الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Nº Réf :……

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la Communication (STIC)

Développement d'un outil qui génère des nouveaux médicaments

Préparé par : **Benabdellah Soumia**
Zaabat Roukia

Devant le jury

Dib Abderrahim	**M.A.A**	**C.U.Abd Elhafid Boussouf**	**Président**
Bouchekouf Asma	**M.A.B**	**C.U.Abd Elhafid Boussouf**	**Rapporteur**
Bouchemal Nerdjas	**M.C.B**	**C.U.Abd Elhafid Boussouf**	**Examinateur**

Année Universitaire : 2016/2017

Remerciements

Après la louange et grâce à Dieu qui nous a donné La force et la patience d'accomplir ce modeste travail.

Je remercie vivement Mlle ***Bouchekouf Asma*** *notre encadreur pour sa présence, son aide.*
Notre vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ce travail en acceptant de l'examiner et de l'enrichir par leurs propositions.

Nous tenons à exprimer mes sincères remerciements à tout le personnel de l'institut de sciences et de la technologie du centre universitaire de Mila surtout les enseignants qui nous ont enseigné durant ces cinq années d'étude.

Nous remercions également toutes les personnes qui nous ont aidés, de près ou de loin pour la réalisation de ce travail en particulier à nos parents, et nos frères, soeurs, amis qui nous avons encouragé.

Dédicace

Je dédie ce modeste travail à :

A mes très chers parents qui m'ont tant soutenu et encouragé dans tous les domaines et surtout pour réaliser ce mémoire, J'espère que Dieu les protège.

A mes frères et ma sœur

A Tous ma famille

A mon binôme ***ROUKIA****.*

A Tous mes amis

A Tous mes enseignants.

SOUMIA

Dédicace

A ma chère maman que j'aime le plus au monde, et mon père qui a toujours été là pour moi

« Mes très chères parents vous êtes pour moi un sujet de fierté, vous m'avez toujours appris le sens de la responsabilité et de la confiance en soi, que Dieu vous garde et vous alloue bonne santé, bonheur et longue vie. »

A mes très chers Frères et Soeurs en témoignage de mon attachement affectif.

A mon fiancé ***BADIS*** *et ça famille*

A mon binôme ***SOUMIA*** *que j`estime beaucoup.*

A tous ceux et celles qui me sont chers.

Je dédie ce modeste travail.

ROUKIA

Résumé :

La production d'un nouveau médicament est un processus long, complexe et trop coûteux. Il peut s'écouler 12 à 15 ans pour développer un nouveau médicament. La conception de médicament assisté par ordinateur (computer aided drug design CADD) est une nouvelle technique qui apparaitre ces derniers années, cette technique permet de développer des nouveaux médicaments en utilisant l'ordinateur, l'utilisation de l'ordinateur dans ce domaine permet de produire des médicaments efficaces en réduisant le temps et le cout de développement, il existe plusieurs approche dans cette technique, parmi ces approches nous avons l'approche de novo drug design. C'est une approche très puissante qui permet de générer des médicaments qui satisfaire certains propriété comme la propriété favorable ADME/TOX. A cause de développement continue dans le domaine de pharmacie, et de la nouveauté de la conception de médicament assisté par ordinateur , nous proposons de réaliser un outil qui permet de générer des nouveaux molécules(médicaments) qui satisfaire certains propriétés , ici nous choisissons de satisfaire deux propriétés , la biodisponibilité orale OBA et la propriété de similarité a un médicament connu en utilisant le coefficient de Tanimoto, notre objectif n'est pas seulement de satisfaire ces deux propriétés mais aussi de définir ce nouveau domaine, de donner une base de recherche et de développement , et d'aider les pharmacies en Algérie de développer des médicament de qualité en réduisant le temps, le cout et les efforts . Notre outil NSGAII-DND est basé sur l'algorithme d'optimisation multi-objectif NSGA-II, en utilisant l'approche de somme pondéré par agrégation des deux objectifs (OBA et le coefficient de Tanimoto), une étude expérimentale en utilisant un ensemble de Library de fragment construire a partir d'une base de donné de la chemoinformatique, et la Library open source java CDK (Chimistry Developement Kit), cette étude donne des résultats prometteuse.

Mots clés : la chemoinformatique, l'optimisation multi-objective, computer aided Drug design, le Drug design, De Novo Drug Design, la bio disponibilité orale, le coefficient de Tanimoto, Chimistry Developement Kit.

Abstract :

Producing a new drug is a complex, time-consuming and very expensive process. It can typically take 12 to 15 years to develop one new drug. the computer-aided drug design (CADD) is a new technique that appear in recent years, this technique allows the development of new drugs using the computer, use the computer in this area can produce effective drugs by reducing the time and cost of development, there are several approaches in this technique, among these approaches we have the approach of de novo drug design. This is a very powerful approach to generate drugs which satisfy some properties as the property favorable ADME / TOX. Because of continuous development in the pharmacy field, and the novelty of the drug design computer assisted, we propose to create a tool to generate new molecules (drugs) that

satisfy certain properties, here we choose to meet two properties, bio oral availability OBA and drug like property (the similarity to a known drug) using the Tanimoto coefficient, our goal is not just to satisfy both properties but also to define this new field, giving a base of research and development, and to assist pharmacies in Algeria to develop quality drugs by reducing the time, the cost and efforts. Our NSGAII-DND tool is based on the multi-objective NSGA-II optimization algorithm, using the approach of weighted sum optimization by aggregating two goals (OBA and Tanimoto coefficient), a experimental study using a set of Library of fragments constructing from a chemoinformatic data base, and the open source java library CDK (Chimistry Developement Kit) is done, this study gives promising results.

Key words: chemoinformatique, the multi-objective optimization, Computer Aided Drug Design, the Drug design, De Novo Drug Design, bio oral availability, Tanimoto coefficient, Chimistry Developement Kit.

الملخص:

إن عملية إنتاج دواء جديد عملية طويلة، معقدة و جد مكلفة، حيث تستغرق من 12 إلى 15عاما لتصنيع دواء واحد فقط. تعتبر صناعة الأدوية المدعمة بالحاسوب تقنية جديدة ظهرت في السنوات الأخيرة. هذه التقنية تسمح بإنتاج أدوية جديدة في وقت معقول وبتكلفة مقبولة. هناك عدة مناهج تحتويها هده التقنية من بينها التصميم دو نوفو وهي تقنية فعالة تسمح بصناعة أدوية ذات نوعية وتلبي بعض الأهداف. ونظرا للتطور الحاصل في ميدان صناعة الأدوية في الجزائر بالإضافة إلى أن ميدان صناعة الأدوية المدعمة بالحاسوب تعتبر مجالا جديدا في الجزائر اقترحنا إنشاء أداة تساعد على صناعة الأدوية باستعمال الحاسوب والتي تلبي هدفين الأول هو مشابهة الجزيء المنتج لجزيء معروف باستعمال معامل تانيموتو والثاني هو خاصية التوافر الحيوي عن طريق الفم باستعمال القاعدة الخماسية لليبينسكي. هدفنا من خلال هده الأداة ليس فقط توفير هاتين الخاصيتين وإنما التعريف بهذا المجال الجديد وإعطاء قاعدة للبحث والتطوير ومساعدة الصيدليين في الجزائر على إنتاج أدوية ذات نوعية وبتكلفة صغيرة وزمن قصير وتوفير الجهد. أداتنا تعتمد على الخوارزمية الجينية متعددة الأهداف باستعمال منهج تجميع الأهداف في هدف واحد معقد. دراسة تجريبية باستعمال مجموعة من المكتبات لجزيئات مستخرجة من قاعدة معطيات كيميائية والمكتبة سيديكا للغة البرمجة جافا.هذه الدراسة تعطي نتائج واعدة.

الكلمات المفتاحية: المعلوماتية الكيميائية، الخوارزمية متعددة الأهداف، صناعة الأدوية المدعمة بالحاسوب، تصميم دو نوفو، التوافر الحيوي عن طريق الفم، القاعدة الخماسية لليبينسكي، معامل تانيموتو.

Table des matières

Introduction générale

1. Contexte du travail et motivations 1
2. Problématique 2
3. Objectifs et schéma de solution 3
4. Organisation du mémoire 4

Chapitre 01 : La chemoinformatique

1. Introduction 6
2. La chemoinformatique 6
2.1. Historique de la chemoinformatique 6
2.2. Définition de la chemoinformatique 7
2.3. Objectif de la chemoinformatique 7
2.4. Les applications de la chemoinformatique 8
3. Le drug design 8
3.1. Définition 8
3.2. Le processus drug design 9
3.2.1. La phase de recherche fondamentale 10
3.2.2. La phase préclinique 10
3.2.3. Les phases cliniques 10
3.2.4. La phase d'enregistrement et de commercialisation 11
3.2.5. La phase clinique IV 11
3.3. Différents type de drug design 11
3.3.1. Ligand-Based Drug Design 12
3.3.2. Structure-Based Drug Design 12
3.3.3. Conception rationnelle de médicaments 12
3.3.4. Drug Design assistée par ordinateur 12
4. Les types d'expériences 12
4.1. In vivo 12
4.2. In vitro 12

5. Les propriétés ADME/TOX et drug-like 13
5.1. La propriété ADME/TOX 13
5.1.1. Définition 13
5.1.2. Les propriétés 14
5.2. La notion drug-like 15
6. La conception des médicaments assistés par ordinateur 15
6.1. Le terme in silico 16
6.2. Le criblage 16
6.2.1. Le criblage à haut débit 16
6.2.2. Le criblage virtuel 17
6.3. Le doking 18
6.4. Le modèle QSAR 18
6.4.1. Définition 18
6.4.2. Objectifs de QSAR 19
6.4.3. Les descripteurs moléculaires 19
6.4.4. La recherche de similarité 20
6.4.5. Représentations des fingerprints 21
7. La méthode de novo drug design 21
7.1. Définition 21
7.2. Le principe 21
7.3. Le processus de novo drug design 22
7.3.1. Contraintes cibles primaires 22
7.3.2. Les stratégies de construction 23
7.3.3. Contraintes cibles secondaires 25
8. Conclusion 25

Chapitre 02 : L'optimisation multi-objectif

1. Introduction 27
2. Définition d'un problème d'optimisation 27
3. Les problèmes d'optimisation mono-objectif 28
4. Les problèmes d'optimisation multi-objectif 28
4.1. Formulation d'un problème d'optimisation multi-objectif 29

4.2. Notions de dominance et optimalité au sens Pareto 30
4.3. Approches de résolution multi-objectif 32
4.3.1. Classification « point de vue décideur » 32
4.3.2. Classification « point de vue concepteur » 33
5. Classification des algorithmes d'optimisation 33
5.1. Algorithmes déterministes / algorithmes stochastiques 34
5.1.1. Algorithmes déterministes 34
5.1.2. Algorithmes stochastiques 34
5.2. Algorithmes d'optimisation locale/globale 34
5.2.1. Algorithmes d'optimisation locale 34
5.2.2. Algorithmes d'optimisation globale 34
5.3. Les méthodes exactes / approchée 35
5.3.1. Les méthodes exactes 35
5.3.2. Les méthodes approchées 35
6. Historique des méthodes de résolution multi-objectif 35
6.1. Méthodes exactes pour l'optimisation multi-objectif 35
6.2. Métaheuristiques pour l'optimisation multiobjectif 35
7. Les algorithmes génétiques 36
7.1. Définition 36
7.2. Les phases d'un algorithme génétique 37
7.2.1. Codage et création de la population initiale 37
7.2.2. Evaluation de la population 38
7.2.3. Les opérateurs génétiques 38
7.2.4. La fonction d'adaptation (fitness) 42
7.2.5. Critère d'arrêt 42
7.3. Pseudo code d'un algorithme génétique 42
7.4. Organigramme d'un algorithme génétique 43
8. Les algorithmes évolutionnaires 43
8.1. Les algorithmes non-élitistes 44
8.1.1. Multiple Objective Genetic Algorithm (MOGA) 44
8.1.2. Non dominate Sorting Genetic Algorithm (NSGA) 45
8.1.3. Niched Pareto Genetic Algorithm (NPGA) 46

8.2. Les algorithmes élitistes 47
8.2.1. Strength Pareto Evolutionary Algorithm (SPEA) 47
8.2.2. Pareto Archived Evolution Strategy (PAES) 48
8.2.3. Pareto Envelope based Selection Algorithm (PESA) 48
8.2.4. Non dominate Sorting Genetic Algorithm II (NSGA-II) 49
9. Conclusion 52

Chapitre 03 : L'optimisation en chemoinformatique

1. Introduction 54
2. L'optimisation dans le domaine de la chemoinformatique 54
3. L'optimisation multi-objectif dans le modèle QSAR 56
3.1. MoQSAR 56
3.2. PUMP-RP 57
3.3. QSAR 3D 57
3.4. MOQSPR 57
4. L'optimisation multi-objectif dans le docking 57
4.1. ClustMPSO 58
4.2. GOLD (Genetic Optimization for Ligand Docking) 58
4.3. NSGA (Non-dominated sorting genetic algorithm) 58
4.4. MOSFOM 59
5. L'optimisation multi-objectif dans le DND 59
5.1. MEGA (Multi-objective Evolutionary Graph Algorithm) 60
5.2. PLD (Pareto Ligand Designer) 61
5.3. MOEA 62
5.4. MOLig 62
6. L'optimisation multi-objectif dans la conception de la librairie 62
7. Les différents outils en chemoinformatique 64
7.1. CDK 65
7.2. RDKit 66
7.3. OpenBabel 66
8. Principales fonctionnalités des outils de la chemoinformatique 66

9. Workflows de la chemoinformatique 66
9.1. Taverna 67
9.2. KNIME (Konstanz Information Miner) 67
10. Les bases de donné de la chemoinformatique 67
11. Conclusion 68

Chapitre 04 : Contribution à la conception De Novo des médicaments

1. Introduction 70
2. L'idée de base 70
3. Formulation de problème 71
4. Description générale de l'architecture de notre outil 71
4.1. L'algorithme NSGA-II 72
4.1.1. Définition 72
4.1.2. Principe détaillé de NSGA-II 73
4.1.3. Organigramme de l'algorithme NSGA-II 78
4.1.4. Pourquoi on choisit NSGA-II ? 79
4.2. Les librairies de fragment 80
4.3. Les évaluateurs 80
4.4. L'outil CDK 80
4.4.1. Définition 80
4.4.2. Les logiciels «JChemPaint» et «Jmol» 81
4.4.3. Principales fonctionnalités de la librairie CDK 81
5. Les étapes détaillées de notre travail 86
5.1. Combinaison acide/amine 86
5.2. Génération de la population initiale P_0 86
5.3. Création de la population enfant Q_0 87
5.4. Combinaison des deux populations P_0 et Q_0 87
5.5. Les opérateurs génétiques 87
5.5.1. L'opérateur de sélection 87
5.5.2. L'opérateur de croisement 88
5.5.3. L'opérateur de mutation 88

5.6. Evaluation des individus (la fonction objectif) 88
5.6.1. La fonction "OBA" (drug-likness) 89
5.6.2. La fonction "TCoeff" 89
5.7. Affichage de la solution optimale 91
6. Description de la dynamique de NSGAII-DND 91
7. Le pseudo-code de l'algorithme NSGAII-DND 93
8. Conclusion 94

Chapitre 5 : les résultats obtenus et étude comparative

1. Introduction 96
2. Les données 96
2.1. Les librairies des fragments 96
2.2. Les molécules de référence 97
3. La configuration des paramètres 98
3.1. L'effet de la taille de population P 99
3.2. L'effet de nombre d'itération NItér 100
4. Les résultats expérimentaux et étude comparative 101
4.1. Résultats obtenus par l'algorithme NSGAII-DND 101
4.1.1. Cas de test 1(Lidocaïne) 101
4.1.2. Cas de teste 2 (Furano_pyrimidine) 104
4.2. Résultats obtenus par l'algorithme MOGA_DND 106
4.2.1. Cas de test 1(Lidocaïne) 106
4.2.2. Cas de teste 2 (Furano_pyrimidine) 107
4.3. Etude comparative 109
4.3.1. Evaluation de Fitness 109
4.3.2. Le temps d'exécution des algorithmes NSGAII-DND et MOGA_DND . 110
5. Conclusion 110

Conclusion générale

Conclusion générale 113

Liste des figures

Chapitre 01 :

Figure 1.1: Le processus de conception de nouveau médicament (temps et cout) 9
Figure 1.2: Les différentes étapes de développement d'un médicament 10
Figure 1.3: Les raisons de rejet des médicaments. 13
Figure 1.4: Processus de conception de médicaments *in-silico* 16
Figure 1.5: Aperçu du criblage virtuel 18
Figure 1.6: La relation entre le modèle QSAR, in silico, in vitro et in vivo 19
Figure 1.7: Le principe de novo Drug design 22
Figure 1.8: Le growing 23
Figure 1.9: Le linking 24
Figure 1.10: La stratégie link/grow 24
Figure 1.11: La stratégie lattice 25

Chapitre 02 :

Figure 2.1: Représentation d'un problème multi-objectif 29
Figure 2.2: La notion de dominance 30
Figure 2.3: Front de Pareto 31
Figure 2.4: Exemple de front de pareto 32
Figure 2.5: Classification des méthodes de résolution 34
Figure 2.6: Codage binaire des variables 37
Figure 2.7: Codage réel des variables 38
Figure 2.8: Evaluation d'un individu 38
Figure 2.9: Exemple de sélection par roulette 39
Figure 2.10: Croisement à un point pour des chromosomes codés en binaire 40
Figure 2.11: Croisement à deux points pour des chromosomes codés en binaire 41
Figure 2.12: Croisement uniforme 41
Figure 2.13: Operateur de mutation 42
Figure 2.14: L'organigramme d'un algorithme génétique 43
Figure 2.15: Exemple de tableau de dominance 44
Figure 2.16: Fonctionnement de l'algorithme NSGA 46
Figure 2.17: Fonctionnement général de l'algorithme SPEA 48

Figure 2.18: Schéma de fonctionnement de PESA 49
Figure 2.19: Calcul d'un représentant (crowding)..... 51
Figure 2.20: Schéma de fonctionnement de NSGA-II 51

Chapitre 03 :

Figure 3.1: Schémas générale de l'algorithme MEGA 61
Figure 3.2: Workflow schématique de la conception de la librairie virtuelle..... 63

Chapitre 04 :

Figure 4.1: L'idée générale de notre travail 70
Figure 4.2: Modélisation de l'approche NSGAII-DND 72
Figure 4.3: Classification des individus en plusieurs fronts de Pareto 74
Figure 4.4: Principe de Crowding distance..... 75
Figure 4.5: Cas hypothétique de l'assignation de la Crowded distance 76
Figure 4.6: Schéma de l'évolution de l'algorithme NSGA –II 77
Figure 4.7: Organigramme de l'algorithme NSGA-II..... 79
Figure 4.8: Diagramme UML expliquant la hiérarchie d'héritage 82
Figure 4.9: Visualisation de structures chimiques 2D 84
Figure 4.10: La représentation des individus en nombre entier..... 87
Figure 4.11: Le croisement des individus 88
Figure 4.12: La mutation d'un individu..... 88
Figure 4.13: Exemple de calcul du coefficient de Tanimoto 90
Figure 4.14: Affichage des structures 2D par guhautils..... 91
Figure 4.15: Architecture générale de l'algorithme NSGAII-DND..... 92

Chapitre 05 :

Figure 5.1: Visualisation 2D d'une molécule dessinée à l'aide de l'applet guhautils 97
Figure 5.2: La structure 2D des 2 molécules de référence 97
Figure 5.3: L'interface pour la configuration des paramètres de l'algorithme 98
Figure 5.4: Interface pour ajouter l'emplacement de la Library acide..... 98
Figure 5.5: Evaluation de la fonction de fitness pour différentes tailles de population P 99
Figure 5.6: Evaluation de la fonction de fitness pour différentes nombres d'itération..... 100
Figure 5.7: L'interface qui affiche les meilleures solutions 102

Figure 5.8: Les structures 2D des meilleures molécules obtenues par l'algorithme NSGAII-DND liées à la molécule de référence "Lidocaïne". 103
Figure 5.9: L'interface qui affiche les meilleures solutions par apport à la molécule de référence "Furano_pyrimidine"..... 104
Figure 5.10: Les structures 2D des meilleures molécules obtenues par l'algorithme NSGAII-DND liées à la molécule de référence " Furano_pyrimidine" 105
Figure 5.11: Les structures 2D des meilleures molécules obtenues par l'algorithme MOGA_DND liées à la molécule de référence "Lidocaïne" 107
Figure 5.12: Les structures 2D des meilleures molécules obtenues par l'algorithme MOGA_DND liées à la molécule de référence "Furano_pyrimidine". 108
Figure 5.13: Comparaison des résultats de fitness pour les deux algorithmes : NSGAII-DND et MOGA_DND liées à la molécule de référence "Lidocaïne". 109
Figure 5.14: Comparaison des résultats de fitness pour les deux algorithmes : NSGAII-DND et MOGA_DND liées à la molécule de référence "Furano_pyrimidine ". 110

Liste des tableaux

Chapitre 01 :

Tableau 1.1: Exemple de mesures de similarités utilisées en chemoinformatique 21

Chapitre 03 :

Tableau 3.1: Quelques méthodes multi-objectifs dans la chemoinformatique 56
Tableau 3.2: Les outils de la chemoinformatique 65
Tableau 3.3: Liste de base de données connue en chemoinformatique 68

Chapitre 05 :

Tableau 5.1: Les valeurs de finess obtenue pour différentes taille de population 99
Tableau 5.2: Les valeurs de finess obtenue pour chaque nombre d'itération ..... .100
Tableau 5.3: Les résultats obtenus pour le cas de test 1par l'algorithme NSGAII-DND. ... 102
Tableau 5.4: Les résultats obtenu pour le cas de test 2 par l'algorithme NSGAII-DND…...104
Tableau 5.5: Les résultats obtenus pour le cas de test 1par l'algorithme MOGA_DND. 106
Tableau 5.6: Les résultats obtenu pour le cas d'étude 2 par l'algorithme MOGA_DND. 108

Líste Des Acronymes

ADME/TOX	**A**bsorption/**D**istribution/ **M**étabolisme /**E**xcrétion/**Tox**icité.
CADD	**C**omputer_**A**ided **D**rug **D**esign.
CDK	**C**hemistry **D**evelopment **K**it.
DD	**D**rug **D**esign.
DND	**D**e **N**ovo **D**rug Design.
EA	**E**volutionary **A**lgorithm.
GA	**G**enetic **A**lgorithm.
GOLD	**G**enetic **O**ptimization for **L**igand **D**ocking.
HTS	**H**igh-**T**hroughput **S**creening.
MEGA	**M**ulti-objective **E**volutionary **G**raph **A**lgorithm.
MOEA/D	**M**ulti **O**bjective **E**volutionary **A**lgorithm Based **D**ecomposition.
MOGA	**M**ulti-**O**bjectif **G**enetic **A**lgorithm.
MOGA_DND	**M**ultiple **O**bjective **G**enetic **A**lgorithm for **D**e **N**ovo **D**rug Design.
MOOP	**M**ulti **O**bjectif **O**ptimization **P**roblem.
MoQSAR	**M**ulti **O**bjectif **Q**uantitative **S**tructure-**A**ctivity **R**elationship.
MOSFOM	**M**ulti-**O**bjective **S**coring **F**unction **O**ptimization **M**ethodology.
NSGA	**N**on-dominated **S**oring **G**enetic **A**lgorithm.
NSGA-II	**N**on-dominated **S**orting **G**enetic **A**lgorithm II.
NSGAII-DND	**N**on-dominated **S**orting **G**enetic **A**lgorithm II for **D**e **N**ovo **D**rug Design.
OBA	**O**ral **B**io-**A**vailability.
PLD	**P**areto **L**igand **D**esigner.
QSAR	**Q**uantitative **S**tructure-**A**ctivity **R**elationship.
QSPR	**Q**uantitative **S**tructure-**P**roperty **R**elationship.
SDF	**S**tructure **D**ata **F**ormat.
SMILES	**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification.
SOOP	**S**ingle **O**bjectif **O**ptimization **P**roblem.
TCoeff	**T**animoto **C**oefficient.

