée et elle perd sa fonction. La structure quaternaire est la forme prise par l'assemblage de plusieurs protéines.

L'ordre d'acides aminés détermine l'identité d'une protéine (l'insuline : MALWMRPLLPLLALLA ... ENYCN) mais la forme que la protéine adopte dans son environnement permet de connaître sa fonction biologique :

- transport (hémoglobine, albumine, transporteurs membranaires)
- structure (spectrine)
- travail mécanique (actine et myosine)
- hormones, récepteurs (insuline, récepteur de l'insuline)
- immunoglobulines (IgG, IgM)

La fonction d'une protéine ne peut être comprise que par sa structure. La structure qualifie le type de forme adoptée par un segment d'acides aminés. Cette forme provient des liaisons chimiques entre les acides aminés et possède certaines propriétés telle la stabilité. Les biologistes reconnaissent deux structures : une hélice α, un feuillet β. Ces deux structures sont les plus stables 2 connues aujourd'hui.

Figure.18 structure 3D d'une protéine

La découverte de la fonction d'une protéine est un axe de recherche majeur en génomique, puisque les processus biologiques sont activés par ces molécules. Par exemple, l'hémoglobine est une protéine assurant le transport de l'oxygène dans le sang.

Un réseau biologique est une représentation de la circulation d'un certain type d'information dans la cellule. Il en existe plusieurs types :

- Réseau génétique ou de régulation : le gène A régule l'expression du gène B
- Réseau d'interaction protéine-protéine : La protéine A interagit physiquement avec la protéine B
- Réseau de signalisation : la protéine A transmet un signal informatif à la protéine B
- Réseau métabolique : l'ensemble des réactions chimiques dans une cellule.

La compréhension des réseaux opérant dans la cellule permet de mieux cerner les phénomènes qui conduisent à une maladie.

II.10) Réplication, transcription et traduction

Les trois processus de base du dogme central introduit par Francis Crick, c'est-à-dire, la réplication (ADN), la transcription (ARN) et la traduction (en protéines) du matériel génétique (figure.19).

- Réplication : réalisation d'une copie (à un seul brin) des instructions du gène impliqué de l'ADN. Elle est assurée par les ADN polymérases.
- Transcription : l'ADN est copié en ARN messager (ARNm). Seules certaines portions de l'ADN sont transcrites, ces séquences sont appelés gènes.
- Traduction : mécanisme qui permet de traduire l'ARNm en protéines par les ribosomes.

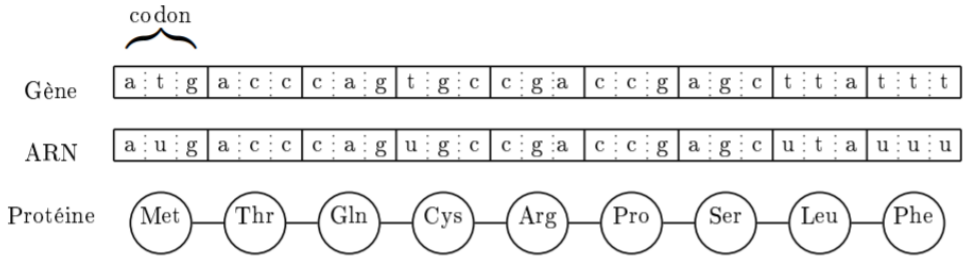


Figure.19 Les trois processus de base (réplication, transcription et traduction)

L'information génétique est ainsi codée dans les séquences d'ADN (support de l'information génétique). L'ADN se réplique grâce à la complémentarité des deux brins, la cellule recopie son génome avant de se diviser en deux cellules filles. Pour ce faire, les deux brins de la molécule d'ADN sont séparés et chacun sert de modèle pour la formation d'un nouveau brin complémentaire. Les gènes de l'ADN sont transcrits en séquences d'ARNm (L'ADN s'exprime par l'intermédiaire des ARNs), elles-mêmes traduites en séquences d'acides aminés pour synthétiser les protéines. L'épissage est le mécanisme par lequel les introns sont éliminés du transcrit primaire.

III) Quelques notions de bioinformatique

Les notions de biologie vues à la section précédente montrent à quel point les systèmes biologiques sont très complexes. Divers formalismes ont été proposés pour modéliser ces systèmes biologiques, à partir de trois types de données organisées dans des bases de données adaptées au traitement informatique:

- Les séquences d'ADN, d'ARN et de protéines
- Les structures d'ARN et de protéines
- Les réseaux (d'interactions entre protéines, de régulations génétiques et métaboliques).

III.1) quelques définitions de la bioinformatique

Plusieurs définitions existent :

- La bioinformatique est la science dédiée à la gestion, l'organisation, la comparaison, la classification, l'analyse et la modélisation de l'information biologique afin de produire des connaissances biologique 1979 par Paulien Hogeweg
- La bioinformatique est le domaine de la science dans lequel la biologie, l'informatique et les technologies de l'information sont combinées en une seule discipline (Définition du NCBI 2001).
- La Bioinformatique est l'application de l'informatique à la gestion et l'analyses des données biologiques. (European Bioinformatic Institute : EBI, 2004).
- Intégration des méthodes mathématiques, statistiques et informatiques pour analyser les données biologiques, biochimiques et biophysiques (Georgia Inst of Tech., USA).
- La bioinformatique est l'étude de l'information biologique quand elle passe de son site de stockage dans le génome aux différents produits des gènes dans la cellule. Elle inclut la création et le développement de technologies informatiques avancées pour les problèmes de la biologie moléculaire. (Stanford University, USA).
- La bioinformatique se réfère spécifiquement à la recherche et à l'utilisation de patterns et de structures dans les données biologiques et au développement de nouvelles méthodes pour accéder aux bases de données. (Virginia Inst Tech., USA).
- Bioinformatique : recherche, développement ou application d'outils informatiques [computationnels] et d'approches pour étendre l'utilisation des données biologique, médicales, comportementales ou sanitaires, y compris [les outils et approches] pour acquérir, entreposer, organiser, archiver, analyser ou visualiser de telles données. (National Institute of Health (NIH), USA).

III.2) Chronologie du développement de la bioinformatique

L'histoire de la bioinformatique peut être résumée par la chronologie suivante :

1951 : Frédérick Sanger détermine la séquence des acides aminés de l'insuline.

1960 : Lien entre séquence & structure (Perutz et al., 1960).

1965 : La divergence et la convergence évolutionnaire dans les protéines (Zuckerkandl & Pauling, 1965).

1965 : Première compilation de protéines ("Atlas of Protein Sequences"). Matrices de substitution (Dayhoff et al., 1965)

1967 : La construction des arbres phylogénétiques (Fitch & Margoliash, 1967).

1970 : Algorithme pour l'alignement global de deux séquences (Needleman & Wunsch, 1970).

1971 : Premier travaux sur le repliement des ARNs (Ninio, 1971).

1972 : Clonage de fragments d'ADN dans un virus, l'ADN recombiné : Paul Berg, David Jackson, Robert Symons

1972 : Le groupe de Fiers en Belgique détermine la séquence d'un gène.

1973 : 1973 Début du génie génétique. Premières expériences de recombinaison génétique.

1973 : Découverte des enzymes de restriction qui coupe spécifiquement l'ADN. Méthode de transfection (introduction d'un ADN étranger) des cellules eucaryotes grâce à un virus (vecteur).

1974 : Programme de prédiction de structures secondaires des protéines (Chou & Fasman).

1977 : séquençage du génome du Bactériophage par F. Sanger

1977 : Premier "package" Bioinformatique (Staden).

1978 -1980 : Séquençage du 1er génome à ADN, le bactériophage phiX174 : Frederick Sanger
Premières bases de données : EMBL, GenBank, PIR.

1980 Première expérience de thérapie génique, non autorisée, par l'Américain Cline.
1981 : Programme d'alignement local de séquences (Smith & Waterman, 1981).
1984 : Amplification de l'ADN : réaction de polymérisation en chaîne (PCR - Karry Mullis)
1985 : "FASTA" : Programme d'alignement local de séquences (Lipman & Pearson, 1985)
1987 : Nouveau vecteur permettant de cloner des fragments d'ADN 20 fois plus grands : le YAC (Yeast Artificial Chromosome) qui rend possible le séquençage de grands génomes.
1988 : Taq polymérase, enzyme thermostable pour la PCR. Création du "National Centre for Biotechnology Information" (NCBI).
1990 : - Clonage positionnel et premier essai de thérapie génique.
- BLAST : Programme d'alignement local de séquences (Altschul et al., 1990)
1991 : - Grail, programme performant pour localiser les gènes (Mural et al., 1991).
- Expressed Sequences Tags (EST) : méthode rapide d'identification des gènes.
1992 : Séquençage complet du chromosome III de levure.
1993 : European Bioinformatics Institute (EMBL). Création à terme du "European Bioinformatics Institute" (EMBL - EBI).
1995 : Première séquence complète d'un micro-organisme H. influenza (Fleischmann et al.,. 1995)
1995 : Analyse du transcriptome : début des puces à ADN
1996 : Séquence complète de la levure (consortium européen).
1997 : - 11 génomes bactériens séquencés
- Evolutions de BLAST : "Gapped BLAST" et "PSI-BLAST" (Alschul et al., 1997).
1998 : Séquençage du 1er organisme pluricellulaire, Caenorhabditis elegans (100 Mb).
2000 : Séquençage du 1er génome de plante, Arabidopsis thaliana.
2001 : Séquençage ("premier jet") complète du génome humain.
2004 : Les ARN interférents (ARNi) sont des ARN synthétisés en laboratoire. Ils permettent, lorsqu'ils sont injectés dans une cellule, de se fixer sur les ARNm à leur sortie du noyau et donc d'empêcher leur traduction en protéines. Cela occasionne une baisse importante de la quantité de cette protéine dans la cellule (Meister et Tuschl, 2004).
L'ARN pourrait donc être utilisé à des fins médicales (tel que les ARNi), notamment dans la régulation de l'expression génétique, comme guide pour des enzymes, dans le contrôle de la réplication des plasmides, etc.
2007-2008 : - Avènement des nouvelles technologies de séquençage à très haut débit, dites de secondegénération et maintenant de 3è génération.
- Prise de conscience du phénomène "big data" (pas seulement en biologie) qui devient peu à peu une discipline scientifique.
- Le projet 1000 Génomes, débuté en 2008, a pour objectif de séquencer le génome de 2500 personnes, afin d'identifier les spécificités de chacun.
2010 : projets « génome » (génomes complets ou en cours de séquençage, métagénomes) cf. GOLD http://www.genomesonline.org/
2019 : Plus de 18900 génomes eucaryotes et procaryotes séquencés et des milliers en cours de séquençage (Genomes OnLine).

III.3) Modélisation des données biologiques

Avec le développement des nouvelles technologies à très haut débit, une vaste quantité de données expérimentales est produite. Le premier objectif de la bioinformatique est de stocker et d'organiser ces données dans des dépôts de données (banques ou bases de données), afin d'analyser et prédire le comportement des systèmes vivants dans des conditions de fonctionnement normales ou pathologiques.

La modélisation est le principal outil utilisé pour l'étude des entités biologiques et leurs interactions. Une première étape dans le travail de modélisation est d'une part de formaliser les entités biologiques qui vont être modélisées, et d'autre part de définir les relations qui existent entre ces entités.

Les modèles font abstraction d'une partie de la réalité, afin de restreindre la complexité de la réalité, c'est-à-dire le nombre d'informations prises en compte, seules les informations considérées comme essentielles sont décrites dans un modèle. L'abstraction de la réalité est basée sur des hypothèses simplificatrices concentrées sur l'aspect biologique qui sera étudié au travers du modèle. Un formalisme est une manière de décrire les entités et leurs relations au sein d'un modèle.

III.3.1) représentation des séquences

Une macromolécule biologique, qu'il s'agisse d'un acide nucléique (ADN, ARN) ou d'une protéine, est le plus souvent représentée par sa séquence, une représentation linéaire, à une seule dimension. Les séquences sont issues principalement de la technique du séquençage (technologie permettant de lire directement la séquence d'ADN). Malgré la complexité des molécules biologiques, les séquences sont représentées par de simples chaînes de caractères. Ainsi l'ADN et l'ARN sont représentés chacun par un ensemble de quatre lettres appelées nucléotides, {A, C, G, T} et {A, C, G, U}. Les protéines sont représentées par un ensemble de 20 lettres, appelées acides aminés ou résidus, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}.

On appelle séquence S sur un alphabet Σ une suite finie et ordonnée d'éléments (lettres) x_i appartenant à Σ, $S = x_1, x_2, \ldots x_n$. On appelle longueur d'une séquence le nombre d'éléments qui la composent, on la note $|S| = n$. Soit S une séquence de longueur n. On appelle sous-séquence de S toute partie de S composée d'un ensemble de caractères consécutifs de S, que nous noterons S_{ij} avec $1 \leq i \leq j \leq n$ la sous-séquence $S_{ij} = x_i \ldots x_j$ avec $|S_{ij}| = j-i+1$. Pour i = j nous avons $S_{i,i} = S_i = x_i$ c'est le $i^{ème}$ élément de la séquence S. On appelle préfixe d'une séquence S de longueur h toute sous-séquence $S_{1,h}$, avec $1 \leq h < n$. Les séquences nucléotidiques ou d'acides aminés sont généralement stockées au format FASTA.

Dans l'analyse de données biologiques, l'analyse de séquences occupe le devant de la scène. En effet les applications les plus courantes de la bioinformatique sont l'analyse et la comparaison de séquences (les alignements multiples de séquences, la prédiction de la structure et de la fonction d'une protéine à partir de sa séquence…).

III.3.2) Représentation de familles de séquences

- **Séquence consensus** : séquence de longueur n contenant, à chaque position, le symbole le plus fréquent à la même position dans l'alignement.

Exemple : n=5 et Σ = {a, c, g, t, –}

```
a c g – t
a c a c t
a g g c –          le consensus a c g c t
g c – c g
```

Alignement

- **Matrice consensus** : matrice $\Sigma \times n$ contenant la fréquence d'apparition de chaque symbole à chaque position de l'alignement. n est la longueur de l'alignement sur l'alphabet Σ.

Exemple : n=5 et Σ = {a, c, g, t, –}

```
a c g – t
a c a c t
a g g c –
g c – c g
```

Alignement

	C1	C2	C3	C4	C5
a	0,75	0	0	0,25	0
c	0	0,75	0	0,75	0
g	0,25	0,25	0,5	0	0,25
t	0	0	0	0	0,5
-	0	0	0,25	0,25	0,25

Matrice consensus

- **Motif** : expression rationnelle décrivant l'ensemble des séquences ou une partie particulièrement conservée

III.3.3) représentation des structures

Ces données sont issues de différentes techniques d'expérimentation : analyse cristallographiques, résonance magnétique nucléaire, cryomicroscopie électronique. Une autre technique de modélisation moléculaire concerne la prédiction de la structure 3D d'une protéine à partir de sa structure primaire (sa séquence), en prenant en compte les différentes propriétés physico-chimiques des acides aminés. De même, la modélisation des structures 3D d'acides nucléiques (à partir de leur séquence nucléotidique) pour les structures d'ARN.

a) Structure secondaire de l'ARN

Les structures secondaires et tertiaires de l'ARN sont bien plus variées et porteuses d'informations fonctionnelles que celles de l'ADN. La structure secondaire d'un ARN est caractérisée par un ensemble de liaisons entre ses bases. En effet, la structure simple brin de l'ARN permet un repliement de la molécule sur elle-même, au moyen d'appariements Watson-Crick (en substituant l'uracile à la thymine), mais aussi d'appariements Wobble ou Hoogsteen. Prédire la structure secondaire a donné lieu à de nombreuses avancées.

- La représentation en squiggle-plot

Elle est structurée en tiges ou hélices (paires de bases contiguës) et boucles (nucléotides non-appariés, entourés d'hélices) (figure. 20).

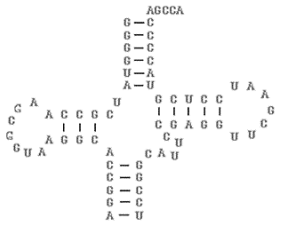


Figure.20 représentation en squiggle-plot

- La représentation en dot-bracket

Chaque structure de taille *n* est représentée par une séquence de caractères qui sont des parenthèses et des points. Chaque paire de bases appariées (*i*, *j*) est représentée par une paire de parenthèse telle que : le caractère à la $i^{ème}$ position est une parenthèse ouverte "(" et le caractère à la $j^{ème}$ position est une parenthèse fermante ")". Chaque base non-appariée est représentée par un point ".". (figure. 21)

```
00 01 02 03 04 05 06 07 08 09 10
 G  C  A  A  U  G  C  U  U  A  A
 .  .  (  (  .  .  .  )  )  .  .
```

Figure.21 Représentation dot-bracket (les bases (02, 08) sont en interactions, (03, 07) aussi.)

Séquence : GGACAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGUCC

Structure secondaire: .(((((((..(.(((((.......))))).).....).((((((.......))))))..)))))).

- La représentation en forme de graphe

La structure secondaire d'un ARN peut être modélisée par un graphe (Structures secondaires de Waterman (Waterman, 1978) composé de sommets {1, . ., n} et doté d'une matrice d'adjacence A = (aij) telle que :

- $a_{i\,i+1} = 1, \forall i \in [1, n-1]$
- $\forall i \in [1, n]$, il existe au plus un $j \neq i \pm 1$ tel que $a_{ij} = 1$
- Si $a_{ij} = 1$, $a_{kl} = 1$ et $i < k < j$, alors $i \leq l \leq j$

b) Structure tertiaire de l'ARN

La structure tertiaire, est la localisation des constituants chimiques de la molécule dans l'espace. Le repliement de l'ARN dans un espace 3D peut être vu comme un processus en deux étapes :

- le repliement en structure secondaire, grâce à des interactions fortes,
- le repliement en structure tridimensionnelle par des interactions tertiaires.

Les fonctions d'une molécule d'ARN sont très étroitement liées à sa structure tridimensionnelle. La structure secondaire, représentant les interactions canoniques entre les paires de nucléotides les plus stabilisatrices Watson-Crick (G–C ou A–U) et Wobble (G–U). Il résulte de ce processus une forme tridimensionnelle complexe essentielle, chez la plupart des ARN, pour la réalisation d'une fonction spécifique. L'étude conjointe de la séquence de bases et de la structure semble donc nécessaire à la compréhension du rôle d'un ARN.

c) Structure secondaire de la protéine

La structure secondaire des protéines consiste en un réseau d'interactions locales entre résidus d'acides aminés, observable à l'échelle atomique et stabilisés par des liaisons hydrogène. Trois types principaux de structures secondaires sont observés (les hélices α, les feuillets β et les tours). Une hélice α ressemble à un enroulement de la chaine principale autour d'un axe virtuel, d'où sa représentation usuelle sous la forme d'un ruban en serpentin. Les feuillets β forment la structure secondaire la plus fréquente après les hélices α puisque 20 à 28% des résidus se retrouvent dans ces structures secondaires. La troisième structure secondaire majeure est une séquence de 3 acides aminés caractérisés par une liaison hydrogène entre le premier et le troisième résidu. Contrairement aux hélices et aux feuillets, les tours sont des structures secondaires très courtes, bien qu'environ 25 à 30% des acides aminés s'y retrouvent.

La position relative (dans l'espace) des différents éléments de structure secondaire les uns par rapport aux autres décrit la structure tertiaire de la protéine.

d) Structure 3D de la protéine

Les différents éléments de la structure secondaire interagissent entre eux pour former la structure tertiaire. La structure tridimensionnelle correspond à la forme générale de la protéine observable à l'échelle de la molécule tout entière. Le repliement d'une séquence d'acides aminés de longueur *L* peut être représenté par une matrice de contacte *Mat(i, j),*on considère qu'il y a contact entre deux résidus si la distance dans l'espace est inferieur un seuil.

$$Mat(i,j) = f(x) = \begin{cases} 1 & si\ i,j\ sont\ en\ contact\ dans\ la\ structure\ 3D \\ 0 & sinon \end{cases}$$

Une représentation simplifiée du repliement est donnée dans la figure 22. Chaque point représente un résidu et un arc rejoint deux résidus s'ils sont en contact lorsque la protéine est repliée.

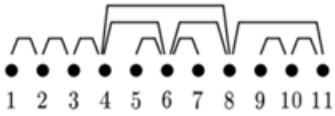


Figure. 22 représentation schématique d'une structure 3D d'une protéine

La prédiction de la structure tertiaire des protéines est un problème très important car elle est étroitement liée à sa fonction. Tout comme la recherche de similitudes entre les séquences se fait en termes d'alignement, c'est-à-dire de la mise en concordance des acides aminés sur la base de leurs similitudes physico-chimiques, on peut aussi superposer des structures et déduire un alignement structural sur la base des appariements 3D. Un résultat fondamental a été de constater que des protéines de séquences voisines se replient généralement d'une façon similaire : une séquence proche donne une structure tridimensionnelle proche, mais la réciproque n'est pas vraie. Si l'on connaît la structure d'une protéine suffisamment proche, homologue, de celle que l'on souhaite étudier, on peut calculer, modéliser, sa structure. Cette constatation a donc permis de s'affranchir des données expérimentales pour modéliser à grande échelle la structure des protéines.

e) structure quaternaire

La nature des protéines est déterminée par leur séquence en acides aminés (la structure primaire). Les acides aminés ayant des propriétés chimiques très diverses, leur disposition le long de la chaîne polypeptidique détermine leur arrangement spatial. Celui-ci est décrit localement par leur structure secondaire, stabilisée par des liaisons hydrogène entre résidus d'acides aminés voisins, et globalement par leur structure tertiaire, stabilisée par l'ensemble des interactions entre les résidus (parfois très éloignés sur la séquence peptidique mais mis en contact spatialement par le repliement de la protéine) ainsi qu'entre la protéine elle-même et son environnement. Enfin, l'assemblage de plusieurs sous-unités protéiques pour former un complexe fonctionnel est décrit par la structure quaternaire de cet ensemble.

On peut distinguer trois grands groupes de protéines en fonction de leur structure tertiaire ou quaternaire : les protéines globulaires, les protéines fibreuses et les protéines membranaires. Presque toutes les protéines globulaires sont solubles et ce sont souvent des enzymes. Les protéines fibreuses jouent souvent un rôle structurel, à l'instar du collagène, constituant principal des tissus conjonctifs, ou de la kératine, constituant protéique des poils et des ongles. Les protéines membranaires sont souvent des récepteurs ou des canaux permettant aux molécules polaires ou électriquement chargées de traverser la membrane.

III.3.4) représentation des réseaux

Les biologistes ne s'intéressent pas seulement aux objets biologiques, mais aussi à leurs interactions (dans les cellules, des milliers de macromolécules sont en interactions). Ces interactions étant des systèmes complexes, leur étude nécessite d'utiliser une représentation simplifiée sous forme de réseaux pour modéliser les relations entre les molécules biologiques. A travers une représentation d'un système sous forme de réseau, les composants sont représentés par des sommets, et les interactions entre ces composants par des arêtes, mais de nombreux détails associés au système étudié sont éliminés. Par exemple les réseaux d'interactions protéine-protéine, peuvent être modélisés par des graphes simples où chaque protéine est représentée sous forme d'un nœud, et les interactions sous forme des arrêts entre ces nœuds (Sharan & Ideker, 2006). L'analyse de ces réseaux est possible grâce au développement de méthodes de classification de graphes.

Il existe plusieurs formalismes mathématiques permettant de modéliser les réseaux de régulation génique. Chaque formalisme permet de représenter avec davantage de précision certains aspects caractéristiques des réseaux. Les modèles d'équations différentielles ordinaires sont probablement le formalisme le plus utilisé pour modéliser les réseaux de régulation génique. Exemple : deux gènes, *a* et *b*, forment un réseau de régulation génique et se régulent de la manière suivante : l'expression de *a* est activée par la présence du produit de l'expression de *b* et par son propre produit d'expression, quand celle de *b* est inhibée par le produit de l'expression de *a* en concurrence avec l'activation due à sa propre expression. Le réseau résultant est représenté dans la

figure 23, les interactions sont représentées avec des arcs orientés : s'il s'agit d'une activation, l'arc se termine par une flèche, s'il s'agit d'une inhibition, l'arc se termine par une barre.

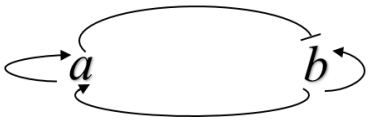


Figure. 23 Réseau de régulation génique de a et b

IV) Bases de données bioinformatique

Les bases de données dédiées à la biologie, Les bases de connaissances biomédicales et les chimio-thèques (banques de données de molécules) sont la matière première à partir de laquelle des méthodes bioinformatique (in silico) vont produire d'autres données et construire de nouvelles connaissances. Les bases de données biologiques sont distinguées par la nature des données qu'elles stockent : séquences d'ADN, de protéines, des gènes, des motifs (fragments de protéine liés à une fonction), des structures, etc. Les fichiers contenant l'information biologique sous la forme de séquences est l'élément central autour duquel les banques de données se sont constituées à l'origine. Ainsi, il existe trois banques principales de séquences d'ADN généralistes : EMBL (gérée par EBI, Europe), GENBANK (NCBI, USA) et DDBJ (Japon), accessibles via Internet. Elles partagent les mêmes données et constituent de ce fait trois points d'entrée d'une seule et même banque mondiale. Le terme banque rappelle ici que les séquences y sont déposées directement par les chercheurs qui les ont obtenues, sous leur seule responsabilité. Ces quantités massives de données sont réalisées dans les laboratoires par expérience traditionnelle (in vivo ou in vitro) ou au moyen de la bioinformatique (in silico). Par exemple les protéines peuvent être obtenues in silico (comme dans la base trEMBL qui contient toutes les protéines déduites à partir des séquences nucléiques contenues dans EMBL) ou isolées à partir des cellules (comme dans la base SWISSPROT). L'origine de la collecte des données est un facteur intéressant pour les biologistes car par exemple la déduction d'une protéine ne permet d'établir qu'une probabilité, et non une certitude sur son existence dans la nature.

La conception d'un système d'information capable de stocker, gérer, intégrer et récupérer ces quantités massives de données est un défi majeur pour les bioinformaticiens. Différents outils et méthodes bioinformatique peuvent ensuite être développés pour l'analyse et l'extraction de connaissances à partir de ces bases de données.

IV.1) Intégration des bases de données

La diversité des modèles et des formats des bases concernées constitue un véritable problème. La question fondamentale est ainsi de savoir comment intégrer ces données biologiques hétérogènes et distribuées, afin de les rendre accessibles et exploitables aussi facilement que si elles figuraient dans une seule et même base. Deux solutions sont envisageables : la première approche dite « fédérative » consiste à ajouter, au-dessus des bases existantes, une couche logicielle qui offre les interfaces nécessaires entre les bases et fasse apparaître l'ensemble comme une seule base virtuelle. La seconde approche est celle des entrepôts de données (data wharehouse), qui consiste à copier les données des différentes bases concernées de leurs bases d'origine et les restructurer au sein d'un schéma unique (Morgat & Rechenmann, 2002). On peut citer par exemple SRS (Etzold et al.1996), Kleisli (Davidson et al., 1997) ou Discovery link développé par la société IBM.

IV.2) Les plus importantes bases de données biologiques

Plusieurs centaines de bases de données dédiées à la biologie moléculaire sont disponibles. Ce nombre augmente chaque année. La revue Nucleic Acids Research consacre chaque année son

numéro de janvier à une revue des bases de données existantes, et maintient un catalogue des bases de données (http://www.oxfordjournals.org/nar/database/c/). On peut distinguer :

- Les bases de données généralistes : elles correspondent à une collecte des données la plus exhaustive et la plus large possible (multi-espèces) et qui offrent un ensemble plutôt hétérogène d'informations.
- Les bases de données spécialisées : elles correspondent à des données plus homogènes (provenant d'une seule espèce) établies autour d'une thématique particulière.

Ces bases de données sont gérées par des organismes mondiaux, les plus importants sont :

- NCBI (National Center for Biotechnology Information),
- EMBL-EBI (European Molecular Biology Laboratory - European Bioinformatic Institute),
- UCSC (University of California at Santa Cruz),
- RCSB (Research Collaboratory for Structural Bioinformatics),
- SIB (Swiss Institute of Bioinformatics).

Les données biologiques stockées dans ces bases de données sont issues d'expériences, d'analyses faites à la main par des chercheurs, de publications et de raisonnements automatiques, et qui peuvent être schématiquement de différentes natures :

- Les données de séquences (ADN, ARN et protéines),
- Les données structurales,
- Les données d'expression (transcriptome et protéomes),
- Les données génomiques,
- Les données fonctionnelles (Données métaboliques issues de la génomique fonctionnelle),
- Les données médicales,
- Les données bibliographiques ...

IV.2.1) Bases de séquences nucléiques

Il existe trois principales bases de séquences nucléiques généralistes, qui sont interconnectées dans le cadre du consortium INSDC (International Nucleotide Sequence Database Collaboration, http://www.insdc.org/) : GENBANK, EMBL et DDBJ (figure. 24). Dans ces banques de données, les séquences sont stockées sous forme de fichiers texte, accessibles par des systèmes d'interrogations : le système Entrez pour Genbank, SRS (Sequence Retrieval System) pour EMBL et GETENTRY pou DDBJ. Avant de publier un article qui décrit une séquence biologique, il est obligatoire de déposer cette séquence dans l'une de ces 3 principales bases de données. Un autre intérêt de ces bases réside dans l'information qui accompagne les séquences (annotations, expertise, bibliographie).

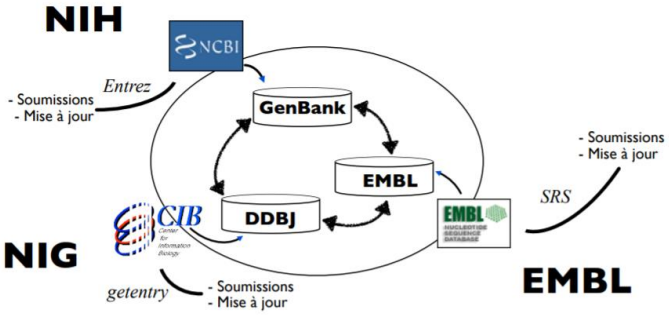


Figure. 24 interconnexions des trois bases de séquences nucléiques
il suffit de consulter une seule de ces banques afin d'accéder au contenu des trois banques
Les séquences sont automatiquement synchronisées entre les trois bases de données.

a) GenBank (Etats-Unis, http://www.ncbi.nlm.nih.gov/genbank/**)**

GenBank est la base de données de séquences génétiques de NIH (National Institutes of Health), créée en 1982 par la société IntelliGenetics et diffusée maintenant par le NCBI (National Center for Biotechnology Information).

b) EMBL (Europe, http://www.ebi.as.uk/embl/**)**

European Molecular Biology Library est une base de données de séquences nucléiques européenne créée en 1980 par l'European Molecular Biology Organisation et diffusée par EBI (European Bioinformatics Instituts, Cambridge UK)

c) DDBJ (Japon, https://www.ddbj.nig.ac.jp/)

La banque japonaise de données ADN Crée et diffusée par National Institute of Genetics (NIG), en 1986.

IV.2.2) Bases de séquences protéiques

Séquences obtenues de deux manières :

- Expérimentalement : séquençage de la protéine isolée à partir de la cellule (long et coûteux).
- in silico : Traduction de séquences d'ADN.

a) Uniprot (http://www.uniprot.org)

Uniprot (Universal protein resource), est le dépôt de données central des séquences et fonctions protéiques regroupant les données de plusieurs bases généralistes de séquences protéiques : SwissProt, TrEMBL et PIR.

b) PIR (Protein Information Resource)

PIR a été établi en 1984 par le National Biomedical Research Foundation afin d'assister les chercheurs dans l'identification et l'interprétation de leurs séquences protéiques, il est localisé à l'université de Georgetown aux États-Unis.

c) SwissProt

SwissProt est une banque de données biologiques de séquences protéiques créée en 1986 par l'institut suisse de bioinformatique. Elle fournit des séquences de protéines fiables, où chaque séquence est expertisée par un expert dans un domaine particulier de la biologie avec des annotations (comme la description de la fonction d'une protéine, ses structures, les maladies associées, etc.).

d) TrEMBL (Traduced EMBL)

TrEMBL est une base de données de séquences protéiques distribuée par l'EBI et traduite automatiquement de la base de séquences nucléiques EMBL.

IV.2.3) Bases de structures

a) PDB (http://www.rcsb.org/pdb/)

Protein Data Bank gérée par (RCSB, USA) contient des informations sur la structure tridimensionnelle des protéines. PDB est une grande base de données qui contient actuellement plus de 100 000 modèles 3D de nombreux types de macromolécules.

b) EBI-MSD (Macromolecular Structure Database at EBI, UK), (http://www.ebi.ac.uk/msd/).

c) NDB (Nucleic Acid structure Datatabase at Rutgers State University of New Jersey, USA), (http://ndbserver.rutgers.edu/NDB/ndb.html).

IV.2.4) Bases génomiques

a) Genome (https://www.ncbi.nlm.nih.gov/genome)

Base de génomes complets appartenant à plus de 307000 organisme (eucaryotes, procaryotes, virus, plasmides, organelles, …) (données de janvier 2020).

b) GOLD (Genomes OnLine Database) (https://gold.jgi.doe.gov/) : base de données qui recense les milliers de génomes séquencés ou en voie de séquençage.

IV.2.5) Bases fonctionnelles (Les bases de motifs protéiques)

a) PROSITE (http://www.expasy.org/prosite).

Elle se compose d'entrées de documentation décrivant les domaines protéiques, les familles et les sites fonctionnels ainsi que les modèles et profils associés pour les identifier. Cette ressource soutient la recherche sur COVID-19 / SARS-CoV-2. Développé par le groupe SwissProt et soutenu par le SIB (Swiss Institute of Bioinformatics).

b) InterPro (https://www.ebi.ac.uk/interpro/).

InterPro est un consortium : pour mieux classer les protéines, il utilise en effet les modèles, les profils et les signatures fournis par 14 bases de données membres (regroupées en une seule ressource) : CATH-Gene3D, SUPERFAMILY, Pfam, SMART, TIGRFAM, PIRSF, SFLD, PANTHER, HAMAP, Prosite, CDD, MobiDB, ProDom, PRINTS.

IV.2.6) Bases de réseaux (d'interactions, de signalisations et métaboliques)

a) DIP (Database of Interacting Proteins), (https://dip.doe-mbi.ucla.edu/dip/Main.cgi).

b) BIND (Biomolecular interaction network database)

c) MIPS (Mammalian protein-protein interaction database)

d) GRID (general repository for interaction datasets)

e) TRANSPATH une base de données de transduction de signaux et de voies métaboliques chez les mammifères. (http://genexplain.com/transpath/)

IV.2.7) Autres bases de données

- ***les bases de données dédiées aux maladies génétiques***

La base de données de référence pour les maladies génétiques est sans conteste **OMIM** (Online Mendelian Inheritance in Man). Cette base de données est née dans les années 1960 grâce au travail de Victor McKusick. OMIM donne de nombreuses informations sur la classification des maladies génétiques, des présentations cliniques et la cartographie génomique de la localisation de la maladie. Il existe d'autres bases de données dédiées aux maladies génétiques. Citons par exemple : **GeneCards** (Weizmann Institute of Science), **Office of Rare Diseases Research** (National Institute of Health).

- ***les bases bibliographiques***

Medline (Medical Litterature Analysis and Retrieval System Online) gérée par la NLM (National Library of Medicine) contient des millions de références de publications du domaine des sciences de la vie. C'est la base de données médicale la plus utilisée dans le monde. Medline est accessible via internet, gratuitement et sans inscription préalable, soit à partir de la page d'accueil de la National Library of Medicine (http://www.nlm.nih.gov), soit à partir de PubMed (http://www.ncbi.nlm.nih.gov/entrez), soit encore à partir de NLM Gateway (http://gateway.nlm.nih.gov). PubMed est une base bibliographique développée par leNational Center for Biotechnology Information (NCBI) de la National Library of Medicine, accessible via internet (http://www.ncbi.nlm.nih.gov/entrez)

A côté de ces bases de données généralistes, de très nombreuses bases de données spécialisées se sont développées, dédiées à l'étude d'une espèce, d'un organisme ou d'une thématique particulière. Telles que **GenoList** à l'Institut Pasteur pour les bactéries E. coli, **Cyanobase** pour la cyanobactérie Synechosystis, ou **TAIR** pour la plante A. thaliana. **FlyBase** pour la drosophile D. melanogaster, **MGD** pour la souris ou encore **GDB** pour l'homme. D'autres bases sont thématiquement spécialisées, par exemple, la base **EPD** (eukaryotic promoter database) rassemble les séquences de promoteurs

eucaryotes, et les bases **INTERPRO** et **eMOTIF** décrivent des motifs et des profils de familles de protéines.