Introduction :

Introduction générale

1. Contexte du travail et motivations :

Ces dernières années, on assiste à une situation problématique pour la recherche pharmaceutique, les investissements sont en augmentation constante, tandis que le nombre de nouvelles entités chimiques introduites sur le marché n'augmente pas. Cela signifie que le coût de mise au point d'un médicament est : tout comme les investissements, en augmentation constante. L'une des raisons de cette augmentation est le durcissement des critères d'acceptation des médicaments par les organismes gouvernementaux.

L'informatique joue un rôle croissant dans la recherche en Chimie. Des secteurs très variés de la recherche fondamentale ou appliquée nécessitent des spécialités du traitement informatique, de l'information chimique, de la modélisation moléculaire ou de la chimie théorique.

La chimie se prête à un traitement informatique car elle est complexe et nécessite des capacités d'acquisition, de traitement et d'archivage considérables. Des bases de données important se constituent à travers le monde pour permettre aux chercheurs de suivre quasiment en temps réel l'avancement de la chimie. Le but est de montrer l'implication de l'informatique dans différentes applications de la chimie. Ce domaine est appelé "Chemoinformatique".

Le terme chemoinformatique est apparu il y a quelques années et a rapidement gagné l'utilisation répandue. La définition la plus large de la chemoinformatique « est l'application des méthodes d'informatique pour résoudre des problèmes chimiques ».

La chemoiformatique à des applications différentes dans des domaines variés et l'application la plus communément admise est dans le domaine de la découverte de médicaments «Drug Discovery» qui est aussi connue sous le nom de conception de médicaments « Drug Design, (DD) » qui existe depuis plus d'une décennie [2].

Le domaine de la «conception de médicaments» peut être défini par un ensemble de processus inventifs capables de trouver de nouveaux médicaments, en allant de l'identification d'une cible biologique d'intérêt thérapeutique jusqu'au patient, dans lequel les essais cliniques succèdent au développement préclinique.

Le médicament ou la molécule thérapeutique est généralement une substance chimique, qu'on absorbe pour guérir une maladie. Plus spécifiquement, un médicament est une petite molécule organique qui active ou inhibe la fonction d'une biomolécule, qui à son tour entraîne un bénéfice thérapeutique au patient [3].

Traditionnellement, les médicaments ont été découverts grâce à la recherche de produits naturels et les librairies chimiques de molécules synthétiques biologiquement actives. Par conséquent, le processus de la découverte de médicaments est un processus long et couteux pour l'industrie [2]. Mais de nos jours, les techniques computationnelles sont utilisées dans l'industrie pharmaceutique afin d'automatiser et d'accélérer le processus de recherche et développement d'un nouveau médicament et afin de réduire son coût de calcul [4]. En effet, la conception de médicaments à l'aide d'outils informatiques connue sous le nom de conception de médicaments assistée par

ordinateur (en anglais sous l'acronyme CADD, Computer Aided Drug Design) ou aussi la conception «in silico» a contribué à l'introduction en phase clinique du processus de découverte de médicaments d'environ 50 composés et à la mise sur le marché de nombreux médicaments [5]. Cette approche de conception rationnelle de médicaments est considérée comme l'un des domaines de recherche de la chemoinformatique qui couvre, à ce jour une large gamme d'applications.

2. Problématique :

La «Conception De novo de médicament» (De novo drug design, DND) est l'une des techniques les plus puissantes de CADD, qui joue un rôle essentiel dans la recherche de composés "drug-like" possibles qui appartient à un grand espace chimique pour sélectionner ceux qui ont le plus de chances de devenir des médicaments en utilisant divers méthodes computationnelles [6]. Généralement, pour la conception De novo de médicaments, deux approches peuvent être considérées, l'une basée sur les atomes pour construire les ligands (les molécules) et l'autre utilise les fragments (petites molécules) [7]. L'approche basée sur les atomes consiste comme son nom l'indique à construire une molécule atome par atome. Ce qui la rend trop coûteuse en temps de calcul par rapport à l'approche basée sur les fragments. Dans notre travail, nous nous intéressons à l'approche de conception De novo de médicaments basée sur des fragments car, celle-ci est la meilleure approche pour construire une nouvelle molécule ayant une bonne valeur de fitness dont la taille de l'espace de recherche peut être réduite de manière significative.

Le but d'utilisation des fragments est que celles-ci ont une complexité moléculaire plus faible que les composés entiers. Cependant, ce problème a une nature combinatoire et sa complexité augmente avec la taille et le nombre de librairies de fragments utilisées ainsi que la taille (le poids moléculaire) de fragments choisis pour construire une molécule. En effet, le nombre de molécules possibles qui peuvent être conçues en reliant les fragments moléculaires croît de façon exponentielle quand le nombre de fragments augmente. Par conséquent, une recherche exhaustive ne saurait être envisageable ainsi qu'une méthode d'optimisation exacte sachant qu'une telle complexité est accentuée par le coût de l'évaluation de fonctions objectives qui est souvent prohibitif. Une optimisation approchée s'impose donc comme l'alternative de choix pour gérer la recherche dans l'espace issu de la combinaison d'un nombre important de fragments pour choisir une molécule vérifiant au mieux les propriétés requises en un temps raisonnable. Notre choix s'est porté sur l'algorithme génétique NSGA-II (Elitist Non-dominated Sorting Genetic Algorithm) [Deb, 2000] comme metaheuristique à utiliser pour accomplir cette tâche d'optimisation.

En outre, il est à noter qu'à chaque étape du processus de découverte de médicaments, un certain nombre de molécules ne doivent pas passer au stade suivant de développement. L'une des causes les plus fréquents d'échec de composés tête de série "lead" dans les derniers stades de ce processus est le manque de considération de multiples contraintes au premier stade de l'optimisation de composés candidats [8]. En effet, pour concevoir des nouvelles entités chimiques pour l'industrie pharmaceutique,

plusieurs objectifs d'une importance pharmaceutique doivent être pris en compte, y compris l'affinité de liaison, drug-likeness (Règles de Lipinski), la similitude à un ligand, l'activité biologique d'une cible, les propriétés pharmacocinétiques connues sous le nom ADME (Absorption, Distribution, Métabolisme, Excrétion), etc. La multitude de contraintes font que la conception De novo de médicaments se présente comme un problème d'optimisation multi-objectif dans lequel une solution candidate doit satisfaire simultanément plusieurs objectifs.

L'optimisation est une discipline très ancienne qui a connu un essor considérable et un souffle continu depuis l'apparition des ordinateurs [9]. C'est une branche des mathématiques consacrée à l'étude du minimum/maximum d'une fonction à une ou plusieurs variables sur un certain domaine de définition. La résolution des problèmes d'optimisation est utilisée dans un grand nombre de domaines : économie, automatique, gestion, traitement du signal, sciences de l'ingénieur, conception optimale, etc. Les méthodes d'optimisation peuvent être réparties en deux catégories : méthodes exactes qui fournissent systématiquement une solution (optimale) au problème traité si une telle solution existe et méthodes approchées qui fournissent une solution approchée au problème traité et ne donnent pas de garantie sur l'optimalité de la solution [10]. L'optimisation multi-objectif, aussi connue sous le nom d'optimisation multi-critère ou d'optimisation multi-attribut, est un processus qui consiste à optimiser simultanément deux ou plusieurs objectifs, soumis à certaines contraintes [11]. Un problème multi-objectif est caractérisé, non pas par une solution unique, mais par un nombre de solutions qui réalisent des compromis sur les différents objectifs.

A cause de la complexité d'optimisation de plusieurs objectifs simultanément la plupart des méthodes d'optimisation dans le domaine de DND préfère d'ignorer la nature multi-objectif et de regrouper les objectifs dans un seul objectif composite en donnant un poids à chaque objectif.

3. Objectifs et schéma de solution :

L'objectif principal de notre travail consiste à offrir donc un support pour concevoir de nouvelles molécules et offrir les meilleures chances de sélectionner des composés "drug-like" en vue de poursuivre leur développement préclinique. Un tel outil doit offrir la possibilité de rechercher dans l'espace des fragments la molécule adéquate, de composer une molécule à partir de fragments et d'évaluer les propriétés d'une molécule conçue. Pour ce faire, nous avons développé un outil nommé NSGAII-DND dont les éléments de base sont l'algorithme multi-objectif NSGA-II , l'ensemble de librairies de fragments moléculaires, l'évaluateur de la biodisponibilité, l'évaluateur de similarité et la librairie open-source CDK (Chemistry Development Kit) [12], un outil développée en Java pour la chemoinformatique et la bioinformatique.

L'idée principale de notre outil consiste donc à concevoir à l'aide de CDK une molécule composite "drug-like" à partir de plusieurs fragments qui doivent satisfaire simultanément deux objectifs d'une importance pharmaceutique qui sont: drug-likness

(ou biodisponibilité) de notre molécule conçue et sa similitude à une molécule de référence. Le premier objectif est calculé à l'aide de l'évaluateur de la biodisponibilité qui se base sur les "règles des 5" de Lipinski nécessaires pour la sélection de composés "drug-like". Tandis que, le deuxième objectif est calculé en utilisant l'évaluateur de similarité qui se base sur le "coefficient de Tanimoto" pour mesurer la similarité entre une molécule conçue et une molécule de référence.

4. Organisation du mémoire :

Dans la suite de ce mémoire, notre travail est organisé autour de cinq chapitres :

- ✓ Dans le premier chapitre, nous présentons le domaine de la chemoinformatique et mettons l'accent sur la conception des médicaments assistée par ordinateur autour de laquelle la suite de ce mémoire s'articule.
- ✓ Dans le deuxième chapitre, nous décrivons le cadre de l'optimisation, les éléments de base relatifs notamment à l'optimisation multi-objectif et les différents méthodes de résolution pour traiter un problème d'optimisation multi-objectif.
- ✓ Dans le troisième chapitre, nous passons en revue les approches d'optimisations multi-objectifs proposées dans la littérature en rapport avec la conception de novo de médicaments, ainsi que la description des outils de la chemoinformatique et leurs différentes fonctionnalités.
- ✓ Le quatrième chapitre est spécialement consacré à la description de notre outil NSGAII-DND conçu pour résoudre les problèmes de conception de novo de médicaments dans le contexte d'optimisation multi-objectif. Ce nouvel outil a donc pour but de construire des nouvelles molécules et offrir les meilleures chances de sélectionner des composés ayant des propriétés "drug-like".
- ✓ Enfin, le cinquième chapitre porte sur la présentation et la discussion des résultats obtenus par la mise en œuvre de notre outil NSGAII-DND, ainsi que la comparaison avec l'outil MOGA_DND [31].

CHAPITRE 01 :

La chemoinformatique

Au sommaire de ce chapitre :

1. Introduction.
2. La chemoinformatique.
3. Le drug design.
4. Les types d'expériences.
5. Les propriétés ADME/TOX et drug-like.
6. La conception des médicaments assistés par ordinateur.
7. La méthode de novo drug design.
8. Conclusion.

1. Introduction :

La découverte de médicaments est un processus long et complexe. On estime qu'il faut entre 12 et 15 ans et environ 800 millions d'euros pour la mise au point d'un médicament. En effet, les coûts de l'évolution des nouvelles entités chimiques mises sur le marché et les investissements des entreprises pharmaceutiques en recherche et développement sont très prohibitifs.

Pour lutter contre cet état de fait, le monde de la recherche pharmaceutique optimise constamment les étapes de son processus de découverte et de mise au point de médicaments. La chemoinformatique est un outil de choix pour diminuer le temps et le coût de développement d'un médicament. Cette discipline peut intervenir à différents niveaux du processus de découverte d'un médicament. La conception de médicament à l'aide d'outils informatiques connue sous le nom de conception de médicaments assistée par ordinateur (en anglais sous l'acronyme CADD, Computer aided Drug Design) a contribué à l'introduction en phase clinique d'environ 50 composés et à la mise sur le marché de nombreux médicaments [5].

La chemoinformatique représente le domaine de notre étude. Aussi, nous présentons dans ce chapitre une introduction mettant en exergue les aspects généraux relatifs à ce domaine. Nous commençons d'abord, par une brève description de cette nouvelle discipline, ensuite, nous présentons le processus drug design, et plus tard, nous expliquons de novo drug design la technique la plus puissante de CADD.

2. La chemoinformatique :

2.1. Historique de la chemoinformatique :

La chemoinformatique est une nouvelle discipline apparue il y a environ 40 ans. Au début des années soixante, dans le but d'élucider la structure de composés chimiques inconnus, les données provenant des méthodes existantes (spectroscopie) ont été mises en commun sur informatique. C'était la naissance de la chemoinformatique [13].

Le projet Dendral initié en 1964 à l'université de Stanford a été le premier à développer des générateurs de structures chimiques à partir de spectres de masses.

Ce n'est qu'à la fin des années 60 que Sasaki à l'université de technologie de Toyohashi et Munk à l'université d'Arizona ont utilisé plusieurs méthodes de spectroscopie afin d'élucider la structure chimique de leurs composés.

En 1969, Corey et Wipke ont présenté un travail similaire concernant les systèmes de représentation des molécules. Peu après d'autres groupes comme Ugi , Hendrickson et Gelernter ont développés des systèmes pour représenter des molécules organiques. La chemoinformatique (qui n'avait pas encore de nom) a été reconnue à la fin des années 60 comme une méthode utile d'analyse des données. Dès lors, beaucoup d'articles concernant cette discipline ont commencé à apparaitre dans les journaux scientifiques.

C'est seulement en 1998 que F.K Brown a défini pour la première fois cette discipline comme étant la chemoinformatique [1].

2.2. Définition de la chemoinformatique :

La chemoinformatique est une discipline scientifique relativement récente qui consiste à étudier et à résoudre les problèmes relatifs à la chimie en appliquant des méthodes informatiques [5].

La notion de chemoinformatique a émergé pour la première fois lors du développement d'un médicament en 1998.

C'est Franck Brown qui l'a défini comme suit :
"Chemoinformatics est le mélange de ces ressources d'information (technologie de l'information et de gestion) pour transformer en informations de données et de connaissances en informations destinées à prendre de meilleures décisions plus rapidement dans le domaine de l'identification des médicaments et l'optimisation du lead" [14].

Puis un peu plus tard, en 1999, G.Paris a donné ainsi la définition suivante : "La chemoinformatique est un terme générique qui regroupe la conception, la création, l'organisation, la gestion, l'extraction, l'analyse, la valorisation, la visualisation et l'utilisation d'information chimique"[15].

Une définition plus large de la chemoinformatique a ensuite été proposée par Gasteiger :
"L'application de méthodes chimiques de l'informatique pour résoudre des problèmes" [16].

2.3. Objectifs de la chemoinformatique :

La chemoinformatique est là pour répondre et aider les chimistes à résoudre certains des problèmes fondamentaux suivants :

1) Concevoir des molécules avec des propriétés désirées : L'objectif majeur d'un chercheur en chimie est de produire des composés possédant les propriétés souhaitées et d'étudier la relation quantitative structure-activité (en anglais : Quantitative structure-activity relationship ou QSAR) qui peut être employée pour la prédiction de la propriété de nouvelles molécules, pour lesquelles la valeur expérimentale n'est pas disponible, voire pour des molécules encore non synthétisées [1].

2) Concevoir la réaction chimique et synthétiser des nouveaux composés : la conception de la réaction comprend les réactifs utilisés pour synthétiser les composés (les produits) issus de cette opération chimique.

3) Analyser et élucider les structures obtenues lors de la réaction : Il est nécessaire d'établir la structure du produit due à la réaction en utilisant les différents outils pour déterminer la structure.

4) Transformer les données en informations et celles-ci en connaissances, afin de prendre la meilleure décision possible [1].

2.4. Les applications de la chemoinformatique :

Selon la définition obtenue par Gasteiger en 2003, la chemoinformatique est l'une des technologies utilisées pour résoudre les problèmes liés à la chimie. Les différents aspects chimiques suivants indiquent quelques applications typiques de la chemoinformatique dans ce domaine [17]:

1) **L'information chimique :**
   - ✓ Stockage et extraction des structures chimiques et les données associées,
   - ✓ Diffusion de données sur l'internet,
   - ✓ Raccordement (réticulation) des données à des informations.

2) **Prédiction des propriétés :**
   - ✓ Tous les domaines de la chimie pour la prédiction des propriétés physiques, chimiques ou biologiques des composés.

3) **L'analyse des données :**
   - ✓ provenant de la chimie analytique pour faire des prédictions sur la qualité et l'origine des composés chimiques étudiés.

4) **Chimie organique :**
   - ✓ Prédiction du cours d'une réaction organique.
   - ✓ Conception de la synthèse organique.

5) **La conception de médicaments (drug design) :**
   - ✓ Identification de nouvelles structures de la tête de série (lead).
   - ✓ Optimisation de la structure des têtes de séries.
   - ✓ Etablissement des relations quantitatives de structure-activité.
   - ✓ Comparaison des librairies (bibliothèques) chimiques.
   - ✓ Définition et analyse de la diversité structurelle des librairies chimiques.
   - ✓ L'analyse des données à haut débit.
   - ✓ L'amarrage moléculaire (Docking) d'un ligand sur un récepteur.
   - ✓ La conception de médicament de novo (De-novo drug design(DND).
   - ✓ La perception du pharmacophore.
   - ✓ La modélisation des propriétés ADME/Tox.
   - ✓ etc.

3. Le drug design :

3.1. Définition :

Drug design, parfois appelé la conception de médicaments comme rationnel (rational drug design) ou conception simplement rationnel, est le procédé de l'invention pour trouver de nouveaux médicaments.

Le médicament : est le plus souvent une petite molécule organique qui active ou inhibe la fonction d'une molécule telle qu'une protéine, qui à son tour entraîne un bénéfice thérapeutique au patient [18].

Un médicament est défini, selon l'article L5111-1 du code de santé public français, de cette manière :

"On entend par médicament toute substance ou composition présentée comme possédant des propriétés curatives ou préventives à l'égard des maladies humaines ou animales, ainsi que toute substance ou composition pouvant être utilisée chez l'homme ou chez l'animal ou pouvant leur être administrée, en vue d'établir un diagnostic médicale ou de restaurer, corriger ou modifier leurs fonctions physiologiques en exerçant une action pharmacologique, immunologique ou métabolique" [19].

Dans le sens le plus fondamental, la conception de médicaments implique la conception des molécules qui sont complémentaires dans la forme et la charge à la cible biomoléculaire avec lesquels ils interagissent, et donc vont se lier à elle. Drug design repose souvent, mais pas nécessairement sur les techniques de modélisation informatique. Ce type de modélisation est souvent désigné comme la conception de médicaments assistée par ordinateur (Computer aided drug design) [18].

3.2. Le processus drug design :

Le développement d'un nouveau médicament est un processus complexe, long, très coûteux, et fortement encadré juridiquement à cause des risques de santé publique. En réalité, seules les plus grandes entreprises de l'industrie pharmaceutique sont capables d'assurer le développement d'un médicament de A à Z. Celui-ci est divisé en plusieurs phases, chacune nécessitant la validation de la précédente pour pouvoir continuer. Il est à noter qu'à chaque étape, un grand nombre de molécules échoue à passer au stade suivant de développement. On estime qu'il faut 10-20 ans de recherche et d'essais cliniques pour mettre au point un médicament.

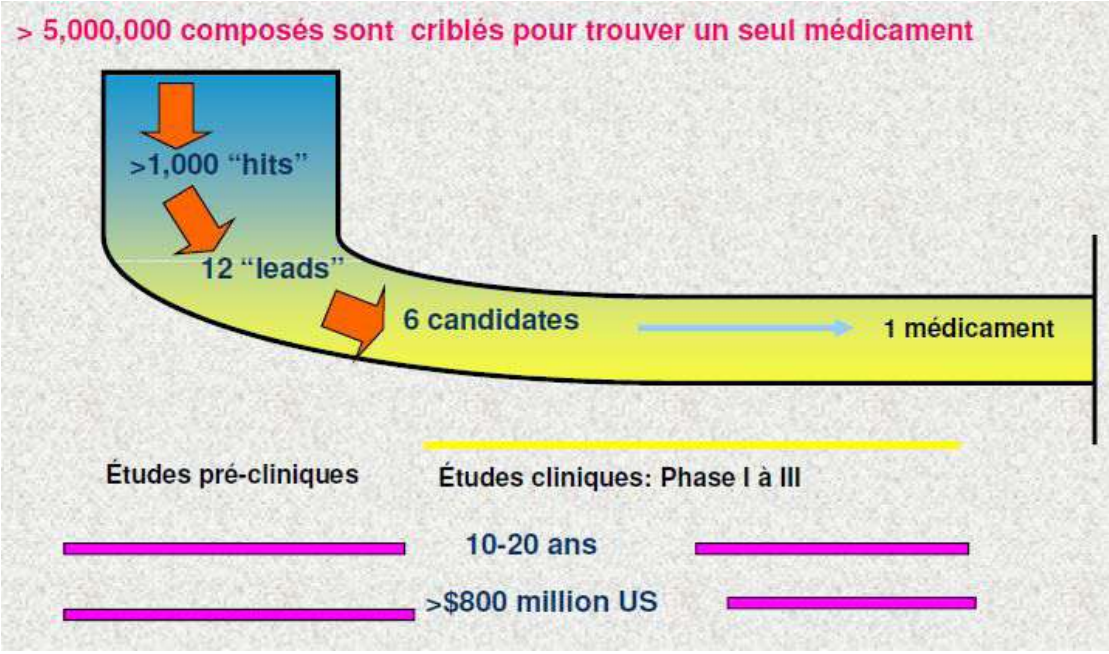


Figure 1.1: le processus de conception de nouveau médicament (temps et cout) [20].

Les principales étapes, présentées dans la Figure 1.2, sont la phase de recherche, la phase préclinique et les différentes phases cliniques :

Figure 1.2 : Les différentes étapes de développement d'un médicament [19].

Une brève description de chacune de ces phases est donnée ci-dessous.

3.2.1. La phase de recherche fondamentale :

Un projet de développement de médicament commence généralement par une phase de recherche qui consiste en l'identification d'une cible impliquée dans une pathologie donnée. Les cibles thérapeutiques sont généralement des protéines, notamment des récepteurs ou des enzymes [19].

3.2.2. La phase préclinique :

Une fois la cible validée, le but est de mettre au point un ligand modulant son activité, en l'activant ou en l'inhibant, afin d'obtenir l'effet curatif souhaité. L'identification de nouveaux composés actifs, souvent appelés hits se fait par le criblage expérimental d'une chimiothèque [19].

Ces hits peuvent provenir de sources naturelles, telles les plantes et animaux, les microbes, ou de sources synthétiques [20].

Les hits primaires confirmés sont ensuite optimisés dans le but d'améliorer leur affinité pour la cible, leur spécificité vis-à-vis d'autres macromolécules biologiques, et leurs propriétés pharmacocinétiques.

Différents filtres, connus sous l'acronyme ADME/Tox, sont appliqués pour éliminer les molécules indésirables le plut tôt possible dans le processus drug design [19].

Les modèles QSAR (Quantitative Structure-Activity Relationship) élaborés à partir des différents pharmacophore jouent un rôle important dans la découverte et la sélection de composés leads (têtes de séries), Ces leads montrent un meilleur potentiel d'action, une meilleure sélectivité, de meilleures propriétés pharmacocinétiques et une toxicité réduite par rapport aux hits initialement sélectionnés [20].

3.2.3. Les phases cliniques :

Il s'agit des premières phases où le composé d'intérêt est administré à l'homme. Au cours du temps, plusieurs phases ont été définies, chacune répondant à une problématique bien spécifique. L'enjeu est de décider si le candidat peut devenir un médicament en tant que tel, en analysant ses bénéfices / risques (efficacité thérapeutique vs. effets secondaires) [19].

3.2.3.1. La phase I :

La phase I consiste à déterminer sur des volontaires sains l'innocuité du composé, les doses maximales tolérées, ainsi que l'apparition éventuelle d'effets secondaires encore inconnus à ce stade.

3.2.3.2. La phase II :

La phase II est la première étape où l'on administre le candidat médicament à des patients, dont l'ordre de grandeur est de quelques centaines. Le but est d'évaluer la réelle efficacité de la molécule ainsi que les doses minimales efficaces - et donc la marge thérapeutique vis-à-vis des résultats obtenus à la phase précédente - tout en répertoriant l'apparition éventuelle de nouveaux effets secondaires.

3.2.3.3. La phase III :

La phase III est conceptuellement similaire à la phase précédente : il s'agit à nouveau d'administrer la molécule à un groupe de patients. Ici, le nombre de personnes est sensiblement plus important, de l'ordre de quelques milliers. Des malades ayant des profils ethniques variés est également souhaitable pour être le plus proche possible de la réalité. Cette phase permet de confirmer la sécurité et l'efficacité du candidat médicament sur la pathologie ciblée. Une comparaison par rapport à un panel de molécules concurrentes déjà présentes sur le marché est également entreprise, notamment pour pouvoir estimer le service médical rendu par le candidat.

3.2.4. La phase d'enregistrement et de commercialisation :

A ce stade, un dossier récapitulant l'ensemble des données des phases précliniques et cliniques est remis aux autorités de santé compétentes, dans le but d'obtenir une autorisation de mise sur le marché(AMM) [19].

3.2.5. La phase clinique IV :

Cette phase relativement récente (également appelée pharmacovigilance) consiste à surveiller, à travers le réseau de médecins et après la mise sur le marché de la molécule, l'apparition d'effets indésirables non détectés précédemment [19].

3.3. Différents types de drug design :

Les méthodes traditionnelles de découverte de médicaments reposent sur in vitro et in vivo des tests d'essais et d'erreurs de substances chimiques sur des cellules ou des animaux d'élevage, et en extrapolant les effets observés à des schémas thérapeutiques potentiels. Les composés sont d'abord criblés pour se rétrécir sur ceux qui ont un certain degré d'activité. Si ces composés partagent des caractéristiques chimiques courantes, un ou plusieurs pharmacophores sont alors développés [21].

À ce stade, les chimistes médicinaux tentent d'utiliser des relations structure-activité pour améliorer certaines caractéristiques (activité contre la cible choisie, réduire l'activité contre des sujets non apparentés) du composé au lead pour le rendre plus efficace et sûr.

Une autre méthode importante pour la découverte de médicaments est la conception rationnelle des médicaments, par laquelle les propriétés biologiques et physiques de la cible sont étudiées, et une prédiction est faite des sortes de produits chimiques qui pourraient entrer dans un site actif.

Il existe quatre méthodes différentes couramment utilisées dans la conception du médicament.

3.3.1. Ligand-Based Drug Design (ou la conception indirecte du médicament):

Elle repose sur la connaissance d'autres molécules qui se lient à la cible biologique d'intérêt, qui peut être utilisée pour dériver un modèle de pharmacophore qui définira les caractéristiques structurelles minimales nécessaires qu'une molécule doit posséder afin de lier à la cible [21].

3.3.2. Structure-Based Drug Design (ou la conception directe du médicament):

Cela repose sur la connaissance de la structure tridimensionnelle de la cible biologique obtenu par des méthodes telles que la cristallographie aux rayons X ou spectroscopie RMN. En utilisant la structure de la cible biologique, on prédit des médicaments candidats qui se lieront avec une affinité élevée et une sélectivité vis-à-vis de la cible. Les graphiques interactifs et l'intuition d'un chimiste médicinal sont utilisés dans ce processus de conception [21].

3.3.3. Conception rationnelle de médicaments :

Il commence par une hypothèse que la modulation d'une cible biologique spécifique peut avoir une valeur thérapeutique par rapport à la conception traditionnelle qui repose sur le procès-et-erreur [21].

3.3.4. Drug Design assistée par ordinateur :

Cette méthode utilise la chimie computationnelle pour découvrir, améliorer, ou étudier les médicaments et les molécules biologiquement actives liées. L'objectif le plus fondamental de cette méthodologie est de prédire l'affinité de liaison d'une molécule donnée à une cible et la cinétique de liaison associée [21].

4. Les types d’expériences :

4.1. In vivo :

In vivo (du latin « au sein de la vie») fait référence à une expérimentation en utilisant un ensemble, un organisme vivant, par opposition à un organisme partiel ou mort. Les études animales et des essais cliniques sont deux formes de recherche in vivo. Test in vivo est souvent employé plutôt que in vitro, car il est mieux adapté pour observer les effets globaux d'une expérience sur un sujet vivant [22].

Bien qu'il existe de nombreuses raisons de croire que les études in vivo ont le potentiel d'offrir un aperçu concluant quant à la nature de la médecine et de la maladie, il y a un certain nombre de façons que ces conclusions peuvent être trompeuses. Par exemple, une thérapie peut offrir un avantage à court terme, mais un mal à long terme.

4.2. In vitro :

In vitro (en latin « dans le verre ») se rapporte à la technique consistant à exécuter une procédure donnée dans un environnement contrôlé à l'extérieur d'un organisme

vivant. De nombreuses expériences en biologie cellulaire sont réalisées en dehors des organismes ou des cellules.

L'une des faiblesses respectueuses des expériences in vitro est qu'ils ne parviennent pas à reproduire les conditions cellulaires précises d'un organisme, en particulier un microbe [22].

5. Les propriétés ADME/TOX et drug-like :

Dans le processus de développement d'un médicament, pour qu' un composé chimique ait une chance de devenir un médicament candidat, il doit certes posséder une bonne activité biologique, mais il doit également avoir des propriétés qui lui confèrent une bonne biodisponibilité, c.-à-d. il doit être facile à administrer et doit atteindre sa cible en concentration suffisante pour être efficace.

En effet, une molécule médicament une fois entrée dans l'organisme, rencontre une série d'obstacles divers sur sa route vers la cible, en rapport avec son absorption, sa distribution, son métabolisme, et son excrétion. Dans l'ensemble, les propriétés d'un médicament concernant son Absorption, sa Distribution, son Métabolisme, et son Excrétion sont souvent désigner sous le terme de propriétés "ADME". A ces propriétés, s'ajoute souvent l'évaluation de la Toxicité (ADME/Tox). Un composé chimique qui possède de bonnes propriétés ADME/Tox est un bon candidat pour être un médicament, il est alors appelé "drug-like" [23].

5.1. La propriété ADME/Tox :

C'est la propriété favorable du médicament, la majorité des médicaments a rejeté à cause de cette propriété, la figure ci-dessous représente les causes de rejet des médicaments :

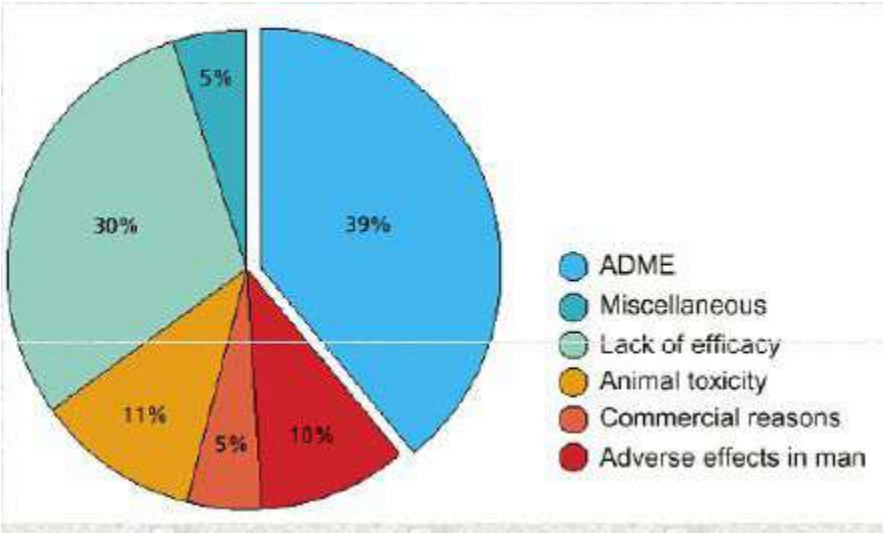


Figure 1.3 : Les raisons de rejet des médicaments [24].

5.1.1. Définition :

ADME est une abréviation de la pharmacocinétique et la pharmacologie pour "absorption, distribution, métabolisme et l'excrétion" et décrit la disposition d'un

composé pharmaceutique dans un organisme. Les quatre critères ont tous une influence aux niveaux de la drogue et de la cinétique de l'exposition au médicament dans les tissus et donc influencer la performance et de l'activité pharmacologique du composé comme un médicament [25].

5.1.2. Les propriétés :

5.1.2.1. [A]bsorption :

Détermine la biodisponibilité d'un compose. Un compose doit généralement franchir un certain nombre de barrières pour atteindre le sang principal vecteur de diffusion. Ce phénomène fait référence au concept de la biodisponibilité orale (Oral Bio-Availability, OBA), lequel se définit mathématiquement comme suit [39] :

$$biodisponibilité(\%) = \frac{Quantité\ de\ produit\ arrivant\ dans\ le\ sang}{Quantité\ de\ produit\ introduit\ dans\ l'organisme} * 100$$

Cette grandeur est définie comme le rapport dans le sang, entre la concentration la plus élevée atteinte par un composé administré oralement, et la concentration atteinte par la même dose du composé injecté directement par voie intraveineuse, sachant que, la biodisponibilité d'un médicament administrée par voie intraveineuse est de 100%. La biodisponibilité peut varier considérablement d'une espèce à l'autre .En dépit de cette variabilité, l'évaluation in silico de l'OBA (la biodisponibilité par voie orale) est faite par l'application des "règles de Lipinski" nécessaires pour la sélection de composés "drug-like".

5.1.2.2. [D]istribution :

Ce critère mesure la capacité d'une molécule à diffuser, par exemple via le flux sanguin, à travers l'organisme. En effet, une molécule doit pouvoir passer d'un compartiment à un autre, afin de pouvoir arriver in fine à l'endroit où sa cible doit être atteinte. La forte liaison à des protéines plasmatiques a un impact négatif sur la distribution d'une molécule [19].

5.1.2.3. [M]étabolisme :

Le métabolisme vise à détecter : (a) la stabilité de la molécule dans l'organisme qui impacte sur son temps d'action et (b) les métabolites de la molécule initiale, à savoir les composés résultant de sa dégradation ou de modifications enzymatiques ayant lieu au sein de l'organisme. Chez l'homme, les cytochromes P450 du foie sont les principales enzymes modifiant les xénobiotiques. Ces derniers sont notamment rendus plus hydrophiles par l'introduction d'atomes d'oxygène. Les métabolites peuvent être inactifs, plus actifs que le composé original et bien entendu potentiellement toxiques, d'où la nécessité de les caractériser et de les étudier [19].

5.1.2.4. [E]xcrétion :

Afin d'éviter les phénomènes d'accumulation, souvent synonyme de toxicité, il faut veiller à ce que les composés administrés, ainsi que leurs métabolites, soient bien excrétés de l'organisme, par exemple Via l'urine ou les selles [19].

5.1.2.5. [T]oxicité :

Comme son nom l'indique, ce critère sert à mesurer la toxicité d'un composé et de ses métabolites. Différents types de toxicité sont évalués, entre autre la cancérogénicité et la tératogénicité. Le test d'Ames est un exemple bien connu mesurant la toxicité d'un composé à travers son pouvoir mutagène / cancérogène [19].

5.2. Notion de composés « drug-like » :

Afin de réduire les coûts de développement de médicaments, il est devenu indispensable d'éliminer les mauvais composés ayant les propriétés physico-chimiques les moins similaires avec les médicaments disponibles sur le marché. De nombreuses études ont été menées sur les propriétés physico-chimiques des médicaments actuellement disponibles sur le marché. Les analyses qui en résultent ont montré l'existence de plusieurs filtres pour obtenir des composés "drug-like". La "règle des 5" de Lipinski est la plus souvent utilisée pour estimer la biodisponibilité d'un composé par voie orale à partir de sa structure bidimensionnelle (2D) [23].

En effet une molécule, possédant une activité biologique, doit respecter (ou être proche) la "règle des 5" pour être un potentiel candidat médicament "drug-like".

Cette règle contient 4 paramètres concernant les propriétés qu'un médicament peut on doit [23] :

- ✓ Masse moléculaire ≤ 500 Da.
- ✓ Log P ≤ 5 (Coefficient de partage entre l'eau et un solvant organique).
- ✓ Nombre de donneurs de liaisons hydrogène ≤ 5.
- ✓ Nombre d'accepteurs de liaisons hydrogène ≤ 10.

6. La conception des médicaments assistés par ordinateur :

La conception de médicaments assistée par ordinateur emploie la chimie computationnelle pour la découverte, l'amélioration et l'étude de médicaments et molécules biologiquement actives. En effet, l'outil informatique aide la conception de médicaments à des étapes spécifiques du processus :

- ✓ Dans l'identification des composés potentiellement thérapeutiques, en utilisant le criblage virtuel « Virtual screening».
- ✓ Dans le processus d'optimisation de l'affinité et de la sélectivité des molécules potentielles vers les têtes de série «lead» ou appelés encore prototypes.
- ✓ Dans le processus d'optimisation du lead de série par rapport aux propriétés pharmacologiques recherchées tout en maintenant une bonne affinité de cette molécule [26].

Toutes ces étapes d'intervention de l'outil informatique sont présentées dans le schéma récapitulatif suivant :

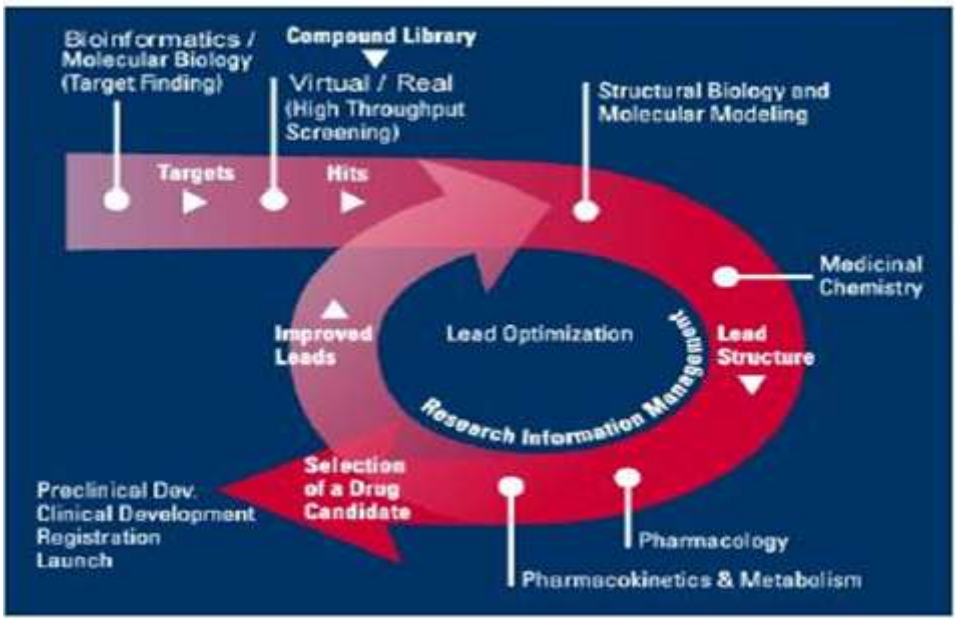


***Figure 1.4 : Processus de conception de médicaments* in-silico [*27*].**

6.1. Le terme in silico :

In silico est une expression utilisée pour signifier "effectuée sur ordinateur ou via la simulation par ordinateur." L'expression silico a été utilisé pour la première fois en public en 1989 dans l'atelier "Cellular Automata: Theory and Applications" à Los Alamos, Nouveau-Mexique. Bien que les études in silico représentent une relativement nouvelle avenue d'enquête, il a commencé à être largement utilisé dans les études qui prédisent comment les médicaments interagissent avec le corps et avec des agents pathogènes. Par exemple, une étude de 2009 utilisé des simulations de logiciels pour prédire comment certains médicaments déjà sur le marché pourrait traiter des souches multiples résistantes aux médicaments et largement résistantes aux médicaments de la tuberculose [22].

6.2. Le criblage :

L'identification d'une cible pharmaceutique peut se faire par différentes méthodes. Une fois la cible identifiée diagnostiquée, il faut tester un ensemble de molécules candidates sur cette cible, selon un processus qualifié de screening. On distingue deux types de criblage : le criblage virtuel, qui est réalisé in-silico, tout en permettant la réalisation de manière rapide et à moindre coût des prédictions de l'activité des molécules. Et le criblage réel à haut débit, quand à lui il permet de tester rapidement «in-vitro» l'activité de composés biologiques, et cela est limité par le nombre de composés à tester en un temps raisonnable et par le coût des tests [26].

6.2.1. Criblage à haut débit :

Le criblage à haut débit, en anglais High-Throughput Screening ou HTS est une méthode d'expérimentation scientifique, il représente une étape très importante dans le processus de découverte de médicaments et relève des domaines de la biologie et de la chimie. Il permet de tester un très grand nombre de composés chimiques (molécules

naturelles ou synthétiques) avec des cibles pharmacologiques afin d'identifier des hits et cela en un temps très court [28].

Le criblage à haut débit met en œuvre des robots capables d'accélérer et d'automatiser des étapes de mise en contact de molécules potentiellement actives en pharmacologie (candidats médicaments) avec un système biologique qui reproduit certains aspects de la vie d'un organisme vivant ou de son dérèglement (maladie). Ce criblage est dit à haut débit car le processus est très rapide. La technologie du criblage à haut débit est issue des progrès de la biologie moléculaire, de l'informatique et de la robotique. Depuis le début des années 90, il est devenu une technique clef de l'industrie pharmaceutique. Actuellement il existe peu d'outil statistique pour garantire la détection de hits avec un haut degré de confiance mais beaucoup de recherches sont entrain de faire afin d'introduire des méthodes statistiques robustes pour valider les hits [28].

6.2.2. Criblage virtuel :

Le criblage virtuel est une technique informatique utilisée en recherche dans le domaine de la conception des médicaments. Il consiste en un test sur de larges librairies de molécules chimiques pour en sélectionner ceux qui ont plus de chance de se transformer en médicaments. Le criblage virtuel est le fruit des avancées scientifiques dans les domaines de la modélisation moléculaire, la chimie combinatoire et la biologie moléculaire. De nos jours typiquement un million de molécules doivent être testé en une courte période de temps, d'où la nécessité d'avoir des méthodes in silico pour faire un criblage rapide et efficace [28].

Le criblage virtuel est plus rapide, moins cher, et permet un meilleur échantillonnage de l'espace chimique que les méthodes à haut-débit expérimentales. Plus précisément, le recours au criblage in silico permet d'ajuster au mieux le nombre de tests expérimentaux en fonction des contraintes budgétaires et temporelles. Ces deux techniques sont néanmoins complémentaires (voir Figure 1.5) car, à l'instar de toutes les approches in silico, le criblage virtuel nécessite une validation expérimentale à posteriori, afin d'évaluer son efficacité et de pouvoir améliorer les paramètres des programmes informatiques utilisés [29].

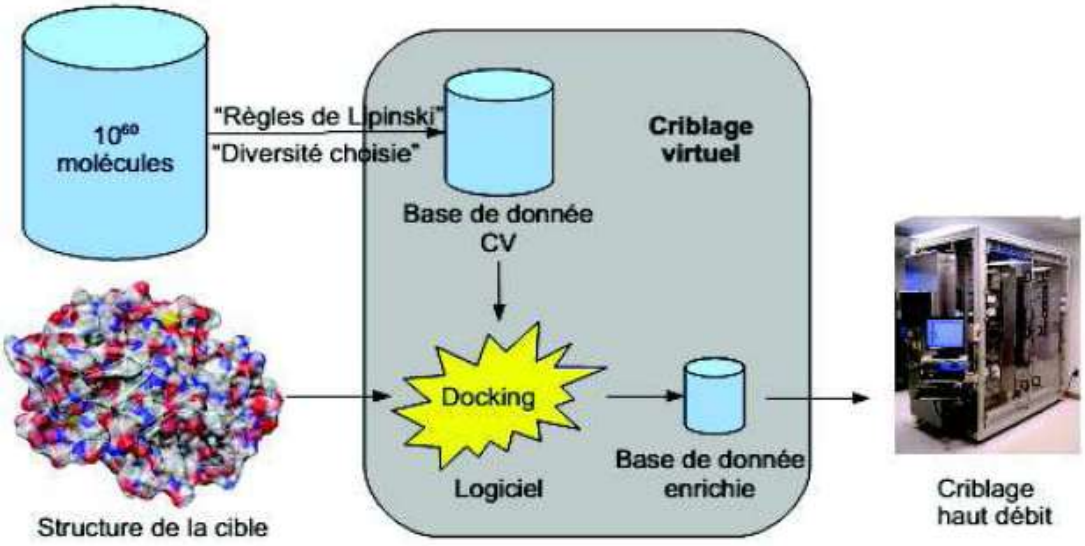


Figure 1.5 : Aperçu du criblage virtuel [29].

6.3. Le docking :

Le docking moléculaire in silico vise à prédire la structure d'un complexe moléculaire à partir des molécules isolées, ce qui est considérablement plus facile à mettre en oeuvre, moins cher et plus rapide [30].

Les logiciels de docking sont donc des outils très utiles en biologie, pharmacie et médecine, car la plupart des principes actifs sont de petites molécules (ligand) qui interagissent avec une cible biologique d'intérêt thérapeutique, généralement protéique (récepteur), afin d'influencer le mécanisme dans lequel cette protéine est impliquée [30].

6.4. Le modèle QSAR :

6.4.1. Définition :

QSAR est un modèle statistique qui relie un ensemble de descripteurs structurels d'un composé chimique à son activité biologique. La présence de caractéristiques particulières augmente la probabilité que le composé soit toxique [33].

Le QSAR a été largement utilisé pour prédire la toxicité des substances en vrac (en particulier les composés de type médicamenteux) mais, à ce jour, les études QSAR pour la prédiction de la toxicité des nanoparticules ont été rarement signalées.

Le système européen de gestion des nouveaux produits chimiques (REACH) favorise les méthodes QSAR comme méthode alternative de test de toxicité [33].

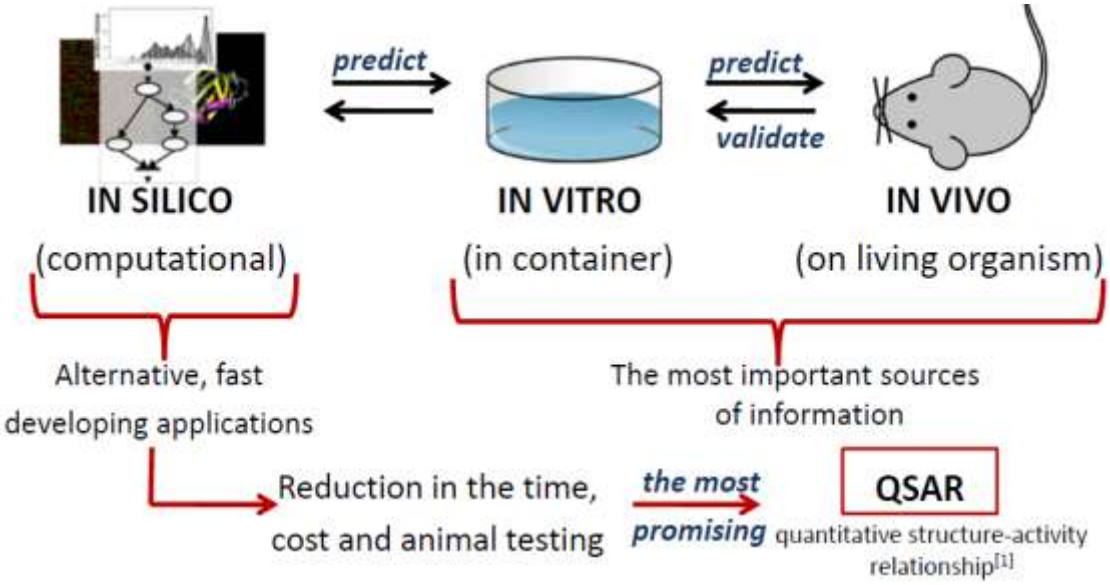


Figure1.6 : La relation entre le modèle QSAR, in silico, in vitro et in vivo [33].