IV.3) Format de fichier

Les différents types de données biologiques sont stockés dans des fichiers avec un format spécifique. Un format de fichier est un moyen standard de coder les informations pour le stockage dans un fichier informatique. L'objectif est de manipuler facilement des séquences dans les bases de données, à l'aide d'un format universel, compatibles avec les traitements de texte. De nombreux formats de fichiers en bioinformatique sont basés sur du texte. Les formats les plus courants sont :

a) Format FASTA

Le format FASTA est relativement souple. Une entrée commence par par le caractère ">", suivi d'un chapeau optionnel. Le chapeau est composé typiquement de l'identifiant de la séquence et d'informations complémentaires optionnelles. La séquence nucléotidique/protéique commence à la ligne suivante et peut couvrir plusieurs lignes continues. Cela permet de mettre plusieurs séquences dans un même fichier.

Exemple :

```
>em|U03177|FL03177 Feline leukemia virus clone FeLV-69TTU3-16.
AGATACAAGGAAGTTAGAGGCTAAAACAGGATATCTGTGGTTAAGCACCTG
TGAGGCCAAGAACAGTTAAACCCCGGATATAGCTGAAACAGCAGAAGTTTC
GCCAGCAGTCTCCAGGCTCCCCA
```

b) Format staden

Le plus ancien et le plus simple : suite des lettres de la séquence par lignes terminées par un retour-à-la-ligne (80 caractères max/ligne). Ce format n'autorise qu'une séquence par fichier.

Exemple **:**

```
SESLRIIFAGTPDFAARHLDALLSSGHNVVGVFTQPDRPAGRGKKLMPSPVKVLAEEKGL
PVFQPVSLRPQENQQLVAELQADVMVVVAYGLILPKAVLEMPRLGCINVHGSLLPRWRGA
APIQRSLWAGDAETGVTIMQMDVGLDTGDMLYKLSCPITAEDTSGTLYDKLAELGPQGLI
TTLKQLADGTAKPEVQDETLVTYAEKLSKEEARIDWSLSAAQLERCIRAFNPWPMSWLEI
EGQPVKVWKASVIDTATNAAPGTILEANKQGIQVATGDGILNLLSLQPAGKKAMSAQDLL
NSRREWFVPGNRLV
```

IV.4) Annotations

La masse d'informations considérables mise à la disposition des biologistes doit être soutenue par des annotations : commentaire du biologiste sur la structure, la fonction et toutes autres informations utiles sur la séquence. L'annotation d'un génome, d'un transcriptome d'un protéome, d'un métabolome ... consiste à documenter de la manière la plus exhaustive les informations issues de ces disciplines. L'annotation est un processus complexe qui peut être subdivisé en trois catégories : l'annotation syntaxique, l'annotation fonctionnelle et l'annotation relationnelle (Stein, 2001) :

- L'annotation structurale ou syntaxique qui permet d'identifier les séquences présentant une pertinence biologique (gènes, signaux, répétitions, …)
- L'annotation fonctionnelle qui permet de prédire les fonctions et produits potentiels des gènes préalablement identifiés (similitudes de séquences, motifs, structures, ...).
- L'annotation relationnelle qui permet enfin de déterminer les interactions que les objets biologiques préalablement identifiés sont susceptibles d'entretenir (familles de gènes, réseaux de régulation, réseaux métaboliques, …).

a. L'annotation automatique s'appuie (essentiellement) sur des comparaisons des séquences à annoter avec les séquences présentes dans les banques de données. Les algorithmes recherchent des

similarités / homologies de séquence, de structure, de motifs, … Ils permettent de prédire la fonction d'une molécule et de transférer automatiquement l'annotation entre les molécules homologues.

b. L'annotation manuelle par des experts qui valident ou invalident la prédiction en fonction de leurs connaissances ou de résultats expérimentaux. L'annotation manuelle est donc tout à fait indispensable. Mais, vue la quantité "astronomique" de données acquises quotidiennement, il est illusoire d'envisager une curation manuelle de l'ensemble des données en temps réel.

V) Outils d'analyse bioinformatique

En raison de la multitude et de la diversité des bases de données disponibles en ligne, les outils bioinformatique disponibles sont également très nombreux, allant de l'assemblage des fragments et de la prédiction de gènes à partir d'une séquence quelconque à l'identification de motifs particuliers (sites de fixation de protéines, etc.), en passant par la traduction automatique des séquences et la prédiction fonctionnelle. Les outils de bioinformatique peuvent se trouver en accès libre via internet (soit à utiliser en ligne, soit à télécharger).

V.1) Assemblage des fragments

La technologie de séquençage ne permet pas de traiter la totalité de la molécule d'ADN à la fois : le génome est donc découpé, au préalable, en fragments qui se chevauchent partiellement. Il faudra ensuite assembler les séquences de ces fragments pour obtenir la séquence du génome dans sa totalité (la molécule d'ADN). L'assemblage consiste à fusionner des fragments d'ADN ou d'ARN issus d'une plus longue séquence afin de reconstruire la séquence originale. Il s'agit d'une étape d'analyse in silico qui succède au séquençage de l'ADN ou de l'ARN d'un organisme. Le nombre d'essais qu'il est nécessaire d'effectuer avant de trouver le véritable emplacement de chaque fragment le long de la macromolécule d'ADN est considérable. Lorsque le génome séquencé est petit (moins de 100 fragments), l'assemblage pouvait être effectué manuellement. Pour les génomes de grande taille, les méthodes automatiques deviennent incontournables, sans lesquelles la reconstitution d'un génome complet ne serait pas accessible dans un délai raisonnable.

- **ABySS** (Jackman et al., 2017) (https://www.bcgsc.ca/resources/software/abyss) est un assembleur de séquences, parallèle et à paires, conçu pour les lectures courtes. La version monoprocesseur est utile pour assembler des génomes d'une taille allant jusqu'à 100 Mbases. La version parallèle est implémentée en utilisant MPI et est capable d'assembler des génomes plus grands.

V.2) Prédiction des gènes

Une fois la séquence d'un génome complet obtenue, débute la phase d'annotation. L'annotation elle-même consiste tout d'abord à rechercher la position des gènes sur cette séquence. La prédiction de gènes à partir des séquences consiste à identifier les zones de l'ADN qui correspondent à des gènes (les introns). Soit par leur similitude avec des gènes déjà connus (méthodes par comparaison), soit par une prédiction en fonction de la séquence (méthodes ab initio) (Mathé et al., 2002).

- **Méthodes utilisées**

• Prédictions ab initio ou de novo (méthodes intrinsèques) qui s'appuient sur des techniques informatiques d'apprentissage automatique utilisant : des modèles de Markov interpolés (exemples d'outils : **Glimmer1.0** (Salzberg et al. 1998), **EasyGene**, **GeneMark**).

• Prédictions basées sur la similarité (méthodes extrinsèques), aussi appelées méthodes par homologie, consistent à comparer la séquence étudiée avec des séquences connues, rassemblées dans les bases de données. Il existe des méthodes extrinsèques qui reposent sur la comparaison des ORF

avec les séquences présentes dans les banques de données (exemples d'outils : **Orpheus**, **Critica**, **Reganor**, ...).

V.3) prédiction des fonctions des gènes/protéines

Une fois que les gènes sont repérés et délimités sur la séquence, il convient d'affecter les fonctions aux protéines codées par les gènes. Les gènes identifiés sont alors traduits en séquences protéiques et la mise en œuvre de méthodes fondées sur la recherche de similarités avec les protéines répertoriées dans les banques de séquences permet de caractériser la fonction biologique.

On procède donc par similarité, en comparant la séquence de la protéine hypothétique avec des protéines homologues de fonctions connues. Les deux logiciels les plus utilisés par les biologistes permettant de repérer les séquences de la banque susceptibles d'avoir des ressemblances biologiques avec la séquence requête sont : **FASTA** (FAST Alignment) (Pearson & Lipman, 1988) et **BLAST** (Basic Local Alignment Search Tool) (Altschul et al., 1997). Et enfin, d'opérer un transfert par similarité, de la fonction biologique présumée.

V.4) Identification et caractérisation des protéines

Le site le plus populaire et le plus complet dédié à l'analyse protéomique reste cependant le site de l'ExPASy (Expert Protein Analysis System, http://www.expasy.org), maintenue par l'Institut Suisse de Bioinformatique.

- **AACompIdent** : (http://us.expasy.org/tools/aacomp/) est un outil qui permet l'identification d'une protéine à partir de sa composition en acides aminés. Il recherche dans les bases de données Swiss-Prot et / ou TrEMBL des protéines dont la composition en acides aminés est la plus proche de la composition en acides aminés donnée.

V.5) Analyse des protéines

AnTheProt pour Windows (http://antheprotpbil.ibcp.fr) est un programme de graphisme moléculaire destiné à la visualisation de protéines, acides nucléiques issus des archives RCSB. Le programme vise à afficher, à enseigner et à générer des images de qualité de publication. Permet de réaliser des images des structures 3D. Il peut toujours être associé à la plateforme générale ANTHEPROT d'analyse de séquences protéiques.

VI) Conclusion

Le nombre de données dans le domaine de la biologie ne cesse d'augmenter en particulier avec le séquençage des génomes de différents organismes. Cette explosion de données impose deux grands défis majeurs : l'acquisition, le stockage, la gestion et la récupération de ces données d'une part et l'analyse biologiques in silico de ces données d'autre part. Le premier défi a été rendu possible grâce à différentes bases de données, accessibles en lignes (GENBANK, EMBL, DDBJ, Uniprot, TrEMBL, …). Le deuxième défi est rendu possible grâce au développement des outils d'analyse comme (BLAST, Glimmer, …) produisant de nouvelles connaissances et prédisant de nouvelles propriétés biologiques. Actuellement la plupart des séquences protéiques ne sont pas obtenues expérimentalement mais à partir d'analyse in silico des données de séquences nucléiques. Les hypothèses sur les fonctions et le rôle des gènes sont de plus en plus issues de l'analyse in silico.

Le séquençage est le plus grand moyen d'acquisition de données biologique. La technologie de séquençage ne permet pas de traiter la totalité de la molécule d'ADN à la fois. Le premier problème qui se pose consiste à assembler les fragments pour obtenir la séquence du génome dans sa totalité, et de repérer ensuite les gènes dans le texte brut du génome constitué par l'enchaînement des nucléotides. Une fois que les gènes sont repérés et délimités sur la séquence, il convient d'affecter une fonction à la protéine correspondante. Ces problèmes sont traités par des outils algorithmiques sophistiqués, dont le développement impose des défis majeurs.

L'un des plus importants défis est de prédire et d'annoter les fonctions de la plupart des produits de gènes. Une première phase dans l'annotation d'une séquence consistant à identifier les gènes de l'organisme (trouver leur localisation précise sur la séquence du génome). Dans une seconde étape, on cherche ensuite à assigner une ou plusieurs fonctions biologiques à chacun de ces gènes hypothétiques. Cette seconde étape est généralement conduite par comparaison des séquences des gènes hypothétiques avec les séquences de gènes de fonction déjà connue. Les gènes identifiés sont alors traduits en séquences protéiques et la mise en œuvre de méthodes fondées sur la recherche de similarités avec les protéines répertoriées dans les banques de séquences permet de caractériser la fonction biologique. On peut s'intéresse ensuite, aux relations qui lient ces objets biologiques : caractérisation des réseaux de régulation et des voies métaboliques.

I) Introduction

La bioinformatique permet de résoudre différents problèmes posés par la biologie. Ceci inclue, l'assemblage du génome, la découverte de gènes, la prédiction de la structure et la fonction des protéines, la prédiction de l'expression des gènes et des interactions protéine-protéine, etc. Ces problèmes biologiques sont généralement complexes, nécessitant l'intervention des méthodes sophistiquées. La résolution de ces problèmes permet de réaliser plusieurs applications dans des domaines variés (l'agriculture, la pharmacologie, la médecine, la virologie, etc.) par exemples : la conception de médicaments, l'identification des candidats gènes et simples nucléotides polymorphismes (SNP) est faite dans le but de mieux comprendre la base génétique de la maladie.

La relation entre les trois types de séquences (ADN, ARN et Protéines) est au cœur de la théorie de la biologie moléculaire (L'ADN est d'abord transcrit en ARNm qui sera traduit en protéine). Si cette relation constitue le « dogme central » de la micro biologie, alors le « dogme central » de la bioinformatique est la déduction par homologie, car la grande partie de l'analyse in silico se fait par comparaison, et que l'un des moyens les plus utilisés pour la comparaison de séquences est l'alignement. Par exemple la fonction des gènes peut être déduite par comparaison avec les gènes homologues connus. La comparaison de gènes homologues est une approche très efficace pour déterminer la fonction et la structure d'une séquence, étudier les processus de l'évolution à l'échelle moléculaire et établir la phylogénie des espèces.

La plupart des problèmes en bioinformatique ont été démontrés NP-difficile. L'enjeu est double, on doit trouver des solutions efficaces à ces problèmes et biologiquement acceptables. Dans ce chapitre, nous allons présenter les principaux problèmes posés en bioinformatique

II) Domaines de la bioinformatique

Les deux principaux domaines de la bioinformatique sont les bases de données et l'analyse de données. Plusieurs branches de l'analyse en bioinformatique se sont constituées avec de nombreux problèmes ouverts et qui peuvent être classifiées selon :

- le type d'objet étudié (génome, transcriptome, protéome...),
- l'échelle des objets étudiés (atomique, moléculaire, cellulaire, organisme, population),
- les méthodes (alignement de séquences, classification, modélisation moléculaire, phylogénie...),
- le concept étudié (séquence, structure, réseau, expression, fonction).

II.1) La bioinformatique des séquences

Cette branche s'intéresse en particulier à l'analyse de séquences qui peut aller de l'identification de gènes ou de régions biologiquement pertinentes dans l'ADN ou dans les protéines aux comparaisons de séquences en passant par la prédiction de motifs ou l'établissement de signatures.

La recherche de similitude entre séquences par comparaison est un élément fondamental qui constitue souvent la première étape des analyses de séquences. Il s'agit de déterminer dans quelle mesure des séquences, génomiques ou protéiques se ressemblent. L'objectif est de révéler des régions proches dans leur séquence en se basant sur le principe de parcimonie, c'est-à-dire en considérant le minimum de changements en insertion, délétion, ou substitution qui séparent les séquences. On peut déduire ainsi, des informations importantes sur la structure et la fonction des protéines ou l'évolution des espèces à l'échelle moléculaire (la phylogénie).

II.2) La bioinformatique structurale

Traite de la reconstruction, de la prédiction, de la modélisation et de l'analyse des structures 2D et 3D des macromolécules biologiques (protéines, acides nucléiques). Comprendre les maladies

au niveau moléculaire, afin de concevoir des médicaments mieux ciblés est l'une des principales retombées attendues de l'analyse de la structure tridimensionnelle des molécules biologiques.

II.3) La bioinformatique des réseaux

S'intéresse aux interactions entre gènes, protéines, cellules, organismes, en essayant d'analyser et de modéliser les comportements collectifs d'ensembles de briques élémentaires du vivant. Les interactions complexes entre les gènes et leurs produits (ARN et protéines) régissant l'activité de la cellule afin de lui permettre de s'adapter en permanence aux variations de son environnement. L'ensemble des interactions entre entités biologiques, qui sont au cœur de la biologie des systèmes, est appelé interactome.

III) Problèmes fondamentaux de la bioinformatique

En 1972, Anfinsen (Anfinsen, 1972) formule l'hypothèse que la structure tertiaire de la protéine est entièrement déterminée par la séquence des acides aminés qui la compose. Le fait que la séquence détermine la structure qui elle-même détermine la fonction de la protéine est appelé le paradigme séquence-structure-fonction. Un moyen de déterminer la fonction d'une protéine consiste alors à comparer sa structure ou sa séquence à des protéines de fonction connue (Sleator & Walsh, 2010) : La fonction aura d'autant plus de probabilité d'être identique que la séquence ou la structure sera similaire. On considère que deux séquence proches ont un ancêtre commun récent et partagent donc, une similarité au niveau de la fonction biologique. On peut ainsi chercher à aligner les séquences génétiques de l'homme et de la souris, pour pouvoir ensuite faire des expériences de génétique chez la souris, et en tirer des conclusions chez l'homme.

Parmi les problèmes fondamentaux de la bioinformatique on trouve : l'alignement des séquences, la prédiction de structure et fonction, la phylogénie moléculaire, la détection de gènes, l'analyse de l'expression des gènes et des réseaux de régulation, etc. Le problème qui a connu le plus grand développement est sans doute l'alignement de séquences.

III.1) Problème d'alignement

Un alignement permet de comparer des séquences biologiques (ADN, ARN ou protéines), afin d'identifier les régions similaires entre eux. L'opération d'alignement consiste à représenter les séquences les une sous les autres, avec une possible insertion d'espaces (Gaps) entre les résidus pour que des caractères identiques ou similaires soient apparier. Trois situations sont possibles pour une position donnée de l'alignement : les caractères sont les mêmes (Identité), les caractères ne sont pas les mêmes (substitution) ou l'une des positions est un espace (Insertion/Délétion). L'objectif est de maximiser le nombre de caractères en commun. Les régions similaires contenant des nucléotides ou des acides aminés conservés peuvent indiquer : une fonction biologique proche, une structure tridimensionnelle semblable, une origine et/ou une histoire d'évolution commune entre les séquences.

D'un point de vue biologique, Les gènes sont normalement transmis d'une génération à une autre sans aucun changement. Cependant, des mutations ont lieu parfois induisant des formes altérées provoquant le changement d'une séquence :

- il arrive qu'une base soit remplacée par une autre (substitution),
- il arrive que des morceaux de séquence disparaissent (délétions), matérialisés par des Gaps
- il arrive que de nouveaux morceaux de séquence soient introduits (insertion).

Ce sont ces phénomènes que les algorithmes d'alignement essayent de modéliser.

III.1.1) Similarité et homologie (aspect de comparaison)

- **L'homologie** implique que les séquences dérivent d'une séquence ancestrale commune et qu'elles ont une même histoire évolutive (fonctions conservées par exemple). deux séquences sont homologues ou elles ne le sont pas. Si A est homologue à B et B homologue à C, alors A est homologue à C (même si A et C se ressemblent très peu). Des séquences homologues dans une même espèce sont dites paralogues, des séquences homologues dans deux espèces différentes sont dites orthologues.
- **La similarité** mesure la ressemblance entre deux ou plusieurs séquences. C'est une quantité qui se mesure en % d'identité (ressemblance exacte).

Taux de similitude = pourcentage d'identité + pourcentage de substitutions conservatives.

Un fort taux de similarité de séquence est une indication forte de l'existence d'une homologie, mais ce n'est pas une preuve. Une faible similarité de séquences (séquences éloignées) ne veut pas dire non-homologie. L'estimation d'une bonne similarité va permettre d'émettre une hypothèse d'homologie à tester. Un alignement multiple de séquences et une analyse phylogénétique sont nécessaires pour établir l'homologie. La comparaison de séquences nécessite donc la mise en œuvre de procédures de calcul et de modèles biologiques permettant de quantifier la notion de ressemblance entre ces séquences.

III.1.2) Les différents types d'alignements

Un alignement de séquences peut être global ou local, réalisé sur deux séquences ou plus.

- **L'alignement global** vise à aligner les séquences données sur toute leur longueur. En revanche, **l'alignement local** tente d'identifier les sous-régions des séquences dans lesquelles leurs configurations coïncident (les zones de forte homologie).
- Un **alignement par paire** consiste à aligner deux séquences biologiques, et qui peut être local ou global. Cependant, dans les cas complexes, la quantité d'information contenue dans deux séquences n'est pas suffisante, et il devient nécessaire d'étendre la comparaison à plusieurs séquences. C'est là l'objet des **alignements de séquences multiples (MSA)**. Il est plus informatif que les alignements de deux séquences.
- **Alignement d'une séquence avec les séquences des banques de données** (global ou local)

  L'alignement avec les banques consiste à comparer une séquence inconnue avec une base de données de séquences connues pour tenter d'en tirer de l'information à partir des séquences similaires trouvées. Les deux logiciels les plus utilisés par les biologistes permettant de repérer les séquences de la banque susceptibles d'avoir des ressemblances biologiques avec la séquence requête sont : FASTA (FAST Alignment) (Pearson & Lipman, 1988) et BLAST (Basic Local Alignment Search Tool) (Altschul et al., 1990, 1997).

L'intérêt de l'alignement global est de révéler les événements évolutifs sur l'ensemble de la longueur des séquences d'intérêt. On recourt par exemple aux alignements globaux quand on veut étudier l'évolution d'une famille de protéines dans son ensemble (l'analyse phylogénétique). Les alignements locaux révèlent les segments conservés entre deux ou plusieurs séquences. On les utilise par exemple pour extraire un domaine conservé à partir d'une famille de séquences homologues (la découverte et la recherche de motifs).

III.1.3) Approches d'alignement

Les séquences très courtes peuvent être alignées à la main (alignement manuel). Cependant les problèmes rencontrés en biologie nécessitent l'alignement de séquences longues qui ne peuvent

être alignées à la main, d'où la nécessité de méthodes algorithmiques (alignement automatique). Deux approches d'alignement automatique existent : l'alignement par score et l'alignement statistique.

L'approche par score cherche à optimiser un score d'alignement. Ce score tient compte d'une part du nombre de résidus identiques entre les séquences alignées et du nombre de résidus similaires sur le plan physico-chimique (par exemple, dans l'alignement de séquences protéiques les deux acides aminés Lysine (K) et Arginine (R) sont très proches). Et d'autre part du nombre de mutation (insertion, délétion et substitution). Il s'agit de quantifier la similitude entre des séquences (d'ADN, de protéines) alignées, en attribuant un certain nombre de points à chaque alignement et on sélectionne l'alignement (ou les alignements) de score le plus élevé.

L'approche statistique pour l'alignement de deux séquences est réalisé par maximisation d'un critère de vraisemblance, dans un contexte de paires de séquences Markov caché. Les modèles d´évolution qui permettent cette approche sont ceux introduits par (Thorne et al., 1991, 1992), ou encore des variantes (Miklos et al., 2004). *Dans cette thèse on s'intéresse à l'approche par score, qui regroupe trois méthodes principales: les méthodes exactes, les méthodes progressives et les méthodes de raffinement itératives.* Une telle démarche consiste à :

- Choisir les séquences à aligner
- Choisir un système de score reflétant le degré de similarité
- Choisir un système de gaps (insertions, délétions)
- Choisir une méthode d'alignement
- Evaluer la qualité de l'alignement (validation)

III.1.3.1) Le choix du matériel à comparer (ADN ou protéine)

Une des questions qui se posent au biologiste lorsqu'il compare des séquences est de savoir sur quel matériel il doit travailler : ADN ou protéine ? La pauvreté de l'alphabet de l'ADN (4 lettres contre 20 pour les protéines) : la probabilité de bon appariement est donc beaucoup plus importante lors de comparaisons ADN/ADN. Tandis que pour deux Acide aminés, cette comparaison est plus fine car elle peut être basée sur des critères physico-chimiques ou sur des taux de mutations naturels. Un acide aminé peut être remplace par un autre sans que la structure ou la fonctionnalité d'une protéine soit grandement altérée. En comparant des séquences d'ADN, on rencontre plus de similitudes dues au hasard (25%) par rapport aux séquences protéiques (5%). Dès que c'est possible, il est préférable de comparer les séquences au niveau protéique.

III.1.3.2) Les systèmes de scores

Pour quantifier la similitude entre séquences, un score est calculé. Celui-ci peut mesurer soit le rapprochement (similarité), soit l'éloignement (la distance) des séquences. Ce score repose sur un système qui permet d'attribuer un coût aux opérations élémentaires (identité, substitution, insertion et délétion). Pour chaque identité l'alignement est récompensé et pour chaque opération de mutation (insertion/délétion (indel) ou substitution) l'alignement est pénalisé. La similarité de deux séquences peut ainsi être calculée en sommant les scores élémentaires de chaque position. Pour l'alignement multiple de séquences la similarité peut être calculée en sommant les similarités des séquences prises deux à deux (sum of pairs (Carrillo & Lipman, 1988). Dès lors, rechercher le(s) meilleur(s) alignement(s) revient à rechercher le(s) alignement(s) réalisant le meilleur score, indiquant la plus grande similarité possible. L'objectif est donc d'obtenir l'alignement de score optimum qui soit l'alignement le plus biologiquement significatif.

Le score élémentaire entre deux (acides aminés ou nucléotides) donnés est un élément d'une matrice de similarité qui rend compte de tous les états possibles en fonction de l'alphabet utilisé dans la description des séquences. Il existe plusieurs de ces matrices avec des modes de construction

différents. Ces matrices sont en général complétées par des fonctions de score pour quantifier l'introduction des indels dans les alignements.

a) Systèmes de scores pour l'ADN

Les substitutions d'acides nucléiques pénalisant toutes les mutations de la même manière (table. 3) n'est pas très significatif du point de vue biologique.

	A	C	G	T
A	**1**	0	0	0
C	0	**1**	0	0
G	0	0	**1**	0
T	0	0	0	**1**

Table. 3 La matrice identité (identité = +1, substitution = 0, gap = -1)

En tenant compte de la proximité des propriétés physico-chimiques des nucléotides : les transitions (AG et CT) se produisent plus fréquemment, et devraient donc être moins pénalisées que les transversions (AC, GT, CG et AT) (figure. 25).

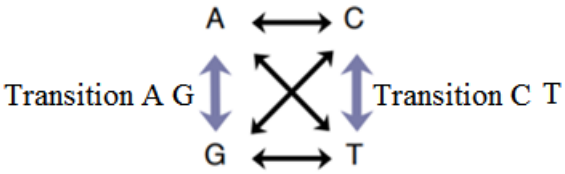


Figure. 25 Modèle d'évolution P (transition) > P(transversion)

Plus généralement, on considère une matrice de scores qui attribue le score élémentaire s (x, y) à l'alignement de la lettre x en face de la lettre y avec gap $s(x, -) = -1$, (table. 4).

	A	C	G	T
A	**1**	0	**0,5**	0
C	0	**1**	0	**0,5**
G	0,5	0	**1**	0
T	0	**0,5**	0	**1**

Table. 4 Matrice de similarité transition-transversion

b) Systèmes de scores pour les protéines

Au niveau protéique, des acides aminés peuvent avoir des propretés physico-chimiques proches. Par conséquence, certaines substitutions peuvent être conservatives, c'est-à-dire un acide aminé peut être remplace par un autre sans que la structure ou la fonctionnalité d'une protéine soit grandement altérée. En tenant compte de cette réalité biologique on peut bâtir des systèmes de scores améliorant la fiabilité des recherches de similitudes protéiques. Dans ce cas, le score de chaque paire d'acides aminés mis en correspondance est extrait d'une matrice de substitution, basée sur la nature physico-chimique des acides aminés ou déduite empiriquement de l'évolution moléculaire. Les matrices de scores qui en découlent permettront d'augmenter la fiabilité des recherches de similitudes protéiques.

- **Matrices physico-chimiques**

Matrices basées sur des propriétés comme (volume, conformation, charge polaire ou non polaire, …). Elles sont issues des protéines ayant des homologies structurelles. Elles sont utilisées pour rechercher la similarité structurelle.

- **Matrices empiriques déduites de l'évolution moléculaire**

Matrices construites à partir de substitutions observées entre acides aminés intervenues au cours de l'évolution, déterminées à partir d'alignements multiples de protéines ayant des homologies

fonctionnelles. Elles sont utilisées pour rechercher la similarité fonctionnelle. Deux grandes familles de matrices de substitutions sont les plus utilisées : Les matrices PAM et BLOSUM.

- **Matrices PAM**x (Percent Accepted Mutation) (Dayhoff et al., 1978) Les PAMx sont Basées sur l'alignement de protéines conservées à plus de 85%. Elles sont Calculées par la probabilité d'observer la mutation $a_1 \rightarrow a_2$ après un temps évolutif donné. Plus la valeur est négative, plus la probabilité est faible, plus le remplacement est rare. Une matrice PAMx est valable pour une certaine distance évolutive x mesurée en PAM = nombre de mutations ponctuelles par 100 acides aminés.

  PAMx : x mutations acceptées pour 100 sites entre les séquences qui ont servi à construire la matrice. Exemple : PAM40, PAM120 et PAM250, …

- **Matrices BLOSUM**x (Blocks Substitutions Matrices) (Henikoff & henikoff, 1993)

  Matrices basées sur des comparaisons par paires utilisant des alignements locaux. Elles créées à partir de domaines comprenant des séquences plus ou moins divergentes (Utilisation près de 2000 domaines conservés provenant de 500 familles de protéines). Matrices plus adaptées pour des protéines distantes du point de vue évolutif. Toutes les paires ayant servi à construire une matrice BLOSUMx ont une identité ≥ à x %.

  BLOSUMx : matrice obtenue à partir de séquences présentant au minimum x% d'identité entre elles. Exemple : BLOSUM45, BLOSUM62 et BLOSUM80, …

D'autres matrices existent dans la littérature comme les matrices basées sur des arbres construits en utilisant le maximum de parcimonie : JTT (Jones et al., 1992). Et les matrices basées sur des arbres construits en utilisant le maximum de vraisemblance : WAG (Whelan & Goldman, 2001). Matrices protéiques liées aux propriétés physico-chimiques Matrice d'hydrophobicité de Levitt (1976) Matrice de structures secondaires de Levin et al. (1986) Matrices utilisant des superpositions de structures 3D (ex: matrices de Johnson et Overington, 1993). Mais Les matrices PAM et BLOSSUM sont les plus utilisées.

- **Le choix d'une matrice protéique**

Le choix d'une matrice de substitution gouverne le système de score donc influence les résultats obtenus. Il est souvent difficile de savoir laquelle doit être utilisée dans les différents programmes de comparaison de séquences protéiques. Les premières études comparatives sur l'utilisation de différentes matrices (Taylor, 1986), (Argos, 1987), (Risler et al., 1988) montraient déjà qu'il n'existe pas de matrice idéale. Ainsi, dans une étude sur les matrices de type PAM, (Altschul, 1991) conseille pour les méthodes d'alignements locaux la matrice PAM40 pour retrouver des alignements courts avec des protéines très semblables et les matrices PAM120 et PAM250 pour des alignements plus longs et de plus faible ressemblance. Il préconise également l'utilisation de la PAM120 lorsque l'on ne connaît pas a priori le degré de ressemblance de deux séquences comme c'est le cas par exemple dans les programmes de recherche de similitudes avec les banques de données.
(Henikoff & Henikoff, 1993) ont évalué plusieurs matrices en utilisant le programme BLAST de recherche de similitude sans insertion-délétion. Leur étude a établi que les matrices dérivées directement des comparaisons de séquences ou des comparaisons de structure sont supérieures à celles qui sont extrapolées du modèle d'évolution de Dayhoff. En particulier ils concluent que la matrice BLOSUM62 permet d'obtenir les meilleurs résultats. En conclusion, il apparaît tout de même que pour des séquences similaires et courtes, il est préférable d'utiliser une matrice BLOSUM élevée ou PAM faible (exemple 40). Pour des séquences divergentes et longues, il est préférable d'utiliser une matrice BLOSUM faible ou PAM élevée. Enfin, pour la comparaison d'une séquence donnée à un ensemble de séquences dans une banque de données, il semble que la matrice BLOSUM62 soit un bon point de départ. Pour tous les logiciels qui utilisent l'alignement de séquences, la matrice

BLOSUM62 est souvent la matrice par défaut. Matrice PAM250 optimale pour distinguer des protéines apparentées de celles dont les similarités sont dues au hasard (matrice de mutation standard de Dayhoff).

La relation BLOSUM / PAM : BLOSUM45 ⟷ PAM250 et BLOSUM60 ⟷ PAM160

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

Table. 5 Matrice de substitution BLOSUM62

	C	G	P	S	A	T	D	E	N	Q	H	K	R	V	M	I	L	F	Y	W
C	12																			
G	-3	5																		
P	-3	-1	6																	
S	0	1	1	1																
A	-2	1	1	1	2															
T	-2	0	0	1	1	3														
D	-5	1	-1	0	0	0	4													
E	-5	0	-1	0	0	0	3	4												
N	-4	0	-1	1	0	0	2	1	2											
Q	-5	-1	0	-1	0	-1	2	2	1	4										
H	-3	-2	0	-1	-1	-1	1	1	2	3	6									
K	-5	-2	-1	0	-1	0	0	0	1	1	0	5								
R	-4	-3	0	0	-2	-1	-1	-1	0	1	2	3	6							
V	-2	-1	-1	-1	0	0	-2	-2	-2	-2	-2	-2	-2	4						
M	-5	-3	-2	-2	-1	-1	-3	-2	0	-1	-2	0	0	2	6					
I	-2	-3	-2	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	4	2	5				
L	-6	-4	-3	-3	-2	-2	-4	-3	-3	-2	-2	-3	-3	2	4	2	6			
F	-4	-5	-5	-3	-4	-3	-6	-5	-4	-5	-2	-5	-4	-1	0	1	2	9		
Y	0	-5	-5	-3	-3	-3	-4	-4	-2	-4	0	-4	-5	-2	-2	-1	-1	7	10	
W	-8	-7	-6	-2	-6	-5	-7	-7	-4	-5	-3	-3	2	-6	-4	-5	-2	0	0	17

Table. 6 Matrice de substitution PAM250

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	X
C	12																				
S	0	2																			
T	0	2	2																		
P	-3	0	0	8																	
A	0	1	1	0	2																
G	-2	0	-1	-2	0	7															
N	-2	1	0	-1	0	0	4														
D	-3	0	0	-1	0	0	2	5													
E	-3	0	0	0	0	-1	1	3	4												
Q	-2	0	0	0	0	-1	1	1	2	3											
H	-1	0	0	-1	-1	-1	1	0	0	1	6										
R	-2	0	0	-1	-1	-1	0	0	0	2	1	5									
K	-3	0	0	-1	0	-1	1	0	1	2	1	3	3	-1							
M	-1	-1	-1	-2	-1	-4	-2	-3	-2	-1	-1	-2	-1	4							
I	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-2	-2	-2	2	4						
L	-2	-2	-1	-2	-1	-4	-3	-4	-3	-2	-2	-2	-2	3	3	4					
V	0	-1	0	-2	0	-3	-2	-3	-2	-2	-2	-2	-2	2	3	2	3				
F	-1	-3	-2	-4	-2	-5	-3	-4	-4	-3	0	-3	-3	2	1	2	0	7			
Y	0	-2	-2	-3	-2	-4	-1	-3	-3	-2	2	-2	-2	0	-1	0	-1	5	8		
W	-1	-3	-4	-5	-4	-4	-4	-5	-4	-3	-1	-2	-4	-1	-2	-1	-3	4	4	14	
X	-3	0	0	-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	-4	-1

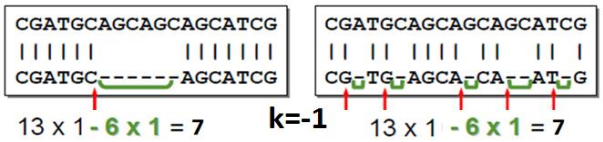
Table. 7 Matrice de substitution Gonnet générée à partir de Swiss-Prot

III.1.3.3) Pénalité des gaps (insertions, délétions ou indel)

L'opération indel présente un coût qu'il faut pouvoir pondérer au plus proche de la réalité biologique. Différents systèmes de mesure ont été proposés :

- **Pénalité fixe**

Attribuer la même pénalité pour chaque position contenant un gap (pénalité = k). La pénalité totale calculée est N*k N : le nombre de gap dans l'alignement.

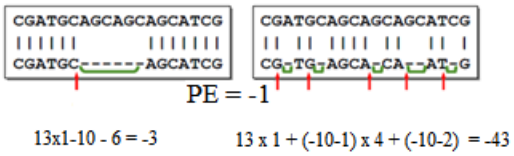


Une pénalité, toujours la même, pour chaque position contenant un gap n'est pas très réaliste. Inspiré des mécanismes évolutifs, la présence d'un gap est plus importante que sa longueur relative. On peut donc séparer cette pénalité en deux, d'une part une pénalité de création ou d'ouverture de gap, d'autre part une pénalité d'extension de gap. La pénalité d'ouverture est beaucoup plus grande que l'extension. Ceci est tout à fait en concordance avec les événements biologiques observés car il peut se produire par exemple une seule délétion de plusieurs bases plutôt que plusieurs pertes indépendantes d'une seule base.

- **Pénalité variable en fonction de la longueur du gap**

Le coût global du gap de longueur L est donné par P = PO + L ∗ PE tel que PO est la pénalité fixe d'insertion (ouverture) indépendante de la longueur et PE est la pénalité fixe d'extension (souvent PO = 10 ∗ PE).

Cette pénalité affinée permet de favoriser un large gap plutôt que de nombreux petits, afin de délimiter des zones homologue plus grandes.



III.1.3.4) Fonctions objectifs

Un système de score est le coût à attribuer aux opérations élémentaires d'alignement (identité, substitution, délétion et insertion). Le score de l'alignement est la somme de toutes les positions calculé par une fonction objectif. Dans le cas de l'alignement par paire, le calcul du score est facile. C'est la somme des scores élémentaires de chaque position entre les deux séquences. Cependant, dans le cas d'alignement multiple, le calcul est plus complexe. Un nombre important de fonctions ont été proposées au cours de ces dernières années. Globalement, elles peuvent être divisées en deux groupes : les fonctions basées sur un score d'alignement par paires et les fonctions basées sur des mesures de conservation de colonne.