6.4.2. Objectifs de QSAR : [33]

- ✓ Prévoir l'activité biologique et les propriétés physicochimiques par des moyens rationnels.
- ✓ Comprendre et rationaliser les mécanismes d'action au sein d'une série de produits chimiques.
- ✓ Économies dans le coût du développement du produit.
- ✓ Les prédictions pourraient réduire l'exigence d'essais sur animaux longs et coûteux.
- ✓ Réduction des tests sur les animaux, réduisant ainsi l'utilisation des animaux et manifestement la douleur et l'inconfort pour les animaux
- ✓ Autres domaines de la promotion de la chimie verte et verte pour accroître l'efficacité et éliminer les déchets.

6.4.3. Les descripteurs moléculaires :

Une molécule peut être représentée par un ensemble de valeurs numériques appelées descripteurs moléculaires (nous appellerons ainsi, son vecteur de description). Ces valeurs peuvent être obtenues expérimentalement mais le plus souvent, elles sont calculées à partir de la structure de la molécule. Par exemple, le poids moléculaire est calculé à partir de celui des atomes constituant le composé chimique. Les descripteurs moléculaires jouent un rôle important pour calculer la similarité entre molécules chimiques. En effet, lorsque deux molécules sont respectivement représentées par deux vecteurs de description, il est possible d'établir une mesure de similarité entre elles par comparaison de leur vecteur descripteur respectif. Cependant ils peuvent être également utilisés en complément dans des méthodes de fouille de données ou d'apprentissage automatique. Plusieurs logiciels sont disponibles pour calculer ces différents descripteurs, comme : QSARIS, Cerius2, VolSurf, Dragon. Enfin des librairies informatiques écrites en Java telles que le CDK (Chemistry Development Kit) [34] ont été développées pour la chemoinformatique et

la bioinformatique. Elles permettent à des utilisateurs avertis d'intégrer le calcul de leurs descripteurs à une plateforme informatique déjà existante.

6.4.4. La recherche de similarité :

Le concept de similarité a été introduit en chimie pharmaceutique vers le début du 20ième siècle.

Aujourd'hui, en chemoinformatique, le concept de «similarité moléculaire» fournit une méthode simple et populaire pour effectuer du criblage virtuel dans les bases de données chimiques. Elle se sert alors des méthodes de traitement de données comme le groupage (clustering) et la fouille de données (data mining). D'autre part, la «diversité moléculaire» explore la manière dont les molécules couvrent un espace chimique déterminé à travers la sélection des composants et la construction de librairies combinatoires. Les mesures de similarité et de diversité moléculaires sont donc complémentaires [35].

La recherche de similarité implique la recherche de toutes les structures dans une base de données qui sont très similaires à une structure donnée. L'utilisation la plus commune est de trouver des composés qui pourraient présenter des propriétés similaires. Par exemple, une recherche de similarité peut impliquer la recherche des structures ayant une similarité supérieure à 0,7 pour une molécule donnée.

L'utilisation classique des mesures de similarité consiste à comparer le vecteur de description d'une molécule cible avec le vecteur de description de chaque molécule d'une base. Les molécules de la base sont alors classées par ordre de similarité décroissante. Il en résulte, en sortie, une liste triée dans laquelle les structures des molécules les plus similaires à la structure cible, appelées les plus proches voisines, sont placées en tête de liste. Ces plus proches voisines sont les structures qui semblent avoir le plus grand intérêt pour l'utilisateur. La sortie peut différer en fonction de la mesure de similarité choisie. En effet, pour mesurer la similarité, on doit :

- ✓ Utiliser une méthode pour représenter les caractéristiques moléculaires (par exemple, les fingerprints)
- ✓ Et les coefficients de similitude qui sont utilisés dans diverses applications qui impliquent la mesure de similarité entre deux molécules.

Les travaux de [36] donnent les mesures de similarité les plus utilisées en chemoinformatique; le tableau 1.1 en présente quelques unes. Où a est le nombre d'éléments égaux à 1 dans le vecteur A de la première molécule, b pour le nombre d'éléments égaux à 1 dans le vecteur B de la deuxième molécule et c le nombre d'éléments communs égaux à 1 dans les deux vecteurs descripteurs A et B.

Nom de la mesure	Expression mathématique
Distance de Hamming	$D_{A,B} = a + b - 2c$
Coefficient de Tanimoto	$C_{A,B} = \frac{c}{a + b - c}$
Distance de Soergel	$D_{A,B} = \frac{a + b - 2c}{a + b - c}$
Coefficient de Dice	$D_{A,B} = \frac{2c}{a + b}$
Coefficient de Cosine	$D_{A,B} = \frac{c}{\sqrt{ab}}$

Tableau 1.1 : Exemple de mesures de similarités utilisées en chemoinformatique.

6.4.5. Représentations des fingerprints :

Les fingerprints moléculaires (ou, empreintes moléculaire) sont un concept largement utilisé en chemoinformatique qui consiste à indiquer la présence ou l'absence des différentes sous-structures dans une molécule [37]. Ils sont des descripteurs largement utilisés pour l'évaluation de similarité ou de diversité entre deux molécules pour trouver les sous-structures communes, etc. En effet, pour comparer deux molécules, le fingerprint est déterminé pour la mesure de similarité et calculé en utilisant des indicateurs de similarité tels que l'indice de Tanimoto.

Les fingerprints sont représentés sous forme des vecteurs de bits (ou valeurs binaires) dans lesquels sont codées les informations sur la structure de la molécule. Chaque bit prend la valeur 0 ou 1. Cette valeur code l'absence ou la présence de certains fragments structuraux ou d'autres caractéristiques. Parfois, un ou plusieurs bits peuvent coder une propriété physico-chimique [38].

7. De novo drug design :

7.1. Définition :

Par définition, De novo est une locution latine signifiant « à partir du début », « rafraîchi» ou « renouvelé». La conception De novo de médicament (De novo drug design, DND) est l'une des techniques les plus puissantes de CADD (Computer-Aided Drug Design) qui joue un rôle essentiel dans la recherche de composés "drug-like" possibles qui appartient à un grand espace chimique pour sélectionner ceux qui ont le plus de chances de devenir des médicaments en utilisant divers méthodes computationnelles [6].

7.2. Le principe :

- ✓ L'approche vise la complémentarité à 3 dimensions a un site de liaison de la protéine cible. Ces propriétés peuvent inclure l'électrostatique, la forme, la taille. Lipophile, l'aromaticité, etc.

- ✓ Il faut utiliser un logicielle pour produire des composés et évaluer les interactions intermoléculaires. Actuellement, la faisabilité chimique n'est pas considérée dans ce processus.
- ✓ Des fragments moléculaires sont joints individuellement à un point d'un ligand partiel dans la cavité de liaison de docking. Une évaluation de la "forme" avec la protéine est déterminée et le scoring peut être marqué. Cette amarrage et la notation des ligands potentiels peut être effectuée manuellement, cependant la génération automatisée des ligands potentiels fournit la vitesse nécessaires pour évaluer de nombreuses structures possibles, et donne une vue impartiale de l'ajustement des fragments. Cette technique peut également être utilisée pour concevoir des liens entre des fragments de molécules plus grosses de fragment accueil [31].

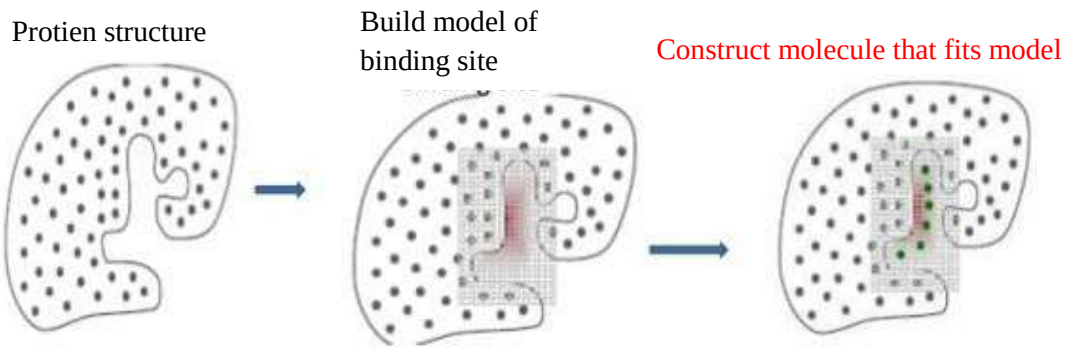


Figure 1.7: Le principe de novo Drug design [32].

7.3. Le processus de novo Drug design :

Le processus De Novo Drug Design ce fait sur 6 étapes chaque étape sera détailler par la suite :

1) Production de contraintes primaires potentiels.
2) Calcul des sites d'interaction.
3) Construire une méthode.
4) Essaye ou notation.
5) les stratégies de recherche.
6) les contraintes cibles secondaires.

7.3.1. Contraintes cibles primaires :

Les contraintes cibles primaires sont les molécules capable de faire une interaction avec le récepteur de ligand, il existe 2 type de contraintes cibles primaires :

1) Receptor base : l'interaction de la forme du récepteur de base pour la conception de médicaments.

2) A base de ligand : la fonction ligand de cible en tant que clé. Dans la conception novo de la structure de la cible devrait être connue d'une haute résolution et la liaison au site doit être bien définie. Cela devrait définit non seulement une contrainte de forme, mais les sites d'interaction hypothétiques généralement constitué de liaisons

hydrogène, interactions non covalentes électrostatiques et autres. Ceux-ci peuvent réduire considérablement l'espace de l'échantillon, comme des liaisons hydrogène et d'autres interactions anisotropes peuvent définir des orientations spécifiques [32].

7.3.2. Les stratégies de construction :

Il existe 4 types de stratégie pour la construction de la nouvelle molécule :

1) Growing
2) Linking
3) La stratégie Link/Grow
4) Méthode basée Lattice.

7.3.2.1. Le growing:

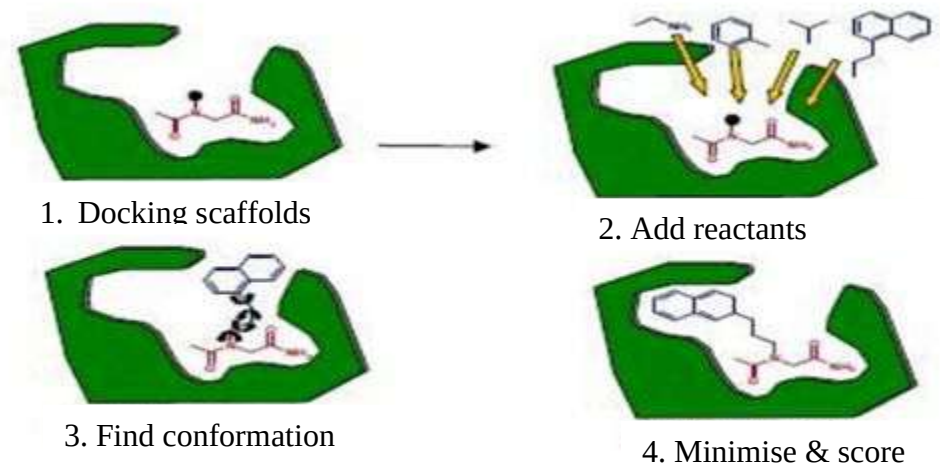


Figure 1.8 : Le growing [32].

- ✓ Un bloc des fragments est le point de départ ou de la graine.
- ✓ Les fragments sont ajoutés pour fournir des interactions de block des fragments avec le site de liaison.
- ✓ Ceux-ci comprennent des chaînes simples d'hydrocarbures, amines, alcools, et même anneau simple.
- ✓ Dans le cas de multiples graines, la croissance est généralement simultanée et continue jusqu'à ce que toutes les pièces ont été intégrées dans une seule molécule [32].

7.3.2.2. Le linking :

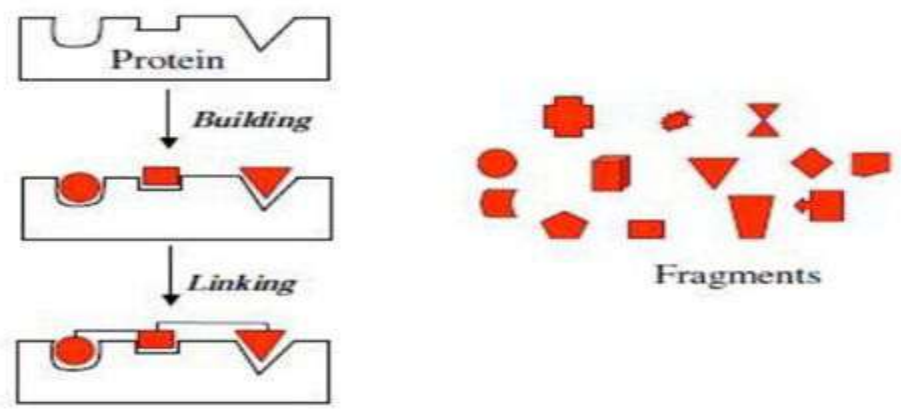


Figure 1.9 : Le linking [32].

- ✓ Les atomes de fragments ou des blocs de construction sont soit placés dans des sites d'interaction clés ou pre_docked utilisant un autre programme.
- ✓ Ils sont reliés entre eux par des règles prédéfinies pour donner une molécule complète.
- ✓ Les groupes de liaison ou linkers peuvent être prédéfinis ou générés pour satisfaire toutes les conditions requises [32].

7.3.2.3. La stratégie Link/Grow :

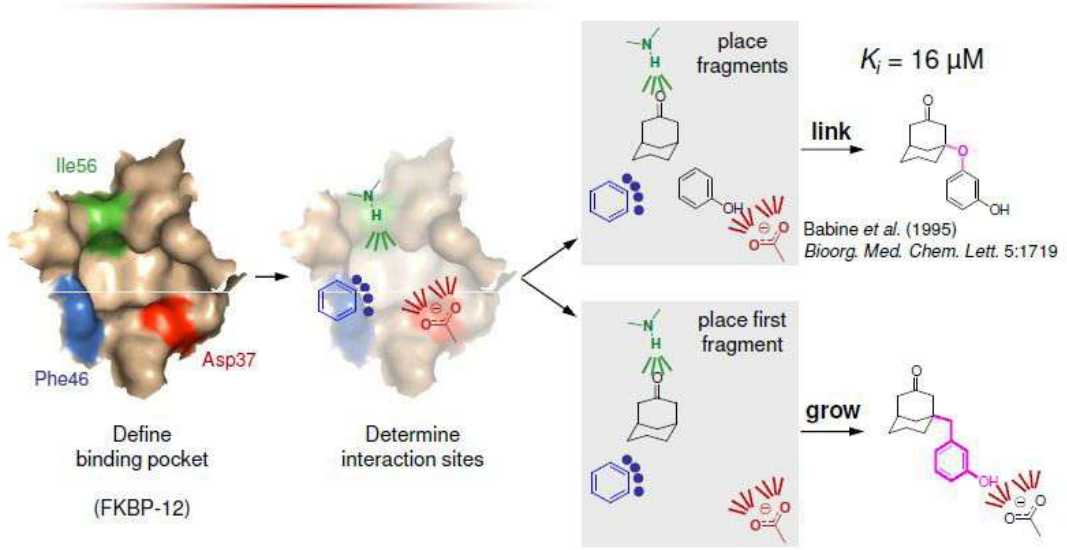


Figure 1.10 : La stratégie link/grow [32].

7.3.2.4. Méthode basée Lattice :

- ✓ Le treillis est placé dans le site de liaison et les atomes autour des sites clés sont joints en utilisant le chemin le plus court.
- ✓ Ensuite, diverses versions dont chacune comprend translation, rotation ou mutation d'atomes sont guidées par la fonction d'énergie potentielle pour aboutir finalement à une molécule cible.

- ✓ Les blocs de construction sont initialement placés au hasard et ensuite par simulation MD (Dynamique moléculaire) autorisés à réorganiser.
- ✓ Après chaque réarrangement certaines obligations ont été brisées et le processus est répété.

Au cours de cette structure haute procédure de notation ont été stockés pour une évaluation [32].

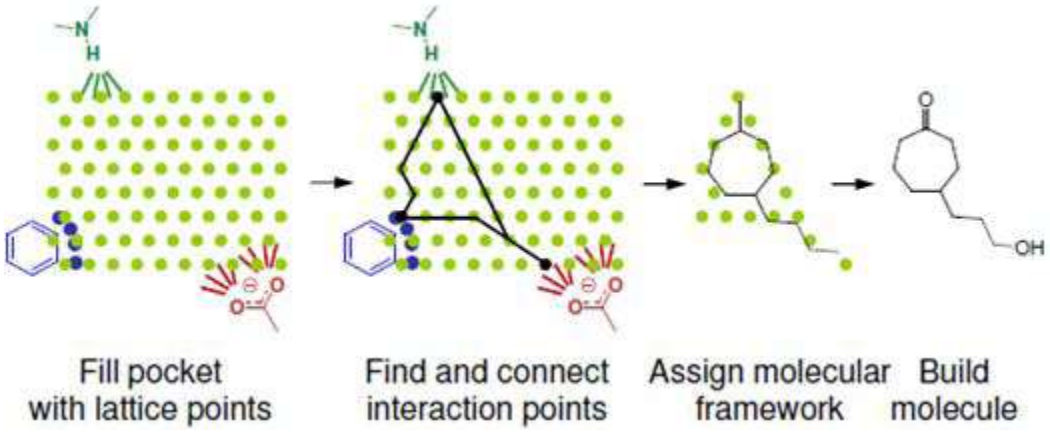


Figure 1.11 : La stratégie lattice [32].

7.3.3. Contraintes cibles secondaires :

L'affinité de liaison à lui seul ne suffit pas pour concevoir des médicaments efficaces il existe d'autre Propriétés essentielles comprennent l'absorption efficace, la distribution, le métabolisme, l'excrétion et la toxicité pour évaluer la performance de la molécule conçu.

8. Conclusion :

Dans ce chapitre, nous avons abordé le domaine de la découverte de médicaments, domaine dans lequel la chemoinformatique joue un rôle central dans toutes les étapes de processus de développement d'un médicament ainsi que dans l'analyse et l'interprétation des données de structures à l'aide d'outils informatiques. L'essor de cette nouvelle approche de conception rationnelle de médicaments dans la chemoinformatique, a permis la mise en place de nombreuses méthodes in silico pour optimiser la durée du processus de développement d'une large gamme d'applications. Son but est donc de réduire le nombre d'échecs et limiter ainsi le coût et le temps de développement pour aider les scientifiques à se concentrer sur les composés les plus prometteurs afin qu'ils puissent minimiser les efforts de tests synthétiques et biologiques.

Dans le chapitre suivant, nous allons présenter un autre aspect important de notre travail à savoir l'optimisation multi-objectif et le calcul évolutionnaire.

CHAPITRE 02 :

L'optimisation multi-objectif

Au sommaire de ce chapitre :

1. Introduction.
2. Définition d'un problème d'optimisation.
3. Les problèmes d'optimisation mono-objectif.
4. Les problèmes d'optimisation multi-objectif.
5. Classification des algorithmes d'optimisation.
6. Historique des méthodes de résolution multi-objectif.
7. Les algorithmes génétiques.
8. Les algorithmes évolutionnaires.
9. Conclusion.

1. Introduction :

De nombreux secteurs de l'industrie sont concernés par les problèmes d'optimisation. En effet, que l'on s'intéresse à l'optimisation d'un système de production, au traitement d'images, à la conception de systèmes, au design de réseaux de télécommunication ou à la bioinformatique et la chemoinformatique nous pouvons être confrontés à des problèmes d'optimisation [40].

Les problèmes d'optimisation issus de problématiques réelles sont la plupart du temps de nature multi-objectif car plusieurs critères sont à considérer simultanément. Optimiser un tel problème relève donc de l'optimisation multi-objectif [45].

Dans ce chapitre, nous décrivons d'abord le cadre de l'optimisation et les éléments de bases relatives, puis nous présentons quelques classifications des algorithmes d'optimisation et l'historique des problèmes d'optimisation multi-objectif. Nous présentons par la suite, les algorithmes évolutionnaires, telles que MOGA, NPGA, PESA, et NSGA-II.

2. Définition d'un problème d'optimisation :

L'optimisation est une discipline très ancienne qui connaît un nouveau souffle depuis l'apparition des ordinateurs. Ses méthodes sont largement utilisées dans de nombreux domaines : économie, automatique, gestion, traitement du signal, sciences de l'ingénieur, conception optimale, etc [9].

a) Un problème d'optimisation est défini par :

- ✓ Un espace de recherche (de décision) : ensemble de solutions ou de configurations constitué des différentes valeurs prises par les variables de décision.
- ✓ Une ou plusieurs fonction(s) dite objectif(s), à optimiser (minimiser ou maximiser).
- ✓ Un ensemble de contraintes à respecter.

Dans la plupart des problèmes, l'espace d'état (décision) est fini ou dénombrable. Les variables du problème peuvent être de nature diverse (réelle, entier, booléenne, etc.) et exprimer des données qualitatives ou quantitatives. La fonction objective représente le but à atteindre pour le décideur. L'ensemble de contrainte définit des conditions sur l'espace d'état que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche (solutions réalisables). La résolution optimale du problème consiste à trouver le point ou un ensemble de points de l'espace de recherche qui satisfait au mieux la fonction objectif. Le résultat est appelé valeur optimale ou optimum. 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 de bonne qualité (la plus proche possible de l'optimum). Il est vital pour déterminer si une solution est meilleure qu'une autre, que le problème introduise un critère de comparaison (une relation d'ordre). Dans la littérature il existe des

problèmes académiques utilisés comme des benchmarks : sac à dos, les fonctions de schaffer, voyageur de commerce, Flowshop, … et des problèmes réels (applications industrielles) : télécommunications, transport, environnement, … etc [40].

b) Un problème d'optimisation est donc caractérisé par:

- ✓ Le domaine des variables de décision: soit continu et on parle alors de problème continu, soit discret et on parle donc de problème combinatoire
- ✓ La nature de la fonction objectif à optimiser: soit linéaire et on parle de fonction linéaire, soit non linéaire et on parle de fonction non linéaire.
- ✓ Sa taille: problème de petite ou de grande taille.
- ✓ Le nombre de fonctions objectifs à optimiser: soit une fonction scalaire et on parle alors de problème mono-objectif, soit une fonction vectorielle et on parle donc de problème multi-objectif.
- ✓ La présence ou non des contraintes: on parle de problème sans contrainte ou avec contrainte.
- ✓ L'environnement: problème dynamique (la fonction objectif change avec le temps) [40].

3. Les problèmes d'optimisation mono-objectif :

Lorsqu'un seul objectif (critère) est donné, 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 Ω des solutions potentielles respectant certaines contraintes et une fonction d'objectif $f: \Omega \to \Psi$ qui associe à chaque solution admissible $s \epsilon \Omega$ une valeur $f(s)$. Résoudre l'instance (Ω, f) du problème d'optimisation consiste à trouver la solution optimale $s^* \epsilon\ \Omega$ qui optimise (minimise ou maximise) la valeur de la fonction objective f. Pour le cas de la minimisation : le but est de trouver $s^* \epsilon\ \Omega$ tel que $f(s^*) \leq f(s)$ pour tout élément $s \epsilon \Omega$. Un problème de maximisation peut être défini de manière similaire [40].