- **La somme des paires SP (Sum of Pairs)** (Carrillo & Lipman, 1988)

La somme des pairs SP est la fonction objectif la plus utilisée dans les méthodes d'alignement multiple. le SP score est égale à la somme de tous les scores d'alignement par paires possibles des séquences prise deux à deux. En utilisant une matrice de substitution et un système de pénalité de Gaps.

$$SP = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sc(S_i, S_j) \quad - \quad \sum Pénalité(Gaps)$$

$$sc(S_i, S_j) = \sum_{k=1}^{m} sc(a_k, b_k)$$

n : le nombre de séquence,

S_i, S_j : deux séquences,

$sc(S_i, S_j)_{i \neq j}$: le score attribué à chaque couple de séquences aligné, calculé par la formule

m : la taille de l'alignement,
a_k, b_k : deux nucléotides ou acides aminés,
$sc(a_k, b_k)$: le score élémentaire entre deux (acides aminés ou nucléotides) résultant de l'alignement de la lettre a_k en face de la lettre b_k, il est attribué par une matrice de similarité ou de substitution.
Pénalité(*Gaps*) : pénalité due à une insertion ou une délétion.

- **La somme des paires WSP (Weight Sum of Pairs)**

Dans quelques problèmes d'alignement multiple de séquences, l'optimisation du SP peut engendrer des alignements incorrects quand il y a un grand nombre de séquences issues de quelques espèces et peu de séquences d'autres espèces. Pour cela des poids sont attribués aux séquences pour diminuer ce biais de tel sorte que les séquences convergentes reçoivent de petits poids est les séquences les plus divergentes reçoivent de grands poids (Altschul et al ,1989). Généralement, les poids sont calculés directement à partir de l'arbre guide construit initialement. Le score pondéré des paires WSP (Weighted Sum of Pairs) est calculé par la formule suivante :

$$WSP = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} sc(S_i, S_j)$$

L'inconvénient majeur de cette fonction est la difficulté d'établir les bons paramètres d'alignement comme la matrice de substitutions et les pénalités de gaps, qui peuvent être déterminés empiriquement par une large analyse d'alignements (Thompson et al, 1994).

- **La fonction COFFEE** (Notredame et al, 1998)

L'idée de base de la fonction Coffee (Consistency based Objective Function for alignment Evaluation) est de générer une librairie de tous les alignements de paires de séquences possibles par un des algorithmes existants, tel que ClustalW (pour n séquences la librairie contiendra n*(n-1)/2 paires). A chaque paire d'alignement est assigné un poids qui représente la somme des résidus correctement alignés. Le score global donnant la qualité d'un alignement est donné par la formule suivante :

$$Score_{Coffee} = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} score(A_{ij})}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} W_{ij} * len}$$

Où n est le nombre de séquences, *len* est la longueur de l'alignement. W_{ij} est le pourcentage d'identité entre deux séquences alignées S_i, S_j représentée dans la librairie. A_{ij} est l'alignement de deux séquences S_i, S_j dans l'alignement multiple global. Score (A_{ij}) est le degré d'identité entre l'alignement A_{ij} et son correspondant dans la librairie d'alignements. Généralement, les algorithmes basés sur cette fonction donnent des alignements de bonne qualité comme le programme T-COFFEE, SAGA.

Des variantes de cette fonction ont été proposées, tels que T-Coffee (Notredame et al., 2000), M-Coffee (Wallace et al., 2006) et 3D-Coffee (O'Sullivan et al., 2004).

- **Le score consensus** (Tompa, 2000)

L'idée de base de cette fonction est de trouver une séquence consensus (SC) qui représentera toutes les séquences de l'alignement. Le SC est utilisé pour déterminer le score de MSA en additionnant le score des paires d'alignements entre chaque séquence de MSA et le SC.
Considérant un alignement de n séquences ($S_1, S_2, \ldots, S_n$) et une séquence consensus ($c_1, c_2, \ldots, c_l$), l étant la longueur de l'alignement courant. Le but est trouver l'alignement qui minimise la somme des erreurs consensus sur toutes les colonnes :

$$SC = \sum_{i=1}^{l} d(i)$$

La distance des sommes $d(i)$ est calculée pour chaque caractère consensus c_i avec tous les autres résidus de la colonne i.

$$d(i) = \sum_{j=1}^{n} d(S_{ji}, c_i)$$

Par exemple le calcul d'un score consensus de l'alignement suivant :

```
- G C T G A T A T A A C T
G G G T G A T - T A G C T
A G C G G A - A C A C C T
```

Etant données les scores suivants pour les différentes situations : d(A,B) = 2 pour A ≠ B, d(A, -) = d(-, A) = 1 pour A ≠ '-', 0 sinon.

```
a1 = - G C T G A T A T A A C T
a2 = G G G T G A T - T A G C T
a3 = A G C G G A - A C A C C T
     ------------------------
consensus   :   - G C T G A T A T A X C T
column value:   2 0 2 2 0 0 1 1 2 0 3 0 0 = 13
```

- **Le score profil** (Reinert, 2003)

Le principe de cette fonction est de déterminer le profil alignement, qui donne la fréquence relative de chaque résidu dans chaque colonne. Il est utilisé pour déterminer le degré de la relation d'une séquence dans une famille de protéine. Le profil est la somme de la fréquence des résidus.

- **Le score entropie**

La mesure d'entropie en MSA est la somme de l'entropie de chaque colonne. Pour chaque colonne j d'un alignement A, l'entropie est calculée par la formule suivante :

$$Entropie(A[j]) = -\sum_{r} N_r * \log(p_r)$$

Où N_r est le nombre d'apparitions du résidu r dans la colonne j. p_r est la probabilité d'apparition du résidu r dans la colonne j, donnée par la formule :

$$p_r = \frac{N_r}{\sum_{r'} N_{r'}}$$

Cette fonction donne des informations sur le degré de variation des informations contenues dans les séquences. Une colonne reçoit un zéro d'entropie si tous les caractères alignés dans la colonne sont identiques. Plus la colonne est variable, plus l'entropie est élevée. L'entropie de colonne est maximum s'il y a des nombres égaux de tous les caractères possibles dans la colonne. Quand le score d'entropie est employé comme fonction objectif, le but est de trouver u alignement qui minimise la somme des scores d'entropie de toutes les colonnes. Le score basé entropie est préféré dans les études statistiques et mathématiques des alignements (Nicolas et al, 2002).

- **Dialign** (Morgenstern, 1999)

Cette fonction utilise des "fragments" d'alignements par paires locaux et non plus des paires de résidus. Chaque paire de fragments est évalué en fonction de sa longueur et de sa qualité. On ne peut généralement pas retrouver toutes les paires de fragments dans l'alignement multiple des séquences. La fonction objectif à maximiser est la somme des valeurs attribuées aux paires de fragments conservées dans l'alignement multiple. Dialign-T (Subramanian et al., 2005) est une amélioration

- **La fonction NorMD** (Thompson et al., 2001)

Cette fonction est en partie basée sur la fonction de somme des paires. Une pondération est apportée suivant la similarité des séquences mais aussi en fonction de leurs longueurs avant

alignement. La valeur est normalisée afin qu'elle ne dépende que de la qualité du résultat, et non de la longueur de l'alignement ou du nombre de séquences.

III.1.3.5) Les principales méthodes d'alignement de séquences

Dans la littérature, un grand nombre d'algorithmes ont été proposés pour traiter le problème d'alignement multiple de séquences. Dans l'approche d'alignement par score, on distingue deux méthodes : les méthodes exactes et les méthodes approchées.

a) Les méthodes exactes

Une méthode exacte naïve suppose être capable de calculer le score de tous les alignements possibles entre les séquences. Cependant, en raison de sa grande complexité temporelle et spatiale, elle n'est pas applicable en pratique. A titre d'exemple le nombre d'alignements possibles de deux séquences de même longueur *n* est donné par la formule : $\sum_{i=0}^{n} C_{n+i}^{i} * C_{n}^{k}$ (Pour $n = 50$ l'espace de recherche est de 1, $53*10^{37}$ alignements possible). La programmation dynamique (PD) permet d'apporter une réponse à ce problème avec une complexité moindre. La PD se base sur le calcul récursif des instances du problème et la construction d'une table dynamiquement, pour conserver tous les résultats intermédiaires obtenus.

En conservant tous les résultats intermédiaires dans une table on évite d'avoir à les recalculer de nombreuses fois. En effet, refaire ainsi les mêmes calculs à chaque itération est à l'origine de la complexité exponentielle des algorithmes naïfs (Derrien, 2008). Malgré l'amélioration apportée par la programmation dynamique, elle ne permet de traiter qu'un nombre limité de séquences de petites tailles.

L'algorithme de Needleman-Wunch (Needleman & Wunch, 1970) basé sur la programmation dynamique, retourne l'alignement optimal global de deux séquences. Cet algorithme est amélioré plus tard par Gotoh (Gotoh, 1982). L'algorithme de Smith-Waterman (Smith & Waterman, 1981) développé pour répondre au besoin d'alignement local est une adaptation de l'algorithme de Needleman-Wunch. Le principe est assez simple : à chaque étape de la construction de l'alignement, on a trois possibilités : soit la prochaine lettre de la séquence X est alignée en face d'un gap, soit la prochaine lettre de la séquence Y est alignée en face d'un gap, soit on aligne la prochaine lettre de la séquence X avec la prochaine lettre de la séquence Y. Parmi ces trois possibilités, on garde celle qui maximise le score total (i.e. le score de l'étape précédente + le coût de cette étape) et on continue.

On peut généraliser l'algorithme de Needleman-Wunsh pour l'alignement de paires de séquences aux alignements multiples de *n* séquences en employant une table de score *n*-dimensionnelle. On est alors garanti d'obtenir une solution optimale du point de vue de la fonction objectif. Cependant la complexité de l'algorithme qui en résulte est telle qu'on ne peut l'appliquer qu'à un petit nombre de séquences de faible longueur. En effet, Avec *k* séquences de longueur *L*, La mémoire requise serait en O(L^k), Le temps requis serait en O ($k^2\ 2^k\ L^k$). Pour 20 séquence de 100 acides aminés : 100^{20} bytes. Le temps requis serait $20^2 2^{20} 100^{20}$. Il existe alors un problème de temps et d'espace mémoire, il est donc nécessaire d'utiliser d'autres types algorithmes pour pouvoir augmenter le nombre de séquences à aligner :

- La méthode MSA (Measurement Systems Analysis) (Lipman et al., 1989) propose une heuristique basée sur l'algorithme exact de Needleman-Wunsch. Elle offre l'avantage de permettre d'augmenter la limite du nombre de séquences pouvant être alignées (il est ainsi possible d'aligner jusqu'à une dizaine de séquences). Plus tard, plusieurs améliorations ont été apportées pour réduire la complexité temporelle et spatiale de MSA (Gupta et al, 95).
- La méthode DCA (Divided and Conquerir Algorithm) (Stoye, et al., 1997) est une amélioration de MSA, basée sur l'idée de diviser les séquences en groupes de sous-séquences

pour être aligner. Les sous-alignements obtenus seront assemblés pour former un alignement multiple de séquences.

Avec DCA, il est possible de réaliser des alignements de plus de 20 séquences. Pour aligner un grand nombre de séquences, Il est donc nécessaire d'utiliser des algorithmes approchés.

b) Les méthodes approchées

MSA ne peut pas être résolu en temps polynomial et l'utilisation de méthodes exactes comme la programmation dynamique n'est pas réalisable même pour un nombre modéré de séquences de grande taille. Par conséquent, d'autres méthodes sont utilisées pour trouver des solutions approximatives à ce problème. Ces méthodes heuristiques (approximatives) n'offrent aucune garantie sur la recherche de solutions optimales. Elles se divisent en deux classes : les méthodes progressives et les méthodes itératives.

- **Les méthodes progressives**

Le principe des méthodes progressives a été introduit par (Hogeweg & Hesper, 1984), il consiste à aligner deux à deux les séquences les plus similaires, puis à ajouter successivement les autres séquences (moins apparentés) un par un à l'alignement jusqu'à ce que le MSA contienne toutes les séquences. Dans (Feng & Doolittle, 1987) une méthode progressive améliorée comporte deux étapes principales. La première étape consiste à construire un arbre guide phylogénétique en fonction de la distance par paire, pour guider l'ordre d'alignement des séquences. Dans la deuxième étape, les séquences sont alignées suivant cet arbre guide.

Les méthodes progressives fournissent rapidement des résultats de bonnes qualités, et permettent donc, la construction d'alignements contenant un grand nombre de séquences. parmi les algorithmes progressifs les plus connus, il y a ClustalW (Thompson et al., 1994), T-Coffee (Notredame et al., 2000) MAFFT (Katoh et al., 2002), Kalign (Lassmann & Sonnhammer, 2005) et MultAlin (Corpet, 1988).

- ClustalW commence par calculer les distances de tous les paires de séquences pour construire une matrice de distances. Cette matrice est utilisée pour construire un arbre guide pour déterminer l'ordre dans lequel les séquences doivent être alignées. Il utilise la méthode de neighbour-joining (Saitou & Nei, 1987) pour générer l'arbre guide, une matrice de poids pour supposer que toutes les séquences à aligner ont le même degré de divergence et il n'attribue pas la même valeur de pénalité de gap vis-à-vis sa position dans la séquence (il distingue entre les gaps du début, du milieu et de la fin). ClustalW sélectionne automatiquement une matrice appropriée en fonction du pourcentage d'identité observé par paire : ID> 35% Gonnet 80, 35%> ID> 25% Gonnet 250 et ID <25% Gonnet 350.
- T-Coffee (Tree-based Consistency objective function for alignment evaluation) commence par combiner l'alignement par paire local et global pour générer une bibliothèque primaire des alignements par paire. Puis une bibliothèque étendue est générée sur la base des nouveaux alignements, ces alignements sont créés si deux séquences sont alignées à une troisième séquence. l'alignement final est généré sur la base de la bibliothèque d'extension, M-Coffee (Wallace et al,. 2006) est une extension de T-Coffee qui utilise la consistance pour approcher un alignement consensus, et un méta-modèle pour créer un MSA en combinant le résultat de nombreux algorithmes individuels en un seul MSA.
- MAFFT est une méthode basée sur les transformées de Fourier rapide FFT (Fast Fourier Transform). FFT sert à détecter les séquences homologues qui aident à construire un bon arbre-guide. Aussi elle sert à réduire la complexité temporelle de l'algorithme. Deux heuristiques différentes, la méthode progressive (FFT-NS-2) et la méthode de raffinement itératif (FFT-NS-i), sont implémentées dans MAFFT.

- Kaling est une méthode progressive qui utilise l'algorithme Wu Manber string-matching (Wu & Manber, 1992) pour estimer la distance entre les séquences. Cette technique permet d'améliorer la précision et la rapidité de l'alignement multiple. Comme amélioration de Kalign il y a le HMM-Kalign (Becker et al., 2007) et Kalign-LCS (Deorowicz et al., 2014).
- MultAlin il considère que l'arbre-guide AG_0 déterminé à l'aide de la matrice de distance ne peut pas toujours être le meilleur. par conséquent, l'alignement A_0 généré à partir de AG_0 est utilisé pour construire le nouvel arbre-guide AG_1. AG_1 est utilisé pour générer un nouvel alignement A_1. ce processus est répété jusqu'à la stabilisation de l'alignement ($AG_{i-1} = AG_i$) (Corpet, 1988).

Dans les méthodes progressives le grand avantage est la rapidité d'exécution et le principal inconvénient est la forte dépendance vis-à-vis de la précision des alignements initiaux par paires. Par conséquent, s'il y a des erreurs dans les premières étapes de l'algorithme, elles ne pourront pas être corrigées dans les étapes ultérieures, elles seront également propagées à l'alignement final.
La principale alternative à l'alignement progressif est la stratégie itérative qui est raisonnablement robuste et beaucoup moins sensible au nombre de séquences.

- **Les méthodes itératives**

Les méthodes itératives (Notredame & Higgins, 1996) sont des heuristiques qui fournissent des résultats de très bonnes qualités en un temps raisonnable. Le principe consiste à aligner toutes les séquences simultanément, en générant des alignements de plus en plus proche de l'optimum après des itérations successives jusqu'à la fin de l'algorithme. L'algorithme doit s'arrêter, lorsqu'aucune amélioration ne peut être ajoutée au résultat ou le nombre d'itérations atteint le maximum.

Les algorithmes évolutionnaires sont l'une des méthodes itératives les plus utilisées pour résoudre les MSA. Les méthodes évolutionnaires itératives ne peut pas concurrencer en termes de vitesse les méthodes d'alignement progressif mais elles sont très flexibles car elles n'ont pas de limitation particulière sur la fonction objectif à optimiser et elles ont l'avantage de pouvoir corriger les séquences initialement mal alignées ; ce qui n'est pas possible avec la méthode progressive. En plus plusieurs algorithmes itératifs peuvent être combinés pour fabriquer des alignements. Les méthodes itératives les plus représentatives sont :

- SAGA (Sequence Alignment by Genetic Algorithm) (Notredame & Higgins, 1996) est un algorithme génétique itératif. La méthode consiste à faire évoluer une population d'alignements et à l'améliorer (mesurée par la fonction objectif WSP) progressivement par les opérateurs de mutations et de croisements.
- Tabu search (Riaz et al., 2004) procède en deux étapes pour réaliser l'alignement multiple : – création d'un alignement initial A_0 par insertion de gaps dans les séquences. – utilisation d'une méthode tabou (Glover and Laguna, 1997) pour déplacer les gaps dans l'alignement. L'ensemble des alignements que l'on peut obtenir en déplaçant un gap dans A_i constitue les voisins de Ai. L'alignement A_{i+1} est le meilleur voisin qui n'appartient pas à la liste tabou. La solution est obtenue lorsque l'algorithme se stabilise.
- MSA-GA (Gondro & Kinghorn, 2007) est une méthode basée sur l'AG dans laquelle la population initiale est generée avec des séquences pré-alignées en utilisant l'algorithme Needleman-Wunsch.

Ces algorithmes nécessitent généralement plus de calculs, et leurs temps d'exécution sont souvent plus importants.

- **Les méthodes hybrides**

Dans les algorithmes itératifs, plusieurs méthodes peuvent être utilisées pour effectuer l'alignement. Certains d'entre eux combinent des méthodes itératives et progressives, telles que

GAPAM (Genetic Algorithm Progressive Alignement Method) (Naznin et al., 2012) et Muscle (Multiple sequence comparison by log expectation) (Edgar, 2004).

Nous avons cité quelques exemples d'algorithmes d'alignement multiple de séquences, mais il en existe de nombreux autres. Et chaque année, de nombreuses publications relatives à des algorithmes d'alignement multiple sont réalisées.

III.1.3.6) Evaluation de performances

La grande difficulté dans l'alignement multiple de séquences est de savoir si biologiquement il est bon. Cette difficile question peut être seulement répondue en utilisant une fonction objectif mathématique capable de mesurer la qualité biologique d'un alignement. Pour cela plusieurs fonctions objectifs ont été proposées tel que la somme des paires SP, Coffee, le score profil, etc. Malheureusement, on ne connait pas à cet instant, une fonction objectif dont l'optimal mathématique est corrélé avec l'optimal biologique. Un moyen utilisé pour tester l'efficacité biologique des méthodes d'alignement est l'utilisation des bases d'alignements de références (benchmarks) comme BaliBase (Thompson et al., 1999), Oxbench (Raghava et al., 2003), SABmark (Van Walle et al., 2005), Homstrad (Mizuguchi et al., 1998). Dans cette section nous allons présenter les principales bases de jeux d'essais. Nous détaillerons plus particulièrement Balibase que nous avons utilisée pour valider notre méthode.

Balibase (Thompson et al., 1999) est la première base des jeux d'essais pour les protéines qui a été massivement citée dans les publications. Elle continue à être une référence incontournable. La première version de Balibase comportait 142 jeux de séquences, repartis en 5 catégories, appelées références. La référence1 correspond à des jeux d'essais classés par longueur (petit, moyen, long), et par pourcentage de similitude (< 20%, < 40% et > 40%). La référence2 présente une séquence orpheline, qui n'a aucune similarité avec les autres séquences. Les jeux d'essais de la référence 1 sont composés de 4 à 6 séquences, alors que ceux des autres références en contiennent plus : en moyenne une dizaine de séquences, et jusqu'à 23 dans certains cas. La version 2 de Balibase reprend les 5 références existantes auxquelles sont ajoutées 4 nouvelles références. Cependant, ces nouvelles références sont uniquement constituées de jeux d'essais et ne contiennent pas de résultat optimal. Les tests que l'on trouve dans la littérature sont réalisés sur les 5 premières références de Balibase 2. La version 3 de Balibase (Thompson et al., 2005) contient également de nouveaux jeux d'essais, Comme pour la version 2, les nouveaux jeux ne sont pas proposés avec l'alignement optimal.

Chaque jeu d'essais de Balibase est proposé avec la meilleure solution. Pour pouvoir réaliser des comparaisons entre le résultat d'un algorithme et la solution optimale ; deux fonctions de comparaison ont également été ajoutées, chacune permettant de montrer un critère de qualité pour les alignements :

- La fonction de comparaison SPS (Sum-of-Pairs Scor) est basée sur le principe de la fonction de somme des paires. Toutes les paires de séquences sont parcourues, aussi bien dans l'alignement de référence que dans l'alignement résultat. Pour chacun des deux alignements, les paires de résidus identiques ont la valeur 1 et les autres ont la valeur 0. Cette méthode consiste donc à déterminer le nombre de paires de résidus identiques entre la référence et le résultat. En divisant cette somme par le nombre total de paires de résidus de la référence, on obtient un pourcentage de similarité entre les paires de résidus des deux alignements. A noter que les gaps ne sont pas comptabilisées, il est donc possible qu'un alignement multiple ait un score de 1 sans être identique à la référence. Réussir à obtenir une paire de résidus identique entre la référence et le résultat ne suffit plus, il faut obtenir l'identité entre tous les résidus d'une même colonne.
- La fonction de comparaison CS (Column Score) est la qualité que l'on attribue d'une colonne complète bien alignée. Le critère de comparaison CS se calcule en faisant la somme de toutes les

colonnes identiques entre l'alignement de référence et l'alignement résultat. Pour obtenir un pourcentage, ce nombre est divisé par le nombre de colonnes de l'alignement de référence.

Il est facile de constater que le critère de comparaison CS est beaucoup plus contraignant que le critère SPS. En effet, il suffit d'un seul résidu mal placé pour que le reste de la colonne soit évalué à 0. Ce phénomène est particulièrement visible pour les jeux d'essais avec une séquence orpheline. Cette séquence étant difficile à aligner avec les autres la valeur CS de l'alignement peut très facilement valoir 0.

Le site Internet de Balibase (http://bips.u-strasbg.fr/fr/Products/Databases /Balibase) propose un programme écrit en langage C (bali score.c) qui permet d'évaluer les valeurs SPS et CS. Il contient également un tableau dans lequel sont déjà calculés des résultats des 10 algorithmes pour chaque jeu d'essais. Ces résultats sont destinés à simplifier les tests sur un nouvel algorithme, et permettent de le comparer aux 10 algorithmes en question. Une fois qu'un MSA est effectué, nous pouvons alors procéder à la dérivation d'un arbre phylogénétique.

VI) Conclusion

Depuis des années, la bioinformatique applique des concepts aussi divers que les (méta)heuristiques, la classification, l'apprentissage, les chaînes de Markov, etc., pour résoudre des problèmes fondamentaux de la bioinformatique (alignement des séquences, prédiction de structure, phylogénie moléculaire, détection de gènes, etc.). Les entités biologiques (séquences, structures, motifs, etc., ...) sont analysées (étudiés, alignés, classés, etc.) afin de tirer les caractéristiques et les rapports entre ces données et produire de nouvelles connaissances biologiques.

Dans l'analyse bioinformatique, deux différentes approches fondamentales peuvent être identifiées, celles basées sur la modélisation comparative et celles basées sur la modélisation de novo. Les méthodes dites de novo se basent sur une analyse directe à partir de données expérimentales sans utilisation d'autres objets. Par exemple, 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. Par contre, Les approches de modélisation comparative sont basées sur l'homologie, elles ne sont donc applicables que s'il existe des objets avec une haute similarité de séquences ou de structures connues. Par exemple, la prédiction des gènes basée sur la similarité/homologie consiste à comparer la séquence étudiée avec des séquences connues, rassemblées dans les bases de données. La similarité de séquences et de structures peuvent fournir des preuves de relations évolutionnaires entre les entités et peuvent indiquer des propriétés fonctionnelles partagées.

Les problèmes que nous avons présentés (alignement, phylogénie, ...) ont tous une grande importance en biologie moléculaire. Chacun de ces problèmes a été démontré comme étant NP-Complet. Devant l'importance de ces problèmes, il est nécessaire de pouvoir obtenir des résultats de bonne qualité pour qu'ils puissent être utilisés par les biologistes. L'objectif de la bioinformatique est de de contribuer à l'amélioration des méthodes de résolution. Les méthodes exactes permettant de les résoudre ne sont bien souvent utilisables que pour des petites instances. En pratique, ce sont les méthodes approchées qui sont employées pour apporter des solutions de bonne qualité.

I) Introduction

Nombreux problèmes rencontrés en bioinformatique et la biologie computationnelle peuvent être formulés comme problèmes d'optimisation et, par conséquent, se prêtent à l'application de puissantes techniques de recherche heuristique (Cohen, 2004), (Mitra, 2005). Traditionnellement, l'optimisation est effectuée par rapport à un seul objectif soit par agrégation ou par optimisation des objectifs séparément, mais la possibilité d'optimiser plusieurs objectifs simultanément est de plus en plus reconnue (Coello Coello et al., 2002). En biologie, il a été démontré que l'optimisation multiobjectif à des avantages significatifs par rapport à une approche à objectif unique (agrégée ou mono-objectif) (Deb & Reddy, 2003), (Curteanu et al., 2006), (Mandal et al., 2005) et (Someren et al., 2003).

L'article de (Handel et al., 2007) passe en revue l'application de l'optimisation multiobjective dans le domaine de la bioinformatique et de la biologie computationnelle. Les auteurs ont identifié cinq contextes distincts, donnant raisons à l'utilisation de l'optimisation multiobjectifs en bioinformatique.

Dans ce chapitre, nous visons à décrire la portée potentielle de méthodes d'optimisation multiobjectif dans des applications en bioinformatique à travers un état de l'art des travaux existants, en particulier l'utilisation des algorithmes d'optimisation évolutionnaires multiobjectif (AEMO). Le chapitre se terminera par quelques pistes prometteuses pour de futures recherches.

II) Contribution de l'optimisation multiobjectif à la bioinformatique

(Handel et al. 2007) ont proposé une catégorisation basée sur différents types de contextes qui soulèvent le besoin d'optimisation multiobjectif dans les applications biologiques.

II.1) Optimisation multiobjectif standard

La première catégorie identifie le contexte « standard » de l'optimisation multiobjectif, où tous les objectifs à optimiser sont clairs et mesurables. Dans ce contexte, les fonctions objectifs ont la primauté, c'est-à-dire ce sont eux qui définissent l'ensemble de Pareto (l'ensemble des solutions est induit par les objectifs). Ainsi, l'utilisation d'une approche multiobjectif, apporte beaucoup d'avantage pour le biologiste aussi bien dans la qualité que dans le choix de la solution à retenir. Il s'agit d'une part, de tirer parti de l'aspect contradictoire des objectifs pour parvenir à un bon compromis (un équilibre) améliorant la qualité (biologique) des solutions et, d'autre part, la possibilité d'obtenir plusieurs solutions en une seule exécution donnant ainsi plus de choix au décideur pour des solutions biologiquement significatif. Un exemple de cette catégorie de problème est l'optimisation des processus biochimiques où des compromis existent entre les aspects de la qualité du produit et du temps de réaction. La polymérisation est d'une importance fondamentale dans les industries chimique et biochimique. C'est un processus de réaction qui relie des petites molécules dans des chaînes de polymères pour former des macromolécules. Une polymérisation optimale est soumise à une gamme d'objectifs contradictoires différents, y compris le degré de polymérisation et le temps de réaction, et elle est donc naturellement adaptée à une approche multiobjectif. Ce problème a été abordé à l'aide d'un certain nombre d'objectifs et d'algorithmes d'optimisation différents (Deb et al., 2004), (Garg & Gupta, 1999), (Mitra et al., 2004), (Curteanu et al., 2006). D'autres exemples d'utilisation de l'optimisation multiobjectif sont des applications liées au processus de fermentation de la bière (Andres-Toro et al., 2004), la fermentation de l'acide citrique d'aspergille noir (Aspergillus niger) (Mandal et al., 2005), la production d'acide gluconique (Halsall-Whitney et al., 2003), la production d'huile dans la levure Yarrowia lipolytica (Muniglia et al., 2003) et l'optimisation des fonctions hépatiques de la sécrétion d'urée et d'albumine dans le cadre métabolique (Sharma et al., 2005). Dans (Higuera et al., 2012) une approche multiobjectif a été

utilisée pour optimiser les paramètres de régulation allostérique des enzymes dans un modèle de cycle de substrat métabolique. Les auteurs ont utilisé deux objectifs : la bonne direction du flux dans un cycle métabolique et le coût énergétique de l'application de l'ensemble de paramètres. L'allostérie est un mode de régulation de l'activité d'une protéine oligomérique par lequel la fixation d'une molécule effectrice en un site, modifie les conditions de fixation d'une autre molécule, en un autre site distant de la même protéine. Ce concept a été formalisé par (Monod et al., 1965). Enfin, Les processus biochimiques sont généralement dynamiques et les cellules ont souvent plus d'un objectif qui sont généralement contradictoires, par exemple, minimiser la consommation d'énergie tout en maximisant la production d'un métabolite spécifique. Par conséquent, une optimisation multiobjectif est nécessaire pour calculer les compromis entre ces objectifs contradictoires (Nimmegeers et al., 2016). Ces auteurs ont conçu un modèle de réseau inspiré de la glycolyse dans lequel un problème d'optimisation multiobjectif est considéré : la minimisation du coût enzymatique et la minimisation du temps de fin avant d'atteindre une concentration minimale de métabolite extracellulaire.

II.2) Optimisation multiobjectif comme outil pour contrebalancer un biais

La deuxième catégorie est celle où l'optimisation multiobjectif est utilisée comme outil pour contrebalancer un biais de mesure affectant une fonction objectif. Un tel biais de mesure est, par exemple, rencontré dans les problèmes d'alignement, où les alignements courts peuvent être obtenus de manière simple et le nombre de désapparièrent (mismatches) augmentent automatiquement avec la longueur de l'alignement. Ce paramètre peut être décrit mathématiquement, comme suit, en supposant qu'un seul objectif (principal) soit optimisé :

$$f(x) = f'(x) + m(g(x))$$

Où f' est une mesure idéale (inconnue) non-biaisée de l'objectif principal, $m\big(g(x)\big)$ est un terme de biais où *m* est une fonction inconnue mais monotone d'une fonction mesurable g et f est la somme mesurable mais biaisée des deux. Dans l'exemple de problème d'alignement, f (la fonction de score utilisée) donne une estimation de qualité biaisée, g est la longueur de l'alignement donné, *m* est supposé être une fonction monotone, et f' est la qualité idéale (mais inconnue) de l'alignement. On veut minimiser $f'(x)$ comme suit :

$$minimiser\ f'(x) = f(x) - m(g(x))$$

Mais, puisque *m* est inconnu, nous ne pouvons pas formuler le problème de cette façon. Cependant, nous pouvons formuler le problème en termes de deux objectifs mesurables :

$$minimiser\ (f(x), -\big(g(x)\big))$$

Par conséquent, le cadre multiobjectif est utilisé comme moyen d'introduire un objectif supplémentaire g pour contrebalancer le biais de l'objectif principal. Notons que les équations ci-dessus peuvent être généralisées à plus d'un objectif primaire, si nécessaire. L'ensemble des solutions optimales de Pareto contiendra certainement la solution souhaitée puisque chaque optimum de Pareto est la meilleure valeur de $f(x)$, étant donné une valeur fixe de $g(x)$. Dans ce scénario, la sélection de la meilleure solution ne dépend généralement pas de préférences, mais sur l'estimation des biais. Des exemples de ce type de problème comprennent la sélection de caractéristiques (feature selection) non supervisées et les problèmes d'alignement de séquences et de structures.

L'optimisation multiobjectif a été introduite comme solution potentielle au problème de la sélection de caractéristiques non supervisées, car elle permet d'optimiser l'un de ces objectifs et de contrebalancer son biais par la minimisation ou la maximisation simultanée de la cardinalité des caractéristiques (Handl & Knowles., 2006), (Kim et al., 2002), (Morita et al., 2003).

Une approche multiobjectif de l'alignement de séquence a été proposée dans (Roytberg et al., 1999). Elle est basée sur une forme modifiée de programmation dynamique. En ce qui concerne

l'alignement des séquences locales, un autre compromis peut être observé entre la longueur des motifs comparés et les scores de qualité obtenus : à l'évidence, le nombre de substitutions défavorables ou de pénalités de gaps tend à augmenter pour les alignements plus longs, c'est-à-dire il y a un biais provoquant la préférence des alignements courts. Dans (Zwir et al., 2002), une approche évolutionnaire multiobjectif a été utilisée pour l'optimisation simultanée de ces deux aspects conflictuels lors de l'identification de courts éléments répétitifs intercalés dans la séquence d'ADN de Tripanosoma cruzi : La méthode a obtenu toutes les solutions identifiées par des approches alternatives mono-objectif et a découvert des compromis efficaces supplémentaires entre les deux objectifs utilisés. Les approches d'optimisation multiobjectif de l'identification des motifs ont également été explorées dans (Rajapakse et al., 2006), dans le but d'intégrer plusieurs sources d'information et dans (Cotik et al., 2005), dans le but de mieux préciser les propriétés des motifs recherchés. Dans les travaux de (Ranjani Rani and Ramyachitra, 2016) l'algorithme d'optimisation de la recherche de nourriture bactérienne BFO (Bacterial Foraging Optimization Algorithm) a été utilisé avec des objectifs multiples : la maximisation de la similarité, le pourcentage de non-gap, les blocs conservés et la minimisation de la pénalité des gaps. La base de données de référence BAliBASE 3.0 a été utilisée pour examiner l'algorithme multiobjectif proposé MO-BFO par rapport à d'autres méthodes largement utilisées Clustal, Omega, Kalign, MUSCLE, MAFFT, algorithme génétique (GA), optimisation des colonies de fourmis (ACO), colonie d'abeilles artificielle (ABC), Optimisation des essaims de particules (PSO) et algorithme génétique hybride avec colonie d'abeilles artificielle (GA-ABC). Les résultats finaux montrent que l'algorithme MO-BFO proposé donne un meilleur alignement que les méthodes les plus largement utilisées.

II.3) Intégration de sources multiples

Dans la troisième catégorie, l'optimisation multiobjectif est utilisée pour intégrer des données (bruyantes) provenant de plusieurs sources. Les problèmes où cette approche est utilisée sont souvent d'origine mono-objectif. Cependant, plusieurs vues bruyantes des données doivent être intégrées, car leur utilisation combinée peut donner de meilleurs résultats que l'utilisation de données provenant d'une seule source d'informations. Mathématiquement, ce paramètre peut être décrit par un ensemble de fonctions objectifs :

$$f_1(x) = f_1'(x) + \bar{n}_1$$
$$\dots$$
$$f_m(x) = f_m'(x) + \bar{n}_m$$

Où la valeur de chaque fonction objectif f_i est égale à la valeur d'une fonction idéale f_i'avec quelques bruit aléatoire inconnu $\bar{n}_i$, pour $i \in 1...m$. et le problème est formulé comme :

$$minimiser\ z = f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

Notons qu'il n'est pas garanti que la solution souhaitée soit parmi les optima de Pareto.

Des exemples de ce type de problème sont l'inférence des arbres phylogénétiques et la classification de données (data clustring) avec plusieurs matrices de dissimilarité.

Les approches traditionnelles de l'inférence d'arbres phylogénétique (les méthodes de matrice de distance, les méthodes de parcimonie maximale et les méthodes de vraisemblance maximale) ne prennent pas en compte l'intégration de multiples ensembles de données provenant de sources différentes. Bien que ces ensembles de données soient souvent bruyants et partiellement conflictuels, on peut généralement supposer qu'ils se complètent mutuellement et qu'ils sont plus informatifs en combinaison que seuls. L'intégration de ces différentes sources d'information se fait le plus souvent à priori ou à posteriori de l'inférence réelle de l'arbre phylogénétique (De Queiroz et al., 1995). L'approche multiobjectif fournit un outil alternatif pour intégrer et arbitrer ces données contradictoires au cours du processus d'inférence et qu'une telle approche peut en fait être plus robuste

et informative que l'intégration a priori ou a posteriori utilisée dans la littérature (Poladian and Jermiin, 2006). Plus récemment (Noutahi and El-Mabrouk, 2018) présentent un nouvel algorithme GATC (Genetic Algorithm for gene Tree Construction), basé sur un algorithme génétique et traite le problème comme celui de l'optimisation multiobjectif de la topologie des arbres de gènes, étant données des contraintes relatives à l'évolution des familles de gènes par mutation de séquences et par gain/perte de gènes. Les auteurs ont montré qu'une telle approche est non seulement efficace, mais appropriée pour la construction d'ensemble d'arbres de référence. L'algorithme présente également l'avantage d'être facilement adaptable pour considérer d'autres sources d'informations (conservation de fonction, ordre des gènes, etc.) sans avoir à formuler un modèle d'évolution tenant explicitement compte de ces informations.

Le clustering est l'une des tâches fondamentales de la classification non supervisée. Des travaux ont montré qu'une approche multiobjectif du clustering peut en effet se traduire par une performance améliorée et robuste à travers des données présentant une gamme de propriétés de données différentes et peuvent être supérieures à certaines approches d'intégration a posteriori (Handl & Knowles, 2007). L'utilisation de l'optimisation multiobjective pour le clustering a également été proposée pour les situations dans lesquelles le critère de clustering est biaisé par rapport au nombre de clusters (Liu et al., 2005) ou où plusieurs sources de données (sous la forme de plusieurs matrices de dissimilarité) devraient être intégrées dans un clustering unique (Dale & Dale, 1994), (Ferligoj and Batagelj, 1992). Récemment dans (Keel et al., 2018) une approche théorique des jeux est proposée pour la construction de réseaux de protéines est adaptée dans le cadre de l'optimisation multiobjectif, et étendue pour incorporer la procédure de raffinement de clustering. La nouvelle méthode, MOCASSIN-prot, a été appliquée à des protéines multi-domaines de cluster de dix génomes. La performance de MOCASSIN-prot a été comparée à deux méthodes de clustering de protéines, le clustering de Markov (TRIBE-MCL) et le clustering spectral (SCPS). Ils ont montré que par rapport à ces deux méthodes, MOCASSIN-prot, qui utilise à la fois la composition du domaine et les informations de similarité de séquence quantitative, génère moins de faux positifs. Il permet d'obtenir des grappes de protéines plus cohérentes sur le plan fonctionnel et de mieux différencier les familles de protéines.

II.4) Approximation des performances par des proxys

La quatrième catégorie comprend les applications dans lesquelles l'objectif sous-jacent «réel» du problème, $f'(x, y)$ est une fonction à la fois de la solution *x* et de certaines variables «cachées» y qui ne sont pas disponibles lors de l'optimisation. Par exemple, dans l'apprentissage supervisé d'un classificateur, *y*, fait référence à la capacité de généralisation du classificateur sur les données futures (qui peuvent être estimé à l'aide d'un ensemble de test après l'optimisation, mais le classificateur ne doit pas être formé à l'aide de ces exemples).

Puisque la fonction f' ne convient pas à une utilisation dans un processus d'optimisation (car y n'est pas disponible), il doit être remplacé par des objectifs « proxys » $f_i(x)$ qui sont des fonctions de *x*. Souvent, ces objectifs «indirects» ne captent que certains aspects d'une bonne solution et différents proxys sont complémentaires les uns par rapport aux autres. Ainsi, il devrait s'attendre à ce que la ou les solutions souhaitées obtiennent un score relativement fort sur tous les objectifs «proxy» et l'approche multiobjectif semble donc utile, bien que la solution ne puisse pas être garantie comme faisant partie de l'ensemble associé de Pareto optimal. Dans ce contexte, c'est la solution qui a la primauté et les objectifs ne sont qu'un moyen d'orienter la recherche afin de découvrir cette solution. Des exemples de ce type de problème comprennent l'apprentissage supervisé des classificateurs, le regroupement des données (clustering) et la prédiction de la structure des protéines.

Lors de l'examen des performances des classificateurs binaires (par exemple, pour la distinction entre tumeur et tissu sain), la sensibilité et la spécificité sont souvent considérées comme mesures informatives de la performance de la classification. Ces deux mesures sont toujours en conflit et, pour un classificateur, le compromis entre les deux peut être représenté sous forme de courbe caractéristique de fonctionnement du récepteur (ROC) (Metz, 1978). Traditionnellement, cette courbe de compromis n'a pas été explicitement optimisée et, à la place, une pondération a priori des deux objectifs a été utilisée pendant l'entraînement. Cependant, il est peu probable qu'une courbe ROC obtenue de cette manière soit optimale au sens optimal de Pareto et des compromis plus favorables entre sensibilité et spécificité peuvent être obtenus grâce à l'utilisation directe de l'optimisation multiobjectif (Kupinski and Anastasio, 1999). Les courbes ROC pour les problèmes multi-classes ont également été optimisés en utilisant l'optimisation évolutionnaire multiobjectif (Everson and Fieldsend, 2006).

Dans la prédiction de la structure des protéines, la formulation d'une fonction énergétique qui modélise de manière réaliste les différentes interactions locales et globales contribuant au repliement des protéines est d'une complexité extrême. Traditionnellement, les fonctions énergétiques empiriques consistent en une somme des différents composants énergétiques contribuant au processus de repliement de la macromolécule ou des pondérations entre ces composantes (Tsai et al., 2003). L'utilisation de l'optimisation multiobjectif peut être une approche bénéfique dans la prédiction de la structure des protéines. Elle a été suggérée par un certain nombre d'auteurs : (Schulze-Kremer. 2003) a suggéré la décomposition de la fonction énergétique en un vecteur à neuf dimensions. Entre autres, l'énergie de torsion, l'énergie de van der Waals, l'énergie électrostatique et un terme d'énergie de pénalité favorisant des modèles de pliage compacts ont été pris en compte et optimisés à l'aide d'un algorithme évolutionnaire multiobjectif. Dans (Cutello et al., 2006a), (Day et al., 2002), une formulation plus simple à deux objectifs basée sur le potentiel énergétique CHARMM (Chemistry at HARvard Macromolecular Mechanics est le nom d'un ensemble de champs de force largement utilisés en dynamique moléculaire) a été proposée, où les interactions locales et non locales ont été traitées comme des objectifs séparés. Des résultats prometteurs ont été obtenus en comparaison avec d'autres algorithmes sur cinq protéines différentes.

II.5) Multi-objectivisation

Pour effectuer une multi-objectivisation, nous devons soit remplacer l'objectif unique original d'un problème par un ensemble de nouveaux objectifs, soit ajouter de nouveaux objectifs en plus à la fonction d'origine. Dans les deux cas, nous voulons être sûrs que l'optimum global du problème d'origine est l'un des points optimaux de Pareto dans la version multiobjectif du problème. Spécifiquement, le problème doit être reformulé de manière à maximiser $k \geq 2$ fonctions objectifs de telle sorte que la relation suivante entre les solutions des deux formulations soit vérifiée (Knowles et al., 2001) :

$$\forall x^{opt} \in X^{opt}, \exists x^* \in X^*, x^* = x^{opt}$$

Où x^{opt} est une solution optimale au problème mono-objectif original, et X^{opt} est l'ensemble de toutes ces solutions, et x^* et X^* se rapportent à la formulation multiobjectif du problème. Cela garantit qu'au moins une des vraies solutions optimales de Pareto sera optimale par rapport à l'objectif principal d'origine et correspondra à la meilleure solution. Un exemple de ce type de problème est l'identification de la structure à partir des données de diffraction des rayons X sur poudre (Putz et al., 1999), (Lanning et al., 2000).

III) Applications des AEMO en bioinformatique

Parmi les nombreuses métaheuristiques actuellement utilisées, les algorithmes évolutionnaires multiobjectif (AEMO) sont clairement les plus populaires dans la littérature spécialisée et ont encore plusieurs opportunités de recherche à offrir aux nouveaux arrivants (Coello Coello, 2017). Dans la littérature on y trouve des applications dans pratiquement toutes les disciplines, y compris la biologie (Coello Coello & Lamont, 2004). L'utilisation des AEMO en biologie a suscité un intérêt croissant, principalement dans le domaine de la bioinformatique (Handl et al., 2007), (Mitra et al., 2006). Une analyse de la littérature présente cinq principaux types d'applications des AEMO en biologie (Jaimes & Coello Coello, 2008).

III.1) Optimisation du système

Il s'agit de certaines applications dans lesquelles il est intéressant de déterminer le degré d'optimalité d'un certain système biologique. Les généticiens réalisent des projets en utilisant un ensemble de SNP (single nucleotide polymorphism), pour, par exemple, rechercher des gènes responsables d'une maladie. Ainsi, avant le lancement du projet, les généticiens doivent sélectionner un sous-ensemble de SNP à partir de grandes bases de données. (Hubley et al., 2002) ont formulé cette tâche comme un problème d'optimisation bi-objectif et ont proposé un algorithme appelé MAGMA (Multiobjective Analyzer for Genetic Marker Acquisition). Les objectifs utilisés par les auteurs sont : minimiser l'écart moyen par rapport à la longueur d'espace idéale entre deux SNP et maximiser la qualité moyenne des SNP. L'algorithme proposé a été testé en utilisant deux vrais problèmes de sélection de SNP avec une bibliothèque relativement petite de SNP, et un problème construit avec une grande bibliothèque contenant un grand nombre de SNP. Le front de Pareto avait dans tous les cas une forme concave, et MAGMA a pu découvrir le vrai front de Pareto dans les trois problèmes.

(Lee et al., 2004) formulent la conception de sonde pour les puces à ADN comme un problème multiobjectif, qui est ensuite résolu par le NSGA-II. La méthode proposée a permis d'obtenir des ensembles de sondes plus fiables que Les biopuces basés sur la réalisation de matrices d'oligonucléotides pré-synthétisés pour la détection du VPH (Human Papilloma Virus).

III.2) Classification

Une grande variété de problèmes en bioinformatique reposent sur l'exécution de tâches de classification (supervisées, non supervisées ou des combinaisons des deux). Deb et Reddy (2003) abordent la classification des données sur le cancer à deux classes en utilisant le NSGA-II. Les auteurs formulent un problème à deux objectifs et un problème à trois objectifs. Le premier problème consiste à minimiser la taille du sous-ensemble de gènes et à minimiser la somme des erreurs de classification dans les échantillons d'apprentissage et de test. Dans le deuxième problème, les erreurs de classification dans les échantillons d'apprentissage et de test sont considérées comme deux objectifs différents. Dans (Liu et al., 2005) les auteurs ont proposé une méthode basée sur l'entropie pour sélectionner des gènes liés aux différentes classes de cancers, réduisant simultanément la redondance entre les gènes. Ce problème bi-objectif est résolu en utilisant une approche d'agrégation résolue par un algorithme glouton. Mitra et ses collaborateurs (Mitra et al., 2006) (Mitra & Banka, 2006), (Banka & Mitra, 2006) ont proposé un cadre évolutif pour le bi-clustering des données d'expression génique dans un contexte multiobjectif. Les deux objectifs considérés étaient la maximisation de la taille du bi-cluster et la maximisation de l'homogénéité. Selon les résultats, ce cadre donne de meilleurs résultats que certaines autres méthodes disponibles dans la littérature (Bleuler et al., 2004), (Cheng and Church 2000), (Yang et al., 2003), (Zhang et al., 2004).

III.3) Alignement de séquence et de structure

L'objectif est d'évaluer les similitudes structurelles entre une certaine macromolécule et une séquence disponible à partir d'une base de données. La recherche se fait à travers une série d'alignements. (Malard et al., 2004) formulent l'identification peptidique de novo comme un problème d'optimisation multiobjectif avec contraintes. Les objectifs considérés dans l'étude sont la maximisation de la similitude entre les portions de deux peptides et la maximisation du rapport de vraisemblance entre l'hypothèse nulle et l'hypothèse alternative. Les contraintes sont traitées comme une fonction objective de la même manière que l'optimisation multiobjective contrainte par algorithme génétique (COMOGA) proposée par (Surry and Radcliffe, 1997). Dans (Ortuno et al., 2012) les auteurs ont implémenté un algorithme évolutionnaire multiobjectif basé sur NSGA-II pour résoudre le problème d'alignement multiple de séquences. Soto et Becerra (Soto et Becerra 2014) ont proposé un algorithme évolutionnaire multiobjectif efficace pour améliorer des séquences pré-alignées. La méthode proposée est validée avec une base de données d'alignements de séquences multiples raffinés et utilise quatre métriques standard pour comparer la qualité des résultats. Dans (Zambrano-Vega et al., 2017) les auteurs ont présenté jMetalMSA, un outil logiciel open source pour résoudre des problèmes d'alignement multiple de séquences avec des métaheuristiques multiobjectif.

III.4) Prédiction et conception de structure

Dans ce cas, l'objectif est de prédire la structure d'une macromolécule, étant donné que les propriétés fonctionnelles des macromolécules dérivent de leur forme tridimensionnelle. Cette forme tridimensionnelle est, à son tour, principalement déterminée par la séquence des bases ou des acides aminés. Des travaux dans (Shin et al., 2005), (Lee et al., 2003) ont utilisé le NSGA-II pour générer un ensemble de séquences d'ADN qui peuvent être utilisées dans la conception de microarray. Les propriétés souhaitables d'une séquence d'ADN sont les mesures de qualité réalisées tout en satisfaisant certaines contraintes. La qualité d'une séquence peut être obtenue en minimisant quatre objectifs : la similitude entre deux séquences de l'ensemble, le nombre de bases pouvant être hybridées entre les séquences de l'ensemble, le degré d'occurrences successives d'une même base et la probabilité de former une structure secondaire. Une bonne séquence doit avoir des propriétés physiques et chimiques similaires. Dans (Day et al., 2002) les auteurs ont utilisé l'algorithme génétique désordonné rapide multi-objectif (MO fmGA) (Zydallis et al., 2001) pour résoudre le problème de prédiction de la structure des protéines. Cette étude est basée sur une technique de minimisation d'énergie qui utilise le CHARMm fonction énergétique. Cette fonction est composée de 10 termes majeurs et afin d'utiliser un cadre multi-objectif, elle a été décomposée en deux objectifs de minimisation : la somme des énergies connectées et la somme des énergies des atomes non connectés. Les variables de décision pour ce problème sont les angles dièdres de la protéine à résoudre. L'algorithme a été appliqué à deux protéines, [Met]-Enkephelin et polyalanine. Pour les deux problèmes, un front de Pareto convexe a été obtenu. Les résultats ont été comparés à ceux obtenus dans une étude précédente utilisant une fmGA à objectif unique (SO fmGA) (Michaud et al., 2001). Pour ce faire, pour chaque vecteur du front de Pareto obtenu les deux valeurs objectives ont été ajoutées pour obtenir une valeur unique. Ensuite, la meilleure valeur objective trouvée a été comparée à la valeur unique obtenue par le SO fmGA. Pour [Met] -Enkephelin, le MO fmGA a trouvé la meilleure solution, tandis que pour la polyalanine, le MO-fmGA se compare favorablement par rapport au SO fmGA.

Dans (Boisson, 2008) une étude sur différentes nouvelles modélisations multiobjectif possibles pour un problème du domaine de l'analyse structurelle des molécules et de leurs interactions : le docking moléculaire flexible.

III.5) Problèmes inverses

Ce sont des problèmes dans lesquels nous avons certaines informations qui ont été générées par un processus biologique et notre objectif est de déduire le système d'origine en utilisant ces informations disponibles. (Poladian and Jermiin, 2006) ont proposé d'utiliser une approche évolutionnaire multiobjectif pour inférer des arbres phylogénétiques intégrant de nombreux types de données disponibles. Comme le soulignent les auteurs, les algorithmes évolutionnaires multiobjectif sont particulièrement adaptés pour obtenir l'inférence phylogénétique pour trois raisons : le grand espace combinatoire associé avec toutes les phylogénies possibles, les résultats contradictoires obtenus en utilisant différents ensembles de données et le fait qu'un seul meilleur arbre peut ne pas révéler toute l'histoire, mais un des meilleurs arbres peut également révéler des informations sur la relation entre deux espèces. Ainsi, cette approche multiobjectif donne une famille d'arbres au lieu d'un arbre unique obtenu par une analyse combinée. Dans cette formulation, chaque objectif du problème correspond à la maximisation de la vraisemblance (likelihood) de l'arbre étant donné un type d'information. La méthode a été appliquée à un problème simple à quatre espèces en utilisant deux ensembles de données. Les auteurs ont conclu que l'inspection visuelle du front de Pareto résultant aidera le biologiste expérimenté à interpréter le conflit entre les ensembles de données et à décider d'un plan d'action. De plus, avec une approche multiobjectif, le praticien n'a pas besoin de déterminer à priori l'importance relative des données.

L'inférence de réseaux de régulation des gènes est un autre type de problèmes inverses. Certains produits génétiques déterminent où, quand et dans quelle mesure un autre gène est exprimé en protéines. Ainsi, les processus cellulaires tels que la croissance, la différenciation et la reproduction cellulaires sont le résultat d'interactions complexes entre les gènes. Les réseaux de régulation des gènes sont utilisés pour représenter ces interactions entre les gènes à l'aide d'un graphe dirigé. La tâche du bioinformaticien est de modéliser ces réseaux à partir de grandes quantités de données de puces à ADN.

IV) Pistes prometteuses pour de futures recherches

Les AEMO ont été appliqués à différents problèmes de bioinformatique. Cependant, il existe d'autres voies possibles pour de futures recherches qui méritent d'être explorées (Jaimes & Coello Coello, 2008) permettant d'améliorer les résultats obtenus. Par exemple l'utilisation d'approches hybrides, l'incorporation des préférences de l'utilisateur, l'utilisation de la connaissance du domaine.

- **Utilisation d'approches hybrides** : l'utilisation de techniques hybrides peut présenter des avantages plus importants pour résoudre des problèmes multiobjectif survenant en biologie. Dans (Boisson, 2008) une modélisation multiobjectif par des algorithmes génétiques hybrides pour le docking moléculaire flexible un problème du domaine de l'analyse structurelle des molécules et de leurs interactions. Dans (Sharma and Rani, 2019) une approche hybride multiobjectif est proposée pour la sélection de gènes en utilisant deux métaheuristiques puissantes, SSA (Salp Swarm Algorithm) et MOSHO (multi-objective spotted hyena optimizer). La technologie des puces à AND est devenue un outil puissant pour la détection et la prévention précoces du cancer. Il aide à fournir une vue d'ensemble détaillée du micro-environnement complexe de maladies. De plus, la disponibilité en ligne de milliers d'analyses d'expression génique a fait de la classification des données de puces à ADN un domaine de recherche actif. Un objectif commun est de trouver un sous-ensemble minimum de gènes et de maximiser la précision de la classification. Quatre classificateurs différents sont formés sur sept ensembles de données de haute dimension en utilisant un sous-ensemble de caractéristiques (gènes), qui sont obtenus après l'application

de l'algorithme de sélection de gène hybride proposé. Les résultats montrent que la technique proposée surpasse considérablement les techniques de pointe existantes.

- **Incorporation des préférences de l'utilisateur** : La plupart des AEMO sont couramment utilisés sous l'hypothèse que l'ensemble optimal de Pareto est nécessaire. Cependant, dans la plupart des applications pratiques, toutes les solutions ne sont pas nécessaires, car les utilisateurs identifient normalement les régions d'intérêt dans le front de Pareto (Handl & Knowles, 2007). Il existe plusieurs manières d'incorporer les préférences de l'utilisateur dans un AEMO de telle sorte que la recherche se limite à une certaine partie du front de Pareto (Coello Coello et al., 2007), (Rachmawati & Srinivasan, 2006), (Wang & Terpenny 2005), (Branke & Deb, 2005), (Cvetkovic & Parmee 2002) et (Coello Coello, 2000).
- **Utilisation de la connaissance du domaine** : l'incorporation de la connaissance du domaine peut améliorer les performances des AEMO adoptés pour résoudre des problèmes complexes. Ces connaissances peuvent être fournies soit a priori, soit peuvent être extraits lors de la recherche (Landa & Coello Coello, 2006). Cette connaissance peut influencer les opérateurs d'un AEMO ou peut être utilisée pour concevoir des procédures heuristiques visait à réduire la taille de l'espace de recherche.

V) Conclusion

Dans ce chapitre, nous avons souligné la large applicabilité de l'optimisation multiobjectif dans les domaines de problèmes biologiques. Les gains de performance et la flexibilité offerte par l'optimisation multiobjectif a été illustré dans l'état de l'art établi. L'intérêt des bioinformaticien pour l'utilisation des AEMO augmente, nous avons exploré l'utilisation des AEMO dans différents applications en bioinformatique. Cependant, l'optimisation multiobjectif dans le domaine de la bioinformatique demeure un champ de recherches actif. L'utilisation d'approches hybrides, l'incorporation des préférences de l'utilisateur, l'utilisation de la connaissance du domaine, sont des voies permettant d'améliorer les résultats obtenus.

I) Introduction

De nombreux problèmes du monde réel dans la plupart des disciplines (y compris la Bioinformatique) impliquent l'optimisation simultanée de plusieurs objectifs concurrents (l'amélioration de certains objectifs implique la détérioration d'autres). Ces problèmes sont appelés multiobjectif et leur résolution comporte un très grand nombre de solutions, dont chacune représente le compromis entre les différents objectifs. Le but final est de trouver l'ensemble des solutions optimal connu sous le nom « l'ensemble Pareto-optimal ». L'image de cet ensemble dans l'espace objectif est appelé « le front Pareto-optimal ». Pendant longtemps, des approches classiques qui transforment les objectifs multiples en un seul objectif ont été utilisées pour résoudre ces problèmes. Elles comprennent les méthodes d'agrégation (Ishibuchi & Murata, 1998), la méthode de contrainte epsilon (Haimes et al., 1971), la méthode de programmation par buts (Charnes et al., 1955), etc. Ces techniques présentent plusieurs limites. Ils sont très sensibles à la forme ou à la continuité du front de Pareto et ont tendance à générer un seul élément de l'ensemble optimal de Pareto par exécution (Coello Coello, 2017). Il devrait être préférable d'optimiser les fonctions objectifs simultanément pour pouvoir capturer en même temps différents aspects de la qualité des solutions. L'approche Pareto, définie à l'origine en économie au XIXe siècle (Pareto, 1896), traite les objectifs de manière équitable (sans favoriser un objectif par rapport à un autre) et peut fournir un ensemble de solutions de compromis en une seule exécution de l'algorithme. Vilfredo Pareto formule le concept que « dans un problème multiobjectif, il existe un équilibre tel qu'il n'est pas possible d'améliorer un critère sans détériorer au moins un des autres critères ».

L'optimisation multiobjectif de Pareto est basée sur le concept de dominance (définie dans l'espace objectif), qui est une relation entre deux solutions. Si la solution A est meilleure que la solution B sur un objectif et mauvaise sur un autre, cela signifie que A et B ne sont pas dominées l'une par rapport à l'autre (incomparables). Si la solution A n'est pas inférieure à la solution B pour tous les objectifs, mais qu'elle est meilleure pour au moins un objectif, cela signifie que la solution B est dominée par A. Une solution appartient au front de Pareto si et seulement si elle n'est pas dominée par toute autre solution. Toutes les solutions non dominées (le front de Pareto) fournissent des informations utiles sur les relations de compromis entre des objectifs contradictoires et permettent à un décideur d'envisager plusieurs alternatives (Milajić et al., 2016). Si le choix de l'approche de Pareto est confirmé, la méthode d'optimisation à mettre en œuvre dépendra de la complexité du problème. Ces méthodes peuvent être classées en méthodes exactes et heuristiques. La complexité temporelle des méthodes exactes rend son utilisation impraticable pour les grandes instances. Les heuristiques ou métaheuristiques fournissent un front de Pareto approximatif pour les grandes instances, dans un temps de calcul raisonnable.

Les algorithmes hybrides peuvent faire bon usage des caractéristiques de différents algorithmes pour obtenir des avantages complémentaires afin d'améliorer les performances et l'efficacité optimales de l'algorithme (Ramadan et Zeineldin, 2014). La plupart des méthodes coopératives ont été essentiellement réalisées entre différentes heuristiques. Cependant, des méthodes exactes peuvent être utiles pour résoudre de grandes instances en hybridant la résolution exacte des sous-problèmes (petites instances) et la résolution heuristique du problème global.

Dans ce projet, nous essayons de profiter (dans la mesure du possible) de l'avantage des méthodes exactes afin d'améliorer la précision de l'alignement multiple de séquences. L'algorithme proposé est un nouveau schéma d'hybridation collaboratif, combinant Mémétique NSGA-II (M-NSGA-II) comme méthode Pareto de recherche globale et l'algorithme de Needleman et Wunsch (NW) comme méthode mono-objectif exacte. NW qui est une méthode de comparaison, coûteuses en temps, elle sera appliquée sur des sous-ensembles de séquences à aligner. L'algorithme M-NSGA-II

incorpore une méthode de recherche locale dans l'algorithme NSGA-II. Nous avons conçu une méthode de recherche locale qui fonctionne sur les positions des gaps pour améliorer tous les descendants produits par NSGA-II. La méthode proposée nommée Needleman-Wunsch Memetic Non-dominated Sorting Genetic Algorithm (NW-M-NSGA-II) a été testée sur Balibase et comparée avec les autres méthodes de la littérature.

II) Alignement multiple de séquences (MSA)

Du point de vue biologique, Les gènes sont normalement transmis d'une génération à une autre sans aucun changement (ce principe de conservation est dû à des contraintes fonctionnelles ou structurales). Les résidus (Acide Nucléique ou Acide Aminé) essentiels aux fonctions sont moins sujets à mutation, ces régions fonctionnelles sont relativement préservées car des mutations (substitution, indel) trop radicales sont catastrophiques. Cependant, des mutations ont lieu parfois induisant une modification dans l'information génétique qui peut être transmissible à la descendance, engendrant ainsi, la diversité entre individus (l'évolution). Mais elles sont aussi à l'origine des maladies génétiques et des prédispositions aux maladies (mutation pathogène). La conséquence d'une mutation dépend de son effet fonctionnel, qui peut être :

- neutre (le gène muté n'entraîne pas de changement dans la structure de la protéine),
- conduire à l'amélioration d'une fonction (diversité, évolution)
- ou à l'altération d'une fonction (effet pathogène).

Donc, la séquence d'acides aminés d'une protéine porte des informations sur la structure et la fonction de la protéine. Cependant, il est difficile de prévoir des résidus importants pour la structure et la fonction à partir d'une seule séquence (méthodes de novo). Un moyen efficace d'extraire de telles informations est la comparaison de séquences homologues, puisque les protéines partageant un ancêtre commun ont souvent une structure et une fonction similaires. Autrement dit, les résidus critiques pour la fonction ou la structure ont été conservées dans des protéines homologues au cours de l'évolution moléculaire.

Ce sont ces phénomènes que les algorithmes d'alignement multiple essayent de modéliser et analyser pour produire de nouvelles connaissances. Le but est d'extrapoler des données obtenues de façon expérimentales sur certaines séquences à d'autres séquences pour lesquelles aucune donnée expérimentale n'est disponible. En d'autres termes, nous pouvons prédire les résidus ou les régions d'alignement sous de fortes contraintes en comparant les séquences d'acides aminés de protéines homologues et en identifiant les régions d'alignement conservés.

Du point de vu bioinformatique, l'alignement multiple de séquences vise à organiser trois ou plusieurs séquences d'ADN, d'ARN ou de protéines de manière à mettre en évidence leurs similitudes. À cette fin, les séquences sont posées les unes sur les autres en ajoutant des gaps afin qu'un modèle de score qui évalue la qualité d'un alignement en fonction du nombre de non-appariements (substitutions ou mismatch), d'appariements (identités ou match) et de gaps (insertions et délétions ou indel) est optimisé. Le modèle de score nécessite la définition d'une matrice de substitution appropriée et d'une pénalité des gaps, qui indiquent respectivement les récompenses pour l'alignement de deux caractères quelconques de l'alphabet et la pénalité pour l'insertion de gaps, respectivement. L'alignement multiple de séquences a plusieurs applications importantes en bioinformatique :

- Il permet d'identifier des sites fonctionnels qui correspondent en général aux régions les plus conservées.
- Il est utilisé pour prédire la structure ou la fonction d'une protéine, si on détecte une homologie avec une protéine de structure ou de fonction déjà connue. la modélisation par homologie de la structure 3D potentielle d'une protéine pour laquelle on ne dispose que de la séquence, consiste à disposer d'au moins une protéine dont la structure 3D est connue qui sert de

"modèle". la séquence étudiée doit bien sûr être proche (homologue) de celle de la protéine modèle. Il faut donc d'abord effectuer des alignements de séquences pour tester la ressemblance (un fort taux de similarité est une indication forte de l'existence d'une homologie).

- Il est utilisé comme point de départ pour les analyses phylogénétiques. Les développements sur notre compréhension des virus sont essentiellement dus à l'étude de la phylogénie (identifier les souches d'un virus et leur origine commune). Par exemple les chercheurs ont pu relier le HIV aux SIV grâce à des études phylogénétiques
- L'identification d'une protéine proche dont la structure est connue d'une autre protéine impliquée dans une maladie afin de concevoir un médicament, …
- Egalement dans le processus d'annotation des séquences.

Donc, plusieurs méthodes de prédiction (structure et fonction), d'inférence (phylogénie), d'identification et d'annotation sont basées sur l'alignement multiple de séquences. La qualité d'un MSA influence fortement le résultat de ces prédictions. Malgré le fait qu'il existe de nombreuses méthodes MSA, des approches d'alignement biologiquement parfait ne sont pas trouvées jusqu'à présent. Il est important de développer des heuristiques permettant de fournir une précision efficace et des résultats d'alignement statistiquement significatifs.

L'optimisation multiobjectif suggère une meilleure façon de résoudre le problème d'alignement (Chowdhury & Garai, 2017). *Dans ce projet, nous avons adopté une approche multiobjectif Pareto, pour résoudre le problème d'alignement multiple de séquences protéique. Les deux fonctions objectifs contradictoires utilisées sont le score de substitution à maximiser et le coût des gaps à minimiser. Une méthode efficace doit établir un bon compromis entre la convergence vers la frontière Pareto et la répartition des solutions le long de la frontière Pareto.*

II.1) Etat de l'art (MSA multiobjectif)

Dans la littérature plusieurs méthodes d'optimisation multiobjectif pour l'alignement multiple de séquences ont été proposées : MOMSA (Seeluangsawat and Chongstitvatana., 2005) introduit un algorithme évolutionnaire multiobjectif pour améliorer les solutions obtenus par Clustal X. Les auteurs ont utilisé deux fonctions objectifs (la fonction de récompense et la fonction de pénalité). La première fonction objectif a été affecté par la valeur GOP>GEP et la deuxième fonction objectif a été affecté par la valeur GOP<GEP. Cofolga2mo (Teneda, 2010) est un programme d'alignement de séquence d'ARN par paires basé sur un algorithme génétique multi-objectif (MOGA). Deux fonctions contradictoires optimisées simultanément la similarité de séquence et la structure secondaire. AlineaGA (Mateus da Silva et al., 2011) est une approche Pareto évolutionnaire biobjectif pour l'alignement multiple de séquences. Les deux objectifs à optimiser sont la somme des paires SP et l'identité (le nombre de colonnes entièrement identiques dans l'alignement). Le coût d'alignement par paires de chaque acide aminé est déterminé par la matrice de notation PAM 350. Une pénalité de gap de -10 est appliquée quand un acide aminé est aligné avec un gap. Dans (Ortuno et al., 2012) un algorithme évolutionnaire multiobjectif basé sur NSGA-II est implémenté afin d'optimiser trois scores différents pour évaluer chaque alignement : le score PAM250, le pourcentage de non-gaps et le pourcentage de colonnes complètement alignées. MO-SAStrE (Ortuno et al., 2013) algorithme multiobjectif basé sur l'algorithme génétique de tri non dominé (NSGA-II), vise à optimiser conjointement trois objectifs: score STRIKE, pourcentage de non-gaps et colonnes totalement conservées. Dans (Rubio-Largo et al., 2015) une métaheuristique multiobjectif mémétique hybride est présentée pour l'alignement de séquences multiples. jMetalMSA (Zambrano-Vega et al., 2017 a) est un outil logiciel open source pour résoudre le problème MSA avec les métaheuristiques multiobjectifs. Il maximise simultanément la fonction de somme des paires pondérée avec des

pénalités de gaps affines (WSP) et le score du nombre de colonnes totalement conservé. M2Align (Zambrano-Vega et al., 2017 b) est une version parallèle et plus efficace de l'optimiseur à trois objectifs MO-SAStrE (Ortuno et al., 2013), capable de réduire le temps de calcul de l'algorithme. Dans (Zambrano-Vega et al., 2017 c) les auteurs considèrent une formulation à trois objectifs de MSA, qui comprend le score STRIKE, le pourcentage de colonnes alignées et le pourcentage de symboles sans gap. Une étude comparative rigoureuse en utilisant quatre métaheuristiques multiobjectifs : NSGA-II, MOCell, GWASF-GA et NSGA-III conclut que NSGA-III offre les meilleures performances globales. BiMuSA (Schenker & Paquete, 2013) une implémentation pour résoudre les problèmes d'alignement biobjectif à séquences multiples. Le problème biobjectif considéré dans ce travail consiste à minimiser le nombre d'indel et à maximiser le nombre de correspondances moins le nombre de mésappariements. MOSAL (Paquete et al., 2014) un outil logiciel qui fournit une implémentation open-source et une application en ligne pour l'alignement de séquences par paires multiobjectifs. Dans (Zhu et al., 2015) Un célèbre cadre d'algorithme évolutif multiobjectif basé sur la décomposition est appliqué pour résoudre MSA.

II.2) les méthodes proposées

DM-NSGA-II (Deep memetic NSGA-II) (Mahdi & Nini, 2021) est un algorithme multiobjectif hybride conçu pour améliorer les résultats de l'algorithme mémétique NSGA-II en résolvant le problème académique du sac à dos multiobjectif. Il combine les avantages des deux approches exacte et heuristique. Le NSGA-II effectue une exploration globale de l'espace de recherche pour trouver différentes régions avec des solutions de bonne qualité. L'algorithme de recherche locale HCFI-LS (Hill-Climbing First-Improvement Local Search) améliore chaque descendant produit par NSGA-II en explorant son voisinage. L'algorithme exact Branch & Bound de type Pareto (B&B-PLS) améliore les performances de l'algorithme mémétique NSGA-II, en appliquant une recherche locale approfondie sur le voisinage de certains points du front de Pareto actuel, afin d'obtenir plus de solutions non-dominées uniformément réparties et mieux convergées du front Pareto-optimal.

Nous allons nous inspirer de cette technique pour traiter le problème MSA multiobjectif. Dans ce cas l'algorithme principal est le NSGA-II renforcé par une heuristique spécifique conçue pour améliorer chaque nouvel individu généré par NSGA-II. Cette heuristique de recherche locale est basée sur le déplacement des gaps à d'autres positions dans la même séquence en gardant la même longueur de l'alignement. Pour la méthode exacte, nous allons utiliser l'algorithme Needleman-Wunsch qui doit être de nature mono-objectif (compte tenu des spécificités du problème MSA).

II.2.1) Formulation mathématique du problème MSA

Étant donné S un ensemble de n séquences $S = \{S_1, S_2, \ldots S_n\}$ ($n \geq 3$) dans le même alphabet Σ et de longueur différentes $|S_i|$ (i=1..n).

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_n \end{bmatrix}$$

$S_i = s_{i1}\, s_{i2} \ldots s_{iL_i}$ est la séquence i de longueur $|S_i|=L_i$ sur l'alphabet Σ.

L'alignement multiple (une représentation matricielle) consiste à insérer S dans une matrice MSA de dimension (n x L) où le nombre de lignes est fixe et égal au nombre de séquences n et le nombre de colonnes L est variable, en introduisant des espaces (gaps) entraînant des décalages. L'objectif est de maximiser le nombre de caractères en commun dans la même colonne.

Initialement, le nombre de colonnes L est égal à la longueur de la plus longue séquence de S. $L = \max(|S_i|)_{i=1..n}$. Pour toute séquence $S_i \in S$ de longueur $L_i < L$, m gaps y sont ajoutés ($m = L - L_i$) i=1..n.

A un moment donné la matrice *MSA* est composée par *n* vecteurs $S'_1, S'_2 \ldots S'_n$ de même longueurs *L* obtenus par insertion d'espaces (gaps " - ") dans $S_1, S_2 \ldots S_n$:

$S'_i = s_{i1}\, s_{i2} \ldots s_{iL}$ est une séquence de longueur $|S'_i| = L$ sur l'alphabet $\Sigma \cup \{-\}$.

$$MSA = \begin{bmatrix} s_{11} & \cdots & s_{1L} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nL} \end{bmatrix}$$

Avec :

- $\forall i, \nexists j$ tel que $s_{ij} = -$ (pas de colonnes avec uniquement des gaps, il apparait au moins une lettre.).
- $\max |S_i| \leq L \leq \sum_{i=1}^{n} |S_i|$. dans la pratique les solutions aux problèmes d'alignement contenaient rarement plus de 50% de gaps par rapport à la plus longue séquence (Rubio-Largo et al., 2017). Donc la limite pratique de *L* est très inférieure à $\sum|S_i|$ (qui reste une limite théorique).
- conservation des séquences initiales en supprimant les gaps.
- On ne peut jamais changer l'ordre relatif de deux caractères dans une séquence.