Malheureusement, dans de nombreuses situations rencontrées en pratique un seul critère peut être insuffisant et il n'est en général pas possible de fournir dans tous les cas une solution optimale dans un temps raisonnable. En effet, la principale difficulté que l'on rencontre en optimisation mono-objectif vient du fait que modéliser un problème sous la forme d'une équation unique peut être une tâche difficile. Ceci exige la considération simultanée de plusieurs objectifs contradictoires. L'optimisation multi-objectif autorise ces degrés de liberté qui manquaient en optimisation mono-objectif [41].

4. Les problèmes d'optimisation multi-objectif :

L'optimisation multi-objectif, aussi connue sous le nom d'optimisation multi-critère est un processus qui consiste à optimiser simultanément deux ou plusieurs objectifs, soumis à certaines contraintes. Un problème multi-objectif est caractérisé, non pas par une solution unique, mais par un nombre de solutions qui réalisent des compromis sur les différents objectifs. Il procède donc par la définition au préalable

des critères de qualité de la solution du problème, puis l'algorithme d'optimisation va résoudre le problème en cherchant les meilleures solutions en fonction de ces critères. Les problèmes d'optimisation multi-objectifs sont dans divers domaines y compris : la conception de produit/processus de fabrication, le domaine automobile, les finances, la chimie, etc. Les méthodes traditionnelles d'optimisation qui sont utilisées pour résoudre les problèmes avec un seul objectif ne conviennent donc pas pour les problèmes multi-objectifs [23].

4.1. Formulation d'un problème d'optimisation multi-objectif :

Formuler un problème d'optimisation revient à d'identifier clairement trois ingrédients : les variables d'optimisation ou de décision, les objectifs du problème et les contraintes. D'une façon générale, le problème d'optimisation multi-objectif est exprimé par l'équation suivant [42,43] :

$$\min(ou\ max) \quad f(x) = [f_1(x), f_2(x), \dots, f_m(x)], \qquad x \epsilon E = R^m\ et\ m > 2$$

$$\begin{cases} g_i(x) \leq 0, & i = 1, \dots, p \\ h_j(x) = 0, & j = 1, \dots, q \\ x_{kMin} \leq x_k \leq x_{kMax}, & k = 1, \dots, n \end{cases}$$

Où $m > 2$ est le nombre de fonctions objectifs et $f(x) = f_1(x), f_2(x), \dots, f_m(x)$ est le vecteur des fonctions à optimiser. Le vecteur $x = (x_1, \dots, x_n)$ est le vecteur des variables de décisions.

$(g_i(x) \leq 0, h_j(x) = 0, x_{kMin} \leq x_k \leq x_{kMax})$ sont respectivement les contraintes d'inégalités, les contrains d'égalités et les contraintes de domaine. Les x_{kMin} et x_{kMax} sont respectivement les bornes inférieure et supérieure du variable de décision x_k.

R^m est considéré comme l'ensemble des solutions réalisables ou faisables. On doit alors distinguer la notion d'espace de recherche, représentant l'ensemble des valeurs pouvant être prises par les variables, et la notion d'espace réalisable, sous-espace des variables satisfaisant les contraintes. L'image de l'espace de recherche par la fonction objectif f est appelée espace des objectifs (ou espace des critères). Ces notions sont représentées dans la Figure 2.1 :

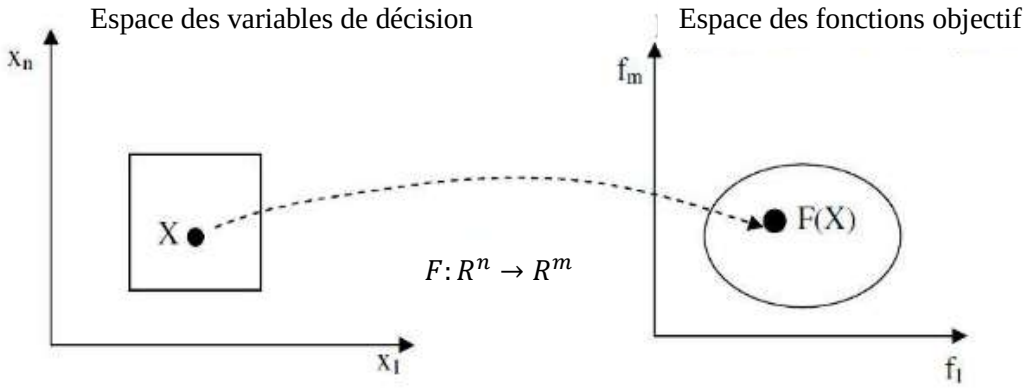


Figure 2.1 : Représentation d'un problème multi-objectif [44].

Une méthode d'optimisation est alors la recherche d'un point ou l'ensemble des points dans l'espace de recherche qui minimisent (ou maximisent) les fonctions objectifs sous l'ensemble des contraintes.

Contrairement à l'optimisation mono-objectif, la solution d'un problème multi-objectif n'est pas unique, mais est un ensemble de solutions non dominées, connu comme l'ensemble des solutions Pareto Optimales (PO) [45].

4.2. Notions de dominance et optimalité au sens Pareto :

Au XIXème siècle, Vilfredo Pereto, un mathématicien italien, formule le concept suivant : dans un problème multi-objectif, il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critères. Cet équilibre a été référencé comme l'optimum de Pareto [64]. Un point x et dit Pareto-optimal s'il n'est dominé par aucun autre point appartenant à E. Ces points sont également appelés solution non inferieure ou non dominées. Pour mieux comprendre le concept de Pareto-optimal ou optimalité de Pareto, introduisons d'abord la notion de "dominance de Pareto".

Définition 1 : Domination de Pareto

Soient deux vecteurs de décision a et b [46] :

a $\succ$ b (a domine b) $\Leftrightarrow f(a) > f(b)$

a $\succcurlyeq$ b (a domine faiblement b) $\Leftrightarrow f(a) \geq f(b)$

a$\sim$b (a est indifférent pour b) $\Leftrightarrow f(a) \not\geq f(b) \wedge f(b) \not\geq f(a)$

Les définitions pour un problème de minimisation $(\prec, \leq, \sim)$sont analogues.

Dans la figure 2.2 (problème de maximisation), le rectangle gris clair entoure la région dans l'espace des objectifs qui est dominée par le vecteur de décision représenté par B. Le rectangle gris foncé contient les vecteurs d'objectifs qui correspondent aux vecteurs de décision dominant la solution associée à B. Toutes les solutions dont le vecteur de décision correspondant est ni dans le rectangle clair, ni dans le rectangle foncé, sont indifférentes pour la solution représentée par B [46].

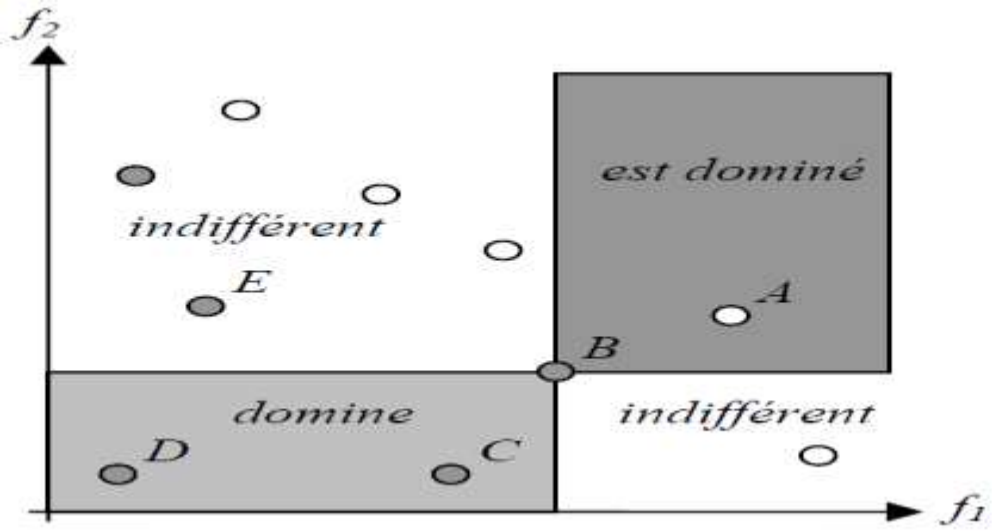


Figure 2.2 : La notion de dominance [46].

Basé sur le concept de domination de Pareto, les critères d'optimalité pour les problèmes d'optimisation multi-objectif peuvent être introduits. Dans la figure 2.1, A est unique parmi B, C, D et E : son vecteur de décision A n'est dominé par aucun autre vecteur de décision. Cela signifie que A est optimal dans le sens où il ne peut être amélioré sur aucun objectif sans causer la dégradation d'au moins un autre objectif. De telles solutions sont appelées solutions optimales de Pareto; parfois le terme de solutions non-inférieures [47] est utilisé.

Définition 2 : Optimalité de Pareto

Un vecteur de décision $x \epsilon X_f$ est dit non dominé par rapport à un ensemble $A \subseteq X_f$ ssi : $\nexists\, a \epsilon A : a \succ x$

Dans la figure 2.1 les points blancs représentent les solutions optimales de Pareto. Elles sont indifférentes les unes des autres. Cela constitue la principale différence avec les problèmes à objectif unique : il n'y a pas une seule solution optimale mais plutôt un ensemble de compromis optimaux. Aucun d'eux ne peut être identifié comme étant meilleur que les autres sans qu'une information de préférence ne soit incluse [46].

Définition 3 : Ensemble optimal de pareto et front de pareto

L'ensemble des solutions optimales de Pareto est appelé ensemble optimal de Pareto ; les vecteurs d'objectifs correspondants constituent la frontière (ou front) de Pareto ou plus généralement la surface de Pareto (figure 2.3) [46].

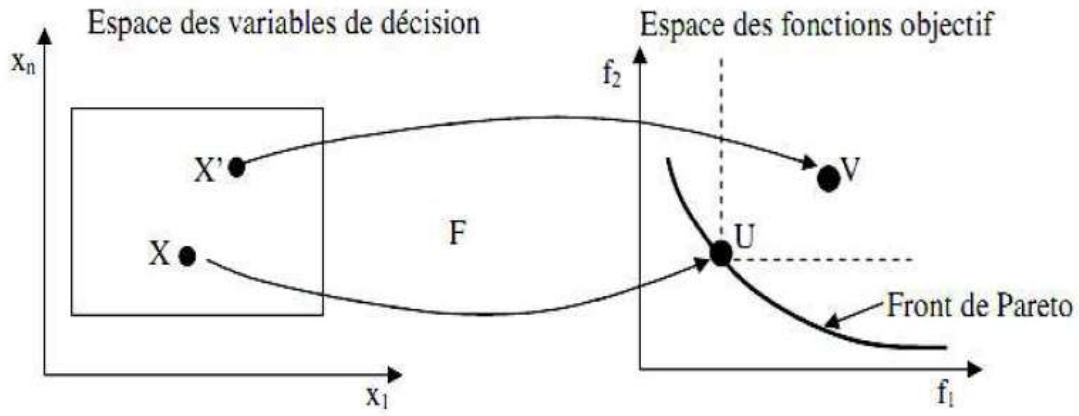


Figure 2.3 : Front de Pareto [44].

Dans la figure 2.4, les points A et B sont deux points du front de Pareto : A ne domine pas B, B ne domine pas A, mais tous les deux dominent le point C (problème de minimisation). Le but de l'optimisation multi-objectif est de déterminer le front de Pareto pour un problème donné [48].

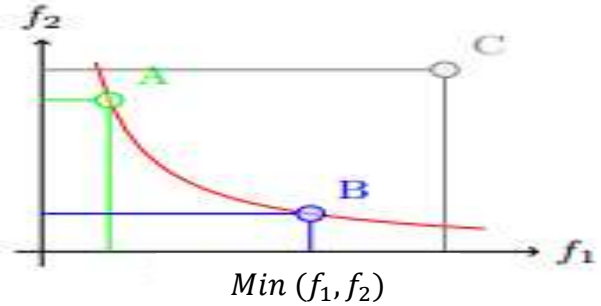


Figure 2.4 : Exemple de front de pareto [48].

4.3. Approches de résolution multi-objectif :

La résolution de problèmes multi-objectifs relève de deux disciplines assez différentes. En effet, résoudre un problème multi-objectif peut être divisé en deux phases [40] :

1) **La recherche des solutions de meilleur compromis** : C'est la phase d'optimisation multi-objectif.
2) **Le choix de la solution à retenir** : C'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi-objective et cela fait appel à la théorie de la décision.

Dans les différentes publications, nous rencontrons deux classifications différentes des approches de résolution de problèmes multi-objectifs. Le premier classement adopte un point de vue décideur, les approches sont classées en fonction de l'usage que l'on désire faire. Le deuxième classement adopte un point de vue concepteur, les approches sont triées de leur façon de traiter les fonctions objectives. Ainsi avant de se lancer dans la résolution d'un problème multi-objectif, il faut se poser la question du type d'approche de résolution à utiliser [40].

4.3.1. Classification « point de vue décideur » :

La solution d'un problème multi-objectif est un ensemble de solutions. Cependant, pour un problème réel, une seule solution pourra être déployée. Un choix par un décideur doit donc être effectué ; le décideur peut intervenir en amont de la résolution, après celle-ci, ou de manière interactive [49] :

4.3.1.1. Les approches a priori :

Le décideur définit ses préférences entre les différents objectifs avant d'utiliser la méthode d'optimisation.

4.3.1.2. Les approches progressives :

Le décideur affine son choix de compromis au fur et à mesure du déroulement de la méthode d'optimisation.

4.3.1.3. Les approches a posteriori :

Le décideur choisit la solution de son choix parmi l'ensemble des solutions fournies par la méthode d'optimisation.

4.3.2. Classification « point de vue concepteur » :

Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum Pareto. Les approches utilisées pour la résolution de problèmes multi-objctifs peuvent être classées en deux catégories [Barichard, 2003] : les approches non Pareto et les approches Pareto [40].

4.3.2.1. Les approches non Pareto :

Ces approches ne traitent pas le problème comme un véritable problème multi-objectif. Elles cherchent à ramener le problème initial à un ou plusieurs problèmes mono-objectifs.

4.3.2.2. Les approches Pareto :

Les approches pareto ne transforment pas les objectifs du problème, ceux-ci sont traités sans aucune distinction pendant la résolution.

5. Classification des algorithmes d'optimisation :

La classification des problèmes d'optimisation variable suivent le point de vue considéré :

- ✓ Algorithmes déterministes / algorithmes stochastiques.
- ✓ Algorithmes d'optimisation locale/globale.
- ✓ Les méthodes exactes/approchée.

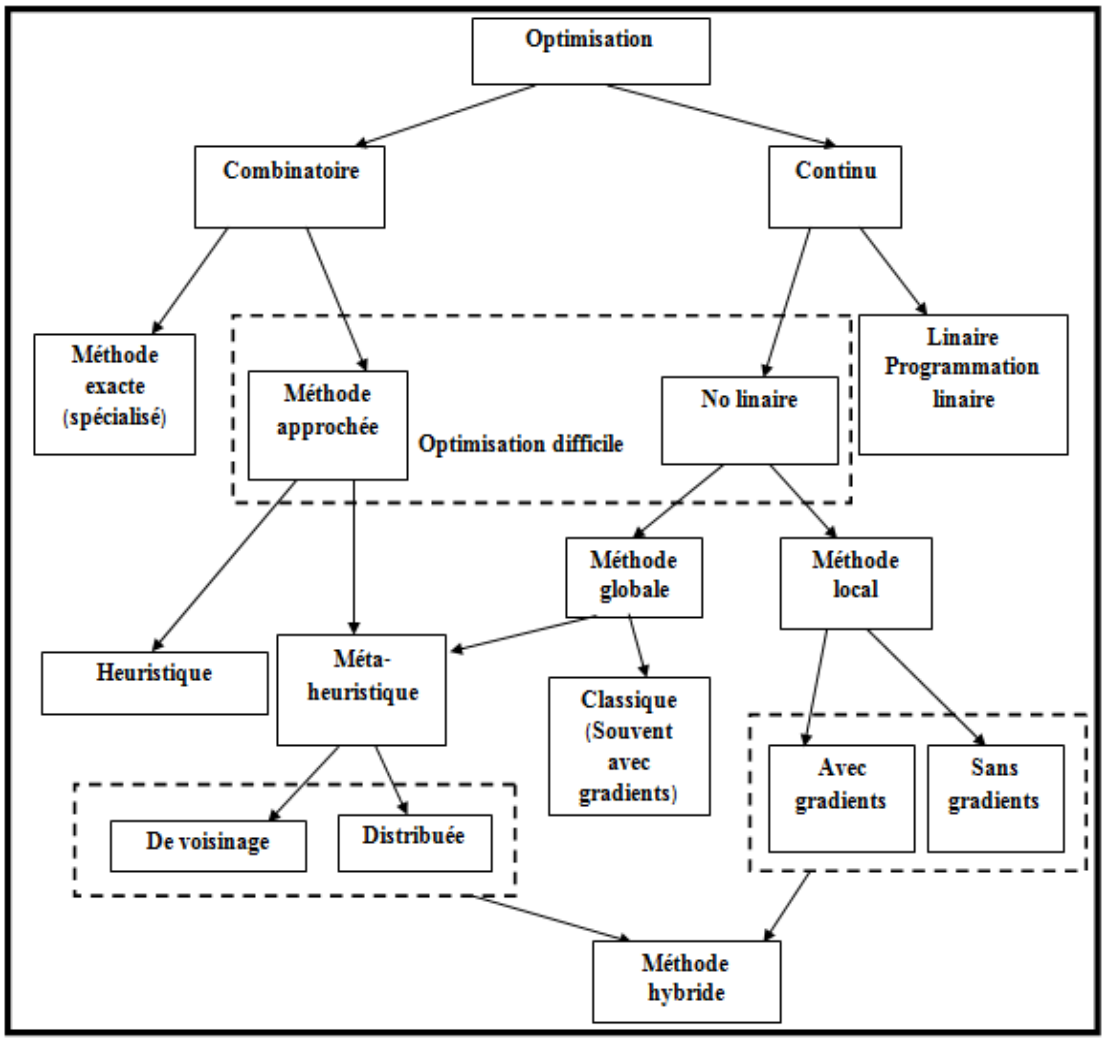


Figure2.5 : Classification des méthodes de résolution.

5.1. Algorithmes déterministes / algorithmes stochastiques :

5.1.1. L'algorithme déterministe : sont basées sur le calcul de la dérivée du problème, ou sur des approximations de cette dérivée. Elles nécessitent donc quelques informations sur le vecteur gradient, soit qu'elles cherchent le point où il est nul, soit qu'elles utilisent sa direction [55].

5.1.2. L'algorithme stochastique : utilisent un ensemble d'actions qui cherchent la solution optimale de façon « aléatoire orientée », sans avoir besoin d'aucune information sur les dérivées ou sur le comportement du problème [55].

5.2. Algorithmes d'optimisation locale/globale :

5.2.1. Algorithmes d'optimisation locale : Tout algorithme piégé par le premier optimum rencontré ne permettent pas d'obtenir une solution proche de l'optimum global en raison de la trop grande cardinalité de l'espace de recherche.

5.2.2. Algorithmes d'optimisation globale : Tout algorithme qui n'est pas sensible aux minima locaux. Algorithme permettant d'obtenir une solution proche de l'optimum global.

5.3. Les méthodes exactes/approchée :

5.3.1. Les méthodes exactes :

Les méthodes exactes permettent de calculer la (ou les) solution(s) optimale(s), on est sûr qu'il n'existe aucune solution de meilleure qualité [50].

5.3.2. Les méthodes approchées :

Certains problèmes d'optimisation demeurent cependant hors de portée des méthodes exactes. Un certain nombre de caractéristiques peuvent en effet être problématiques, comme l'absence de convexité stricte (multi-modalité), l'existence de discontinuités, une fonction non dérivable, présence de bruit, etc.

Dans de tels cas, le problème d'optimisation est dit difficile, car aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable, on devra alors faire appel à des heuristiques et des métaheuristiques permettant une optimisation approchée [51].

5.3.2.1. L'heuristique :

Cette classe ne permet pas de donner une garantie d'optimalité ; cependant, elles sont bien adaptées aux problèmes complexes de grande taille. En effet, une heuristique permet de calculer une solution approchée en un temps raisonnable comparativement à une méthode exacte [50].

5.3.2.2. La métaheuristique :

Ce sont des heuristiques dédiées à un problème donné et celles plus génériques, ces dernières peuvent être appliquées à un large panel de problèmes [50].

6. Historique des méthodes de résolution multi-objectif :

6.1. Méthodes exactes pour l'optimisation multi-objectif :

Dans la littérature, plusieurs méthodes exactes multi-objectifs basées sur le branch and bound [Sen, 1988], sur l'algorithme A* [Stewart, 1991] et la programmation dynamique [Carraway, 1990], la méthode à deux phases [Ulungu, 1995] ont été proposées pour résoudre de petits problèmes à deux objectifs. Pour les problèmes à plus de deux critères ou de grandes tailles, il n'existe pas de procédures exactes efficaces, étant données les difficultés simultanées de la complexité NP-difficile, et le cadre multi-objectif des problèmes [40].

6.2. Métaheuristiques pour l'optimisation multi-objectif :

Les métaheuristiques ont été largement utilisées pour la résolution de problèmes multi-objectifs. Ceci est d'autant plus vrai pour les algorithmes évolutionnaires travaillant sur des populations de solutions et étant bien adaptés pour générer des fronts de solutions. Ainsi Deb publiait dès 2001 un livre sur l'utilisation des algorithmes évolutionnaires pour l'optimisation multi-objectif [45].

Un autre livre publiée par Coello Coello et Lamont montre le spectre des applications multi-objectifs abordées par les algorithmes évolutionnaires [Coello,

2004] .Ces applications vont du design d'appareils électromagnétiques, au design de circuits logiques, à l'optimisation de problèmes de tournées en passant par l'analyse de promoteurs dans le domaine de la bio-informatique.

L'approche Pareto a été introduite initialement dans les algorithmes génétiques par Goldberg [Goldberg, 1989]. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble compromis de solutions (le front Pareto). 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 [40].

- ✓ **NSGA:** Nondominated Sorting Genetic Algorithm by Srinivas and Deb, 94.
- ✓ **NPGA:** Niched Pareto Genetic Algorithm by Horn, Nafpliotis and Goldberg, 94.
- ✓ **MOGA:** Multi-Objective Genetic Algorithm by Fonseca & Fleming 93. Murata & Ishibuchi 95
- ✓ **MGK:** Morita-G.-Katoh Algorithm by Morita & Katoh, 98, 2001.
- ✓ **SPEA:** Strength Pareto Evolutionary Algorithm by Zitzler & Thiele, 98.
- ✓ **PAES:** Pareto Archived Evolution Strategy by Knowles & Corne, 99.

Ainsi, les algorithmes évolutionnaires dédiés à l'optimisation combinatoire multi-objectif ont bien évolué ces dernières années. Cependant ces méthodes ne restent que des approches heuristiques et si sur des applications réelles elles semblent être performantes, l'objectif est toujours de chercher à améliorer les résultats. Dans cet objectif, une approche prometteuse concerne la coopération de différentes méthodes et en particulier la coopération entre méthodes exactes et méthodes heuristiques.

Puis plus tard, Patrick Siarry publiait un livre sur les principales metaheuristiques : Recuits simulé, recherche avec tabous, recherche à voisinages variables, méthodes GRASP, algorithmes évolutionnaires, fourmis artificielles, essaims particulaires et autres méthodes d'optimisation.

7. Les algorithmes génétiques :

7.1. Définition :

Un algorithme génétique est un algorithme stochastique itératif qui utilise une population d'individus représentant les solutions potentielles du problème d'optimisation à résoudre. Cette population va évoluer de générations en générations : les individus les « mieux adaptés » auront plus de chance de se reproduire et donc de transmettre leurs caractéristiques héréditaires [56].