Aligner plusieurs séquences homologues revient donc à maximiser la conservation entre des séquences par l'introduction de *gaps*. La résolution par une méthode de score du problème d'alignement multiple soulève trois questions fondamentales :

- Le choix de l'ensemble de séquences à aligner.
- Le choix d'une fonction objectif permettant d'évaluer la qualité d'un alignement. Obtenir l'alignement de score optimum qui soit l'alignement le plus biologiquement significatif.
- Le choix d'une stratégie de recherche (trouver un bon alignement en un temps raisonnable).

II.2.2) le choix de l'ensemble de séquences à aligner

Une des questions qui se posent au biologiste lorsqu'il compare des séquences est de savoir sur quel matériel il doit travailler : ADN ou protéine ? Cette question a été abordée dans le chapitre (II) section (III.1.3.1) où il est indiqué qu'il est préférable de comparer les séquences au niveau protéique. Dans cette thèse, nous nous intéressons au processus d'alignement global d'un groupe de séquences de protéines pour obtenir un alignement de séquences multiples (MSA) significatif.

Pour le choix des séquences à aligner, c'est un problème typiquement biologique. C'est au biologiste de déterminer quel ensemble de séquences faut-il aligner (selon l'objectif de l'étude). MSA est un outil très fort utilisé pour répondre pratiquement à la plupart des problèmes posés par la biologie moléculaire. Il y a donc, différentes raisons et objectifs pour effectuer un MSA. Le choix donc, des séquences à aligner (tâche du biologiste) doit prendre en compte des objectifs de l'étude et de la complexité du problème. *Dans cette thèse, nous avons choisi MSA sur des séquences protéiques de Balibase.*

II.2.3) Le choix des fonctions objectifs

Avoir un bon alignement de séquences multiples aidera les biologistes à trouver les bonnes réponses à de nombreux problèmes. C'est le rôle de la fonction objectif d'évaluer la qualité d'un alignement. Elle permet de calculer un score (coût) attribuée à chaque alignement qui doit mesurer sa qualité (le savoir biologique que cet alignement nous apporte). De façon idéale, une fonction doit assigner à un alignement optimal un score traduisant l'intérêt biologique de l'information qu'il contient. Plusieurs fonctions objectifs ont été proposées tel que la somme des paires SP, T-COFFEE, le score profil, etc. Malheureusement, nous ne connaissons pas encore, une fonction objectif

mathématique capable de calculer la réalité biologique d'un alignement. Le seul moyen utilisé pour tester l'efficacité biologique des méthodes d'alignement est l'utilisation des bases d'alignements de références (benchmarks) comme BaliBase (Thompson et al., 1999). Ces dernières contiennent des familles de séquences dont l'alignement multiple optimal (du point de vue biologique) est connu et généralement crée à la main.

Une bonne fonction objetif doit mesurer la qualité d'un alignement en tenant compte des connaissances biologiques. Dans cette perspective, les chercheurs ont développé des systèmes de scores améliorant la fiabilité de la fonction objectif du point de vue biologique. Dans ce cas, le score de chaque paire d'acides aminés est extrait d'une matrice de substitution basée sur la nature physico-chimique des acides aminés ou déduite empiriquement de l'évolution moléculaire. Ces matrices sont généralement construits en observant les mutations dans de grands ensembles d'alignements, soit basés sur des séquences, soit sur des structures. Par exemple, le savoir biologique de la similarité selon le modèle évolutionnaire est résumé implicitement d'une manière empirique dans les matrices de substitution Blosum. Le score d'un alignement dépend donc, de la matrice de substitution (contenant les récompenses de similarité) et la pénalité des gaps, en générale combiné et calculé dans une seule fonction objectif (comme dans la fonction SP).

Le problème d'alignement multiple de séquences a été étudié par de nombreux chercheurs, comme un problème d'optimisation mono-objectif en utilisant la fonction objectif SP (sum of pairs) pour évaluer la qualité de l'alignement (la solution). Rien ne garantit que la seule solution optimale obtenue possède un sens biologique. Donc, il est souvent préférable de lister plusieurs solutions intéressantes. L'approche multiobjectif qui a la possibilité d'obtenir plusieurs solutions en une seule exécution permet d'atteindre cet objectif. En plus, les solutions du front de Pareto obtenues sont incomparables entre eux (i.e. tous les individus sont candidats pour une solution biologiquement significative).

Une façon naturelle d'évaluer la qualité des alignements consiste à tenter de générer autant d'identités que possible tout en limitant le nombre de gaps. Autrement dit maximiser la similarité et minimiser le nombre de gaps (il est claire que ces deux objectifs sont contradictoires). Comme il y a un compromis entre ces deux objectifs, souvent nous ne pouvons pas obtenir une solution unique qui a à la fois le meilleur score de similarité de séquence et le meilleur score des pénalités simultanément. Au lieu d'une fonction objectif scalaire combinant toutes les valeurs de récompense et de pénalité, nous utilisons des objectifs à optimiser simultanément et qui sont contradictoires. Toute amélioration d'un objectif peut détériorer l'autre objectif. Par exemple, un MSA avec un meilleur score de similarité peut contenir un très grand nombre de gaps, ce qui est biologiquement insignifiant ; ou peut contenir des nombres inférieurs de colonne de conservation qui est également indésirable. Ainsi, obtenir un seul alignement ayant tous les objectifs optimisés simultanément est impossible.

L'aspect contradictoire des objectifs permet de parvenir à un compromis (un équilibre) qui peut améliorer la qualité (biologique) des solutions (On sait bien que, c'est l'insertion des gaps qui permet de maximiser la similarité, si on insert des gaps sans contrainte on peut obtenir une meilleur solution mathématique, mais elle peut mener à un contre sens biologique). Optimiser simultanément ces deux objectif peut parvenir à un équilibre améliorant la qualité biologique de MSA.

$$f_1 = max \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sc(S_i, S_j)$$

$$f_2 = \min \sum_{i=1}^{n} coût\_gaps\ (S_i)$$

$$sc(S_i, S_j) = \sum_{k=1}^{m} sc(a_k, b_k)$$

$$coût\_gaps(S_i) = \sum_{r} (po + l * pe)$$

- n : le nombre de séquence,
- S_i, S_j : deux séquences,
- $sc(S_i, S_j)_{i \neq j}$: le score attribué à chaque couple de séquences aligné.
- m : la taille de l'alignement.
- a_k, b_k : la paire d'acides aminés mis en correspondance des deux séquences respectivement S_i et S_j.
- $sc(a_k, b_k)$: le score élémentaire entre deux (acides aminés ou nucléotides) résultant de l'alignement de la lettre a_k en face de la lettre b_k, il est attribué par une matrice de similarité ou de substitution. *sc(-,-) = 0, sc(a,-) = sc(-,a) =* -1 et *sc(a, b) = sc(b, a)= Mat(a,b).*
- Il y a $\frac{k(k-1)}{2}$ paires par scores à calculer pour chaque colonne.
- Le coût des gaps de la séquence S_i est donné par la somme des coûts de tous les ilots de gaps formés dans la séquence S_i, calculé par la formule de pénalité affinée (*po + l * pe*). Cette pénalité affinée permet de favoriser un large gap plutôt que de nombreux petits (c'est une réalité biologique).
- *l est la* longueur d'un ilot de gaps contigu.
- *r* est le nombre d'ilots de gaps dans la séquence S_i. Par exemple la séquence acg---ttac-ccta--aa contient 3 ilots, le premier est de longueur *l* = 3, le deuxième est composé d'un seul gap (*l* = 1) et dans le troisième *l* = 2. Donc, *Coût*_gaps (acg---ttac-ccta--aa) = (*po+3*pe*) + (*po+1*pe*) + (*po+2*pe*).
- *po* est la pénalité fixe d'ouverture indépendante de la longueur.
- *pe* est la pénalité fixe d'extension (souvent *po* = 10 $*$ *pe* et *pe* =1).

Dans ce travail, nous allons utiliser les matrices de substitution BLOSUMx (x=30, 62 ou 100) selon l'instance du problème étudié. Dans notre formulation multiobjectif, l'ensemble Pareto contient la meilleure solution d'une formulation mono-objectif de la somme des paires mais aussi de nombreux autres alignements qu'il n'est pas possible de trouver du tout par l'approche mono-objectif.

L'autre probléme de l'alignement multiple est liée à l'optimisation de ces fonctions objectifs. Le calcul mathématique de l'alignement multiple appartient à la classe des problèmes NP-difficile. En conséquence, toutes les méthodes courantes d'alignement multiple sont des méthodes approchées et aucune d'elles ne peut garantir la solution optimale.

II.2.4) Le choix d'une stratégie de recherche

De nombreux algorithmes qui effectuent la tâche d'aligner un groupe de séquences protéiques existent, et de nouveaux outils sont constamment développés et publiés (problème qui reste encore ouvert). L'approche exacte détermine un alignement optimal, mais elle ne peut être utilisée que pour des séquences de petites tailles. Les méthodes progressive, sont reconnues d'être très rapides et donnent des résultats assez satisfaisants mais leur inconvénient est le fait de s'arrêter sur les minima locaux et si une erreur est commise au début de l'alignement, elle va se propager sur l'alignement final. L'approche itérative est une manière très simple et efficace permettant d'améliorer des méthodes d'alignement multiples (Notredame, 2002). L'inconvénient majeur est le temps d'exécution élevé par rapport aux méthodes progressive.

Les MSA à base de score, partent du postulat que les meilleurs résultats du point de vue statistique sont aussi les plus significatifs du point de vue biologique (la motivation derrière ce

principe est la matrice de substitution utilisée dans le système de score). Or ce n'est pas toujours le cas car des résultats biologiquement intéressants peuvent être non significatifs sur un plan statistique. L'utilisation d'une seule fonction objectif permettant d'assigner un score à chaque alignement, et de fournir comme résultat le seul alignement du meilleur score, peut ne pas intéresser le biologiste. Alors que l'approche multiobjectif utilisant deux ou trois fonctions antagonistes traitées de manière équitable, fournissant un ensemble d'alignements incomparables entre eux, devrait être préférable pour le biologiste.

NSGA-II est un AEMO qui a été démontré comme l'un des algorithmes les plus efficaces et les plus célèbres pour l'optimisation multiobjective, cependant, la capacité de convergence de NSGA-II est limitée (Zhang et Ma, 2015). Par conséquent, il est impératif de l'améliorer en augmentant sa vitesse de convergence et en améliorant sa précision de résolution (Xiaoyun & Yi, 2018). Pour améliorer les performances des algorithmes évolutionnaires, Moscato (1989) a introduit le concept d'algorithmes mémétique, en combinant la capacité de recherche globale de l'algorithme évolutionnaire et la méthode d'optimisation de recherche locale (LS). Il s'avère que c'est un moyen efficace de résoudre l'optimisation multiobjectifs (Fang et al., 2018). Les AEMO Memetic peuvent accélérer la convergence et obtenir un front de Pareto approximatif de haute performance (Gong et al., 2016).

Pour répondre à ces objectifs, nous avons conçu une méthode de recherche locale qui traite les gaps comme un caractère spécial afin d'améliorer l'alignement. Généralement, l'événement d'insertion/délétion (gap) joue un rôle très important dans le MSA, et les gaps mal placés sont un désastre pour la signification biologique. Cette méthode nommée recherche locale pour le placement des gaps GPLS (Gaps Placement Local Search) incorporée dans l'algorithme NSGA-II pour constituer l'algorithme Mémétique NSGA-II (M-NSGA-II). GPLS est appelé pour améliorer chaque descendant produit par NSGA-II.

Pour augmenter la précision de certains alignements produits par M-NSGA-II, nous allons appliquer l'algorithme exact de Needleman et Wunsch (NW) sur des sous-ensembles de quelques alignements du front de Pareto actuel. L'algorithme proposé nommé Needleman-Wunsch Memetic Non-dominated Sorting Genetic Algorithm (NW-M-NSGA-II) est un nouveau schéma d'hybridation collaborative.

III) NW-M-NSGA-II (méthode proposée)

La méthode NW-M-NSGA-II est composée de l'algorithme M-NSGA-II appliqué sur tout l'espace de recherche (problème complet) et de l'algorithme NW (Needleman et Wunsch) appliqué sur des petits sous-espaces de recherche (sous-problèmes).

III.1) M-NSGA-II

Le M-NSGA-II proposé (figure. 26) incorpore une heuristique de recherche locale GPLS dans NSGA-II. La méthode GPLS tente de trouver les emplacements appropriés des gaps pour améliorer les nouveaux individus produits par NSGA-II.

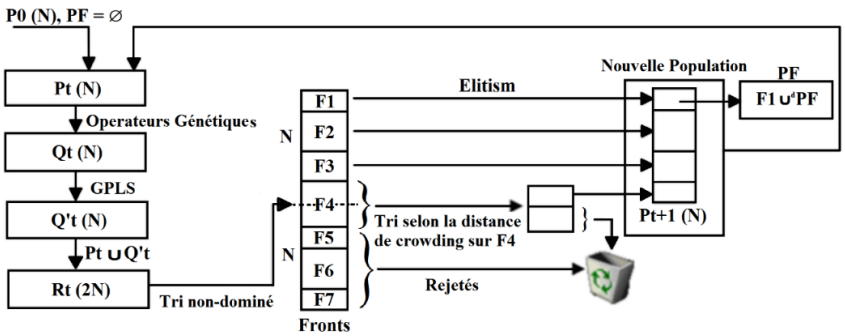


Figure. 26 M-NSGA-II

a) critère d'arrêt de la recherche

Un bon critère d'arrêt permet à l'algorithme de révéler sa capacité à poursuivre la recherche tout en améliorant les résultats. La spécificité du problème d'alignement de séquences nous a imposé de choisir un critère d'arrêt reflétant le sens biologique du problème. Ce critère sera déterminé par le nombre maximum de longueur d'alignement atteint par l'insertion de gaps (opération de mutation). L'opérateur de mutation augmente la taille de l'alignement en ajoutant un gap supplémentaire dans chaque séquence après un certain nombre de générations successives sans amélioration *nbrg*. Ainsi de suite on incrémente la taille jusqu'à satisfaction du critère d'arrêt défini par la longueur d'alignement maximale (le nombre maximum de colonnes) :

$$longmax = \frac{5}{4}\max(|S_i|)$$

Cela permet de calculer à l'initialisation le nombre maximum de gaps autorisés par rapport à la séquence la plus longue de l'ensemble de séquences à aligner. Le choix de 1,25 comme facteur permet à l'alignement d'être 25% plus long que la séquence la plus longue. Ce choix reposait sur le constat que la plus part des alignements de références de Balibase contenaient rarement plus de 25% de gaps dans la séquence la plus longue.

b) La représentation chromosomique d'un alignement

Nous avons appliqué un codage binaire représentant les positions de gaps par 0 et les résidus par 1. Une telle représentation améliore la technique de recherche pour des séquences très longues et permet d'optimiser l'espace mémoire stockant les alignements.

--aaacggct-----a	**0011111111000001**
aaaaact---acgtc-	**1111111000111110**
-------cgcc-g---	**0000000111101000**

c) La population initiale

La procédure de recherche commence par une population initiale P0 de taille N générée aléatoirement. Un alignement aléatoire est obtenu en insérant des gaps au hasard dans les séquences, sauf dans la plus large. Ainsi, la population initiale P0 contiendra N alignements de longueur L = max $|Si|_{i = 1..n}$, (*n* est le nombre de séquences à aligner).

Exemple :

S_1: aaacggcta	**$\|S_1\|$=9**	**aaa-cg--gcta**	**111011001111**
S_2:aaaaactacgtc	**$\|S_2\|$=12**	**aaaaactacgtc**	**111111111111**
S_3:cgccg	**$\|S_3\|$=5**	**---cgc-c--g-**	**000111010010**

L = max | Si |=12, les alignements dans la population P0 doivent avoir la même taille 12. Donc, on insert aléatoirement (12 – 9 = 3) gaps dans S1 et (12 – 5 = 7) gaps dans S3. Pour la séquence S1 on génère 3 nombres aléatoires de [1...12], soient (4, 7, 8) et pour la séquence S3 on génère 7 nombres aléatoires de [1...12], soient (1, 2, 3, 7, 9, 9, 12).

d) Le processus de recherche

A chaque génération t, un nouvel ensemble d'alignements Qt (population de descendants) de taille N est créé à partir de Pt à l'aide d'opérateurs génétiques. La population (Qt) est améliorée par l'heuristique GPLS produisant la population Q't. Une population combinée Rt = Pt ∪ Q't est formée pour obtenir 2N alignements. Rt est alors classé en différents fronts de non-domination (F1, F2…), suivant le principe du tri non dominé (Ranking). Une nouvelle population (Pt+1) est formée en ajoutant les fronts complets (le meilleur front F1 (élitisme) suivi du deuxième front F2 et ainsi de suite) tant que la taille de Pt+1 ne dépasse pas N. Ensuite, une procédure de crowding (d'encombrement) est appliquée sur le premier front suivant non inclus dans (Pt+1), le F4 par exemple. Le but de cette procédure est d'insérer les meilleurs individus de F4 dans la population Pt+1

jusqu'à ce qu'elle atteigne la taille N. Ce processus itératif se poursuit jusqu'à ce que le critère d'arrêt soit satisfait.

- **Ranking et Crowding (mécanisme de sélection)**

NSGA-II intègre un opérateur de sélection, basé sur le Ranking et le crowding. Le Ranking est défini dans l'espace de recherche, utilisé pour estimer la qualité d'un alignement en attribuant une valeur scalaire unique (la fitness), au vecteur des objectifs. Il consiste à classer les individus en leur donnant un rang et d'inclure dans Pt+1 les meilleurs individus (de rangs faibles). La distance de Crowding (surpeuplement**)** est définie dans l'espace de recherche, utilisé pour estimer la densité au voisinage d'un individu. Elle permet d'inclure dans Pt+1 les (N - |Pt+1|) individus de Fi les mieux répartis au sens de la distance de crowding.

Chaque individu *i* de la population a deux attributs : rang de non domination i_{rank} et distance crowding d_i. Un opérateur de comparaison défini en fonction de ces deux attributs permet de guider le processus de la sélection avec la répartition uniforme des solutions Pareto.

Soient deux individus *i* et *j*, on dit que *i* est meilleur que *j* si :

$$[(i_{rank} < j_{rank})] \quad \text{ou} \quad [(i_{rank} = j_{rank}) \text{ et } (d_i > d_j)].$$

Avec cette relation pour la comparaison de deux solutions non dominées appartenant à deux fronts Pareto, on préfère la solution appartenant au front Pareto d'ordre le plus faible. Sinon, dans le cas où les deux solutions appartenant au même front de Pareto (le dernier front pour compléter la taille de la population parent), on choisit la solution qui a la distance crowding la plus élevée. Le tri de Crowding des points de dernier front pour compléter la taille N est pris dans l'ordre décroissant de leurs valeurs de distance crowding, et les points de la partie supérieure de la liste ordonnée sont choisis.

- **Elitisme**

NSGA-II assure qu'à chaque nouvelle génération, les meilleurs individus rencontrés (F1) soient conservés. M-NSGA-II est caractérisé par l'utilisation d'une population externe PF (Initialement PF = ∅) pour stocker les solutions non-dominées. Pour chaque génération t, PF est mis à jour (PF = F1$\cup^d$ PF) en utilisant l'opérateur d'union non dominé ($\cup^d$) qui est basé sur le concept de dominance de Pareto. A$\cup^d$ B est un ensemble contenant toutes les solutions non-dominées les unes par rapport aux autres. Afin de maintenir la validité des solutions, les colonnes qui ne contiennent que des gaps sont supprimées.

- **L'opérateur de sélection pour la reproduction**

L'opérateur de sélection pour la reproduction utilisé, permet de choisir aléatoirement les couples à partir de la population actuelle.

- **L'opérateur de croisement**

Pour garder le principe de conservation des tailles d'alignement dans la même génération, nous avons opté pour le croisement horizontal. Cet opérateur consiste à sélectionner au hasard une ligne parmi les *n* séquences des deux parents pour les diviser en deux groupes. Enfin, les groupes sont échangés entre ces deux parents pour créer deux nouveaux individus (figure. 27).

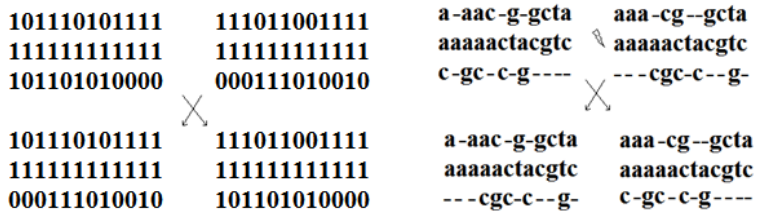


Figure. 27 croisement horizontal

- **L'opérateur de mutation**

L'objectif de l'opération de mutation est de maintenir la diversité au sein de la population, après épuisement de la recherche (exploration et exploitation) dans l'espace de taille *L*. Cet objectif est garanti par l'opérateur d'incrémentation de gap qui permet d'insérer un gap dans tous les alignements de la population Pt, pour obtenir le nouvel espace composé des alignements de taille *L*+1. L'operateur de mutation est effectué après un certain nombre de générations successives sans amélioration *nbrg*.

NB : Après chaque création d'un nouvel individu (par croisement, mutation ou amélioration), une procédure de vérification de la validité de l'alignement est lancée. Cette procédure détecte les colonnes contenant que des gaps et les déplace vers la première colonne (pour assurer le principe de conservation de la taille des alignements dans la population actuelle). Cette colonne ne sera pas comptabilisée dans le score des fonctions objectifs. Afin de maintenir la validité des solutions, les colonnes qui ne contiennent que des gaps sont supprimées avant d'être installées dans le front de Pareto PF.

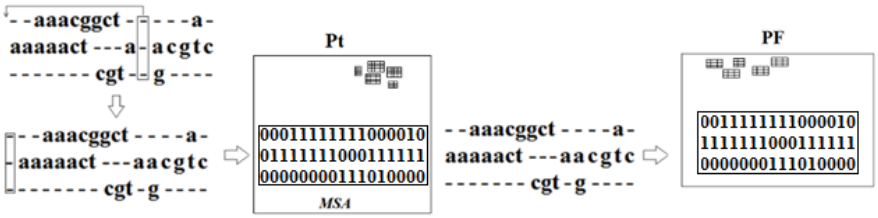


Figure. 28 traitements des colonnes de gaps

e) L'Heuristique GPLS

Le problème capital dans la plupart d'algorithmes d'alignement est le placement inapproprié des gaps dans les séquences. Du point de vue biologique il est moins probable que des gaps de la même taille se produisent à des positions différentes. Dans ce cas, on doit combiner les gaps proches en un seul bloc

RP---CVCPV	RP---CVCPV
RPCACP---V	RP---CACPV
KPCVCPRQLV	KPCVCPRQLV
moins significative	plus significative

Dans la réalité, il est moins probable que deux gaps apparaissent très proches l'un de l'autre.

RP-C--CVPV	RP---CCVPV
RPCACPL-PV	RPCACPL-PV
KPCVCPRQLV	KPCVCPRQLV
moins significative	plus significative

Un autre type de mauvais placement des gaps est la présence des îles de caractères entouré par des gaps.

RPC----CVC----PVC	RPCCVC--------PVC
moins significative	plus significatives

Au niveau de l'évolution, il vaut mieux avoir des gaps groupés.

Le problème de calcul d'un MSA peut être vu (du point de vu biologique) comme un problème d'insertion des gaps aux bons endroits. Pour améliorer le résultat d'un MSA, il faut optimiser non seulement le nombre de gaps à l'intérieur d'un alignement mais aussi leurs positions (qualité biologique). L'heuristique de recherche locale GPLS est conçue pour effectuer ce travail en déplaçant ou en fusionnant les gaps dans les séquences alignées.

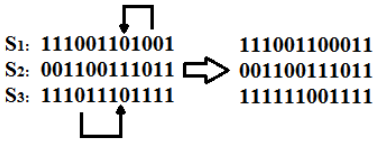
1) *Fusion de gaps*.

Cet opérateur commence par générer un nombre aléatoire (de 1 à *n*) pour désigner le nombre de séquences ($1 \leq nb \leq n$) impliqué dans cette opération. Ensuite il sélectionne les *nb* séquences au hasard pour chaque descendant produit par NSGA-II. Enfin il choisit les gaps les plus proches et les fusionne ensemble dans chaque séquence élue. Le rattachement est effectué à gauche ou à droite.

Exemple : *nb* = 2, S_1 contient 3 blocs de gaps et S_3 contient 2 blocs.

S1: 111001101001
S2: 001100111011
S3: 111011101111

Les deux bl plus proches ($bloc_i$, $bloc_{i+1}$) d'une séquence sont identifiés par le minimum de 1 les séparant. Pour S_1 distance ($bloc_1$, $bloc_2$) =2, distance ($bloc_2$, $bloc_3$) =1, donc on fusionne le $bloc_2$ vers le $bloc_3$ ou l'inverse. Le résultat de la fusion donne l'alignement suivant :



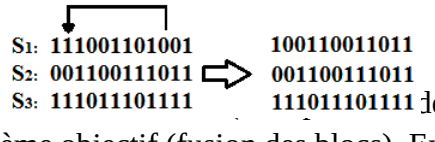
Cet opérateur améliore le deuxième objectif puisque la concaténation des gaps permet de minimiser la pénalité affinée (favorise un large gap plutôt que de nombreux petits). C'est un facteur d'intensification. Si cette nouvelle solution améliore le premier objectif alors elle domine l'ancien alignement, sinon les deux alignements sont incomparables.

2) *Changement de position des gaps*

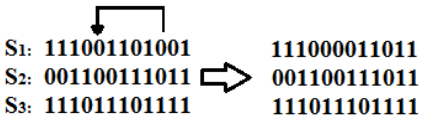
Cet opérateur commence par générer un nombre aléatoire (de 1 à *n*) pour désigner le nombre de séquences ($1 \leq nb \leq n$) impliqué dans cette opération. Ensuite il sélectionne les *nb* séquences au hasard pour chaque descendant produit par NSGA-II. Enfin il sélectionne un bloc de gaps puis une position au hasard. Le bloc de gaps est ensuite déplacé vers la position choisie au hasard. Dans le cas de déplacement proprement dit, cet opérateur garde la même valeur du deuxième objectif, car nous aurons le même nombre de blocs de gaps avec le même nombre de gaps d'extension pour chaque bloc. Mais le premier objectif peut être amélioré (par ce simple déplacement), dans ce cas la nouvelle solution domine l'ancienne.

Exemple : S1: 111001101001
S2: 001100111011
S3: 111011101111
nb=1, S_1, $bloc_3$, position 2.

Le résultat du déplacement donne l'alignement suivant :



Dans le cas de déplace de destination contient un gap) alors cet opérateur améliore le deuxième objectif (fusion des blocs). Exemple (*nb*=1, S_1, $bloc_3$, position 5)



III.2) NW (algorithme de Needleman et Wunsch)

La méthode exacte de programmation dynamique NW réalise un alignement mathématiquement parfait avec une complexité de calcul très élevée (qui augmente exponentiellement avec la taille de l'alignement). Pour cette raison nous allons appliquer cette méthode sur un petit sous ensemble de quelques alignements du front Pareto actuel selon une probabilité μ après épuisement de la recherche dans l'espace d'alignement de taille *L* et avant le passage à l'espace d'alignement de taille *L*+1 par l'operateur de mutation. Le principe général de

l'algorithme consiste à remplir pas à pas une table de gauche à droite et de haut en bas. Pour l'alignement de deux séquences de longueur ℓ_1 et ℓ_2, on utilise une table de taille [ℓ_1+1, ℓ_2+1].

Pour le calcul du score optimal, on utilise la matrice de similarité Blousm*x* (*BM*) et les gaps à coût constant (pour les gaps à coût affine, rien ne permet de savoir s'il s'agit d'une ouverture ou d'une extension de gaps). La fonction objectif utilisée est définie par le système de score *nw*(*x*, *y*) permettant d'estimer l'appariement de deux symboles $x, y \in \Sigma \cup \{-\}$.

$$nw(x,y) = \begin{cases} BM(x,y) & si\ x \neq - \quad et \quad y \neq - \\ 0 & si\ x = - \quad et \quad y = - \\ -1 & si\ x = - \ ou\_exclusif \quad y = - \end{cases}$$

L'algorithme opère en trois étapes :

- Initialisation

Tab [0, 0] = 0

Tab [*i*, 0] = *Tab* [*i*-1, 0] + *nw* (x_i, -) pour tout *i* de 1 à ℓ_1

Tab [0, *j*] = *Tab* [0, *j*-1] + *nw* (-, y_j) pour tout *j* de 1 à ℓ_2

- Calcul des scores et remplissage de la matrice

$$Tab[i,j] = max \begin{cases} Tab[i-1, j-1] + nw(x_i, y_j) \\ Tab[i-1, j] + nw(x_i, -) \\ Tab[i, j-1] + nw(-, y_j) \end{cases}$$

Pour chaque case, 3 cas possibles pour calculer le score maximum : substitution, insertion ou délétion (par rapport à une séquence).

Exemple l'alignement de deux séquences MPRCLCQR et PYRCKCR :

	–	M	P	R	C	L	C	Q	R
–	0	-2	-4	-6	-8	-10	-12	-14	-16
P	-2	-1	1	-1	-3	-5	-7	-9	-11
Y	-4	-3	-1	0	-2	-4	-6	-8	-10
R	-6	-5	-3	2	0	-2	-4	-6	-5
C	-8	-7	-5	0	5	3	1	-1	-3
K	-10	-9	-7	-2	3	4	2	0	-2
C	-12	-11	-9	-4	1	2	7	5	3
R	-14	-13	-11	-6	-1	0	5	6	8

Table. 8 Matrice de construction de l'alignement

cas : score(-,-)=0, score(x,-)=-2, score(x,y)=-1, score(x,x)=2

- Calcul de l'alignement en remontant dans la matrice (traçage en arrière)

Traçage des flèches vers l'origine, marquez un chemin de la cellule en bas à droite vers la cellule en haut à gauche en suivant la direction des flèches. À partir de ce chemin, la séquence est construite selon ces règles : Une flèche diagonale représente une correspondance ou un mésappariement, de sorte que la lettre de la colonne et la lettre de la ligne de la cellule d'origine s'alignent. Une flèche horizontale ou verticale représente un indel. Les flèches horizontales aligneront un gap sur la lettre de la ligne, les flèches verticales aligneront un gap sur la lettre de la colonne. S'il y a plusieurs flèches parmi lesquelles choisir, elles représentent un embranchement des alignements. Si deux branches ou plus appartiennent toutes à des chemins allant du coin inférieur droit à la cellule supérieure gauche, ce sont des alignements également envisageables. Dans ce cas, notez les chemins comme candidats d'alignement. Needleman et Wunsch nomment ce passage le chemin des scores maximum : on commence par le plus haut score, vers le plus haut score parmi les trois cases (*i*-1, *j*-1) (*i*-1, *j*) et (*i*, *j*-1) et ainsi de suite jusqu'à la case (1,1). Dans l'exemple précèdent, on commence par la case (7,8) ayant le plus haut score = 8 (table. 9). Dans ce cas, les scores des 3 cases (6,7) (6,8) et (7,7) sont respectivement 5, 3 et 6. Donc, le parcours de la matrice sera vers la case (7,7) où le score est le plus grand soit 6. Le parcours final de la matrice transformée est le suivant :

	—	M	P	R	C	L	C	Q	R
—	0	-2	-4	-6	-8	-10	-12	-14	-16
P	-2	-1	1	-1	-3	-5	-7	-9	-11
Y	-4	-3	-1	0	-2	-4	-6	-8	-10
R	-6	-5	-3	2	0	-2	-4	-6	-5
C	-8	-7	-5	0	5	3	1	-1	-3
K	-10	-9	-7	-2	3	4	2	0	-2
C	-12	-11	-9	-4	1	2	7	5	3
R	-14	-13	-11	-6	-1	0	5	6	8

Table. 9 Chemin de l'alignement

Le mouvement diagonal qui correspond au passage de la case (i, j) à la case (i+1, j+1). C'est le mouvement que l'on privilégie. Le mouvement vertical qui correspond au passage de la case (i, j) à la case (i, j+1), ce qui donne une insertion sur la séquence en i. Le mouvement horizontal qui correspond au passage de la case (i, j) à la case (i+1, j), ce qui donne une insertion dans la séquence en j. L'alignement optimal est :

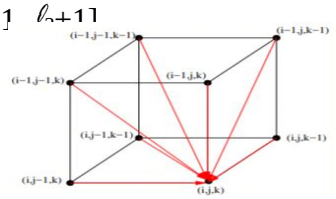
M	P	—	R	C	L	C	Q	R
—	P	Y	R	C	K	C	—	R

La méthode exposée pour 2 séquences est généralisable pour un nombre quelconque n de séquences en employant une table de score n-dimensionnelle. La représentation graphique est toutefois difficile à réaliser même pour n=3. Sur n séquences, pour calculer le score maximum nous avons 2^n-1 cas possibles :

Pour n = 2 nous avons 3 possibilités :

$(i-1, j-1)$	$(i-1, j)$
$(i, j-1)$	(i, j)

De même pour n = 3 nous avons 7 possibilités : L'alignement de trois séquences de longueur ℓ_1, ℓ_2 et ℓ_3 utilise un cube de taille [ℓ_1+1, ℓ_2+1, ℓ_3+1]



- Initialisation

$Tab\,[0, 0, 0] = 0$

$Tab\,[i, 0, 0] = Tab\,[i\text{-}1, 0, 0] + SP\,(x_i, -, -)$ pour tout i de 1 à ℓ_1

$Tab\,[0, j, 0] = Tab\,[0, j\text{-}1, 0] + SP\,(-, y_j, -)$ pour tout j de 1 à ℓ_2

$Tab\,[0, 0, k] = Tab\,[0, 0, k\text{-}1] + SP\,(-, -, z_k)$ pour tout j de 1 à ℓ_3

$$Tab[i,j,0] = max\begin{cases} Tab[i-1,j-1,0] + nw(x_i, y_j) \\ Tab[i-1,j,0] + nw(x_i, -) \\ Tab[i,j-1,0] + nw(-, y_j) \end{cases}$$ Pour tout i de 1 à ℓ_1 et j de 1 à ℓ_2

$$Tab[i,0,k] = max\begin{cases} Tab[i-1,0,k-1] + nw(x_i, z_k) \\ Tab[i-1,0,k] + nw(x_i, -) \\ Tab[i,0,k-1] + nw(-, z_k) \end{cases}$$ Pour tout i de 1 à ℓ_1 et k de 1 à ℓ_3

$$Tab[0,j,k] = max\begin{cases} Tab[0,j-1,k-1] + nw(y_i, z_j) \\ Tab[0,j-1,k] + nw(y_j, -) \\ Tab[0,j,k-1] + nw(-, z_k) \end{cases}$$ Pour tout j de 1 à ℓ_2 et k de 1 à ℓ_3

- Calcul des scores et remplissage de la matrice

$$Tab[i,j,k] = max \begin{cases} Tab[i-1,j-1,k-1] + SP(x_i, y_j, z_k) \\ Tab[i-1,j,k] + SP(x_i, -, -) \\ Tab[i,j-1,k] + SP(-, y_j, -) \\ Tab[i,j,k-1] + SP(-, -, z_k) \\ Tab[i,j-1,k-1] + SP(-, y_j, z_k) \\ Tab[i-1,j,k-1] + SP(x_i, -, z_k) \\ Tab[i-1,j-1,k] + SP(x_i, y_j, -) \end{cases}$$

Le score SP : $SP(x_i, y_j, z_k) = nw(x_i, y_j) + nw(x_i, z_k) + nw(y_j, z_k)$,

$SP(-, y_j, z_k) = nw(-, y_j) + nw(-, z_k) + nw(y_j, z_k)$,

$SP(x_i, -, -) = nw(x_i, -) + nw(x_i, -) + nw(-, -) = -1 + (-1) + 0 = -2$ (gaps à coût constant)

Le principe d'alignement exposé pour deux et trois séquences peut être généralisé à un nombre de séquences $n > 3$ quelconque. Le principe reste le même, il est nécessaire d'utiliser une table en dimension n, dont la taille est le produit cartésien des longueurs de chacune des séquences $[\ell_1+1, \ell_2+1, ..., \ell_n+1]$.

Le meilleur alignement multiple de n séquences est calculé à l'aide d'un hyper-cube n-dimensionnel Tab. Définissant $Tab\ [p_1, p_2, ..., p_n]$ comme étant le meilleur score pour aligner les préfixes de longueurs $p_1, p_2, ..., p_n$ des séquences $S_1, S_2, ..., S_n$ respectivement.

$Tab\ [0, 0, ..., 0] = 0$

$Tab[p_1, p_2, \dots, p_n] = max_{\varepsilon \in \{0,1\}^n, \varepsilon \neq 0}\{Tab[p_1 - \varepsilon_1, p_2 - \varepsilon_2, \dots, p_n - \varepsilon_n] + SP(\varepsilon_1 x_{p_1}, \dots, \varepsilon_n x_{p_n})\}$

$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) \in \{0,1\}^n$ est un vecteur qui indique les directions de progression de l'alignement dans l'hyper-cube, ($si\ \varepsilon_i = 0\ alors\ \varepsilon_i x_{p_i} = -$). La taille de l'hyper-cube est ($\prod_{i=1}^{n}(l_i + 1)$), où l_i est la longueur de S_i, où le calcul pour chaque entrée considère $2^n - 1$ autres.

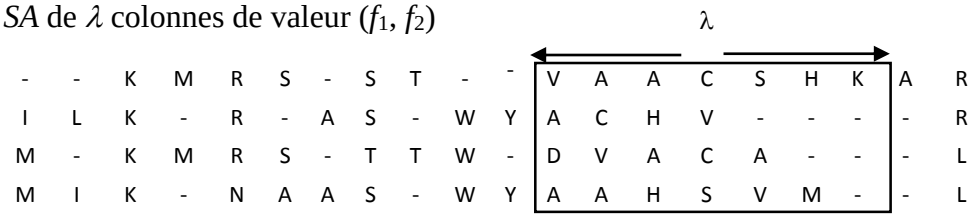
Remarque : l'algorithme MSA ainsi que l'algorithme DCA sont référencés dans la littérature dans la catégorie des algorithmes exacts. Toutefois, dans les deux cas il s'agit de méthodes basées sur des heuristiques, l'appellation « algorithmes exactes » est abusive. L'appellation « algorithmes basés sur une méthode exacte » semble plus appropriée (Derrien, 2008).

Pour profiter de la puissance de la méthode exacte tout en respectant un temps de calcul raisonnable nous avons développé une heuristique progressive basée sur l'algorithme de Needleman-Wunsch. Cette heuristique fonctionne plus rapidement que la méthode exacte, mais sans aucune garantie de trouver le meilleur alignement possible, elle garantit plutôt une solution proche de l'optimale. Cette méthode est appliquée sur des sous problèmes générés à partir de certaines solutions du FP pendant le processus d'optimisation, contribuant à améliorer la qualité des solutions. Elle est essentiellement basée sur l'alignement par paire, étendu pour aligner une séquence avec un alignement. Elle commence par aligner à la fois les trois ou les deux sous-séquences les plus longues, puis d'ajouter les autres une par une.

Les principales étapes de l'utilisation de l'algorithme progressif basé sur la méthode exacte de Needleman-Wunsch peuvent être résumées comme suit :

- ***Générer des sous-problèmes de petite taille λ*** :

à partir d'un alignement $A \in$ PF de taille L et de valeur objectif (F_1, F_2), déterminer un sous-alignement SA de λ colonnes de valeur (f_1, f_2)



Les λ colonnes sont enlevées de l'alignement complet et les colonnes restantes (L-λ) sont fixées à leurs valeurs (F_1 - f_1, F_2 - f_2)

-	-	K	M	R	S	-	S	T	-	-	A	R
I	L	K	-	R	-	A	S	-	W	Y	-	R
M	-	K	M	R	S	-	T	T	W	-	-	L
M	I	K	-	N	A	A	S	-	W	Y	-	L

Un sous problème sera déterminé en supprimant tous les gaps dans le sous-alignement, afin d'obtenir les sous séquences à aligner par la méthode exacte mono-objectif. Cela permet de créer de nouvelles instances de la solution incomplète

V	A	A	C	S	H	K		**s1:**	V	A	A	C	S	H	K
A	C	H	V	-	-	-	⇒	**s2:**	A	C	H	V			
D	V	A	C	A	-	-		**s3:**	D	V	A	C	A		
A	A	H	S	V	M	-		**s4:**	A	A	H	S	V	M	

- appliquer NW mono-objectif sur le sous alignement

Aligner à la fois les 3 sous-séquences les plus longues par la méthode NW

s1:	V	A	A	C	S	H	K
s4:	A	A	H	S	V	M	
s3:	D	V	A	C	A		

Le résultat de l'alignement :

-	V	A	A	C	S	H	K
-	-	A	A	H	S	V	M
D	V	A	-	C	A	-	-

Aligner la sous-séquence s2 avec l'alignement trouvé en appliquant l'algorithme NW. Nous pouvons construire la matrice de score initiale en ajoutant la quatrième séquence comme le montre la table. 10. Il est intéressant de noter que les deux étapes de l'algorithme (c'est-à-dire l'étape d'initialisation et de remplissage) ont été adaptées pour prendre en charge l'alignement progressif d'une séquence avec un alignement.

- Initialisation

Tab [0, 0] = 0

Tab [i, 0] = *Tab* [i-1, 0] + *nw* (x_i, -,-,-) pour tout i de 1 à ℓ_1

Tab [0, j] = *Tab* [0, j-1] + *nw* (-, y_{j1}, y_{j2}, y_{j3}) pour tout j de 1 à ℓ_2

$$nw(x,y) = \begin{cases} BM(x,y) & si\ x \neq - \quad et \quad y \neq - \\ 0 & si\ x = - \quad et \quad y = - \\ -1 & si\ x = - \ ou\_exclusif \ y = - \end{cases}$$

si on prend *BM* = Blosum62

Pour la première colonne chaque caractère sera aligné à trois gaps.

Tab[1, 0]= *Tab*[0,0] + *nw*(A,-,-,-) = 3**nw*(A,-) = -3.

Tab[2, 0]= *Tab*[1,0] + *nw*(C,-,-,-) = -3 + 3**nw*(C,-) = -6. Etc.

Pour la première ligne

Tab [0,1] = *Tab*[0,0] + *nw*(-,-,-,D) = 3* *nw*(-,D)= -3.

Tab [0, 2]= *Tab* [0, 1] + *nw* (-, V, -, V)

= -3 + *nw*(-,V) + *nw*(-,-)+ *nw*(-,V) + *nw*(V,-) +*nw*(V,V)+ *nw*(-,V)

= -3 - 1 + 0 – 1 – 1 + blosum62 (V, V) – 1 = -7 + 4 = -3

- construction

$$Tab[i,j] = max\begin{cases} Tab[i-1,j-1] + nw(x_i, y_{j1}, y_{j2}, y_{j3}) \\ Tab[i-1,j] + nw(x_i, -, -, -) \\ Tab[i,j-1] + nw(-, y_{j1}, y_{j2}, y_{j3}) \end{cases}$$

		-	-	V	A	A	C	S	H	K	
		-	-	-	A	A	H	S	V	M	y_{j2}
		-	D	V	A	-	C	A	-	-	y_{j3}
	-	0	-3	-3	6	6	6	9	2	-3	
	A	-3	-6	-2	21	20	20	23	16	11	
	C	-6	-9	-6	17	22	38	41	34	29	
	H	-9	-11	-10	13	18	34	40	40	35	
	V	-12	-15	-2	9	14	30	36	36	35	

x_i *Table. 10 Matrice de construction de l'alignement*

Le résultat du sous alignement *SA'* est :

-	V	A	A	C	S	H	K
-	-	A	A	H	S	V	M
D	V	A	-	C	A	-	-
-	-	A	-	C	-	H	V

Recalculer le score du nouveau sous-alignement obtenu dans l'espace multiobjectif (f'_1, f'_2) :

Si (f'_1, f'_2) domine ou incomparable à (f_1, f_2)

Alors

Placer le nouvel alignement dans la solution incomplète pour obtenir l'alignement *A*' (F_1-f_1+f'_1, F_2-f_2+f'_2)

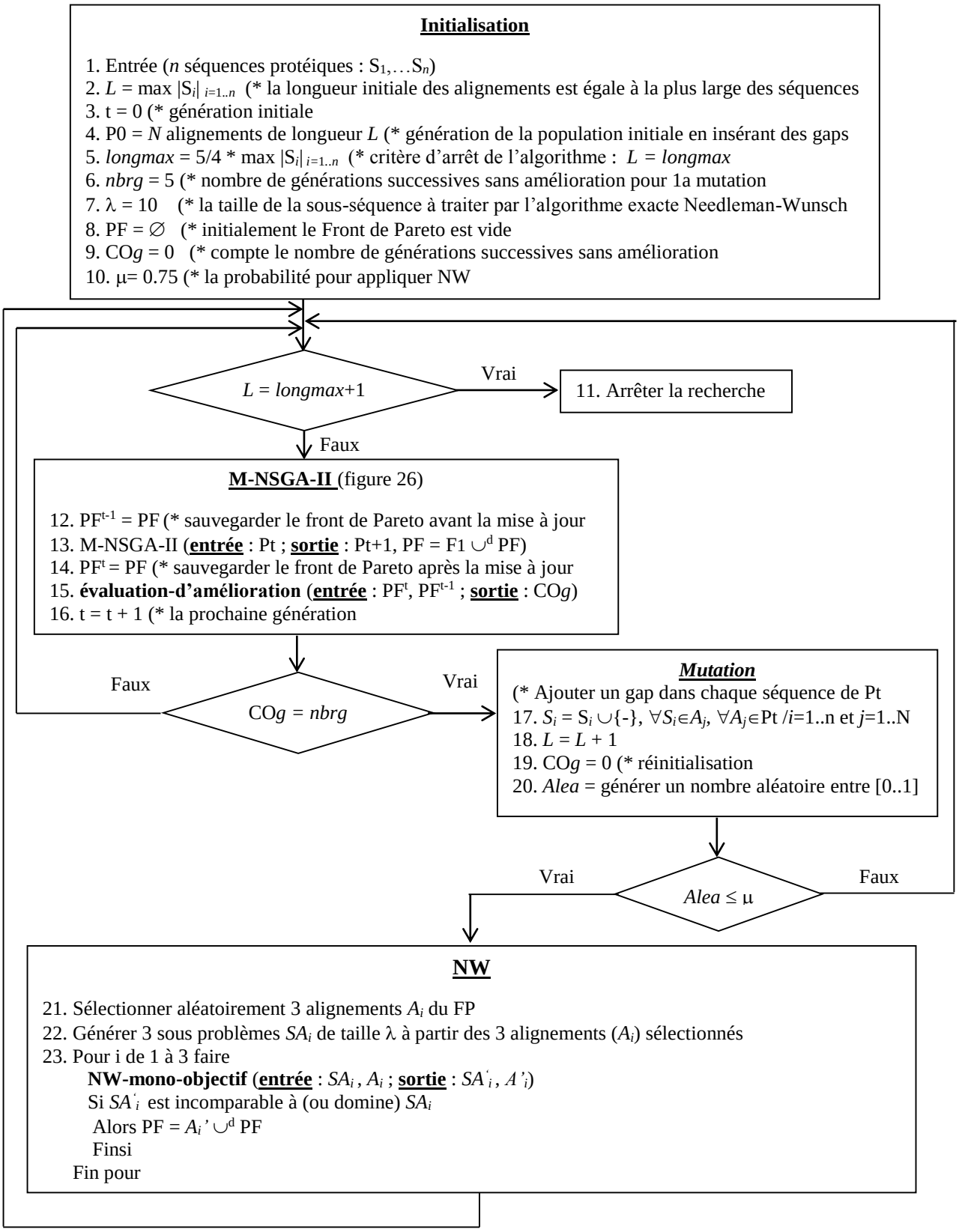
PF = *A'* $\cup^d$ PF

Fin si

La nouvelle solution *A*' après application de l'algorithme NW sur le sous problème de taille λ :

-	-	K	M	R	S	-	S	T	-	-	-	V	A	A	C	S	H	K	A	R
I	L	K	-	R	-	A	S	-	W	Y	-	-	A	A	H	S	V	M	-	R
M	-	K	M	R	S	-	T	T	W	-	D	V	A	-	C	A	-	-	-	L
M	I	K	-	N	A	A	S	-	W	Y	-	-	A	-	C	-	H	V	-	L

III.3) schéma général de l'algorithme hybride NW-M-NSGA-II



- Procédure évaluation-d'amélioration

Cette procédure permet de suivre l'évolution des points non dominés, générés entre les itérations successives. Principalement, il y a trois aspects pour déterminer l'amélioration d'un ensemble de solutions (Okabe et al., 2003): la cardinalité (le nombre de solutions), la convergence (précision) et la diversité (distribution et étalement). Un grand nombre de métriques a été proposé

dans la littérature pour comparer les ensembles de solutions (fronts de Pareto). L'hypervolume (HV) (Zitzler and Thiele, 1999) est la métrique la plus utilisée dans la littérature, puisque HV (ou S-métrique) est la seule métrique unaire capable de mesurer les trois aspects (précision, diversité et cardinalité).

Pour mettre à jour le compteur CO*g*, il est nécessaire de comparer les solutions non dominées obtenues à la génération t avec celles obtenues à la génération t-1, le long des itérations. Par conséquent, HV ne convient pas en raison de sa complexité de calcul. Ainsi, la cardinalité et une métrique intuitive basée sur le point idéal (IP) ont été utilisées. Ils ne prennent pas beaucoup de temps et sont faciles à appliquer (Mahdi and Nini, 2021). La cardinalité est utilisée pour mesurer l'amélioration de la diversité tandis que l'IP est utilisé pour mesurer l'amélioration de la convergence (intuitivement, il y a une amélioration de la convergence si IP (PF^t) domine IP (PF^{t-1})). Mais la métrique IP proposée n'est pas cruciale pour déterminer l'amélioration de la convergence lorsque IP(PF^t) et IP(PF^{t-1}) sont incomparables. Dans ce cas, une métrique binaire nommée C-métrique (Zitzler & Thiele, 1999) est utilisée pour fournir des informations sur la convergence. C(PF^t, PF^{t-1}) donne la fraction des solutions dans PF^{t-1} qui sont dominées au moins par une solution dans PF^t. Ainsi, C(PF^t, PF^{t-1}) = 1 signifie que toutes les solutions de PF^{t-1} sont dominées par au moins une solution de PF^t, tandis que C(PF^t, PF^{t-1}) = 0 implique qu'aucune solution de PF^{t-1} n'est dominé par une solution dans PF^t. C(PF^t, PF^{t-1}) > C(PF^{t-1}, PF^t) indique une amélioration de la convergence.

$$C(PF^t, PF^{t-1}) = \frac{|\{b \in PF^{t-1} / \exists a \in PF^t : a \ \boldsymbol{dominates}\ b\}|}{|PF^{t-1}|}$$

évaluation-d'amélioration (**entrée** : PF^t, PF^{t-1} ; **sortie** : CO*g*)
Si $|PF^t| > |PF^{t-1}|$ (* cardinalité
Alors CO*g* = 0 (* amélioration de la diversité
Sinon si IP (PF^t) **domine** IP (PF^{t-1})
Alors CO*g* = 0 (* amélioration de la convergence
Sinon si IP (PF^t) **incomparable** to IP (PF^{t-1})
Alors si C (PF^t, PF^{t-1}) > C (PF^{t-1}, PF^t)
Alors CO*g* = 0 (* amélioration de la convergence
Sinon CO*g* = CO*g* + 1 (* pas d'amélioration
Finsi
Sinon CO*g* = CO*g* + 1 (* pas d'amélioration IP (PF^{t-1}) domine IP (PF^t)
Finsi
Finsi
Finsi

- Complexité temporal de l'algorithme NW-M-NSGA-II

Au-delà de deux séquences, le problème devient rapidement très complexe car l'espace des alignements possibles explose. Le NW-M-NSGA-II proposé est composé de deux algorithmes principaux : M-NSGA-II et NW. La complexité temporelle de M-NSGA-II est $O(MN^2)$, où N est la taille de la population et M le nombre d'objectifs (la complexité temporelle de GPLS est constante). Par contre, NW a une complexité exponentielle, dans le pire des cas de $O(k^2\, 2^k\, \lambda^k)$, avec *k* séquences de longueur λ, où λ=10 est la taille du sous-problème. Dans le pire des cas, la complexité totale est ($O(MN^2)$+ $O(k^2\, 2^k\, \lambda^k)$). En pratique, la complexité temporelle est bien moindre, car la fréquence d'utilisation de NW est réalisée après 5 générations sans amélioration avec la probabilité 0.75.

VI) Evaluation des performances

La stratégie d'optimisation étant établie, il nous reste à évaluer la qualité de la méthode proposée. Deux types de mesure sont considérés pour évaluer les performances d'un algorithme d'optimisation : le temps d'exécution et la qualité de la solution obtenue. Pour les méthodes exactes

(mono ou multiobjectif) seule la mesure du temps est considérée puisque la qualité des solutions est absolue (la solution optimal ou le front de Pareto optimal). Cependant, le temps de calcul pour trouver cette solution ou ce front de Pareto par des méthodes exactes augmente de façon exponentielle avec la taille du problème. C'est ce temps de calcul excessif qui classe MSA dans les problèmes NP-difficile. En terme de complexité temporelle, on considère généralement qu'un algorithme est plus efficace qu'un autre si son temps d'exécution du cas le plus défavorable à un ordre de grandeur inférieur. Dans le cas des méthodes approchées (mono ou multiobjectif) les deux types de mesures sont considérés. En effet, le but est de développer des algorithmes les plus efficaces possible en temps de calculs (raisonnable), en qualité de solutions produites et pouvant traiter des problèmes de grande tailles.

En optimisation mono-objectif, la présence d'ordre total entre les solutions rend la mesure (la comparaison) de qualité évidente. En optimisation multiobjectif, la mesure de qualité nécessite l'évaluation d'un ensemble de solutions de compromis. La convergence vers le front de Pareto et la préservation de la diversité des solutions sont les deux propriétés importantes pour l''évaluation d'un algorithme en termes de qualité des solutions obtenues. De nombreux indicateurs de performances ont été proposés dans la littérature (La mesure S, La métrique C, la mesure de contribution, l'entropie,…).

Dans le problème d'alignement multiple de séquences l'enjeu est double, on doit trouver des solutions mathématiquement efficaces (mesure S, métrique C,…) et biologiquement acceptables. Un moyen utilisé pour tester l'efficacité biologique des méthodes d'alignement est d'effectuer des statistiques sur des bases d'alignements de références comme Balibase, qui permettent d'estimer la signification biologique des résultats. Pour estimer les performances de notre algorithme nous allons établir une stratégie d'évaluation (un protocole expérimental).

VI.1) protocole expérimental

Dans cette section, nous allons détailler la méthodologie expérimentale suivie : -Trouver le jeu de paramètres menant à des meilleures performances de l'algorithme. - Montrer la performance de l'approche multiobjectif par rapport à l'approche mono-objectif. - Présenter l'avantage de l'approche hybride en utilisant une méthode exacte. - La comparaison de la signification biologique des algorithmes proposés avec les autres méthodes de la littérature. Les algorithmes proposés ont été implémentés dans MATLAB R2016b, sur PC (i5, 3230M, CPU 2.60GHz 2.60 GHz, RAM 4 Go et Win7 64 bits).

VI.1.1) Le réglage des paramètres

Afin d'adapter au mieux le comportement de la méthode au problème posé, il est important de trouver le jeu de paramètres menant à des meilleures performances de l'algorithme. Il est donc indispensable d'étudier l'influence de chaque paramètre sur le comportement de l'algorithme. Cependant, le réglage du comportement de l'algorithme en fonction de ses paramètres déterminants est une tâche fastidieuse et coûteuse en temps.

Les paramètres déterminants dans la méthode proposée sont :

- *N* : la taille de la population,

- *longmax* : La longueur maximale d'alignement atteint pour arrêter le processus de recherche.

- *nbrg* : nombre de générations successives sans amélioration pour incrémenter la longueur de l'alignement.

- la matrice de similarité : Le choix d'une matrice de substitution gouverne le système de score donc influence les résultats obtenus. La matrice de similarité utilisée est Blosum*x* (tel que $x = 30, 62$ ou 100) selon l'instance du problème étudié.

- nombre de sous problèmes :

- λ : la taille des sous problèmes

- μ : la probabilité d'évocation de l'algorithme exacte NW.

Le critère d'arrêt $longmax$ est fixé à $\frac{5}{4}\max(|S_i|)$, Ce choix est basé sur le fait que la plus part des alignements de références de Balibase (une signification biologique) contenaient rarement plus de 25% de gaps dans la séquence la plus longue. Afin d'ajuster empiriquement les paramètres affectant la qualité des résultats, plusieurs exécutions ont été effectuées sur l'instance BB11001 de l'ensemble de données RV11 de Balibase qui est composé de quatre séquences (1aab_, j46_A, 1k99_A, 2lef_A). RV11 est l'un des ensembles de données les plus informatifs de Balibase (Kemena & Notredame, 2009).

- la matrice Blosum62 donne de meilleurs résultats pour l'instance BB11001 par rapport à Blosum30 et Blosum100.

- Un des paramètres déterminant est le *nbrg* qui compte le nombre de générations successives sans amélioration, afin d'incrémenter la longueur des alignements, en ajoutant un gap dans chaque ligne (*L*=*L*+1). Si la valeur de *nbrg* est grande on risque de trop explorer (inutilement) l'espace de recherche, qui est défini par les alignements de taille *L*, et le temps d'exécution risque alors d'être trop long. A l'inverse pour *nbrg* faible on risque de ne pas avoir le temps nécessaire pour bien exploiter l'espace de solutions de taille *L*, et l'incrémentation se fait très rapidement. Ce paramètre permet de révéler la capacité de l'algorithme à poursuivre la recherche tout en améliorant les résultats. Pour trouver la bonne valeur de *nbrg*, 10 exécutions ont été effectuées en utilisant l'algorithme M-NSGA-II sur l'instance BB11001 avec une population de taille *N*=10, en faisant varier graduellement *nbrg*.

M-NSGA-II: BB11001, N=10, Blosum62.										
	nbrg =1	*nbrg* =2	*nbrg* =3	*nbrg* =4	*nbrg* =5	*nbrg* =6	*nbrg* =7	*nbrg* =8	*nbrg* =9	*nbrg* =10
Nbr-Gén-avec-Amélioration	25.4	75.9	147.6	229.7	293.3	368.4	475.9	494.2	489.4	513.6
Nbr-Gén-sans-Amélioration	18	52.4	103.4	164.2	204.5	272.2	379.1	477.6	502.2	533.7
Nbr-Total-Gén	43.4	129.1	251	393.9	497.8	640.6	855	971.8	991.6	1047.3
% du nombre de générations avec amélioration	58.53%	58.79%	58.80%	58.31%	**58.92%**	57.51%	55.66%	50.85%	49.35%	49.04%

Table. 11 Le pourcentage du nombre de générations participant à l'amélioration

La table 11, montre la moyenne en nombre de générations (avec et sans) amélioration de dix exécutions en fonction du nombre de générations successives sans amélioration (*nbrg*). Selon les résultats l'operateur de mutation augmentant la taille de l'alignement chaque 5 génération successive sans amélioration. Notons que ce nombre peut ne pas donner le même résultat pour une autre instance du problème.

- Il est clair qu'une population de petite taille permet d'un côté, un bénéfice considérable en temps de calcul global et d'un autre côté, elle augmente le risque de convergence prématurée. Par conséquent, il est important de déterminer d'une manière empirique une population aussi petite que possible. Pour cela, 10 exécutions ont été effectuées en utilisant l'algorithme M-NSGA-II sur l'instance BB11001, avec *nbrg*=5 et en faisant varier la taille *N* = 10, 20, 30, 40, 50, etc. en mesurant l'hypervolume HV (la surface délimitée par le front et le point Nadir). Chaque front de Pareto est alors formé par l'union non dominée des 10 fronts obtenus ($PF = \cup_{i=1..10}^{d} Pf_i$) avec le temps moyen des 10 exécutions.

M-NSGA-II: BB11001, nbrg=5, Blosum62.										
N	10	20	30	40	50	60	70	80	90	100
Temps (s)	3.2	5.18	6.75	9.76	11.88	15.83	22.76	24.96	30.61	33.75
HV	8065	10221	14696	17543	16844	17602	17864	18273	18935	18346
Q = HV/HV_{10}		1.27	1.82	2.18	2.09	2.18	2.22	2.27	2.35	2.27
r = Q/temps		0.245	**0.270**	0.223	0.176	0.138	0.097	0.091	0.077	0.067

Table. 12 le rapport qualité-temps en fonction de la taille de la population

Dans la table 12, Q représente le taux d'amélioration (en terme de métrique S) par rapport au résultat obtenu pour *N*=10 et cela, pour chaque *N* de 20 à 100. Le terme *r* représente le rapport entre le taux Q et le temps moyen. Plus ce rapport est grand meilleur est le résultat (dans ce cas la taille *N*=30 donne le meilleur rapport qualité-temps).

- Pour réduire le temps de calcul coûteux de l'algorithme exact, la contribution de NW est appréciée selon la probabilité $\mu = 0.75$ après l'operateur de mutation (i.e. après 5 génération successive sans amélioration). Trois sous-problèmes de taille $\lambda = 10$ sont générés à partir des solutions choisis du front Pareto actuel.

VI.1.2) Performance de l'approche multiobjectif par rapport au mono-objectif

Dans cette étude nous allons montrer la performance de l'approche multiobjectif par rapport à l'approche mono-objectif. L'algorithme mono-objectif utilisé dans cette étude est l'algorithme mémétique (GPLS-GA) incorporant un algorithme génétique et l'heuristique GPLS. Cette méthode utilise la même stratégie de l'algorithme multiobjectif M-NSGA-II proposé.

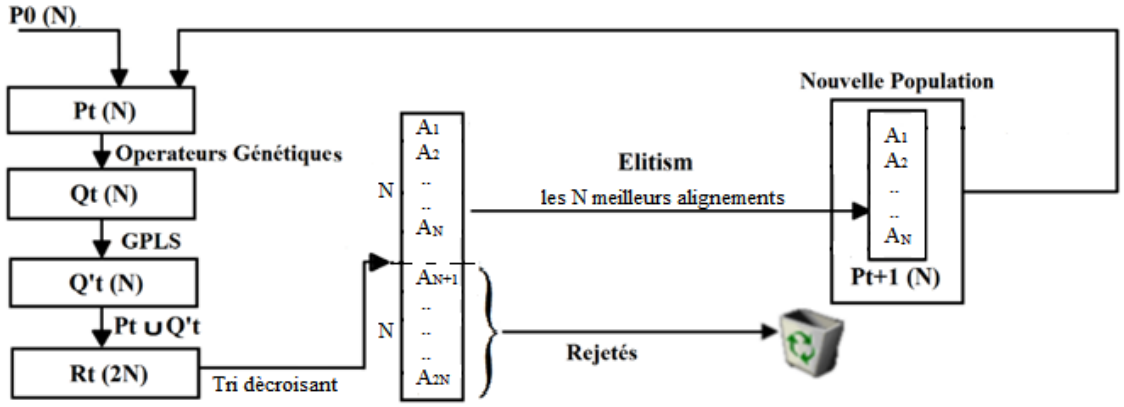


Figure. 29 Schéma général de l'algorithme mono-objectif GPLS-GA

L'algorithme mono-objectif utilise la fonction scalaire SP qui combine les valeurs de récompense et de pénalité, permettant ainsi, d'obtenir un seul alignement.

$$SP = \max[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} sc(S_i, S_j) - \sum_{i=1}^{n} coût\_gaps\ (S_i)]$$

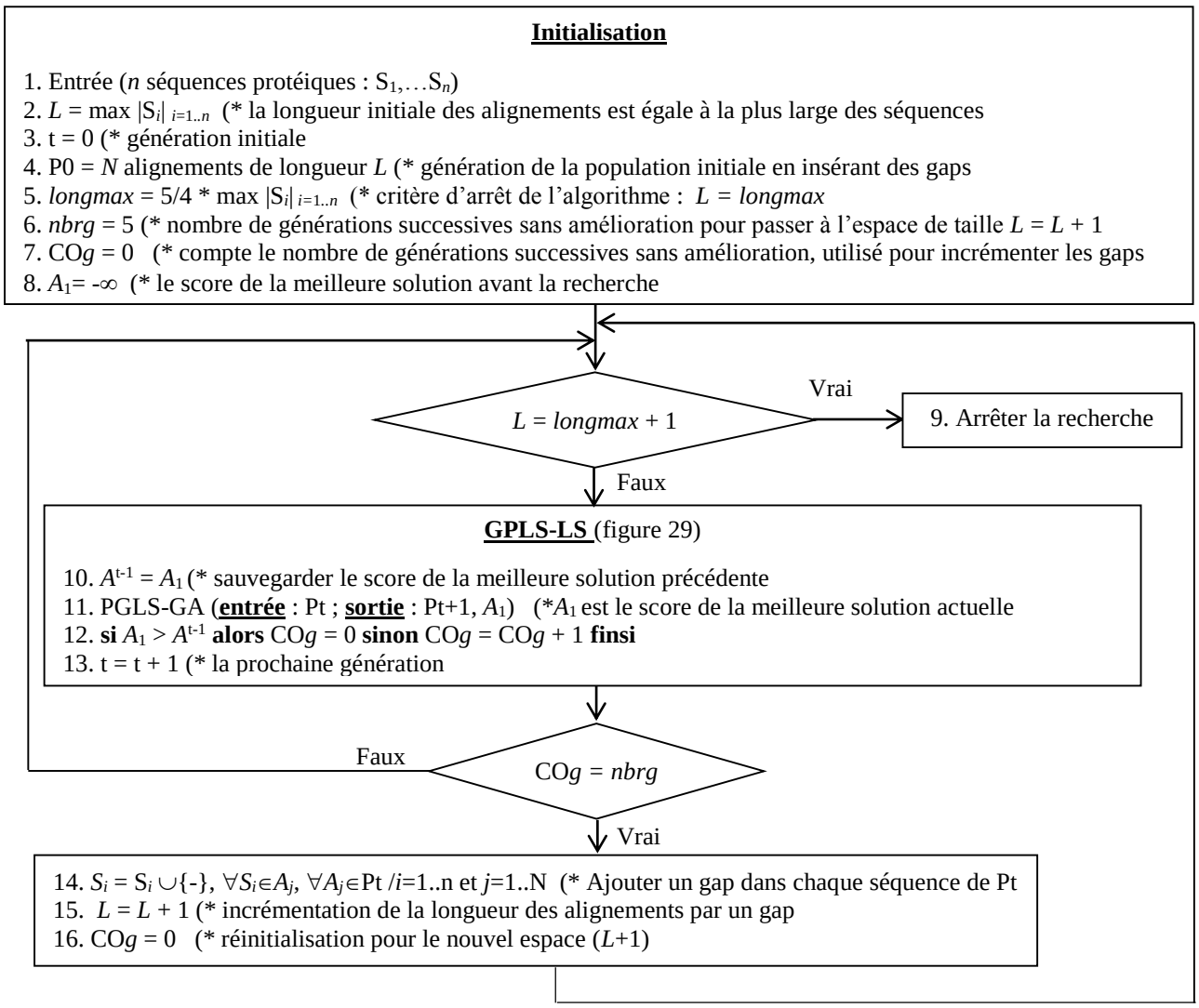
En multiobjectif nous utilisons les deux termes (récompense et pénalité) de SP séparément, comme deux fonctions objectifs à optimiser simultanément et qui sont contradictoires.

$$f_1 = max \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} sc(S_i, S_j) \qquad f_2 = \min \sum_{i=1}^{n} coût\_gaps\ (S_i)$$

L'optimisation de ces deux fonctions simultanément permet ainsi, d'obtenir un ensemble de solutions intéressantes. Cet ensemble de Pareto peut contenir la meilleure solution d'une formulation mono-objectif, mais aussi de nombreux autres alignements qu'il n'est pas possible de trouver du tout par l'approche mono-objectif. Le calcule (la soustraction) $f_1 - f_2$ pour toutes les solutions du front

Pareto permet de lister les scores SP de la formulation mono-objectif : SP_i = max $(f_1 - f_2)_i$ pour i=1..$|FP|$, ($|FP|$ est la cardinalité de l'ensemble du front de Pareto). SP_i est la meilleure solution (mathématique) obtenue par la méthode multiobjectif pour la formulation mono-objectif.

L'algorithme GPLS-GA



Nous appliquons ces deux méthodes au sous ensemble de données RV11 (BB11001, BB11005 et BB11013) de Balibase. BB11005 est composé de quatorze séquences (1b8g_A, 1lc5_A, 1bw0_A, 1d2f_A, 1dty_A, 2dkb_, 2gsa_A, 1ohv_A, 1b5o_A, 1fg3_A, 1h1c_A, 1jg8_A, 1ax4_A, 1ajs_A). BB11013 est composé de cinq séquences (1idy_, 1hst_A, 1tc3_C, 1aoy_, 1jhg_A). La table 13 présente les résultats obtenus (pour 10 exécutions) montrant la supériorité de l'approche multiobjectif par apport a l'approche mono-objectif du point de vu mathématique. Du point de vu biologique la variété de solutions du front Pareto nous donne souvent le choix pour une solution préférée.

	BB11001 (Blosum62)	BB11005 (Blosum30)	BB11013 (Blosum30)
GPLS-GA (SP)	208	1376	-577
M-NSGA-II (SP_i = max $(f_1 - f_2)_i$)	**237**	**3843**	**-466**

Table. 13 les résultats en termes de SP monoobjectif

VI.1.3) évaluation de l'apport d'une méthode exacte dans l'hybridation

Dans cette étude nous allons montrer l'avantage de l'approche hybride en utilisant une méthode exacte. Nous allons comparer M-NSGA-II et NW-M-NSGA-II sur le sous ensemble de données RV11 (BB11001, BB11005, BB11013) de Balibase. La figure 30 montre la boîte à moustaches, qui illustre la distribution des valeurs d'hypervolume calculées par M-NSGA-II et NW-M-NSGA-II sur les instances BB11001, BB11005 et BB11013 au cours de 30 différentes exécutions (si l'effectif de l'échantillon est trop petit, les quartiles et les valeurs aberrantes apparaissant dans la boîte à moustaches risquent de ne pas être significatifs).

D'après les résultats de la distribution de la métrique d'hypervolume illustrée dans la figure 30, il est clair que NW-M-NSGA-II donne de meilleurs résultats (convergence et diversité). On voit que sur les 30 fronts Pareto obtenus, 23,33% en BB11001, 6,67% en BB11005 et 26,67% en BB11013 produits par NW-M-NSGA-II donnent de meilleurs résultats que tous les fronts produits par M-NSGA-II.

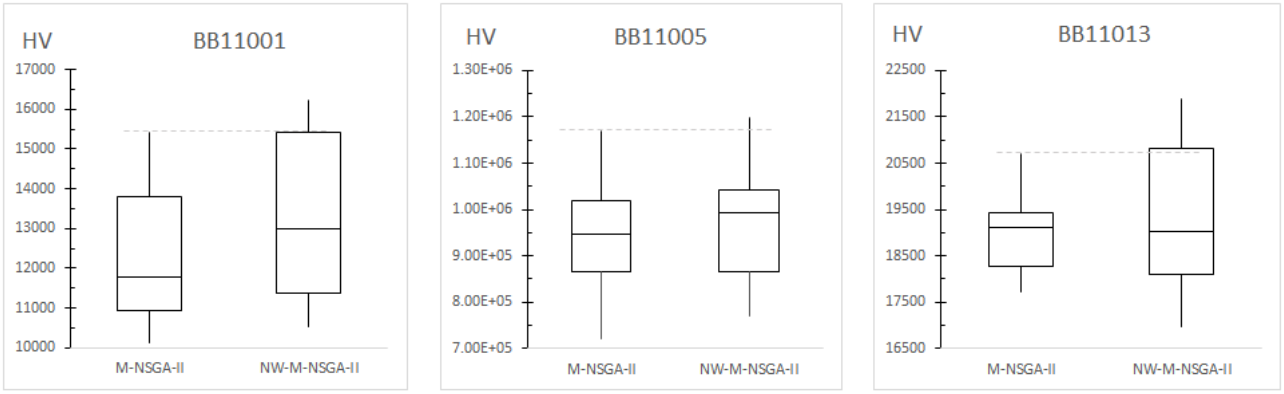


Figure. 30 la distribution des valeurs d'hypervolume calculées par M-NSGA-II et NW-M-NSGA-II

La table 14 contient les résultats de la moyenne des 30 hypervolumes avec le temps moyen obtenus. Ces résultats confirment l'efficacité de l'approche NW-M-NSGA-II par rapport à M-NSGA-II en termes de moyenne des HV mais avec un temps de calcul plus grand.

	BB11001		BB11005		BB11013	
	M-NSGA-II	NW-M-NSGA-II	M-NSGA-II	NW-M-NSGA-II	M-NSGA-II	NW-M-NSGA-II
Moy(HV)	1.23E+04	**1.33E+04**	9.60E+05	**9.69E+05**	1.89E+04	**1.92E+04**
Moy(temps) s	**7.12**	11.86	**19,65**	31.32	**8.44**	13.56

Table. 14 la moyenne des 30 hypervolumes avec le temps moyen obtenus

VI.1.4) Comparaison de la méthode proposée avec des méthodes de la littérature

Il est important pour comparer les résultats obtenus de prendre en compte le point de vue biologique. Pour cela, des jeux de tests ont été créés à partir de séquences réelles, et ils ont été alignés par des biologistes. Des bases de jeux d'essais regroupent ces alignements, avec pour chaque jeu le meilleur résultat possible d'un point de vue biologique. Dans cette étude, les résultats obtenus par notre méthode sont comparés à des méthodes bien connues publiées dans la littérature. En raison du fait qu'il n'y a pas de critère concret pour évaluer la qualité d'un algorithme donné, des benchmarks standards, tels que Balibase, OXBench, PREFAB et SMART, sont fournis pour mesurer l'efficacité de différents algorithmes MSA. Balibase est la base de données la plus utilisée, qui possède des alignements de référence sélectionnés à la main par des biologistes (considérés corrects), utilisée pour attester la qualité des logiciels d'alignement multiple de séquences. Elle fournit deux mesures permettant d'évaluer la précision de l'alignement obtenu par rapport à l'alignement de référence le SPS et le CS. Elle contient une application en langage C, appelée bali_score, qui mesure le score SPS (Sum of Pairs) et TC ou CS (Total Column ou Column Score) de l'alignement de test par rapport à l'alignement de référence. Le score de somme de paires (SPS) est le rapport entre le nombre de paires

de résidus bien alignés dans l'alignement obtenu et ceux de l'alignement de référence. Le score total de la colonne (CS) est le nombre de colonnes bien alignées par rapport au nombre de colonnes de l'alignement de référence (CS est équivaut à SPS dans le cas de deux séquences). En général ($0 \leq CS \leq SPS \leq 1$).