Dans le vocabulaire employé par les algorithmes génétiques, les termes comme individu, population, gène, chromosome, parent, enfant, reproduction, croisement et mutation sont utilisés :

- ✓ L'environnement se rapporte à l'espace de recherche qui définit l'ensemble des configurations possibles des paramètres de la fonction à optimiser.
- ✓ Un individu dans cet environnement est une configuration possible des paramètres.
- ✓ Un ensemble d'individus forme une population.

- ✓ Chaque individu peut être représenté par un chromosome qui est composé d'une chaîne de gènes contenant les caractéristiques génétiques de cet individu.
- ✓ Le gène étant la partie élémentaire d'un chromosome représente un trait de caractère ou une fonction particulière. Dans le codage binaire, un gène vaut 0 ou 1.
- ✓ Fonction d'adaptation : est la fonction à optimiser ;
- ✓ Adaptation : mesure la qualité de la solution (l'individu) par rapport au problème ;
- ✓ Reproduction : un enfant hérite des gènes de ses parents ;
- ✓ Génération : l'ensemble de la population à un moment donné du processus ;

Le but : est de trouver (atteindre) l'individu le mieux adapté.

7.2. Les phases d'un algorithme génétique :

Les algorithmes génétiques sont basés sur les phases suivantes :

1) Création d'une population initiale.
2) Evaluation des individus de la population.
3) Sélection des meilleurs individus.
4) Reproduction : Croisement et mutation.
5) Formation d'une nouvelle génération.

Retour au 2 si non convergence.

7.2.1. Codage et création de la population initiale :

La première étape dans la mise en œuvre des algorithmes génétiques consiste donc à définir et à coder convenablement les variables du problème. En fonction du problème étudié, les codes utilisés peuvent être binaires ou réels [58].

- ✓ Les codages binaires ont été très employés à l'origine. A chaque variable xi du problème à optimiser correspond un gène défini par une séquence de symboles (nombres binaires, entiers ou réels, etc.). Un ensemble de gènes (les variables du problème) forme un chromosome. La figure 2.6 illustre un exemple de codage binaire sur 8 bits d'un chromosome dont les variables sont définies dans les domaines suivants :

$$x_1 \in [0,2];\ x_2 \in [3,5];\ x_3 \in [0,1];\ x_4 \in [0,5];\ x_5 \in [1,2];$$

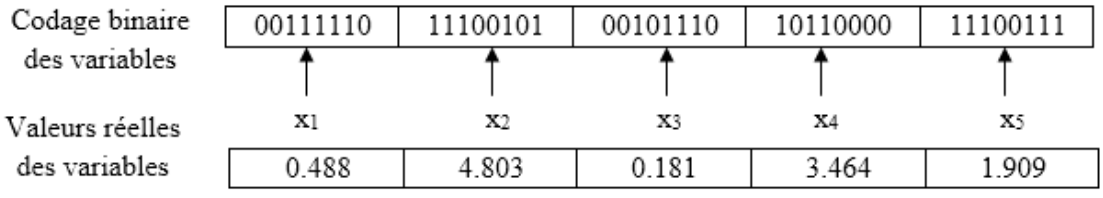


Figure 2.6 : Codage binaire des variables.

- ✓ Le codage réel est désormais largement utilisé, notamment dans les domaines applicatifs, pour l'optimisation de problème à variables continués. Il permet d'exploiter la gradualité des fonctions qui utilisent des variables continues, c'est-à-dire qu'une légère variation des variables se traduit par une légère variation de

la fonction. Le chromosome dans le codage réel est un vecteur de nombres réels, où chaque gène représente une variable du problème. Le raffinement des solutions est ainsi facilité par rapport au codage binaire [59]. La figure 2.7 présente un exemple de codage réel des variables.

	x_1	x_2	x_3	x_4	x_5
Valeurs réelles des variables	1.22	4.51	0.905	3.46	4.55

Figure 2.7: Codage réel des variables.

L'algorithme génétique démarre avec une population initiale d'individus dans le codage retenu. Le choix des individus conditionne fortement la rapidité de l'algorithme.

7.2.2. Evaluation de la population :

Cette étape consiste à évaluer chaque solution contenue dans la population, la mesure de performance des solutions s'appuie sur la valeur de la fonction objectif (appelée aussi, "fitness" ou "fonction d'évaluation de individu") [60]. Cette étape permet de classer les solutions, afin de déterminer les solutions qui seront sélectionnées pour construire une nouvelle population de solutions. La figure 2.8 donne un exemple de l'évaluation d'un individu (codé sur 8 bits). Cette évaluation consiste à transformer la chaine 0/1 du chromosome en une valeur réelle, appelée valeur d'adaptation. Si cette fonction est uniquement basée sur la fonction objective du problème, on utilisera le terme de fonction de coût pour la designer.

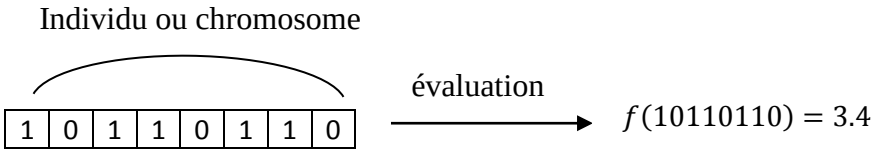


Figure 2.8 : Evaluation d'un individu.

La valeur d'adaptation obtenue pour chaque chromosome lors de l'évaluation est utilisée, notamment lors des opérations de sélection, pour choisir les individus amenés à se reproduire par des croisements ou à être utilisés par d'autres operateurs génétiques.

7.2.3. Les opérateurs génétiques :

Les algorithmes génétiques utilisent principalement trois types d'opérateurs génétiques qui sont la "sélection", le "croisement" et la "mutation". Ces opérateurs sont appliqués aux individus de la population avec l'espoir d'améliorer leurs caractéristiques génétiques et de les amener à s'adapter à leur environnement [61].

7.2.3.1. L'opérateur de sélection :

La sélection a pour objectif d'identifier les individus qui doivent se reproduire. Cet opérateur ne crée pas de nouveaux individus mais identifie les individus sur la base de leur fonction d'adaptation (fitness), les individus les mieux adaptés sont sélectionnés alors que les moins adaptés sont écartés. Cette opération est peut être la plus importante dans les algorithmes génétiques puisqu'elle permet aux individus d'une

population de survivre, de se reproduire ou de mourir [23]. Il existe plusieurs techniques de sélection sont proposées dans la littérature, nous présentons quatre :

a. Sélection par roulette (wheel) :

La sélection des individus par le système de la roulette s'inspire des roues de loterie. A chacun des individus de la population est associé un secteur d'une roue. L'angle du secteur étant proportionnel à la qualité de l'individu qu'il représente. Vous tournez la roue et vous obtenez un individu. Les tirages des individus sont ainsi pondérés par leur qualité. Et presque logiquement, les meilleurs individus ont plus de chance d'être croisés et de participer à l'amélioration de notre population [62].

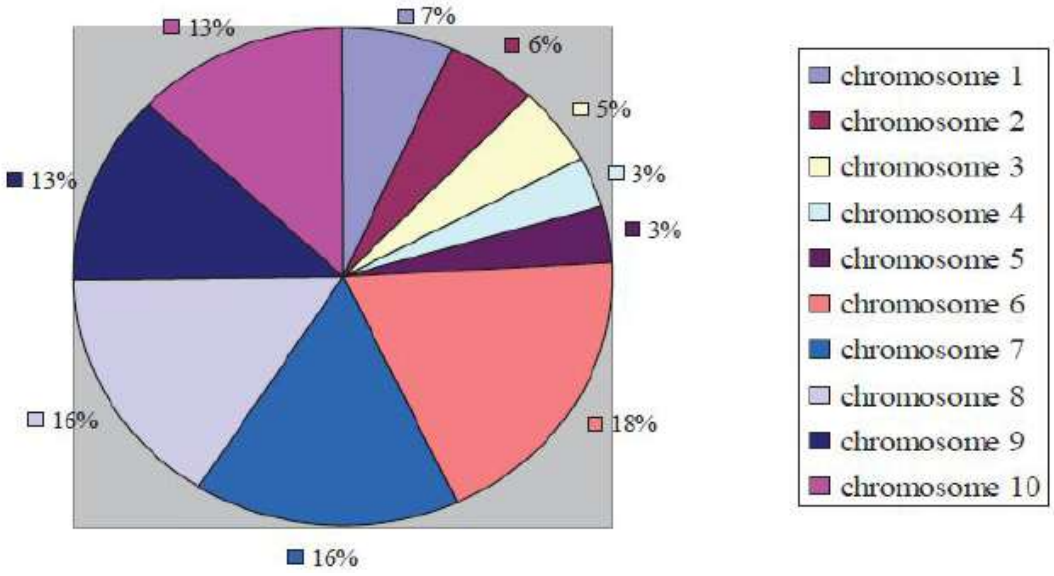


Figure 2.9 : Exemple de sélection par roulette [56].

b. Sélection par tournoi :

Contrairement à la sélection par roulette biaisée, la sélection par tournois ne fait pas appel à la valeur de la fonction de l'adaptation des individus mais se contente de leur classement relatif. C'est une méthode qui augmente les chances pour les individus de faible efficacité de participer à l'amélioration de la population. Le principe consiste à choisir aléatoirement parmi la population un groupe de q individus et celui ayant la valeur d'adaptation la plus élevée sera retenu pour participer à la reproduction. L'opération est répétée autant de fois qu'il y a d'individus à sélectionner [23].

c. Sélection par rang :

C'est une procédure similaire à celle de la sélection par roulette mais, la différence est que la proportion est calculée sur les rangs et non sur la valeur de la fonction d'adaptation. L'ensemble des individus est représenté sur un segment de droite dont les valeurs sont comprises entre 0 et 1. On effectue d'abord un tri des individus par ordre croissant de la fonction d'adaptation. On leur attribue ensuite un rang, l'individu de moins bonne valeur d'adaptation obtient le rang 1 et celui de meilleure valeur d'adaptation le rang n (où n est la taille de la population) [23].

d. L'élitisme :

A la création d'une nouvelle population, il y a de grandes chances que les meilleurs chromosomes soient perdus après les opérations d'hybridation et de mutation. Pour éviter cela, on utilise la méthode d'élitisme. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Ensuite, on génère le reste de la population selon l'algorithme de reproduction usuel. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions [63].

7.2.3.2. L'opérateur de croisement :

Le phénomène de croisement, appelé aussi recombinaison, vise à orienter la recherche vers les régions prometteuses de l'espace de recherche. Pour cela, les individus enfants (descendants) crées par croisement héritent les meilleures caractéristiques de leurs parents et peuvent exploiter au mieux l'espace de recherche. Cependant, aucun gène nouveau n'est introduit dans la population, cet opérateur effectue seulement une combinaison des gènes des deux individus parents pour donner deux nouveaux chromosomes d'individus enfants. Il a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes pour générer un ou plusieurs nouveaux individus qui peuvent se rapprocher de la solution optimum [23]. De nombreux types de croisement existent dans la littérature, les opérateurs les plus connus en fonction du nombre de fragments de chromosome échangés sont: le croisement à un point, croisement à deux points et croisement uniforme.

a. Le croisement à un point :

Dans cet opérateur deux individus se croisent et s'échangent des portions de leur information en un seul point. Pour effectuer ce type de croisement sur des chromosomes de M gènes, la population courante est divisée en deux sous populations de même taille et chaque couple formé par un membre (un parent) provenant de chaque sous population participe à un croisement avec une probabilité Pc souvent supérieure à 0.5. Puis, on tire aléatoirement une portion de chacun des parents. On échange ensuite les deux sous-chaines terminales de chacun des deux chromosomes, ce qui produit deux nouveaux individus enfants. La figure 2.10 illustre un croisement à un point.

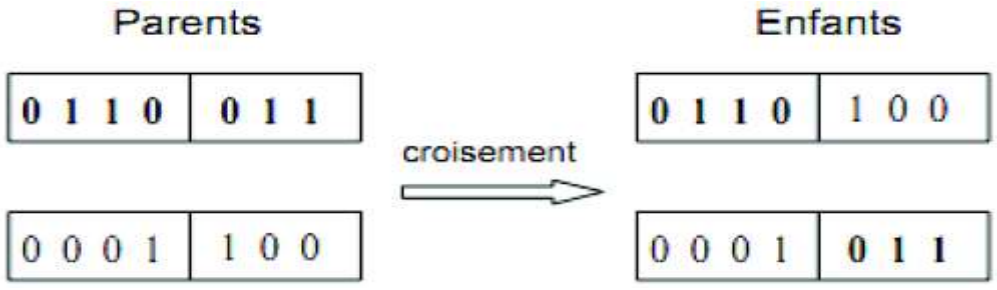


Figure 2.10: Croisement à un point pour des chromosomes codés en binaire.

b. Le croisement à deux points :

Contrairement à l'opérateur précédent, il y a au moins deux points de croisement. La figure 2.11 ci-dessous illustre ce type de croisement.

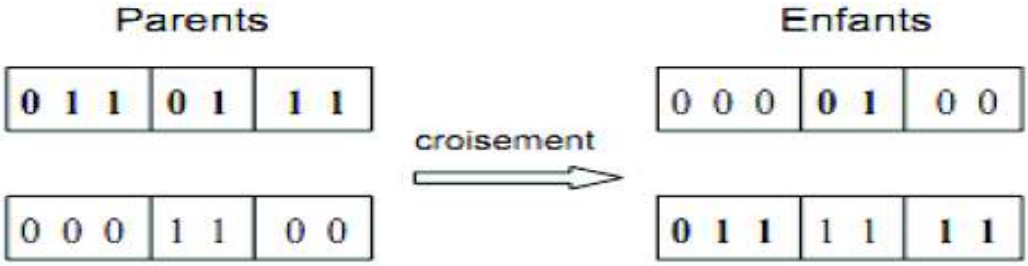


Figure 2.11 : Croisement à deux points pour des chromosomes codés en binaire.

Cette procédure peut être facilement étendue à plusieurs points, on parle alors de croisement multi-points.

c. Le croisement uniforme :

La mise en œuvre de ce type de croisement est fort simple, elle consiste à définir de manière aléatoire un "masque" de croisement, c'est-à-dire une chaîne de bits (composé de 0 et 1) de même longueur que les chromosomes des parents sur lesquels il sera appliqué. Ce masque est destiné à savoir, pour chaque locus, de quel parent le premier fils devra hériter du gène s'y trouvant. Lorsque le bit du masque vaut 0, le premier enfant hérite le bit correspondant du premier parent, sinon il hérite celui du second parent. Le second enfant est le complémentaire du premier. Ce croisement (Figure 2.12) peut être considéré comme une généralisation du croisement multi-point sans connaissance préalable du point de croisement.

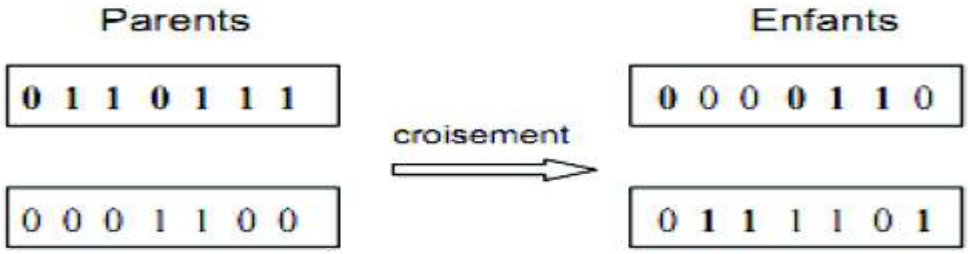


Figure 2.12 : Croisement uniforme.

7.2.3.3. L'opérateur de mutation :

Cet opérateur génétique est généralement appliqué après le croisement des parents et aide à fournir de nouvelles caractéristiques génétiques, difficiles à obtenir par le seul opérateur de croisement. Elle consiste à modifier aléatoirement un petit nombre de gènes, comme illustre dans la figure 2.13 :

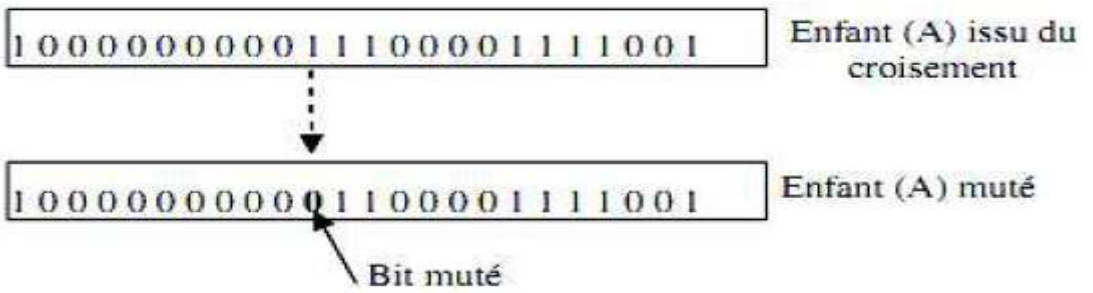


Figure 2.13 : Opérateur de mutation.

7.2.4. La fonction d'adaptation (fitness) :

C'est la fonction à optimiser :

- ✓ Son expression peut être immédiate et assez simple (fonction mathématique), mais ceci n'est pas toujours le cas. On peut donc être amener à en considérer une approximation.
- ✓ A définir en fonction du problème.

7.2.5. Critère d'arrêt :

Le cycle de génération et de sélection de population est répété jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère peut être notamment un nombre maximum de générations, un temps maximal de calcul, une valeur de fitness minimale, ou une convergence vers une solution satisfaisante. Nous pouvons également faire des combinaisons des critères d'arrêt précédents.

7.3. Pseudo code d'un algorithme génétique :

Un algorithme génétique générique à la forme suivante [57] :

1) Initialiser la population initiale P.
2) Evaluer P.
3) TantQue (Pas Convergence) faire :
   a) P' = Sélection des Parents dans P.
   b) P' = Appliquer Opérateur de Croisement sur P'.
   c) P' = Appliquer Opérateur de Mutation sur P'.
   d) P = Remplacer les Anciens de P par leurs Descendants de P'.
   e) Evaluer P

FinTantQue

7.4. Organigramme d'un algorithme génétique :

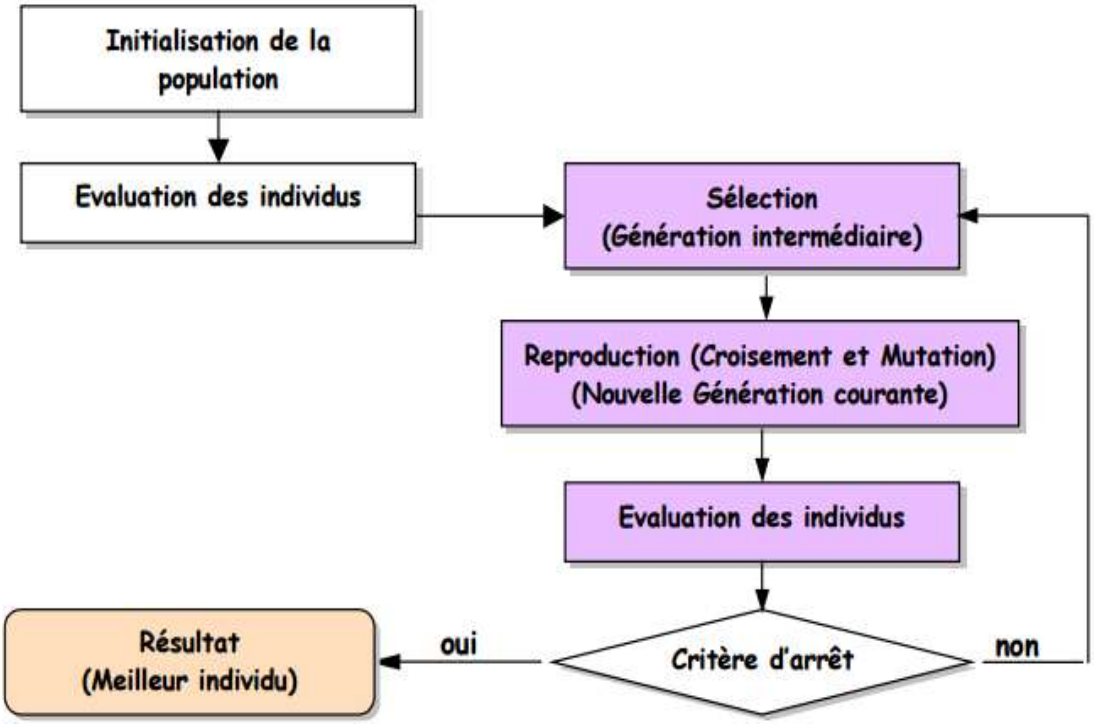


Figure2.14 : L'organigramme d'un algorithme génétique [56].

8. Les algorithmes évolutionnaires :

Plusieurs travaux ont été menés au cours des dernières années pour l'application des méthodes évolutives aux problèmes d'optimisation multi-objectif. Ces algorithmes évolutionnaires ont la capacité à trouver plusieurs solutions optimales de Pareto en une seule exécution et peuvent aider à maintenir la diversité de la population en utilisant des opérateurs génétiques. Cependant, nous distinguons deux grandes catégories d'algorithmes évolutionnaires multi-objectifs: les algorithmes élitistes (Non-dominated Sorting Genetic Algorithm-II (NSGA-II), Strength Pareto Evolutionary Algorithm (SPEA), Multi-Objective Messy GA (MOMGA), etc.) qui utilisent le principe d'archivage et les algorithmes non-élitistes (Vector Evaluated Genetic Algorithm (VEGA), Multi-Objective Genetic Algorithm (MOGA), NSGA, Niched-Pareto Genetic Algorithm (NPGA)) qui n'utilisent pas ce principe. Ils peuvent être classés en fonction de leur génération sur lesquels ils se sont développés [52].

L'élitisme consiste à maintenir une population externe, qui permet d'archiver le meilleur ensemble des points non dominés découverts jusqu'ici. Cet ensemble est mis à jour continuellement pendant la recherche, et participe avec une certaine probabilité à l'étape de sélection. Cette méthode permet ainsi une intensification de la recherche. Actuellement, les algorithmes élitistes obtiennent de meilleurs résultats sur un grand nombre de problèmes multi-objectifs.

8.1. Les algorithmes non-élitistes :

Les approches que nous allons voir ne conservent pas les individus Pareto optimaux trouvés au cours du temps. Elles maintiennent difficilement la diversité sur la frontière Pareto. La convergence des solutions vers la frontière de Pareto est lente.

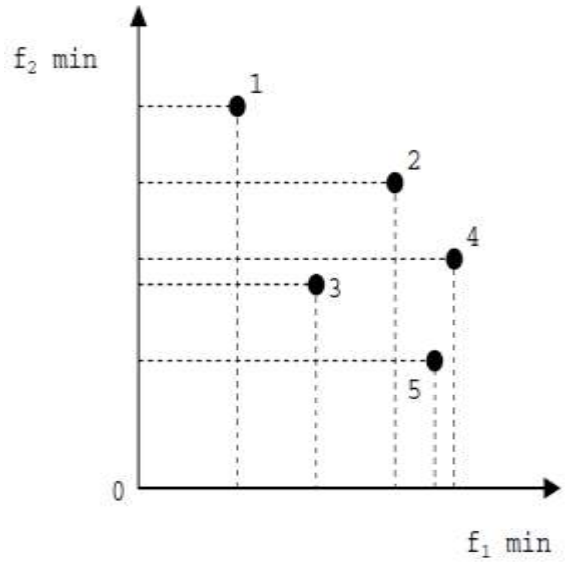
8.1.1. Multiple Objective Genetic Algorithm (MOGA):

En 1993 Fonseca et Fleming ont proposé une méthode dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent [Fonseca and Fleming 1993]. Ensuite, ils utilisent une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant même rang [54].