Dans la littérature, la plupart des outils d'alignement ont utilisé la version 2.0 de Balibase (Chowdhury & Garai, 2017). Elle contient huit ensembles de référence. La référence1 contient un nombre de séquences équidistantes. La référence2 contient les séquences très divergentes (une séquence orpheline). La référence3 est constituée de groupes de séquences avec moins de 25% d'identité. La référence 4 contient les extensions de terminal N/C (longues insertions de gap aux extrémités). La référence 5 contient de longues insertions de gap au milieu. Les références 6 à 8 contiennent respectivement des répétitions, des permutations circulaires et des protéines transmembranaires.

Afin d'examiner les performances des méthodes proposées (GA-GPLS, M-NSGA-II et NW-M-NSGA-II), nous les comparons avec des méthodes bien connues dans la littérature semblables (basés sur l'algorithme génétique) ainsi que l'outil CLUSTALW, qui est le plus fréquemment utilisé. Chaque algorithme proposé est exécuté 10 fois, et le meilleur de leurs résultats est enregistré. Les valeurs des différents algorithmes de la littérature sont collectées à partir des articles : MSA-GA (Gondro & Kinghorn, 2007), SAGA (Notredame & Higgins, 1996), GAPAM (Naznin et al., 2012), VDGA (Naznin et al., 2011), MOMSA (Zhu et al., 2015) et IBBOMSA (Yadav & Banka, 2016) qui sont basés sur l'optimisation multiobjectif. Les tables 15 à 17 montrent les résultats sur des sous-ensembles de référence 1, 2 et 3 respectivement de Balibase 2.0. Les chiffres en gras signifient les meilleurs scores SPS.

NOM	CLUSTALW	MSA-GA	SAGA	GAPAM	VDGA	MOMSA	IBBOMSA	GPLS-GA	M-NSGA-II	NW-M-NSGA-II
1idy	0.500	0.427	0.342	0.565	0.573	0.215	**0.574**	0.557	0.549	0.563
1tvxA	0.042	0.295	0.278	0.316	0.316	0.053	**0.423**	0.245	0.314	0.318
1uky	0.392	0.443	0.672	0.402	0.464	0.515	0.588	0.591	**0.674**	0.618
Kinase	0.479	0.295	**0.862**	0.487	0.548	0.850	0.783	0.712	0.757	0.769
1ped	0.592	0.501	0.746	0.498	0.482	0.739	0.827	0.725	**0.829**	0.822
2myr	0.296	0.212	0.285	0.317	0.359	0.437	0.468	0.439	0.442	**0.541**
1ycc	0.643	0.650	0.837	0.845	0.839	**0.934**	0.827	0.889	0.893	0.912
3cyt	0.767	0.772	0.908	**0.911**	0.898	0.815	0.893	0.852	0.898	0.884
1ad2	0.773	0.821	0.917	0.956	0.959	0.956	0.928	0.911	0.917	**0.965**
1ldg	0.880	0.895	**0.989**	0.963	0.946	**0.989**	0.826	0.927	0.948	0.944
1fieA	0.932	0.843	0.947	0.963	0.960	0.982	**0.985**	0.813	0.962	0.974
1sesA	0.913	0.620	0.954	0.982	0.962	0.958	**0.992**	0.932	0.947	0.949
1krn	0.895	0.908	0.993	0.960	0.960	**1.000**	0.928	0.932	0.913	0.908
2fxb	**0.985**	0.941	0.951	0.970	0.978	0.936	0.980	0.943	0.921	0.931
1amk	0.945	0.965	0.997	**0.998**	0.984	0.995	0.946	0.956	0.956	0.952
1ar5A	0.946	0.812	0.971	**0.974**	0.968	0.960	0.924	0.936	0.941	0.932
1gpb	0.947	0.868	0.982	0.983	0.984	0.986	**0.989**	0.943	0.963	0.968
1taq	0.826	0.525	0.931	0.945	**0.959**	0.948	0.912	0.859	0.901	0.898
Moy	0.709	0.655	0.809	0.780	0.786	0.793	0.822	0.787	0.818	**0.825**
% MS	5.56%	0.00%	11.11%	16.67%	5.56%	16.67%	**27.78%**	0.00%	11.11%	11.11%

Table. 15 les résultats SPS sur des sous-ensembles de ref1

NOM	CLUSTALW	SAGA	GAPAM	VDGA	MOMSA	IBBOMSA	GPLS-GA	M-NSGA-II	NW-M-NSGA-II
2pia	0.766	0.763	0.826	0.85	**0.973**	0.934	0.823	0.886	0.902
1pamA	0.757	0.623	0.859	0.863	0.959	**0.972**	0.845	0.932	0.964
1aboA	0.650	0.489	0.796	0.791	0.84	**0.842**	0.764	0.817	0.811
1idy	0.515	0.548	0.989	**0.992**	0.974	0.927	0.852	0.915	0.944
1csy	0.154	0.154	0.764	**0.885**	0.854	0.858	0.621	0.787	0.798
1r69	0.675	0.475	0.965	0.934	0.945	**0.979**	0.832	0.935	0.954
1tvxA	0.552	0.448	0.92	0.974	0.936	**0.982**	0.843	0.910	0.904
1tgxA	0.727	0.773	0.878	0.878	0.952	**0.963**	0.823	0.934	0.942
1ubi	0.482	0.492	0.767	0.794	0.921	0.897	0.713	**0.932**	0.925
1wit	0.557	0.694	0.851	0.875	0.92	0.912	0.801	**0.922**	0.912
2trx	0.870	0.870	**0.986**	**0.986**	**0.986**	0.947	0.913	0.967	0.983
1sbp	0.217	0.374	0.765	0.782	0.881	0.927	0.623	0.923	**0.929**
1havA	0.480	0.448	0.879	0.884	0.897	**0.900**	0.754	0.845	0.871
1uky	0.656	0.476	0.808	0.891	0.94	**0.952**	0.767	0.897	0.911
2hsdA	0.484	0.498	0.796	0.856	0.92	0.925	0.765	0.894	**0.928**
3grs	0.192	0.282	0.746	0.781	0.85	0.872	0.645	0.767	**0.877**
Kinase	0.848	0.867	0.799	0.888	0.94	**0.945**	0.835	0.886	0.879
1ajsA	0.324	0.311	0.899	0.906	0.901	0.911	0.702	0.834	**0.920**
1cpt	0.660	0.776	0.875	0.869	0.887	**0.894**	0.827	0.865	0.824
1lvl	0.746	0.726	0.781	0.819	**0.946**	0.927	0.842	0.913	0.897
1ped	0.834	0.835	0.912	0.947	0.972	**0.978**	0.91	0.925	0.936
2myr	0.904	0.825	0.822	0.83	0.966	0.962	0.875	**0.969**	0.942
4enl	0.375	0.739	0.896	0.899	0.915	0.92	0.792	0.910	**0.932**
Moy	0.584	0.586	0.851	0.877	0.925	**0.927**	0.790	0.894	0.908
% MS	0.00%	0.00%	4.35%	13.04%	13.04%	**43.48%**	0.00%	13.04%	21.74%

Table. 16 les résultats SPS sur des sous-ensembles de ref2

NOM	CLUSTALW	SAGA	GAPAM	VDGA	MOMSA	IBBOMSA	GPLS-GA	M-NSGA-II	NW-M-NSGA-II
Kinase	0.619	0.758	0.825	0.890	0.891	0,834	0.798	0.843	**0.916**
1pamA	0.743	0.579	0.835	0.853	**0.924**	0.869	0.831	0.884	0.894
1idy	0.273	0.364	0.601	0.599	0.460	0.602	0.488	0.467	**0.615**
1r69	0.524	0.524	0.709	0.765	0.878	**0.888**	0.713	0.749	0.841
1ubi	0.146	0.585	0.386	0.414	0.661	**0.711**	0.489	0.588	0.543
1wit	0.565	0.484	0.758	0.873	**0.889**	0.793	0.767	0.712	0.837
1uky	0.130	0.269	0.468	0.526	0.639	0.663	0.459	0.56	**0.670**
1ajsA	0.163	0.186	0.311	0.453	0.542	0.575	0.391	**0.586**	0.542
1ped	0.627	0.646	0.775	0.893	0.913	0.924	0.806	**0.931**	0.914
2myr	0.538	0.494	**0.813**	0.651	0.728	0.746	0.678	0.766	0.787
4enl	0.547	0.672	0.800	0.866	0.816	**0.870**	0.792	0.811	0.824
Moy	0.443	0.506	0.662	0.708	0.758	**0.764**	0.656	0.718	0.762
% MS	0.00%	0.00%	9.09%	0.00%	18.18%	**27.27%**	0.00%	18.18%	**27.27%**

Table. 17 les résultats SPS sur des sous-ensembles de ref3

Nous constatons qu'il il existe de grandes variances dans les scores individuels. Cela confirme le théorème du « No Free Lunch » : il n'existe pas de méthode d'optimisation individuelle qui sera meilleure que toutes les autres sur tous les problèmes ou toutes les instances possibles d'un problème donné. À notre avis, il n'est pas possible de tirer des conclusions significatives sur la performance relative des différentes méthodes sur le globale des instances étudiées. La ligne (Moy) montre le score SPS moyen obtenu par les outils d'alignement décrits sur chaque sous-ensemble d'instances traitées. La ligne (%MS) présente le pourcentage du nombre de meilleur score obtenu pour chaque algorithme. En termes de score moyen NW-M-NSGA-II est meilleur sur le sous-ensemble étudié de ref1 et meilleur en termes du nombre de meilleur score par un pourcentage de 27.27% sur le sous ensemble étudié de ref3. On peut conclure sur les sous-ensembles d'instances étudiées que les deux algorithmes proposés présentent un très bon comportement sur le plan individuel ou moyen.

Il convient de souligner que les différences de performances entre les méthodes n'apparaissent que lorsqu'elles sont moyennées sur un grand nombre de cas de test. Pour cela nous allons utiliser la moyenne des scores SPS et CS pour chaque méthode sur l'ensemble de données (ref1, ref2, ref3, ref4, ref5) de Balibase 2.0, en raison de leurs performances avec d'autres algorithmes connexes. La table 18 contient les valeurs des différents algorithmes de la littérature qui sont obtenus en se référant à diverses études de la littérature : (Edgar, 2004), (Cutello et al., 2006b), (Zhang et al, 2005), (Taheri & Zomaya, 2010) et (Dabba et al., 2019).

	Ref 1 (82)		Ref 2 (23)		Ref 3 (12)		Ref 4 (12)		Ref 5 (12)	
Algorithme	**SPS**	**CS**	**SPS**	**CS**	**SPS**	**CS**	**SPS**	**CS**	**SPS**	**CS**
NW-M-NSGA-II	0.898	0.523	0.902	0.544	0.816	0.454	0.944	0.489	0.926	0.697
M-NSGA-II	0.875	0.465	0.905	0.465	0.794	0.487	0.928	0.515	0.925	0.654
GPLS-GA	0.804	0.512	0.875	0.399	0.701	0.364	0.803	0.426	0.832	0.531
CLUSTALW	0.861	0.773	0.932	0.568	0.753	0.460	0.834	0.522	0.859	0.638
SAGA	0.841	0.000	0.586	0.000	0.506	0.000	0.289	0.000	0.642	0.000
RBT-Km	**0.915**	**0.871**	**0.954**	**0.794**	**0.924**	**0.830**	0.944	0.884	**0.987**	**0.979**
PROBCONS(ir=100)	0.911	0.853	0.942	0.616	0.840	0.635	0.937	0.811	0.974	0.893
DIALIGN	0.811	0.709	0.893	0.359	0.684	0.344	0.897	0.762	0.940	0.843
MUSCLE	0.887	0.808	0.935	0.563	0.825	0.564	0.876	0.609	0.968	0.902
PRALINE	0.904	0.839	0.940	0.610	0.764	0.558	0.799	0.539	0.818	0.686
NWNSI	0.867	0.788	0.923	0.514	0.787	0.514	0.904	0.742	0.963	0.859
T-COFFEE	0.866	0.774	0.934	0.561	0.785	0.487	0.918	0.730	0.958	0.903
MAFFT	0.867	0.781	0.924	0.502	0.788	0.504	0.916	0.727	0.963	0.859
PSALIGN[TCOFFEE]	0.884	0.805	0.936	0.583	0.785	0.548	0.891	0.684	0.973	0.900
PSALIGN[PROBCONS]	0.901	0.840	0.940	0.617	0.809	0.522	0.901	0.697	0.980	0.936
MULTALIN	0.834	0.729	0.517	0.440	0.303	0.385	0.292	0.223	0.627	0.462
KALIGN	0.850	0.000	0.920	0.000	0.790	0.000	0.880	0.000	0.920	0.000
FFTNSI	0.838	0.732	0.908	0.496	0.708	0.350	0.793	0.451	0.947	0.831
PRIMEpcw,mea	0.789	0.629	0.925	0.439	0.856	0.547	0.923	0.603	0.890	0.521
Hybrid CSA	0.827	0.653	0.919	0.413	0.786	0.362	0.705	0.319	0.836	0.569
PILEUP8	0.832	0.000	0.429	0.000	0.323	0.000	0.710	0.000	0.639	0.000
PRRN	0.748	0.563	0.902	0.405	0.822	0.483	0.860	0.487	0.822	0.421
ALIGN-M	0.766	0.000	0.884	0.000	0.684	0.000	0.911	0.000	0.917	0.000
SPEM	0.908	0.839	0.934	0.573	0.814	0.569	**0.974**	**0.908**	0.974	0.923
SB_PIMA	0.821	0.000	0.379	0.000	0.267	0.000	0.794	0.000	0.508	0.000
ML_PIMA	0.810	0.000	0.371	0.000	0.372	0.000	0.705	0.000	0.572	0.000
POA	0.666	0.451	0.857	0.265	0.733	0.343	0.805	0.412	0.754	0.323
MOMSA-W	0.844	0.771	0.925	0.557	0.766	0.488	0.871	0.617	0.936	0.802
IMSA	0.834	0.653	0.921	0.413	0.786	0.362	0.73	0.319	0.73	0.319
MOAFS	0.891	0.825	0.916	0.532	0.778	0.494	0.884	0.628	0.923	0.770

Table. 18 Les scores SPS et TC moyens pour chaque méthode sur chaque sous-ensemble Balibase 2.0

La table 19 pressente le classement des algorithmes en termes de SPS moyen et CS moyen sur la totalité des ensembles ref1, ref2, ref3, ref4 et ref5. Les résultats obtenus montrent que les algorithmes proposés sont compétitifs par rapport aux autres méthodes de pointe MSA en termes de précision SPS. En termes de CS les résultats sont moyens.

	Algorithme	SPS
1	RBT-Km	0.945
2	SPEM	0.921
3	PROBCONS(ir=100)	0.921
4	PSALIGN[PROBCONS]	0.906
5	MUSCLE	0.898
6	**NW-M-NSGA-II**	0.897
7	PSALIGN[TCOFFEE]	0.894
8	T-COFFEE	0.892
9	MAFFT	0.892
10	NWNSI	0.889
11	**M-NSGA-II**	0.885
12	MOAFS	0.878
13	PRIMEpcw,mea	0.877
14	KALIGN	0.872
15	MOMSA-W	0.868
16	CLUSTALW	0.848
17	PRALINE	0.845
18	DIALIGN	0.845
19	FFTNSI	0.839
20	ALIGN-M	0.832
21	PRRN	0.831
22	Hybrid CSA	0.815
23	**GPLS-GA**	0.803
24	IMSA	0.800
25	POA	0.763
26	PILEUP8	0.587
27	SAGA	0.573
28	ML_PIMA	0.566
29	SB_PIMA	0.554
30	MULTALIN	0.515

	Algorithme	CS
1	RBT-Km	0.872
2	PROBCONS(ir=100)	0.762
3	SPEM	0.762
4	PSALIGN[PROBCONS]	0.722
5	PSALIGN[TCOFFEE]	0.704
6	T-COFFEE	0.691
7	MUSCLE	0.689
8	NWNSI	0.683
9	MAFFT	0.675
10	MOAFS	0.650
11	MOMSA-W	0.647
12	PRALINE	0.646
13	DIALIGN	0.603
14	CLUSTALW	0.592
15	FFTNSI	0.572
16	PRIMEpcw,mea	0.548
17	**NW-M-NSGA-II**	0.541
18	**M-NSGA-II**	0.517
19	PRRN	0.472
20	Hybrid CSA	0.463
21	MULTALIN	0.448
22	**GPLS-GA**	0.446
23	IMSA	0.413
24	POA	0.359
25	SAGA	0.000
26	KALIGN	0.000
27	PILEUP8	0.000
28	ALIGN-M	0.000
29	SB_PIMA	0.000
30	ML_PIMA	0.000

Table. 19 le classement des algorithmes en termes de SPS moyen et CS moyen

En plus de la précision, le temps requis pour calculer l'alignement multiple de séquences est également un facteur important. Nous allons comparer quelques algorithmes de la littérature avec les méthodes proposées. Le temps affiché représente le temps total sur les 141 instances de Balibase 2.0 (ref1, ref2, ref3, ref4 et ref5).

Méthode	SPS	CS	Temps (s)
NW-M-NSGA-II	0.897	0.541	517
M-NSGA-II	0.885	0.517	308
GPLS-GA	0.803	0.446	184
MUSCLE	0.898	0.689	97
MUSCLE-p	0.883	0.727	52
T-COFFEE	0.892	0.691	1500
NWNSI	0.889	0.683	170
CLUSTALW	0.848	0.592	170
FFTNSI	0.839	0.572	16

Table. 20 Les scores SPS et TC moyens pour chaque méthode, ainsi que le temps total en secondes.

Le table 20 montre que FFTNSI et MUSCLE-p et MUSCLE ont le temps d'exécution le plus court parmi les principales méthodes comparées. MUSCLE, NW-M-NSGA-II et T-COFFEE ont les meilleures performances en termes de SPS moyen. MUSCLE-p, T-COFFEE et MUSCLE ont les meilleures performances en termes de CS moyen.

Enfin, les résultats de l'expérience menée sur Balibase 2.0 confirment que notre méthode fournit une grande précision statistique significative en termes de scores SPS, une moyenne précision statistique en termes de scores CS et un temps d'exécution assez raisonnable.

VI.2) Résultats et discussion

Dans cette section la méthodologie expérimentale suivie est détaillées. Dans le réglage du comportement de l'algorithme en fonction de ses paramètres déterminants, nous avons utilisé l'instance BB11001 de l'ensemble de données RV11 de Balibase 3.0. Notons que ces paramètres fixés peuvent ne pas donner le même résultat pour une autre instance du problème MSA. Donc une étude approfondie de chaque instance pour choisir les meilleurs paramètres est nécessaire (travail fastidieux et très couteux en temps).

La comparaison de GPLS-GA et M-NSGA-II pour montrer la performance de l'approche multiobjectif par rapport à l'approche mono-objectif est plus crédible car les deux algorithmes utilisent la même stratégie de recherche et les mêmes valeurs des paramètres. En plus la soustraction des deux valeurs des fonctions objectifs ($f_1 - f_2$) permet de passer à la formulation mono-objectif facilement. La comparaison de NW-M-NSGA-II et M-NSGA-II permet de montrer l'avantage d'ajouter une méthode exacte à une métaheuristique selon notre schéma, mais avec un temps de calcul relativement élevé.

Dans la comparaison des algorithmes proposés avec les autres méthodes de la littérature en termes des deux scores Balibase, nous avons utilisé l'ensemble de données (ref1, ref2, ref3, ref4, ref5) en raison de leurs performances avec d'autres algorithmes connexes. Cette expérimentation, nous a mené à constater que chaque algorithme a ses propres avantages et inconvénients lorsqu'il fait face à des instances particuliers de séquences. Ce qui peut rendre difficile pour un problème d'alignement donné de faire une sélection rationnelle d'un outil d'alignement approprié pour n'importe quel ensemble de séquences. Il convient de souligner que les différences de performances entre les meilleures méthodes n'apparaissent que lorsqu'elles sont moyennées sur un grand nombre de cas de test. Les résultats des expériences montrent que les algorithmes (NW-M-NSGA-II, M-NSGA-II) sont comparables à d'autres algorithmes dans le score de la somme des paires SPS, tout en montrant également que les scores CS sont moyens.

Conclusion et perspectives

Ces dernières décennies, l'avancement technologique ne cesse de faire croître la quantité de données biologiques disponible, qui ne peut être traitée sans l'aide de la bioinformatique. Le défi de la bioinformatique est de pouvoir analyser et interpréter ces quantités massives de données pour mieux comprendre ces processus biologiques. L'analyse in silico de ces données biologiques pour produire des connaissances significatives implique le développement de nouvelles méthodes performantes, rapides et de qualité. Plusieurs travaux existants passent en revue les raisons de l'utilisation de l'optimisation multiobjectif dans le domaine de la bioinformatique et de la biologie computationnelle. Le travail présenté dans cette thèse est essentiellement consacré à l'analyse par comparaison de séquences biologiques. L'un des moyens les plus utilisés pour la comparaison de séquences est l'alignement. Ce puissant outil est d'importance cruciale pour les biologistes car il permet de répondre à plusieurs questions posées. La réalisation d'un bon alignement multiple de séquences augmente significativement la qualité des prédictions. Un alignement sera considéré comme bon s'il fait concorder un nombre élevé de positions (identité ou substitution conservative) avec un nombre minimal d'insertions ou délétions (ce sont les deux fonctions objectifs considérés dans cette thèse).

Les méthodes traditionnelles de résolution de MSA traitent ce problème comme problème d'optimisation mono-objectif. L'utilisation d'une seule fonction objectif permettant d'assigner un score à chaque alignement, et de fournir comme résultat le seul alignement du meilleur score, peut ne pas intéresser le biologiste. L'enjeu est double, on doit trouver des solutions efficaces à ces problèmes et biologiquement acceptables. Les expériences démontrent que la caractéristique la plus favorable de l'approche multiobjectif proposée réside dans sa capacité à générer, pour toute instance MSA, plus d'un seul alignement proche-optimal. Cette fonctionnalité aidera le décideur à évaluer et à sélectionner l'alignement de séquences multiples biologiquement pertinent. Ce que nous ne pouvons pas réaliser avec une seule solution.

L'objectif principal de ce travail de recherche est d'observer la contribution du multiobjectif dans la résolution du problème MSA de séquences protéiques en termes de qualité biologique et de scores mathématiques des séquences alignées. De nombreux algorithmes d'alignement multiple de séquences protéiques à base de score existent, et de nouveaux outils sont constamment développés et publiés. Les méthodes exactes déterminent l'alignement optimal, mais ne peut être utilisées que pour de petites séquences. Les méthodes progressives sont reconnues comme très rapides et donnent des résultats assez satisfaisants mais leur inconvénient est le fait de s'arrêter aux minimums locaux. L'approche itérative est une manière très simple et efficace permettant d'améliorer des méthodes d'alignement multiples. Le NSGA-II est un algorithme évolutionnaire multiobjectif basé sur l'approche Pareto qui est l'un des algorithmes les plus efficaces pour l'optimisation multiobjectif. L'hybridation est une technique basée sur l'idée que plusieurs méthodes combinées de manière appropriée peuvent produire de meilleurs résultats que si elles étaient appliquées séparément. Nous avons conçu une méthode de recherche locale GPLS qui fonctionne sur les positions des gaps pour améliorer tous les descendants produits par NSGA-II (M-NSGA-II). Pour augmenter la précision de certains alignements produits par M-NSGA-II, nous avons appliqué l'algorithme exact de Needleman et Wunsch sur des sous-alignements de quelques solutions du front de Pareto. L'inconvénient de cette hybridation est l'augmentation du temps de calcul. La stratégie de recherche gouvernée par le paramètre *nbrg* (un certain nombre de générations successives sans amélioration) permet une meilleure exploration et exploitation de l'espace de recherche. Cet espace est déterminé par les alignements de taille *L* (pour *L,* allant de max$|S_i|$ à *longmax* avec un pas de 1 gap : géré par l'operateur de mutation). Après avoir épuisé la recherche dans l'espace d'alignement de taille *L*, par l'algorithme

M-NSGA-II, comme dernier soutien on fait appel à la méthode exacte avec une probabilité de μ=0.75, pour essayer d'améliorer les solutions obtenues jusqu'à présent, avant de passer à L+1.

Enfin, l'approche d'optimisation multiobjectif apporte principalement l'avantage de fournir un ensemble d'alignements qui représentent le compromis entre insertion/délétion et la mise en correspondance des symboles entre les séquences, avec plusieurs caractéristiques souhaitables pour le problème d'alignement multiple de séquences. Malgré cela, il existe encore un besoin de nouvelles méthodologies impliquant des moyens pour surmonter les performances limitées des outils MSA. Pour la méthode proposée, le premier effort futur devrait se concentrer sur l'amélioration de la métrique CS, en introduisant le score du nombre de colonnes totalement conservé comme troisième fonction objectif. Et d'adopter ensuite, l'approche hyper-heuristique qui nous semble plus prometteuse car elle sélectionne la bonne méthode (parmi un groupe de méthodes) durant le processus de recherche.

Références

Ahmia, I. and Aïder, M. (2019). A novel metaheuristic optimization algorithm: The monarchy metaheuristic. Turkish Journal of Electrical Engineering and Computer Sciences 27(1):362-376. DOI : 10.3906/elk-1804-56.

Altschul, S. F., Carroll, R. J. and Lipman, D. J. (1989). Weights for data related by a tree. Journal Molecular Biology 207 pp. 647-653.

Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J. (1990). Basic local alignment search tool. J. Mol. Biol. 215: 403 – 410.

Altschul, S.F. (1991). Amino acid substitution matrices from an information theoretic perspective. J. Mol. Biol. 219:555–565.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research, 25(17), 3389–3402. https://doi.org/10.1093/nar/25.17.3389

Andres-Toro B., Giron-Sierra J.M., Fernandez-Blanco P., Lopez-Orozco J.A., and Besada-Portas E. (2004) Multiobjective Optimization and Multiobjective Control of the Beer Fermentation Process with the Use of Evolutionary Algorithms. J. Zhejiang Univ. SCIENCE, vol. 5, no. 4, pp. 378-389.

Argos, P.A. (1987). Sensitive procedure to compare amino acid sequences. J. Mol. Biol. 193:385–396.

Augerat, P., Belenguer, J.M., Benavent, E., CorberÃ¡n, A., and D. Naddef. (1998) Separating capacity constraints in the {CVRP} using tabu search. European Journal of Operational Research, 106(2-3) :546 – 557.

Azimi, Z.N. (2005). Hybrid heuristics for examination timetabling problem. Applied Mathematics and Computation, 163(2) :705 – 733.

Banka, H. and Mitra, S. (2006). Evolutionary biclustering of gene expressions. Ubiquity, 7(42):1–12

Basseur, M. (2005). Conception D'algorithmes Coopératifs Pour L'optimisation Multiobjectif : Application Aux Problèmes D'ordonnancement De Type Flow-Shop. Université des sciences et technologies de Lille U.F.R. D'I.E.E.A. thèse pour obtenir le grade de Docteur de l'U.S.T.L.

Basseur, M., Seynhaeve, F. and Talbi, EG. (2002). Design of Multi-objective Evolutionary Algorithm: Application to the Flow-shop Scheduling Problem. In Congress of evolutionary computation. CEC'02, pages 1151-1156, Honolulu, Hawaii, USA.

Beasley, D., Bull, D. and Martin, R. (1993). A sequential niche technique for multimodal function optimization. Evolutionary Computation, 1(2):101–125.

Becker, E., Cotillard, A., Meyer, V., Madaoui, H. and Guirois, R. (2007). Hmm.Kalign : a tool for generating sub-optimal hmm alignments. Bioinformatics. 23(22), pp. 3095-3097.

Bellman, R. *Dynamic Programming*, Princeton, Princeton University Press, 1957. Réimpression 2003, Dover Publication, Mineola, New-York, (ISBN 0-486-42809-5).

Bent, R. and Hentenrych, P.V. (2004). A two-stage hybride local search for the vehicle routing problem with time windows. Transportation science, 38(4): 515-530.

Berro, A. (2001). Optimisation multiobjectif et stratégie d'évolution en environnement dynamique. Thèse de doctorat.

Bezerra L.C.T., López-Ibáñez M., & Stützle T. (2013). An analysis of local search for the bi-objective bidimensional knapsack problem. In Proceedings of the 13th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP2013). Springer.

Biyanto, T., Fibrianto, H., Nugroho, G., Listijorini, E., Budiati, T. & Huda, H. (2015). Duelist Algorithm: An Algorithm Inspired by How Duelist Improve Their Capabilities in a Duel.

Biyanto, T., Matradji,, Irawan, S., Febrianto, H., Afdanny, N., Rahman, A., Gunawan, K., Pratama, D. and Bethiana,T. (2017). Killer Whale Algorithm: An Algorithm Inspired by the Life of Killer Whale. Procedia Computer Science. 124. 151-157. 10.1016/j.procs.2017.12.141.

Biyanto, T., Matradji, Syamsi, M., Fibrianto, H., Afdanny, N., Rahman, A., Gunawan, K., Pratama, J., Malwindasari, A.,Abdillah, A., Bethiana, T. and Putra, Y. (2017). Optimization of Energy Efficiency and Conservation in Green Building Design Using Duelist, Killer-Whale and Rain-Water Algorithms. IOP Conference Series: Materials Science and Engineering. 267. 012036. 10.1088/1757-899X/267/1/012036.

Blattner, F.R., Plunkett, G., Bloch, C.A., Perna, N.T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J.D. Rode, C.K., Mayhew, G.F., Gregor, J., Davis, N.W., Kirkpatrick, H.A., Goeden, M.A., Rose, D.J., Mau, B. and Shao, Y. (1997). The complete genome sequence of Escherichia coli K-12. Science, vol. 277, p. 1453-1462 (PMID 9278503).

Bleuler, S., Preli´c A and Zitzler E. (2004). An EA Framework for Biclustering of Gene Expression Data. In: Congress on Evolutionary Computation (CEC 2004), 166–173. IEEE, Piscataway, NJ

Blum, C., Roli, A. and Sampels M. (2008). Hybrid metaheuristics: an emerging approach to optimization, volume 114. Springer.

Boisson. J-C. (2008). Parallel multi-objective algorithms for the molecular docking problem. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB).

Branke, J. and Deb, K. (2005). Integrating User Preferences into Evolutionary MultiObjective Optimization. In: Jin Y (ed.) Knowledge Incorporation in Evolutionary Computation, 461–477. Springer, Berlin Heidelberg. ISBN 3-540-22902-7

Caraway, R.L., Morin, T.L., & Moskowitz, H. (1990). Generalized dynamic programming for multicriteria optimization. European Journal of Operational Research, 44(1), 95-104.

Charnes, A., Cooper, W. W. and Ferguson, R. O. (1955). Management Science. vol. 1, issue 2, 138-151.

Cheng Y, Church G (2000). Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol, 8:93–103

Chowdhury. A and Garai G. (2017). A review on multiple sequence alignment from the perspective of genetic algorithm. Genomics 109 419–431. Elsevier Inc. available at ScienceDirect Genomic.

Coello Coello, C. A., Christiansen, A. D. and Aguirre, A. H. (1995). Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithms. In Proceedings of the Seventh International Conference on Tools with Artificial Intelligence (ICTAI'95), pages 20-23, Herndon, VA, USA, November.

Coello Coello, C.A. (2000). Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In: 2000 Congress on Evolutionary Computation, vol. 1, 30–37. IEEE Service Center, Piscataway, New Jersey.

Coello Coello C.A., Van Veldhuizen D.A. and Lamont G.B. (2002). Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic.

Coello Coello C.A., and Lamont G.B. (2004). Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapour. ISBN 981-256-106-4.

Coello Coello C.A., and Lamont G.B. and Van Veldhuizen D.A (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York, second edn. ISBN 978-0-387-33254-3.

Coello Coello, C. A. (2017). Recent results and open problems in evolutionary multiobjective optimization. In Proceedings of 6th *International Conference on Theory and Practice of Natural Computing (TPNC 2017).* Springer.

Cohen J. (2004). "Bioinformatics, an Introduction for Computer Scientists," ACM Computing Surveys, vol. 36, no. 2, pp. 122-158.

Cook, S. (1971). The complexity of theorem proving procedures. In Proceedings of the third annual ACM symposium on Theory of Computing, pages 151–158.

Cormen, T.H., Leiserson, C.E., Rivest, R. L. and Stein, C. (2002). Introduction à l'algorithmique 2ème édition, chapitres 2, 3, 34, série Sciences Sup, Édition DUNOD.

Corne, D.W. and al, (2001). PESA II: Region-based Selection in Evolutionary Multiobjective Optimization. in Procedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), p. 283-290.

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustring. Nucleic acids research. 16(22), pp. 10881-10890.

Cotik V., Zaliz R.R., and Zwir I. (2005). A Hybrid Promoter Analysis Methodology for Prokaryotic Genomes. Fuzzy Sets and Systems, vol. 152, pp. 83-102, 2005.

Cotta, C., Aldana, J.F., Nebro, A.J. and Troya. J.M. (1995) Hybridizing genetic algorithms with branch and bound techniques for the resolution of the TSP. In D.W.Pearson, N.C.Steele, and R.F.Albrecht, editors, Artificial Neural Nets and Genetic Algorithms 2, page 277-280. Springer-Verlag.

Cotta, C. and Troya, J.M. (2003). Embedding Branch and Bound within Evolutionary Algorithms. Dept. Lenguajes y Ciencias de la Computacion, University of Malaga ETSI Informatica (3.2.49), Campus de Teatinos, 29071 - Malaga, SPAIN.

Cotta, C. (2006). Scatter search with path relinking for phylogenetic inference. European Journal of Operational Research, 169(2):520 – 532. Feature Cluster on Scatter Search Methods for Optimization.

Croes, G. A. (1958). A method for solving traveling salesman problems. Operations Research, vol. 6, no. 6, pages 791-812.

Curteanu S., Leon F., and Galea D. (2006). Alternatives for Multiobjective Optimization of a Polymerization Process. J. Applied Polymer Science.

Cutello V., Narzisi G., and Nicosia G. (2006a). A Multi-Objective Evolutionary Approach to the Protein Structure Prediction Problem. J. Royal Soc. Interface, vol. 3, no. 6, pp. 139-151.

Cutello, V. Lee, D. Nicosia, G. Pavone, M. and Prizzi, I. (2006b). Aligning Multiple Protein Sequences by Hybrid Clonal Selection Algorithm with Insert-Remove-Gaps and BlockShuffling Operators," in Lecture Notes in Computer Science. vol. 4163 Berlin / Heidelberg: Springer, pp. 321-334.

Cvetkovic D, Parmee IC (2002). Preferences and their Application in Evolutionary Multiobjective Optimisation. IEEE Transactions on Evolutionary Computation, 6(1):42–57.

Dabba, A., Tari, A K. and Zouache, D. (2019). Multiobjective artificial fish swarm algorithm for multiple sequence alignment. INFOR Information Systems and Operational Research 58(4):1-22.

Dale M.B. and Dale P.T. (1994). Classification with Multiple Dissimilarity Matrices. Coenoses, vol. 9, no. 1, pp. 1-13.

Dantzig, G. (1963). Linear programming and extensions. Princeton University Press.

Davidson et al., (1997) International Journal of Data Libraries, 1 :36-53.

Day RO, Zydallis JB and Lamont GB (2002). Solving the Protein structure Prediction Problem through a Multi-Objective Genetic Algorithm. In: Proceedings of IEEE/DARPA International Conference on Computational Nanoscience (ICCN'02), 32–35

Dayhoff, M., Richard V., Chang, Marie A., and Sochard, R. (1965). Atlas of Protein Sequence and Structure.

Dayhoff M. O., Schwartz R. M. and Orcutt B. C. (1978). A model of evolutionary change in proteins. Atlas of Protein Structure, 5, Suppl. 3, 345-352.

De Queiroz A. de, Donoghue M.J. and Kim J. (1995). Separate versus Combined Analysis of Phylogenetic Evidence. Ann. Rev. Ecology and Systematics, vol. 26, pp. 657-681.

Deb, K. and Reddy, A. (2003). Reliable classification of two-class cancer data using evolutionary algorithms. BioSystems, 72(1):111–129.

Deb, K., Mitra, K., Dewri, R. and Majumdar, S. (2004).Towards a Better Understanding of the Epoxy Polymerization Process Using MultiObjective Evolutionary Computation. Chemical Eng. Science, vol. 59, no. 20, pp. 4261-4277,

Deb, K. and Jain, S. (2014). An evolutionary many-objective optimization algorithm using reference-pointbased nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, 18(4):577–601.