Soit un individu x_i à la génération t, dominé par pi(t) individus. Le rang de cet individu est :

$$rang(x_i, t) = 1 + p_i(t)$$

Tous les individus non dominés sont de rang 1. Dans l'exemple de la figure 2.15, le tableau de dominance est le suivant :



Point	Dominé par	Rang
1	aucun	1
2	3	2
3	aucun	1
4	3 et 5	3
5	aucun	1

Figure 2.15 : Exemple de tableau de dominance [54].

L'évaluation de la fitness de chaque individu se fait de la façon suivante [53]:

1) Tri de la population selon le rang.
2) Attribuer la fitness des individus par interpolation du meilleur (rang 1) au pire (rang $n \leq N$) dans le sens proposé par David E. Goldberg (Goldberg 1989) selon une certaine fonction, généralement linéaire.
3) Faire la moyenne des fitnesses des individus ayant le même rang, de sorte que chacun d'entre eux soit échantillonné à la même vitesse. Ceci permet de maintenir la fitness global de la population.

Cette méthode obtient des solutions de bonne qualité et son implémentation est facile. Mais les performances sont dépendantes de la valeur du paramètre σ_{share} utilisé

dans le sharing. Dans son article Fonesca explique comment choisir au plus juste la valeur σ_{share}.

La technique de sharing, appelée heuristique de partage, a été proposée par Goldberg [Goldberg 1989]. Le but de cette technique est d'éviter le regroupement des individus autour d'un optimum local lors de la recherche de l'optimum global dans un problème multimodal [54].

8.1.2. Non dominate Sorting Genetic Algorithm (NSGA):

Dans la méthode NSGA proposée par [Srinivas, 1993], le calcul de la de la fitness s'effectue en séparant la population en plusieurs groupes en fonction du degré de domination au sens de Pareto de chaque individu [40].

- ✓ **Pseudo code de NSGA [53]**

```
Génération de la population initiale
repeat
    while Population n'est pas classée do
        Recherche des individus non dominés
        Calcul fitness factice (artificielle)
        Sharing
    end while
  Sélection
  Croisement
  mutation
until Critère d'arrêt réalisé
```

✓ **Principe de NSGA :**

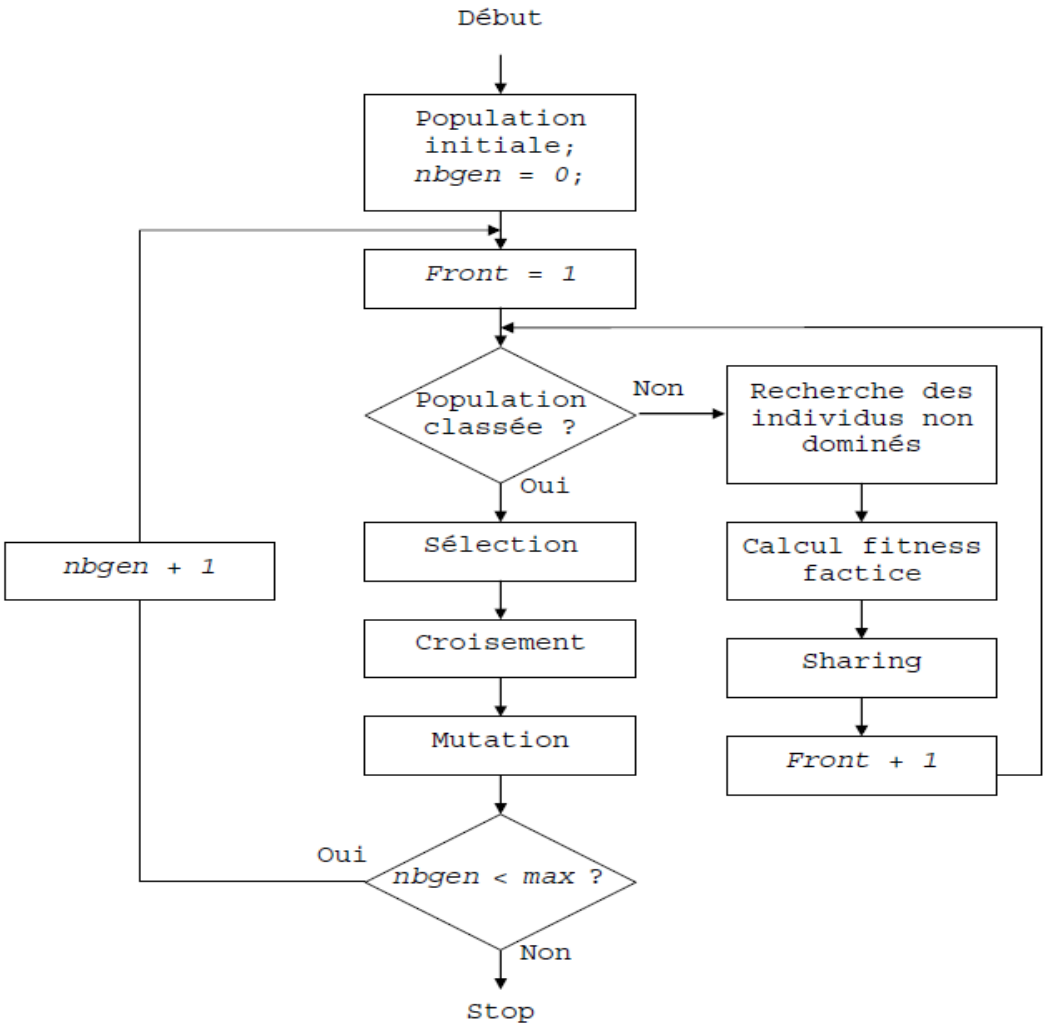


Figure 2.16 : Fonctionnement de l'algorithme NSGA [54].

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 [40].

8.1.3. Niched Pareto Genetic Algorithm (NPGA):

Cette méthode proposée par Horn et Napfliotis utilise un tournoi basé sur la notion de dominance de Pareto [Horn and Napfliotis 1993]. Elle compare deux individus pris au hasard avec une sous-population de taille tdom également choisie au hasard. 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 [54].

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 Napfliotis établissent le constat suivant [40] :

- ✓ Si t_{dom} ≈1% de N (nombre d'individus de la population), il y a trop de solutions dominées.
- ✓ Si t_{dom} ≈ 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.

Outre les paramètres de sharing l'utilisation de cette méthode ajoute un paramètre supplémentaire à fixer par l'utilisateur, t_{dom}. Ce dernier permet de régler aisément la pression sur la population. Les solutions trouvées sont de bonne qualité et cette approche est plus rapide que les approches précédentes car le sharing n'est appliqué que sur une portion de la population.

8.2. Les algorithmes élitistes :

Dans le domaine de l'optimisation multi-objectif, la sélection élitiste consiste à maintenir une seconde population appelée archive, contenant les solutions non dominées trouvées au cours des différentes générations de l'algorithme évolutionniste. Les individus de cette population participe avec une certaine probabilité à l'étape de sélection et donc à la reproduction de nouveaux individus. En outre des techniques de regroupements sont employées pour limiter la taille de cette archive [40].

8.2.1. Strength Pareto Evolutionary Algorithm (SPEA):

En 1998 Zitzler et Thiele proposent la méthode élitiste SPEA [Zitzler, 1998] basée sur le concept Pareto. Pour réaliser cet élitisme, SPEA maintient une archive externe contenant le meilleur front de compromis rencontré durant la recherche. Toutes les solutions de l'archive participe à la sélection. Une méthode de clustering est utilisée pour réduire l'ensemble de Pareto sans supprimer ses caractéristiques [40].

- ✓ **Algorithme SPEA [53]**

Initialiser la population $P0$
Créer l'archive externe vide $\bar{P} = \emptyset$
repeat
    Calcul de la valeur d'adaptation pour tous les individus de $P \cup \bar{P}$
    Selection dans $P_t \cup \bar{P}$ en fonction de la valeur d'adaptation
    Opération de croisement
    Opération de mutation
    Mise à jour de $\bar{P}$ à partir des individus non dominés de P
until critère d'arrêt réalisé

- ✓ **Fonctionnement général :**

La première étape consiste à créer une population initiale (constituée par exemple d'individus générés 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. A 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 phase de clustering est appliquée dans le but de garder les meilleurs représentants [40].

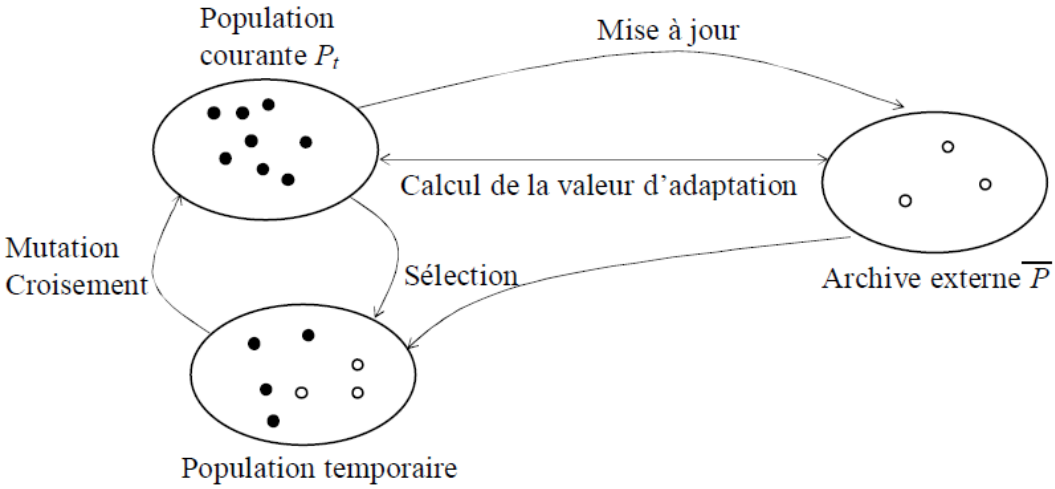


Figure 2.17 : Fonctionnement général de l'algorithme SPEA.

8.2.2. Pareto Archived Evolution Strategy (PAES):

La méthode PAES, conçue et mise en oeuvre par D. Knowles et Joshua David W.Corne (Knowles et Corne 2000) est une approche assez simple. Son principe n'est pas basé sur une population car elle n'utilise qu'un seul individu à la fois pour la recherche des solutions. Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales [53].

L'algorithme de PAES se présente en trois parties [54] :

1) Génération d'une solution candidate.
2) Fonction d'acceptation de la solution candidate.
3) Archivage des solutions non dominées.

Cette méthode est relativement simple à mettre en oeuvre. 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 [40].

8.2.3. Pareto Envelope based Selection Algorithm (PESA):

La méthode PESA a été également proposée par Knowles et Corne [Knowles and Corne 2000]. Elle définit deux paramètres concernant la taille des populations d'individus : P_I (taille de la population interne) et P_E (taille de la population externe ou archive) [54].

✓ **Algorithme PESA :**

1) Génération aléatoire et évaluation de la population P_I, et initialisation de P_E ($P_E = \emptyset$).
2) Transfert de tous les individus non dominés de P_I dans P_E.
3) Si le critère d'arrêt est réalisé alors
   On retourne P_E comme ensemble de solutions.
   Sinon
   On supprime tous les individus de P_I et on recrée P_I de la façon suivante :
   On sélectionne deux parents dans P_E avec la probabilité p_c, on produit un enfant par croisement puis on le mute. Avec la probabilité $(1-p_c)$, on sélectionne un parent et on le mute pour produire un autre enfant.
4) On recommence au 2)

Une solution courante de P_I peut entrer dans l'archive P_E si elle est non dominée dans P_I et si elle est non dominée dans P_E. Une fois, la solution insérée dans l'archive, on supprime tous les membres de l'archive qu'elle domine [54].

✓ **Le fonctionnement :**

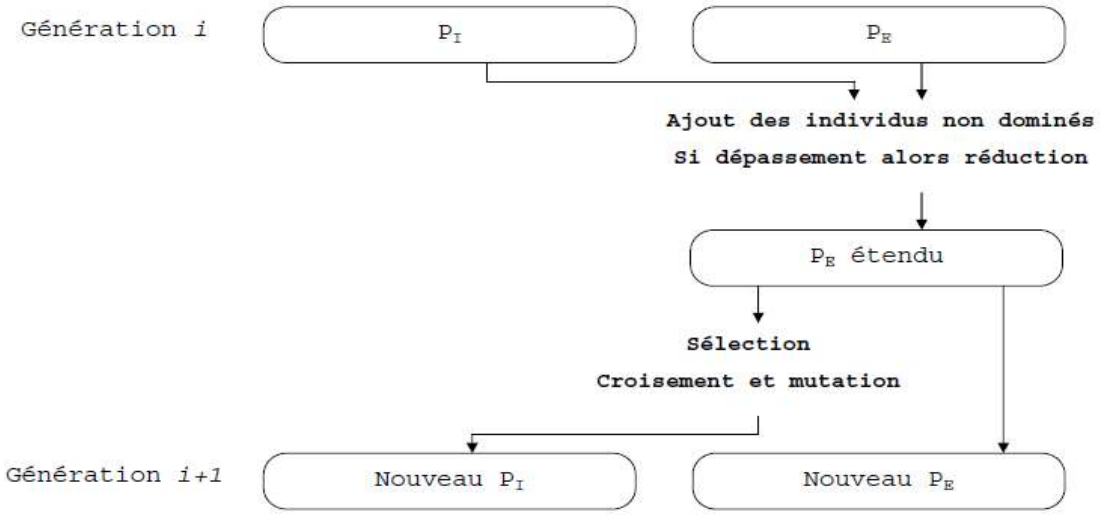


Figure 2.18 : Schéma de fonctionnement de PESA [54].

8.2.4. Non dominate Sorting Genetic Algorithm II (NSGA-II):

L'algorithme NSGA-II proposé par Deb en 2002, apparait comme l'un des algorithmes les plus efficaces pour trouver l'ensemble optimal de pareto avec une excellente variété des solutions. Il est basé sur les trois caractéristiques suivantes : il utilise le principe de l'élitisme, il favorise les solutions non dominées, et il utilise une variété explicite des solutions.

Le NSGA-II commence par une génération aléatoire d'une population initiale P_0 de N individus parents. A la génération t, une population Q_t de N enfants est créée à

partir de la population parent P_t en utilisant les opérateurs génétiques (sélection, croisement, mutation).

Ensuite, les deux populations sont combinées pour former une nouvelle population R_t de taille 2N. La recherche des solutions non dominées permet de classer les individus de R_t en plusieurs fronts de rangs différents. Elle est effectuée de la manière suivante : Chaque individu de R_t est comparé à tous les autres individus par le concept de dominance. Les individus non dominée appartiennent au front de rang 1, le front de pareto. En éliminant, temporairement, ces individus de l'ensemble de recherche, l'algorithme est itéré pour fournir le front de rang 2, et ainsi de suite. La nouvelle population parent P_{t+1} est alors construite avec les N individus appartenant aux fronts de rang les plus faibles. Pour le dernier front, il y a plus de solutions que de places restantes dans la nouvelle population P_{t+1}. Les individus sont alors triés selon leur distance de crowding.

✓ **Algorithme NSGA-II [40]**

1) Initialiser les populations P_0 et Q_0 de taille N
2) Tant que critère d'arrêt non rencontré faire
   Création de $R_t = P_t \cup Q_t$
   Calcul des différents fronts F_i de la population R_t par un algorithme de ranking
   Mettre $P_{t+1} = \emptyset$ et $i = 0$
   Tant que $|P_{t+1}| + |F_i| < N$ faire
   $P_{t+1} = P_t \cup F_i$
   $i = i + 1$
   FinTantQue
   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
FinTantQue

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 d_i d'un point particulier 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 2.19 est représenté l'hypercube en deux dimensions associé au point i. Un algorithme de calcul de la distance de crowding est détaillé dans [Deb, 2001]. 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 *di* calculés, il ne reste plus qu'à les trier par ordre décroissant et à sélectionner les individus possédant la plus grande valeur de crowding [40].

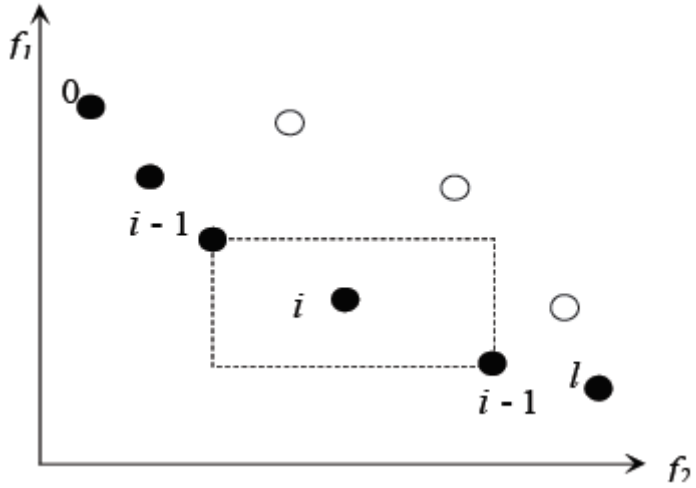


Figure 2.19: Calcul du distance de crowding [40].

✓ **Le fonctionnement :**

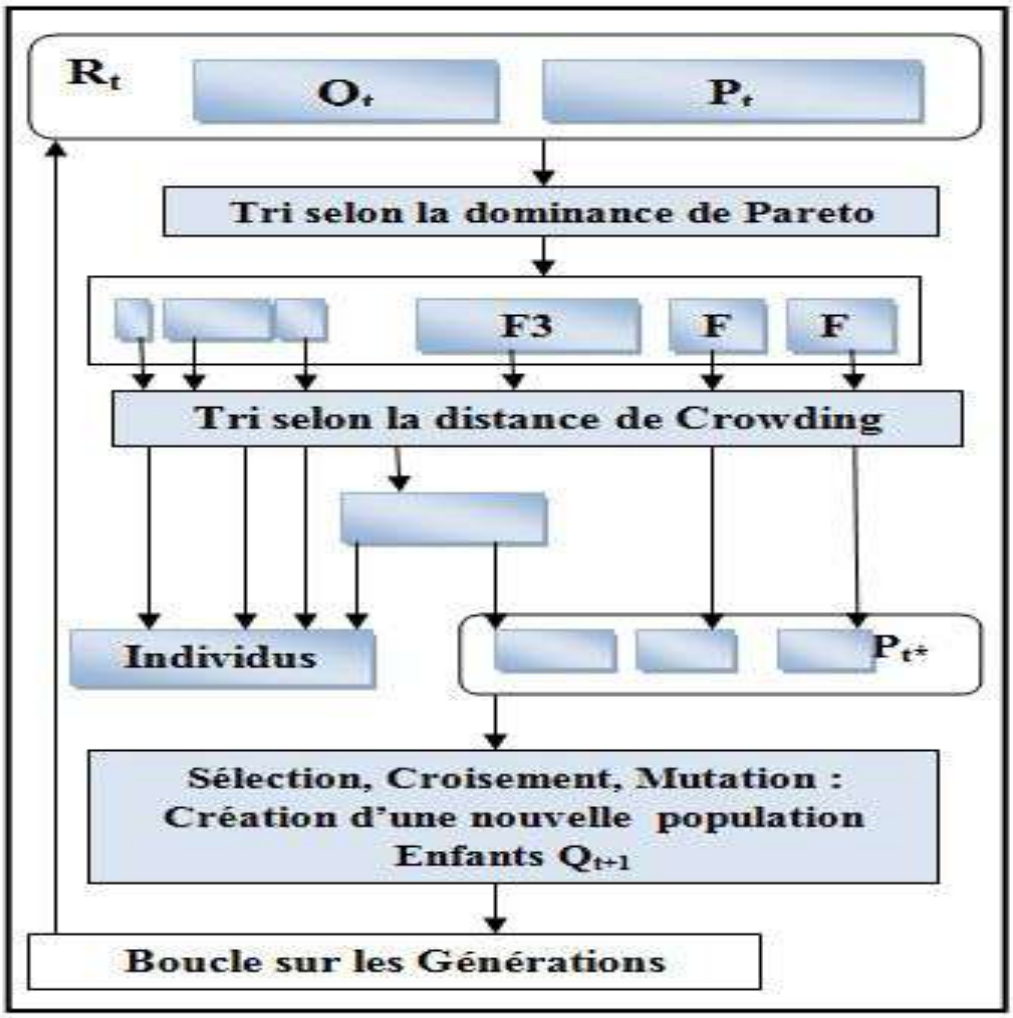


Figure. 2.20: Schéma de fonctionnement de NSGA-II.

Cette nouvelle version de NSGA a permis de réduire la complexité de l'algorithme à O(k,N2), de créer une méthode plus élitiste et de supprimer les paramètres de sharing [40].

9. Conclusion :

Dans ce chapitre, nous avons présenté un ensemble de définitions permettant de mieux appréhender les concepts et les méthodes présents dans ce manuscrit et notamment ceux des méthodes d'optimisation multi-objectif. Celles-ci permettent de résoudre certains types de problèmes et d'obtenir la solution optimale en un temps raisonnable.

Dans le chapitre suivant, nous allons procéder à notre contribution dans le contexte de la chemoinformatique notamment, l'approche de conception De novo de médicaments, dans laquelle nous allons appliquer une de ces méthodes d'optimisation pour concevoir des nouvelles molécules thérapeutiques.

CHAPITRE 03 :

L'optimisation en chemoinformatique

Au sommaire de ce chapitre :

1. Introduction.
2. L'optimisation dans le domaine de la chemoinformatique.
3. L'optimisation multi-objectif dans le modèle QSAR.
4. L'optimisation multi-objectif dans le docking.
5. L'optimisation multi-objectif dans le DND.
6. L'optimisation multi-objectif dans la conception de la librairie.
7. Les différents outils en chemoinformatique.
8. Principales fonctionnalités des outils de la chemoinformatique.
9. Workflows de la chemoinformatique.
10. Les bases de donné de la chemoinformatique.
11. Conclusion.

1. Introduction :

La découverte de médicaments est un problème multi-objectif difficile où de nombreux objectifs pharmaceutiquement importants doivent être satisfaits de manière adéquate pour trouver une solution.

Le problème se caractérise par des espaces de solutions vastes et complexes, encore plus perplexes par la présence d'objectifs contradictoires. Des méthodes d'optimisation multi-objectifs, conçues spécifiquement pour résoudre ces problèmes, ont été introduites dans le domaine de la découverte de médicaments il y a plus d'une décennie et ont régulièrement gagné en acceptation [67].

Ce chapitre passe en revue les dernières méthodes et applications multi-objectifs rapportées dans la littérature, en particulier la modélisation quantitative de la structure-activité, le docking, la conception de novo et la conception de la bibliothèque. En outre, nous abordons les récentes ressources de la chemoinformatique qui sont disponibles et qui permettent la manipulation des structures chimiques ; pour cela, nous présentons les différents outils en chemoinformatique(CDK, RDKit, Open Babel, etc.), ensuite nous abordons tous les environnements de workflow en chemoinformatique à savoir KNIME, Taverna ; enfin, nous résumons dans un tableau toutes les bases de données récentes en chemoinformatique.

2. L'optimisation dans le domaine de la chemoinformatique :

L'amélioration du profil d'une molécule pour le processus de découverte de médicaments nécessite l'optimisation simultanée de nombreux objectifs souvent concurrents.

Traditionnellement, les méthodes chemoinformatiques standard ignoraient ce problème et se concentrent sur l'optimisation séquentielle de chaque propriété biologique ou chimique (c'est-à-dire un seul objectif). Cette approche, connue sous le nom d'optimisation d'objet unique (SOOP), s'efforce de découvrir une seule solution optimale au problème d'optimisation [65].

Implicitement, les méthodes SOOP supposent que la solution optimale pour un objectif sera également l'optimum pour tous les autres objectifs impliqués dans le profilage d'une molécule.

Cependant, lorsque ces autres objectifs sont contradictoires, comme c'est souvent le cas dans la découverte de médicaments, l'optimum individuel correspondant aux nombreux objectifs peut varier considérablement. Les méthodes d'optimisations multi-objectifs (MOOP) introduisent une nouvelle approche pour obtenir l'optimalité sur la base des compromis entre les différents objectifs.