Deorowicz, S., Debudaj-Grabysz, A. and Gudys, A. (2014). Kalign-lcs a more accurate and faster variant of kalign algorithm for the multiple sequence alignment problem. In Man-Machine Interactions, Springer, pp. 495-502.

Derrien, V. (2008). Heuristiques pour la résolution du problème d'alignement multiple. Thèse de doctorat. Université d'Angers. N° d'ordre 885

Dhaenens, C. (2005). Optimisation Combinatoire Multiobjectif: Apport Des Méthodes Coopératives Et Contribution A L'extraction De Connaissances. Université des sciences et technologies de Lille U.F.R. D'I.E.E.A. thèse pour obtenir le grade d'habilitation à diriger des recherches de l'U.S.T.L.

Djikstra, E. W., (1959). A note on two problems in connexion with graphs. Numerische Mathematik, vol. 1, 1959, p. 269-271.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Trans. on Man. Cyber. Part B, 26 :29–41.

Dumitrescu, I., & Stützle, T. (2003). Combinations of local search and exact algorithms. In proceeding of EvoWorkshops on Applications of Evolutionary Computation. Springer.

Ebrahimi, A. and Khamehchi, E.. (2016). Sperm Whale Algorithm: an Effective Metaheuristic Algorithm for Production Optimization Problems. Journal of Natural Gas Science and Engineering. 29. 10.1016/j.jngse.2016.01.001.

Edgar, R.C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research. 32(5), pp.1792-1797. doi: 10.1093/nar/gkh340. Print 2004.

Erickson M., Mayer A. and Horn J., (2001). The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. Dans : Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, et David Corne, éditeurs, First International Conference on Evolutionary MultiCriterion Optimization, pp. 681-695. Springer-Verlag. Lecture Notes in Computer Science, N o 1993, pp. 681-695.

Etzold et al.(1996), Methods Enzymol., ,266 :114-28

Everson R.M. and Fieldsend J.E. (2006). Multi-Class ROC Analysis from a Multi-Objective Optimisation Perspective," Pattern Recognition Letters, vol. 27, pp. 918-927.

Feng, D.-F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25, 351-360

Ferligoj A. and Batagelj V. (1992). Direct Multicriterion Clustering. J. Classification, vol. 9, pp. 43-61.

Fitch, W.M. and Margoliash, E. (1967). Construction of phylogenetic trees. Science, 155:279– 284.

Fleischmann, R., Adams, M., White, O., Clayton, R., Kirkness, E., Kerlavage, A., Bult, C., Tomb, J., Dougherty, B., and Merrick, J. (1995). Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science, 269 5223, 496-512.

Fonseca, C.M. (1995). Multiobjective genetic algorithms with applications to control engineering problems. PhD thesis, University of Sheffield.

Fourman, M.P. (1985). Compaction of symbolic layout using genetic algorithm. In proceedings of the first international conference on genetic algorithms (ICGA)

French, A. P., Robinson, A. C., and Wilson, J. M.. (2001). Using a hybrid geneticalgorithm/branch and bound approach to solve feasibility and optimization integer programming problems. Journal of Heuristics, 7:551–564.

Garg S. and Gupta S.K. (1999). Multiobjective Optimization of a Free Radical Bulk Polymerization Reactor Using Genetic Algorithm. Macromolecular Theory and Simulations, vol. 8, pp. 46-53.

Geem, Z.W., Kim, J H. and Loganathan G.V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. Simulation, 76 (2), 60-68.

Ghorbani, N. and Ebrahim, B. (2014). Exchange market algorithm. Applied Soft Computing. 19. 177–187. 10.1016/j.asoc.2014.02.006

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & Operations Research, 13(5) :533–549.

Glover, F. and Laguna, M. (1997). Tabu search. Kluwer Academic Publishers.

Goldberg, D.E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, pages 41- 49. Lawrence Erlbaum.

Goldberg D.E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley publishing company.

Gondro, C. and Kinghorn, B.P. (2007). A simple genetic algorithm for multiple sequence alignment, Genet. Mol. Res. 6 964–982.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. Journal of Molecular Biology, Vol. 162 :705–708.

Guedas, B., Depince, P., Gandibleux, X. (2010). Vers un algorithme évolutionnaire multiobjectif ad-hoc pour l'optimisation multidisciplinaire. ROADEF, Toulouse, France.

Gupta, S. K., Kececioglu, J. and Schäffer, A. A. (1995). Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. J. Comput. Biol, 2: pp. 459-472.

Halsall-Whitney H., Taylor D., and Thibault J. (2003). Multicriteria Optimization of Gluconic Acid Production Using Net Flow. Bioprocess and Biosystems Eng., vol. 25, pp. 299-307.

Handl J. and Knowles J. (2006) Feature Subset Selection in Unsupervised Learning via Multiobjective Optimization. International Journal Computational Intelligence Research, vol. 2, no. 3, pp. 217-238.

Handl, J. and Knowles J. (2007). An Evolutionary Approach to Multiobjective Clustering. IEEE Trans. Evolutionary Computation, vol. 11, no. 1, pp. 56-76.

Handl, J., Kell D.B. and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. IEEE-ACM Transactions on Computational Biology and Bioinformatics, 4(2):279–292

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4, vol. 4, no 2, 1968, p. 100–107 (DOI 10.1109/TSSC)1968.300136).

Henikoff, S. and Henikoff J.G. (1993). Performance evaluation of amino acid substitution matrices. Proteins Struct. Funct. Genet. 17:49–61.

Hesper, B. and Hogeweg, P. (1970). Bioinformatica: een werkconcept. Kameleon 1(6): 28–29. (In Dutch.) Leiden: Leidse Biologen Club.

Higuera C, Villaverde AF, Banga JR, Ross J, Morán F (2012). Multi-Criteria Optimization of Regulation in Metabolic Networks. PLOS ONE 7(11): e41122. https://doi.org/10.1371/journal.pone.0041122

Hiriart-Urruty, J.B. (2013) convex analysis and optimization in the past 50 years: some snapshots. Special issue in the series Optimization and Its Applications, Springer.

Hogeweg, P., Hesper, B (1984). The alignment of sets of sequences and the construction of phyletic trees: An integrated method. J Mol Evol 20, 175–186.

Holland, J. (1975). Adptation in natural and aritificial systems. University of Michigan Press.

Horn, J., Nafpliotis, N. and Goldberg, D.E. (1994). "A Niched Pareto Genetic Algorithm for Multiobjective Optimization". In : Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol. 1, pp. 82-87, Piscataway, New Jersey. IEEE Service Center.

Hubley, R., Zitzler, E., Siegel, A. and Roach, J. (2002). Multiobjective Genetic Marker Selection. In: Advances in Nature-Inspired Computation: The PPSN VII Workshops, 32–33. University of Reading, UK.

Ishibuchi, H. and Murata, T. (1998). A multi-objective genetic local search algorithm and its application to the flowshop scheduling. IEEE transactions on systems, Man and Cybernetics, 28: 392- 403.

Jackman, SD., Vandervalk, BP., Mohamadi, H., Chu, J., Yeo, S., Hammond, SA., Jahesh, G., Khan, H., Coombe, L., Warren, RL. and Birol, I. (2017) ABySS 2.0: assemblage économe en ressources de grands génomes à l'aide d'un filtre Bloom. Genome Research, 27: 768-777.

Jahuira, C.A.R. (2002). Hybride genetic algorithm with techniques applied to TSP. In second international workshop on intelligent systems design and application, pages 119-124. dynamic Publishers.

Jaimes, A.L. and Coello Coello, C.A. (2008). An Introduction to Multi-Objective Evolutionary Algorithms and Some of Their Potential Uses in Biology. In book: Applications of Computational Intelligence in Biology (pp.79-102). DOI: 10.1007/978-3-540-78534-7_4.

Johnson, DB. (1977). Efficient algorithms for shortest paths in sparse networks », Journal of the ACM, vol. 24, no 1, p. 1–13 (DOI 10.1145/321992.321993).

Jourdan, L., Basseur, M., & Talbi, E.G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. European Journal of Operational Research, 199, 620–629.

Jozefowiez, N. (2004). Modélisation et résolution approchée de problèmes de tournées de véhicules. PhD thesis, Université des Sciences et Technologies de Lille.

Kantorovich, L.V. (1960). Mathematical Methods of Organizing and Planning Production. Management Science, vol. 6, no. 4, pages 366-422.

Karaboga, D. (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report - TR06. Technical Report, Erciyes University.

Katoh .K, Misawa .K, Kuma K. I. and Miyata T., 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Research 30(14):3059-3066. doi: 10.1093/nar/gkf436.

Keel B N., Deng B. and Moriyama E N. (2018). MOCASSIN-prot : une approche de clustering multi-objectifs pour les réseaux de similarité de protéines. *Bioinformatics*, Volume 34, Numéro 8, 15 avril 2018, Pages 1270-1277.

Kemena, C. and Notredame, C. (2009). Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics, vol. 25, no. 19, pp. 2455–2465, 2009.

Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimization. IEEE International Conference on Neural Networks, Perth, Australia.

Kim Y., Street W.N., and Menczer F. (2002). Evolutionary Model Selection in Unsupervised Learning. Intelligent Data Analysis, vol. 6, no. 6, pp. 531-556.

Kirkpatrick, S., Gelatt, Jr. D. and Vecchi Mario P.(1983). Optimization by simulated annealing. Science, 220(4598):671-680.

Klau, GW., Ljubic,I., Moser, A., Mutzel,P., Neuner,P., Pferschy,U., Raidl, G. and Weiskircher, R. (2004). Combining a Memetic Algorithm with Integer Programming to Solve the Prize-Collecting Steiner Tree Problem. Favoritenstraße 9–11/186, 1040 Vienna, Austria

Knowles, J.D., & Corne, W. (1999). The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization, In Proceedings of 1999 Congress on Evolutionary Computation (CEC'99). IEEE.

Knowles, J.D., Corne, W., & Oates, M. J. (2000). The Pareto-envelope based selection algorithm for multiobjective optimization. In proceedings of the sixth international conference on parallel problem solvi*ng from nature.*

Knowles J.D., Watson R.A., and Corne D.W. (2001). Reducing Local Optima in Single-Objective Problems by Multi-Objectivization. Proc. First Int'l Conf. Evolutionary Multi-Criterion Optimization, pp. 269-283.

Krueger, J.M. (1990). Somnogenic activity of immune response modifiers. Trends in Pharmacological Sciences, 11(3):122 – 126.

Kupinski M.A. and Anastasio M.A. (1999). Multiobjective Genetic Optimization of Diagnostic Classifiers with Implications for Generating Receiver Operating Characteristic Curves," IEEE Trans. Medical Imaging, vol. 18, no. 8, pp. 675-685.

Kurwase, F. (1984). A variant of evolution Strategie for vector optimisation, Ph, D Thesis, Vnderbilt University, Nashville, Tennesse.

Lambert, C. (2003). Développement d'une méthode automatique fiable de modélisation de la structure tridimensionnelle des protéines par homologie et application au protéome de Brucella melitensis. Thèse de doctorat. FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX NAMUR.

Land, A.H and Doig, A.G. (1960). An automatic method of solving discrete programming problems. Econometrica: Journal of Econometric Society, page 497-520.

Landa B.R. and Coello Coello C.A. (2006). Solving Hard Multiobjective Optimization Problems Using ε-Constraint with Cultured Differential Evolution. In: Runarsson TP, Beyer HG, Burke E, Merelo-Guerv´os JJ, Whitley LD, Yao X (eds.) Parallel Problem Solving from Nature - PPSN IX, 9th International Conference, 543–552. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland

Lanning O.J., Habershon S., Harris K.D., Johnston R.L., Kariuki B.M., Tedesco E., and Turner G.W. (2000) "Definition of Guiding Function in Global Optimization: A Hybrid Approach Combining Energy and R-Factor in Structure Solution from Powder Diffraction Data," Physics Letters, vol. 317, pp. 296-303.

Lassmann, T., and Sonnhammer, E. L. (2005). Klign an accurate and fast multiple sequence algorithm. BMC bioinformatics 6(1), p 298.

Lee, I., Kim, S. and Zhang, B. (2003). DNA sequence optimization using constrained multi-objective evolutionary algorithm. Evolutionary Computation, CEC'03.

Lee, I., Kim, S. and Zhang, B. (2004). Multi-objective Evolutionary Probe Design Based on Thermodynamic Criteria for HPV Detection. Lecture Notes in Computer Science, 3157:742–750.

Li, F., Zhao, C. and Wang, L. (2014). Molecular-targeted agents combination therapy for cancer: developments and potentials. Int J Cancer 134(6):1257–1269.

Li, M., Zhao, H., Weng, X and Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search. Advances in Engineering Software. 92. 65-88. 10.1016/j.advengsoft.2015.11.004.

Lipman DJ. and Pearson WR. (1985). Rapid and sensitive protein similarity searches. Science. 22 ; 227(4693):1435-41. doi:10.1126/science.2983426. PMID: 2983426.

Lipman, D. J., Altschul, S, F. and Kececioglu, J. D. (1989). A tool for multiple sequence alignment. Proc. Nat. Acad. Sci. USA 86: pp. 4412-4415.

Liu Y., Oezyer T., Alhajj R., and Barker K. (2005). Integrating MultiObjective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering. Informatica, vol. 29, pp. 33- 40.

Liu X, Krishnan A, Mondry A (2005). An Entropy-based gene selection method for cancer classification using microarray data. Feedback

Loughlin, D. and Ranjithan, S. (1997). The neighborhood contraint method : A genetic algorithm based multiobjective optimization technique. In Back, T., editor, Seventh Int. Conf. on Genetic ICGA'97,

Mahdi, S., & Nini, B. (2021). Improved Memetic NSGA-II Using a Deep Neighborhood Search. *International Journal of Applied Metaheuristic Computing (IJAMC), 12*(4), 138-154. http://doi.org/10.4018/IJAMC.2021100108

Malard J, Heredia-Langner A, Baxter D, Jarman K and Cannon W (2004). Constrained de novo peptide identification via multi-objective optimization. Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International

Mandal C., Gudi R.D., and Suraishkumar G.K. (2005). Multi-Objective Optimization in Aspergillus Niger Fermentation for Selective Product Enhancement. Bioprocess and Biosystems Eng., vol. 28, pp. 149-164.

Martin, R. (1990) Single-interval learning by simile within a simulated hebbian neural network. Computers & Mathematics with Applications, 20(4-6) : 217 – 226.

Martin, O.C., Otto, S.W. and Felten. E.W. (1991). Large-step markov chains for the traveling salesman problem. Complex Systems, 5(3) :299–326.

Martin, O.C., and Otto, S.W. (1996). Combining simulated annealing with local search heuristics. Annals of Operations Research, 63(1): 57-75.

Mathé, C., Sagot, M-F., Schiex,T. and Rouzé, P. (2002). Current methods of gene prediction, their strengths and weaknesses, *Nucleic Acids Research*, vol. 30, nº 19, p. 4103-4117.

Mateus da Silva, F.J., Sanchez Pérez J.M., Gomez Pulido, J.A., Miguel A. and Rodriguez, V. (2011). Parallel Niche Pareto AlineaGA – an Evolutionary Multiobjective approach on Multiple Sequence Alignment Journal of Integrative Bioinformatics, 8(3):174.

Meister G. and Tuschl T. (2004). Mechanisms of gene silencing by double-stranded RNA. Nature. 16;431(7006):343-9. doi: 10.1038/nature02873. PMID: 15372041.

Meng X., Liu Y., Gao X. and Zhang H. (2014) A New Bio-inspired Algorithm: Chicken Swarm Optimization. In: Tan Y., Shi Y., Coello C.A.C. (eds) Advances in Swarm Intelligence. ICSI 2014. Lecture Notes in Computer Science, vol 8794. Springer, Cham. https://doi.org/10.1007/978-3-319-11857-4_10

Metz C.E. (1978). Basic Principles of ROC Analysis. Seminars in Nuclear Medicine, vol. 8, no. 4, pp. 283-298.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. Journal of the American Statistical Association, vol. 44, no. 247, pages 335-341.

Meunier, H., Talbi, EG. Reininger, P. (2000). A multiobjective genetic algorithm for radio network optimization. In CEC, volume 1, page 317-324. IEEE service center.

Meunier, H. (2002). Algorithmes évolutionnaires parallèles pour l'optimisation multi-objectif de réseaux de télécommunications mobiles. Thèse de doctorat.

Michaud SR, Zydallis JB, Lamont GB and Pachter R (2001). Scaling a genetic algorithm to medium-sized peptides by detecting secondary structures with an analysis of building blocks. In: Proceedings of the First International Conference on Computational Nanoscience, 29–32

Miklos I., Lunter G. A. and Holmes I. (2004). A Long Indel Model For Evolutionary Sequence Alignment. Molecular Biology and Evolution, 21(3) :529–540.

Miramontes, P. (1989). ADN et ARN physico - chimiques, contraintes Automates cellulaires et évolution moléculaire. Rapport, atelier "Cellular Automata: théorie et applications" à Los Alamos, Nouveau - Mexique.

Mitra, K., Majumdar, S. and Raha, S. (2004). Multiobjective Dynamic Optimization of Epoxy Polymerization Process. Computer and Chemical Eng., vol. 28, no. 12, pp. 2583-2594.

Mitra S. (2005), "Computational Intelligence in Bioinformatics," Trans. Rough Sets, pp. 134-152.

Mitra, S., Banka, H. and Pal, S. (2006). A MOE framework for Biclustering of Microarray Data. Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, 1154–1157.

Mitra, S. and Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition, 39(12):2464–2477 65.

Mizuguchi, K., Deane, C.M., Blundell, T.L. and Overington, J.P.(1998). Homstrad : A database of protein structure alignments for homologous families. Protein Science, Vol. 7 :2469– 2471.

Monod J., Wyman J et Changeux J-P. (1965). On the nature of allosteric transitions: A plausible model. *Journal of molecular biology*, Pages 88-118 DOI https://doi.org/10.1016/S0022-2836(65)80285-6

Morgat, A. and Rechenmann, F. (2002) Biological data and knowledge modeling Med Sci (Paris); 18 : 366–374.

Morgenstern, B. (1999) Dialign2 : improvment of the segment-to-segment approach to multiple sequence alignment. Bioinformatics, Vol. 15 (3) :211–218, 1999.

Morita M., Sabourin R., Bortolozzi F., and Suen C.Y. (2003). Unsupervised Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Word Recognition. In Proceeding Seventh Int'l Conf. Document Analysis and Recognition, pp. 666-671

Muniglia L., Kiss L.N., Fonteix C., and Marc I. (2003). Multicriteria Optimization of a Single-Cell Oil Production. European J. Operational Research, vol. 153, no. 2, pp. 360-369.

Nakarani, S. and Tovey, C. (2004). HONEY BEE Algorithm: A Biologically Inspired Approach to Internet Server Optimization. A biological inspired approach to internet server optimization. 13-15.

Naznin, F., Sarker, R. and Essam, D. (2011).Vertical decomposition with genetic algorithm for multiple sequence alignment, BMC Bioinf. 12, 353.

Naznin, F., Sarker, R. and Essam, D. (2012). Progressive Alignment Method Using Genetic Algorithm for Multiple Sequence Alignment. IEEE Transactions on Evolutionary Computation 16(5), pp, 615-631. DOI: 10.1109/TEVC.2011.2162849

Needleman, S.B. and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48(3): 443–453.

Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. The Computer Journal, Volume 7, Issue 4, Pages 308–313, https://doi.org/10.1093/comjnl/7.4.308.

Neshat, M., Sepidnam, G. and Sargolzaei, M. (2012). Swallow swarm optimization algorithm: A new method to optimization. Neural Computing and Applications. 23. 10.1007/s00521-012-0939-9.

Nicolas, HB., Ropelewski, AJ. and Deerfield, DW. (2002). Strategies for multiple sequence alignment. Biotechniques, 32: 572-578.

Nimmegeers P., Telen D., Logist F and Impe J V. (2016). Dynamic optimization of biological networks under parametric uncertainty. BMC Systems Biology. Volume : 10 20 DOI 10.1186/s12918-016-0328-6

Ninio, J. (1971) Kinetic amplification of enzyme discrimination. Biochimie, 57, 587- 595.

Notredame, C and Higgins D. G. (1996). SAGA: Sequence Alignment by Genetic Algorithm. *Nucleic Acids Research*, Volume 24, Issue 8, Pages 1515–1524, https://doi.org/10.1093/nar/24.8.1515.

Notredame C., Holm L and D.G. (1998). COFFEE: An objective functions for multiple sequence alignments. Bioinformatics, Vol. 14, No. 5, pp. 407-422.

Notredame, C., Higgins, D.G. and Heringa, J. (2000). T-coffee : A novel method for multiple sequence alignments. Journal of Molecular Biology, Vol. 302 :205–217.

Notredame, C. (2002). Recent progresses in MSA: a survey". pharmacogenomic, 3:pp. 1–14.

Noutahi E. and El-Mabrouk N. (2018). GATC: A Genetic Algorithm for gene Tree Construction under the Duplication-Transfer-Loss model of evolution. BMC Genomics 19(Suppl 2):102 doi:10.1186/s12864-018-4455-x

Olivier, C. (2014). Langages formels. Vuibert.

Ortuno F., Florido J. P., Urquiza J. M., Pomares H., Prieto A. and Rojas I. (2012). Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II. WCCI 2012 IEEE World Congress on Computational Intelligence. Brisbane, Australia.

Ortuno, F. M., Valenzuela, O., Rojas, F., Pomares, H., Florido, J. P., Urquiza, J. M. and Rojas, I. (2013). Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. Bioinformatics, vol. 29, no. 17, pp. 2112–2121.

O'Sullivan, O. Suhre, K., Abergel, C., Higgins, D.G. and Notredame, C. (2004) 3dcoffee : Combining protein sequences and structures within multiple sequence alignments. Journal of Molecular Biology, Vol. 340 :385–395.

Padberg, M. and Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM review, 33(1) :60-100.

Papadimitriou, C.H. (1994). Computational Complexity. Addison Wesley.

Paquete, L., Matias, P., Abbasi, M. and Pinheiro, M. (2014). MOSAL: software tools for multiobjective sequence alignment. Source Code for Biology and Medicine, 9:2.

Perutz, M., Rossmann, M. and Cullis, A. et al. (1960). Structure of Hæmoglobin: A Three-Dimensional Fourier Synthesis at 5.5-Å. Resolution, Obtained by X-Ray Analysis. Nature 185, 416–422. https://doi.org/10.1038/185416a0

Poladian L and Jermiin L (2006). Multi-objective evolutionary algorithms and phylogenetic inference with multiple data sets. Soft Computing-A Fusion of Foundations, Methodologies and Applications, 10(4):359–368

Portmann, M.C., Vignier, A., Dardihac, D. and Dezalay, D. (1998). Branch and bound crossed with GA to solve hybride flowshops. European Journal of Operational Research, 107(2): 389-400.

Puchinger, J., & Raidl, G.R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification. In Proceedings of International Work-conference on the Interplay between Natural and Artificial Computation (IWINAC 2005). Springer.

Putz H., Schoen J.C., and Jansen M. (1999). Combined Method for Ab Initio Structure Solution from Powder Diffraction Data. Journal of Applied Crystallography, vol. 32, pp. 864-870.

Rachmawati, L. and Srinivasan, D. (2006). Preference Incorporation in Multiobjective Evolutionary Algorithms: A Survey. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), 3385–3391. IEEE, Vancouver, BC, Canada

Raghava, G.P.S., Searle, S.M.J., Audley, P.C., Barber, J.D. and Barton, G.J. (2003). Oxbench : A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics, Vol. 4 :47.

Rajapakse M., Schmidt B., and Brusic V. (2006). Multi-Objective Evolutionary Algorithm for Discovering Peptide Binding Motifs. Proc. Fourth European Workshop Evolutionary Computation and Machine Learning in Bioinformatics, pp. 149-158.

Ranjani Rani R. and Ramyachitra D. (2016). Multiple sequence alignment using multi-objective based bacterial foraging optimization algorithm. Biosystems,Volume 150, Pages 177-189.

Ramstein, G. (2012). Application de techniques de fouille de données en Bio-informatique Ecole polytechnique de l'université de Nantes.

Rao, R. V., Savsani, J.V. and Balic, J. (2012). Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Engineering Optimization Volume 44, 2012 - Issue 12, Pages 1447-1462

Rashedi, E., Nezamabadi, H and Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. Information Sciences, Volume 179, Issue 13, Pages 2232-2248.

Reinert, K. (2003). Lecture notes on biological sequence analysis. Algorithmishe Bioinformatik.

Riaz, T., Wang, Y. and Li, K.B. (2004). Multiple sequence alignment using tabu search. In CRPIT '29 : Proceedings of the second conference on Asia-Pacific bioinformatics, pages 223–232. Australian Computer Society, Inc.

Risler J.L., Delorme M.O., Delacroix H. and Henaut A. (1988). Amino acid substitutions in structurally related proteins: A pattern recognition approach. Determination of a new and efficient scoring matrix. J. Mol. Biol. 204:1019–1029.

Ritzel, B. et al,. (1994). Using Genetic Algorithm to Solve a Multiple Objective Groundwater Pollution Containment Problem. Water Ressources Research 30, p. 1589-1603.

Roytberg M.A., Semionenkov M.N., and Tabolina O.Y. (1999). ParetoOptimal Alignment of Biological Sequences. Biofizika, vol. 44, no. 4, pp. 581-594.

Rubio-Largo, A., Vega-Rodrıguez, M. A. and Gonzalez-Alvarez, D. L. (2015). A Hybrid Multiobjective Memetic Metaheuristic for Multiple Sequence Alignment. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution, volume 4, issue 4, pp. 406-425.

Salzberg, S., Delcher, A., Kasif, S. and White, O. (1998). Microbial gene identification using interpolated Markov models, Nucleic Acids Research 26: 2, 544-548.

Schaffer, J.D. (1984). Multiple objective optimization with vector evaluated genetic algorithms. PhD thesis, Vanderbilt University.

Schenker, S. and Paquete, L. (2013). BiMuSA: An implementation for biobjective multiple sequence alignment problems CISUC Technical Report TR2013/03.

Schott, J.R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Schulze-Kremer S. (2003). Application of Evolutionary Computation to Protein Folding with Specialized Operators. Evolutionary Computation in Bioinformatics, pp. 163-191.

Seeluangsawat, P. and Chongstitvatana, P. (2005). A Multiple Objective Evolutionary Algorithm for Multiple Sequence Alignment. In proceedings of the 7th annual conference on Genetic an evolutionary computation, ACM, pp, 477-478

Sen, T., Raizadeh, M.E., & Dileepan, P. (1988). A branch-and-bound approach to the bicriterion scheduling problem involving total Flow time and range of lateness. Management Science, 34(2), 254-260.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. International Journal of Computational Science and Engineering. 6. 132-140. 10.1504/IJCSE.2011.041221.

Sharan, R. and Ideker, T. (2006).Modeling cellular machinery through biological network comparison. Nature biotechnology, 24, 427-433.

Sharma N.S., Ierapetritou M.G., and Yarmush M.L. (2005) Novel Quantitative Tools for Engineering Analysis of Hepatocyte Cultures in Bioartificial Liver Systems. Biotechnology and Bioeng., vol. 92, no. 3.

Sharma A. and Rani R. (2019). C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. Computer Methods and Programs in Biomedicine Volume 178, September 2019, Pages 219-235

Shin S, Lee I, Kim D and Zhang B (2005). Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing. Evolutionary Computation, IEEE Transactions on, 9(2):143–158

Sleator, RD. and Walsh, P. (2010). An overview of in silico protein function prediction. Archives of Microbiology, 192(3):151155.

Smith, T.F. and Waterman, M.S. (1981). Identification of Common Molecular Subsequences. Journal of Molecular Biology. 147: 195–197.

Spendley, W., Hext, G.R. and Himsworth, F.R. (1962). Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. Technometrics, 4, 441-461. http://dx.doi.org/10.1080/00401706.1962.10490033.

Srinivas, N. & Deb, K. (1994). Muiltiobjective optimization using non-dominated sorting in genetic algorithms. *Journal of* Evolutionary Computation, 2(3), 221-248.

Stewart, B.S., & White, C.C. (1991). Multiobjective A*. Journal of the ACM, 38(4), 775-814.

Someren, E.P., Wessels L.F.A., Backer E., and Reinders M.J.T., (2003) "Multi-Criterion Optimization for Genetic Network Modeling," Signal Processing, vol. 83, pp. 763-775.

Soto, W. and Becerra, D. (2014). A Multi-Objective Evolutionary Algorithm for Improving Multiple Sequence Alignments. Springer International Publishing Switzerland. S. Campos (Ed.): BSB 2014, LNBI 8826, pp. 73–82.

Stein, L. (2001). Genome annotation: from sequence to biology. Nature Reviews Genetics 2(7):493-503. doi: 10.1038/35080529.

Sternberg, M. J., Bates, P. A., Kelley, L. A. and MacCallum, R. M. (1999). Progress in protein structure prediction: assessment of CASP3." Curr Opin Struct Biol 9(3): 368-73.

Stoye, J., Moulton, V. and Dress, A.W. (1997). DCA: an efficient implementation of the divideand-conquer approach to simultaneous multiple sequence alignment. Comput. Apll. Biosci, 13 (6):625–626.

Stutzle, T. and Hoos, H. (1997). Max-min ant system and local search for traveling salesman problem. In Evolutionary Computation, IEEE International Conference on, page 309-314.

Subramanian, A.R., Weyer-Menkhoff, J., Kaufmann, M and Morgenstern, B. (2005). Dialign-t : An improved algorithm for segment-based multiple sequence alignment. BMC Bioinformatics, 6 : 66.

Surry P and Radcliffe N (1997). The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. Control and Cybernetics, 26(3):391–412

Taheri, J., & Zomaya, A.Y. (2010). RBT-Km: K-Means clustering for Multiple Sequence Alignment. *ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010*, 1-8.

Talbi, E.G., Hafidi, Z., Kebbal, D. and Geib, J.M. (1998). A fault-tolerant parallel heuristic for assignment problems. Future Generation Computer Systems, 14(56) : 425 – 438, 1998.

Tanaki, H et al,. (1995). Multicriteria Optimization by Genetic Algorithmin: a case of scheduling in hot rolling process. In proceeding of the third APORS, p. 374-381.

Taneda, A. (2010). Multi-objective pairwise RNA sequence alignment. Graduate School of Science and Technology, Hirosaki University, Hirosaki, Aomori 036-8561, Japan *Vol. 26 no. 19, pages 2383–2390 doi:10.1093/bioinformatics/btq439*

Taylor, W.R. (1986). The classification of amino acid conservation. J. Theor. Biol. 119:205–218.

Thompson, J., Higgins, D. & Gibson, T. (1994). ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Research. 22, pp, 4673-4690.

Thompson J.D., Plewniak F. and Poch O. (1999). Balibase: A benchmark alignments database for the evaluation of multiple sequence alignment programs. *Bioinformatics, Vol. 15 :87–88.*

Thompson, J.D., Plewniak, F., Ripp, R., Thierry and Poch, O. (2001). Towards a reliable objective function for multiple sequence alignments. Journal of Molecular Biology, Vol. 314 :937–951.

Thompson, J.D., Koehl, P., Ripp, R. and Poch, O. (2005) Balibase 3.0 : Latest developments of the multiple sequence alignment benchmark. PROTEINS : Structure, Function, and Bioinformatics, 61 :127–136.

Thorne J.L., Kishino H. and Felsenstein J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. Journal of Molecular Evolution. 33 : 114–124.

Thorne J.L., Kishino H. and Felsenstein J. (1992). Inching toward reality: an improved likelihood model of sequence evolution. Journal of Molecular Evolution, 34 : 3–16.

Tomczak K et al., (2015). The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol (Pozn), vol. 19(1A), A68-77 (PMID 25691825, DOI 10.5114/wo.2014.47136)

Tompa, M. (2000). Introduction to multiple sequence alignment. University of Washington, Seattle, Technical report.

Treviranus, G.R. (1802). Biologie: Oder philosophie der lebenden natur. *Fur Naturforscher und Aerzte, Gottingen: Johann Friedrich Rower, vol. 1*:4.

Tsai J., Bonneau R., Morozov A., Kuhlman B., Rohl C.A., and Baker D. (2003). An Improved Protein Decoy Set for Testing Energy Functions for Protein Structure Prediction, Proteins, vol. 52, pp. 76-87.

Ulungu, E.L. and Teghem, J. (1995). The two phases method : An efficient procedure to solve biobjective combinatorial optimization problems. Foundation of computing and decision science, 20 :149–156.

Van Walle, I., Lasters, I. and Wyns, L.(2005). Sabmark - a benchmark for sequence alignment that cover the entire known fold space. Bioinformatics, Vol. 21(7) :1267–1268.

Veldhuizen, A.V and Lamont, G.B. (2000). On measuring multi-objective evolutionary algorithm performance. In Congress of evolutionary computation. Piscataway. New jersey. Volume 1, page 204-211.

Voudouris, C. and Tsang, E. (1999). Guided local search. European Journal of Operational Research, 113(2) :469–499, 1999.

Wallace. I.M, O'Sullivan. O, Higgins D.G and Notredame, C. (2006) M-Coffee: combining multiple sequence alignment methods with T-Coffee *Nucleic Acids Research*, Volume 34, Issue 6, Pages 1692–1699.

Wang J. and Terpenny, JP. (2005). Interactive Preference Incorporation in Evolutionary Engineering Design. In: Jin Y (ed.) Knowledge Incorporation in Evolutionary Computation, 525–543. Springer, Berlin Heidelberg. ISBN 3-540-22902-7.

Wolpert, D.H. and Macready, W.G. (1997). No Free Lunch Theorems for Optimization. IEEE Transactions on evolutionary computation, vol. 1, no. 1.

Woodruff, D. L. (1999). A chunking based selection strategy for integrating meta-heuristics with branch and bound. In S. Voss et al., editors, Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization, pages 499–511.

Wu, S. and Manber, U. (1992). Fast text searching : allowing errors. Communications of the ACM. 35(10), pp 83-91.

Yadav, R K. and Banka, H. (2016). IBBOMSA: An Improved Biogeography-based Approach for Multiple Sequence Alignment. Evolutionary Bioinformatics. 12, 237–246. doi: 10.4137/EBO.S40457.

Yang J, Wang H, Wang, W, Yu P (2003). Enhanced biclustering on expression data. Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on, 00:321–327

Yang, X.S. (2008). Nature-Inspired Metaheuristic Algorithms, Luniver Press, UK.

Yang, X.S. and Deb, S. (2009). Cuckoo Search via Levey Flights. 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings. 210 - 214. 10.1109/NABIC.2009.5393690.

Yang, X.S (2013). Bat Algorithm: Literature Review and Applications. International Journal of Bio-Inspired Computation. 5. 10.1504/IJBIC.2013.055093.

Zambrano-Vega, C., Nebro, A.J., García-Nieto, J. and Aldana-Montes, J. F. (2017 a). A Multi-objective Optimization Framework for Multiple Sequence Alignment with Metaheuristics. In: Bioinformatics and Biomedical Engineering. Springer Professional.

Zambrano-Vega, C., Nebro, A.J., Garcıa-Nieto, J. and Aldana-Montes, J. F. (2017 b). M2Align: parallel multiple sequence alignment with a multi-objective metaheuristic. Bioinformatics, 33(19), 3011–3017 doi: 10.1093/bioinformatics/btx338

Zambrano-Vega, C., Nebro, A.J., García-Nieto, J. and Aldana-Montes. J. F. (2017 c). Comparing multiobjective metaheuristics for solving a three objective formulation of multiple sequence alignment. Progress in Artificial Intelligence pp 1–16 doi:10.1007/s1374801701166

Zheng, YJ. (2015). Water wave optimization: A new nature-inspired metaheuristic, Computers & Operations Research, Volume 55, Pages 1-11

Zhang, Z, Teo, A, Ooi B, Tan K (2004). Mining deterministic biclusters in gene expression data. Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on, 283–290.

Zhang, M., Fang, W. Zhang, J. and Chi, Z. (2005). MSAID: multiple sequence alignment based on a measure of information discrepancy. Computational Biology and Chemistry, vol. 29, pp. 175–181.

Zhu, H., He, Z. and Jia, Y. (2015). A novel approach to multiple sequence alignment using multi-objective evolutionary algorithm based on decomposition. IEEE J Biomed Health Inform. 20(2), 1–11.

Zitzle, E., & Thiele. L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 3(4), 257-271.

Zitzler, E., Deb, K. and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary computation. 8. 173-95. 10.1162/106365600568202.

Zuckerkandl, E., and Pauling, L. (1965). Evolutionary Divergence and Convergence in Proteins.

Zwir I., Zaliz R.R., and Ruspini E.H. (2002). Automated Biological Sequence Description by Genetic Multiobjective Generalized Clustering. Annals New York Academy of Sciences, vol. 980, pp. 65-82.

Zydallis JB, Veldhuizen DAV and Lamont GB (2001). A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA–II. In First International Conference on Evolutionary Multi-Criterion Optimization, 226–240. Springer-Verlag. Lecture Notes in Computer Science No. 1993