MOOP cherche à découvrir un ensemble de compromis satisfaisants qui peuvent à leur tour être utilisés pour découvrir la solution globale optimale en optimisant simultanément de nombreuses propriétés dépendantes. Les méthodes MOOP ont été récemment introduites dans le domaine de la chemoinformatique [65]. Ces méthodes

ont été adoptées dès le début en raison des exigences de la découverte de médicaments.

Les médicaments sont essentiellement des molécules avec un profil pharmacologique qui compromet de nombreux objectifs pertinents, y compris la puissance, la sélectivité, la pharmacocinétique et la toxicité [66].

Conformément à l'évolution du champ, les premières applications ont utilisé des méthodes MOOP composites et, peu après, des méthodes basées sur Pareto utilisant des algorithmes développés pour d'autres domaines.

Depuis lors, plusieurs applications couvrant presque tous les domaines de la chimie informatique et chemoinformatique ont été rapportées qui présentent des méthodes plus avancées, souvent conçues sur mesure pour les exigences de processus de découverte de médicaments [67].

Un aperçu détaillé des applications récentes de la technologie MOOP dans ce domaine est présenté ci-dessous en mettant l'accent sur les méthodes basées sur Pareto. Le tableau 3.1 résume les applications représentatives choisies :

Reference	MO type	Search method	Remarks/objectives
Quantitative structure–activity relationships			
Nicolotti et al. (2009)	Pareto-based	Evolutionary (genetic) algorithm	Selection of docking poses for 3D QSAR model construction
MO-QSPR: Manoharan et al. (2010)	Desirability, aggregative	Expert-driven search, desirability plot	QSAR and QSSR models
Soto et al. (2009)	Pareto-based and aggregative	Evolutionary (genetic) algorithm	Feature selection for QSAR model construction
Hii et al. (2011)	Pareto-based	Evolutionary (genetic) programming	*In silico* toxicity prediction; 'goodness-of-fit', model complexity
MTAR: Bajorath et al. (2012)	Visualization	Interactive, expert-driven visual inspection	Multiple target activity
DAD/TAD: Medina-Franco et al. (2011)	Visualization	Interactive, expert-driven visual inspection	Two or three target activities
Docking			
Mardikian et al. (2007)	Pareto-based	Evolutionary (genetic) algorithm	Protein-ligand bonding; van der Waals interactions
Boisson et al. (2008)	Pareto-based	Evolutionary (genetic) algorithm	Energy, surface
MOSFOM: Li et al. (2009)	Pareto-based	Evolutionary (genetic) algorithm	Energy, contact
De novo design and inverse QSAR			
MEGA: Nicolaou et al. (2009)	Pareto-based	Evolutionary (genetic) algorithm	Ligand and target based objectives
PLD: Ekins et al. (2010)	Pareto-based	Evolutionary (genetic) algorithm	Predicted activity, ADME related properties
Molecule Commander: van der Horst et al. (2012)	Desirability and Pareto-based	Evolutionary (genetic) algorithms	High predicted affinity, pharmacophore model, predictive models
MOLig: Sengupta et al. (2012)	Pareto-based	Simulated annealing	Internal energy, target interaction, similarity to scaffold, oral bioavailability
Library design			
Sharma et al. (2011)	Weighted-sum, aggregative	Deterministic annealing	Diversity, representativeness of compounds
MEGALib: Nicolaou et al. (2011)	Pareto-based	Evolutionary (genetic) algorithm	Selectivity between related targets

Venhorst et al. (2010)	Pareto-based	Evolutionary (genetic) algorithm	Structural diversity, chemical tractability, etc.
MSDS: Meinl et al. (2011)	Composite, aggregative	Iterative, greedy search	Activity, representativeness of compounds
CNS MPO: Wager et al. (2010)	Desirability, aggregative	Multi-parameter optimization	Fundamental physicochemical parameters (6)
MO-PSO: Namasivayam et al. (2012)	Pareto-based	Particle Swarm Optimization	SAR discontinuity index; ADME-related properties

Tableau 3.1 : Quelques méthodes multi-objectif dans la chemoinformatique [67].

3. L'optimisation multi-objectif dans le modèle QSAR :

Depuis la première proposition dans les années 1960, les modèles Quantitatifs Structure-Relation d'Activité (QSAR) ont gagné l'acceptation généralisée par la communauté de découverte de médicament. Les modèles QSAR associent des descripteurs moléculaires aux propriétés biologiques en utilisant des techniques statistiques et / ou des algorithmes d'intelligence computationnelle. Typiquement, les modèles QSAR ont été utilisés à des fins d'interprétation, c'est-à-dire pour identifier les relations structure-activité dans les données disponibles. Une deuxième utilisation a été comme modèles prédictifs, c'est-à-dire pour la prédiction de la propriété biologique de nouvelles structures chimiques non testées [92].

Les techniques MOOP ont été utilisées dans la modélisation QSAR au cours de la dernière décennie, on va citer quelques méthodes :

3.1. MoQSAR :

Nicolotti et al ont utilisé une variante d'une EA appelée programmation génétique multi-objectif qui utilisait le classement de Pareto pour optimiser les modèles QSAR [92]. Un certain nombre d'objectifs contradictoires, y compris la précision du modèle, le nombre de termes, la complexité interne et l'opportunité, c'est-à-dire l'interprétation des descripteurs utilisés dans le modèle, ont été pris en considération.

La technique, appelée multi-objectif QSAR (MoQSAR), représente les modèles QSAR en utilisant une structure dite de données arborescentes, où les feuilles représentent des valeurs constantes ou des variables de descripteurs moléculaires et les nœuds internes représentent des opérateurs mathématiques tels que plus, moins ou multiplication. Dans chaque itération de l'algorithme, les modèles sont développés, les représentations des arbres sont traduites en équations QSAR standard et la performance de ces équations est calculée pour chaque objectif. Les modèles sont classés par Pareto et un score de condition physique est déterminé. Le résultat est une collection de modèles représentant des compromis différents des objectifs.

Nicolotti et al ont indiqué que l'approche MoQSAR fournissait une famille de solutions représentant divers compromis des objectifs, permettant à l'utilisateur de sélectionner l'équation la plus adaptée à une application spécifique. De plus, les fonctions objectives ne nécessitaient pas d'étalonnage supplémentaire pour chaque nouveau jeu de données avec cette approche de classement de Pareto.

3.2. PUMP-RP :

Stockfisch a proposé une technique multi-objectif non évolutive appelée la méthode de partition partielle récursive multi-propriété (PUMP-RP) partiellement unifiée pour générer des modèles QSAR [95]. La méthode PUMP-RP est une extension de la méthode de classification des arbres de décision qui peut répondre à plusieurs objectifs. Elle a été utilisée avec succès pour construire des modèles pour analyser les relations de sélectivité [96]. Cette approche ressemble à une analyse lexicographique et nécessite un ordre a priori des objectifs. Les objectifs sont optimisés séquentiellement jusqu'à ce qu'aucun progrès ne soit réalisé. Il est intéressant de noter que la commande des objectifs est cruciale pour la méthode parce que les différents ordres produisent des modèles différents. Cette dépendance à l'ordre des objectifs est un inconvénient potentiel de la méthode car la relation entre les objectifs n'est pas toujours claire et donc les objectifs de classement sont souvent subjectifs.

3.3. QSAR 3D :

Un algorithme génétique multi-objectifs pour la sélection des poses de docking appropriées pour la construction de modèles 3D QSAR est proposé. La méthode optimise deux objectifs, à savoir la corrélation des scores d'accostage à l'activité biologique et l'écart quadratique moyen de la raie moyenne par rapport à un inhibiteur co-cristallisé défini. L'ensemble des modèles QSAR équivalents produits est ensuite regroupé et un ensemble représentatif est sélectionné. La méthode a été appliquée avec succès à l'analyse d'un ensemble de données bien connu que les auteurs avaient déjà examiné en utilisant d'autres approches. Les résultats montrent que la méthode a été très efficace dans l'ensemble et, en particulier, la génération de modèles QSAR 3D avec une bonne sensibilité [93].

3.4. MOQSPR :

Manoharan et al. Proposent la QSPR multi-objectifs (MOQSPR), qui en combinaison avec FB-QSAR, une méthodologie QSAR basée sur des fragments, est utilisée pour l'interprétation des données et la conception de nouveaux composés [94]. La méthode est basée sur la préparation de modèles QSAR, le calcul des fonctions de désirabilité individuelles et leur combinaison via la moyenne géométrique. Ensuite, un diagramme de désirabilité est utilisé pour identifier les gammes de valeurs optimales des descripteurs qui ont une influence majeure sur les réponses prédites et, ainsi, appuient l'interprétation des données biologiques disponibles. Les auteurs ont également utilisé les résultats de MOQSPR pour sélectionner des fragments spécifiques avec des valeurs de descripteurs optimales et suggèrent des modèles analogiques prometteurs qui compromettent l'activité et la sélectivité.

4. L'optimisation multi-objectif dans le docking :

Le problème d'amarrage moléculaire consiste à trouver la meilleure position et orientation d'un ligand (petite molécule) dans le récepteur (protéine). De nouveaux médicaments peuvent être conçus de manière plus efficace en s'appuyant sur la

simulation d'amarrage assistée par ordinateur qui consiste à rechercher des ligands qui se fixent avec une très bonne affinité au récepteur [68].

Le succès d'un programme d'amarrage dépend de la méthode de recherche et la fonction de scoring employées pour trouver les conformations géométriques d'énergie minimale du complexe protéine-ligand.

En effet, l'application des méthodes de résolution des problèmes d'optimisation multi-objectif (Multi-Objective Optimization Problem, MOOP) permet de trouver rapidement l'ensemble d'optimum global en utilisant des fonctions de scoring multi-objectifs. La plupart des fonctions les plus couramment utilisées pour l'évaluation de l'énergie d'interaction sont établies en fonction de scores empiriques correspondant à une combinaison de paramètres pondérés.

Plusieurs approches qui se basent sur les fonctions de scoring multi-objectifs, ont été utilisées comme méthodes de recherche pour le problème de docking entre autres:

4.1. ClustMPSO :

Janson et al. Ont décrit une application d'optimisation de l'amarrage appelée ClustMPSO (basée sur l'algorithme d'optimisation des essaims de particules [PSO]) qui minimise deux objectifs: l'énergie intermoléculaire entre la protéine et le ligand et l'énergie intramoléculaire du ligand [77, 78].

Dans l'application ClustMPSO, les particules sont groupées à chaque itération et divisées en plusieurs essaims sur la surface de compromis des objectifs. L'objectif de cette approche est de trouver plusieurs configurations d'amarrage intéressantes et quasiment optimales en plus de la meilleure configuration d'amarrage. Janson et al ont rapporté que cette application a été testée sur deux expériences d'amarrage en utilisant des données réelles et a été trouvé à surpasser les techniques standard.

4.2. GOLD (Genetic Optimization for Ligand Docking):

La méthode GOLD est un programme de docking automatique qui utilise les algorithmes évolutionnaires pour explorer l'espace conformationnel du complexe ligand-récepteur. Ce programme consiste à générer plusieurs conformations (une population d'individus), chacune correspondant à un ensemble de variables (par exemple angles dièdres, donneurs de liaisons hydrogène, accepteurs de liaisons hydrogène), c.-à-d., chaque type de conformation est codé dans un chromosome. Ces conformations sont évaluées par une fonction de score et les meilleures sont conservées. Leurs variables sont alors modifiées aléatoirement : soit par mutation, soit par croisement, formant une nouvelle génération de conformations [69].

4.3. NSGA (Non-dominated sorting genetic algorithm):

Il a été développé par Deb(2000) [70] et appliqué plus tard à des problèmes de la découverte de médicaments, y compris le docking de ligand. S.Mardikiana démontré dans sa thèse de doctorat (Université de Sheffield, 2007) que l'application des interactions et la liaison électrostatique et de hydrogène permet de combiner la protéine avec le ligand dans un système NSGA bi-objective.

Les études menées par Boisson et al. [71] utilisent aussi une fonction de score bi-objective dans un algorithme génétique pour le docking de ligand, mais cette fois, avec des termes d'énergie et de surface. Ce travail de recherche applique l'algorithme NSGA-II comme précédemment, mais considère aussi l'algorithme IBEA (Indicator-Based Evolutionary Algorithm) [72] qui a donné de meilleures performance que l'algorithme NSGA-II dans cette application.

4.4. MOSFOM (Multi-Objective Scoring Function Optimization Methodology):

Plus récemment, l'algorithme MOSFOM (Multi-Objective Scoring Function Optimization Methodology) a été développé par Li et al [73]. Cet algorithme de docking multi-objectif consiste à combiner deux fonctions de scores: le Contact et l'énergie du programme DOCK [74]. La fonction Contact est la somme des contacts des atomes lourds entre le ligand et le récepteur. La fonction énergie basée sur le champ de force AMBER [75] est composée de deux termes électrostatique et stérique.

$$\text{Energie} = E_{\text{électrostatistique}}(d) + E_{\text{stérique}}(d)$$

Où, E est l'énergie qui dépend de la distance (d) entre deux points susceptibles d'interagir entre elles, le terme électrostatique est de type potentiel de Coulomb et le terme stérique est de type Lennard-Jones [76].

La méthode MOSFOM utilise le Pareto-ranking et donne des résultats plus performants que ceux obtenus en utilisant une fonction de scoring simple.

5. L'optimisation multi-objectif dans le DND :

La conception de novo est un effort pour concevoir des composés chimiques en utilisant la connaissance sur le site cible pharmaceutique ou les ligands disponibles [65]. Afin de concevoir des produits qui sont réellement intéressants pour le chercheur en découverte de médicaments, plusieurs autres objectifs pharmaceutique importants doivent être pris en compte, y compris la sélectivité de la cible spécifique, l'accessibilité synthétique, et certains propriétés pharmacocinétiques connues sous le nom ADME (Absorption, Distribution, métabolisme, excrétion) , etc.

Traditionnellement, la plupart des méthodes de conception De novo se concentrent particulièrement sur la conception de molécules qui satisfont un seul objectif (mono-objectif), soit la prédiction de l'affinité de liaison à une protéine connue ou, la similitude à un ligand et ignorent la nature multi-objectif de la découverte de médicament nécessaire à la prédiction des propriétés pour produire un composé "drug-like"[79].

Récemment, suite à l'introduction de la technologie MOOP [80] de nombreuses nouvelles méthodes du DND sont apparues dans la littérature qui tentent de concevoir des molécules satisfaisant des objectifs multiples prédéfinis et produire ainsi des solutions de candidates avec une plus grande chance de construire un composé "drug-like" le plus efficace.

Parmi ces méthodes qui reconnaissent l'existence de plusieurs objectifs distincts, il y a :

5.1. MEGA (Multi-objective Evolutionary Graph Algorithm):

Nicolaou et al. décrivent MEGA (Multi-objective Evolutionary Graph Algorithm), une méthode qui combine les algorithmes évolutionnaires avec les techniques de recherche locale afin de permettre l'utilisation des connaissances de problèmes spécifiques lors de la recherche et d'amélioration des performances et d'évolutivité [81]. Ils proposent un cadre algorithmique général pour la conception de structures chimiques (graphes moléculaires) qui répondent à plusieurs exigences d'importance pharmaceutique.

L'algorithme applique tous les objectifs disponibles sur un premier ensemble de composés pour obtenir une liste des points (scores) pour chaque individu. La liste des points est ensuite utilisée pour l'élimination des solutions avec des valeurs en dehors de la plage autorisée par les filtres définis par l'utilisateur. Le classement individuel et la diversité de la population bénéficient d'une considération spéciale à travers l'application d'un processus de classification opérant sur les structures chimiques. Les solutions parents sont échantillonnées à partir de chaque cluster en utilisant un classement individuel (individual ranking) assurant ainsi la sélection de diverses molécules. De nouvelles solutions sont générées en utilisant la mutation et le croisement spécifique de graphe.

MEGA a également introduit l'utilisation de l'archive Pareto pour préserver toutes les solutions prometteuses trouvés tout au long de l'évolution et assure que les résultats contiendra des meilleures solutions trouvées. Les principales composantes de MEGA sont décrites en détail dans la figure 3.1 ci-dessous. On note que l'archive Pareto va stocker une population d'élite des solutions à chaque génération.

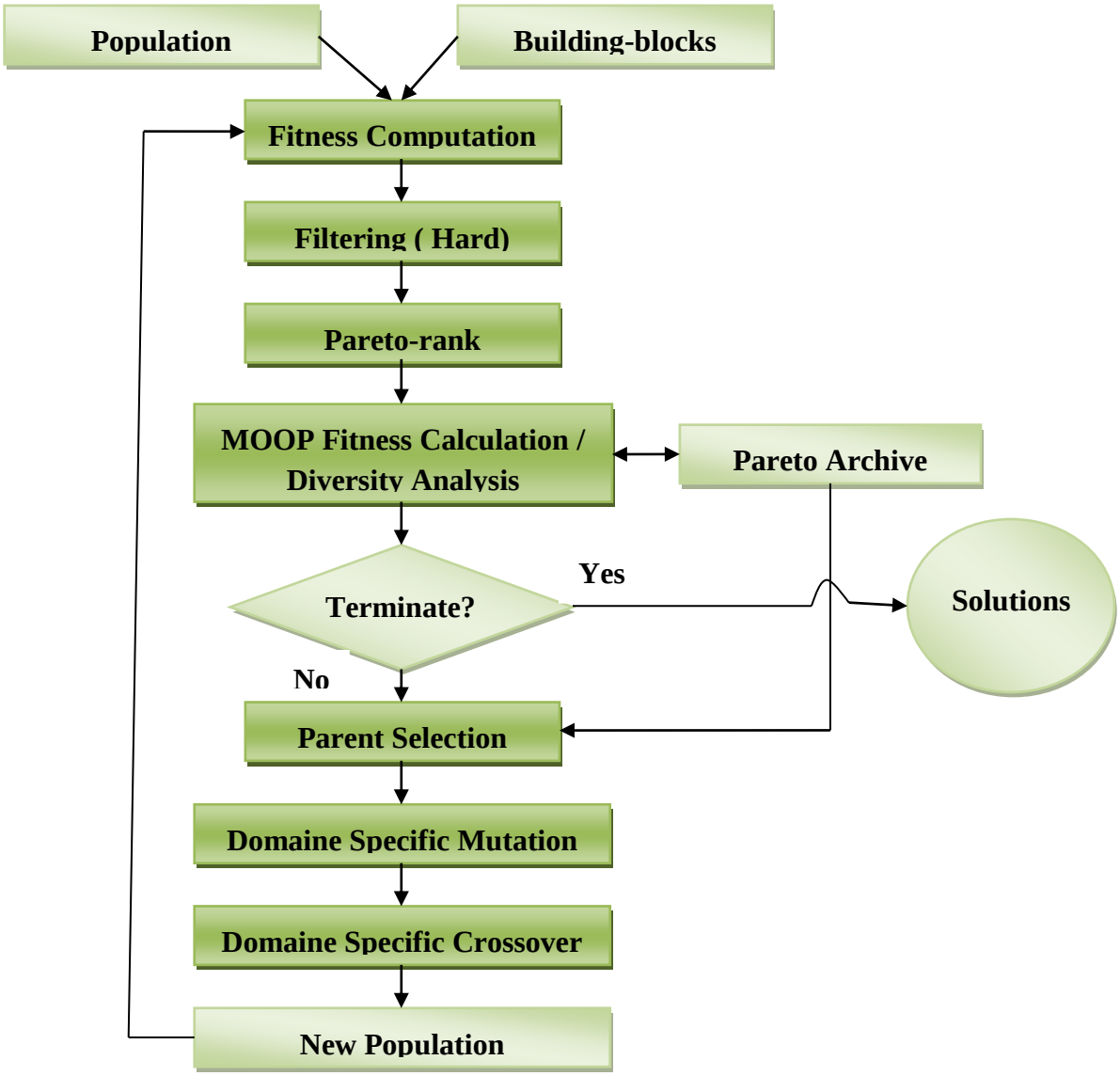


Figure 3.1: Schémas générale de l'algorithme MEGA [81].

Les résultats obtenus par Nicolaou et al. indiquent que l'utilisation du nichage et de l'élitisme, contribuent à l'amélioration des performances du système DND proposé. En outre, ce mécanisme de nichage est spécialement conçu pour préserver la diversité de chémotype de la population des solutions, et l'élitisme pour éviter la perte de solutions prometteuses. La diversité de structure chimique a également été abordée dans les travaux de Kruisselbrink et al. qui utilisent un opérateur basant sur les mesures de similarité de composés pour assurer l'évolution de diverses structures des niches de molécules [82].

5.2. PLD (Pareto Ligand Designer) :

Une approche similaire proposé par Ekins et al. est basée sur le développement de PLD (Pareto Ligand Designer) [83]. La méthode prend en entrée un ensemble de molécules de référence et calcule les valeurs des objectifs qui sont utilisés pour définir chaque rang de Pareto. Les solutions non-dominées sont identifiées et stockées dans l'archive de Pareto. Dans l'étape suivante, la population active est évoluée en utilisant

un ensemble extensif de transformations moléculaires et des molécules qui survivent à un nombre des propriétés qui forment une nouvelle population active. Le processus itère ensuite pour calculer les solutions aux divers rangs de Pareto et non-dominées et continue jusqu'à ce que des critères d'achèvement sont satisfaites. Les auteurs présentent les résultats de l'application du PLD pour des expériences d'optimisation de deux, trois et quatre objectifs tout en conservant l'activité biologique [83].

5.3. MOEA (Molecule Commander) :

Récemment, Van Der Horst et al. décrit une méthode MOEA qu'on appelle « Molecule Commander » et l'applique pour générer un potentiel Antagonistes du récepteur Adénosine A1 [84]. Cette méthode utilise plusieurs objectifs, y compris un modèle de Pharmacophore et un modèle de Machine à Support de Vecteurs (Support Vector Machine) basé sur des empreintes moléculaires pour les sous-types de récepteur Adénosine. Les objectifs ont été regroupés en groupes logiques au moyen de l'indice de désirabilité pour réduire la dimension du problème. Le rang de Pareto a été utilisé en combinaison avec le nichage pour sélectionner les composés qui seront utilisés comme entrées pour le prochain cycle de génération. Il est intéressant de souligner que le processus de nichage utilisé applique le regroupement (clustering) dans l'espace des paramètres pour assurer la sélection des diverses structures chimiques. Similaires aux méthodes décrites ci-dessus, les nouvelles structures sont développées en utilisant des techniques de perturbation moléculaire. Des seuils relatifs aux propriétés physico-chimiques ainsi que des règles chimiques ont été utilisés comme filtres pour éliminer les composés avec un profil indésirable.

5.4. MOLig :

Dans [85], une méthode MOLig utilisant le recuit simulé multi-objectif est présentée. MOLig représente des solutions utilisant une structure arborescente et emploie une collection d'opérateurs de perturbation comme la substitution, la suppression, et l'extension pour générer des nouvelles conceptions de produits chimiques. Dans l'exemple cité, MOLig a été utilisé pour optimiser simultanément l'énergie interne d'un candidat, son énergie d'interaction avec la cible donnée, sa similarité chimique pour un échafaudage de référence et sa biodisponibilité par voie orale. Les auteurs rapportent que la méthode était comparé pour plusieurs méthodes "De novo design" et a affiché de meilleures performances.

6. L'optimisation multi-objectif dans la conception de la librairie :

La méthode de la conception de la librairie (library design) se base généralement sur deux variantes: un ensemble de structures chimiques qui comporte des réactifs chimiques et des règles pour guider la synthèse virtuelle de produits chimiques (voir Figure 3.2). Cependant, l'ensemble initial peut être constitué de grandes librairies de composés réels ou virtuels. L'objectif est la diversité, qui permet d'obtenir une collection de produits représentatifs de l'espace chimique.

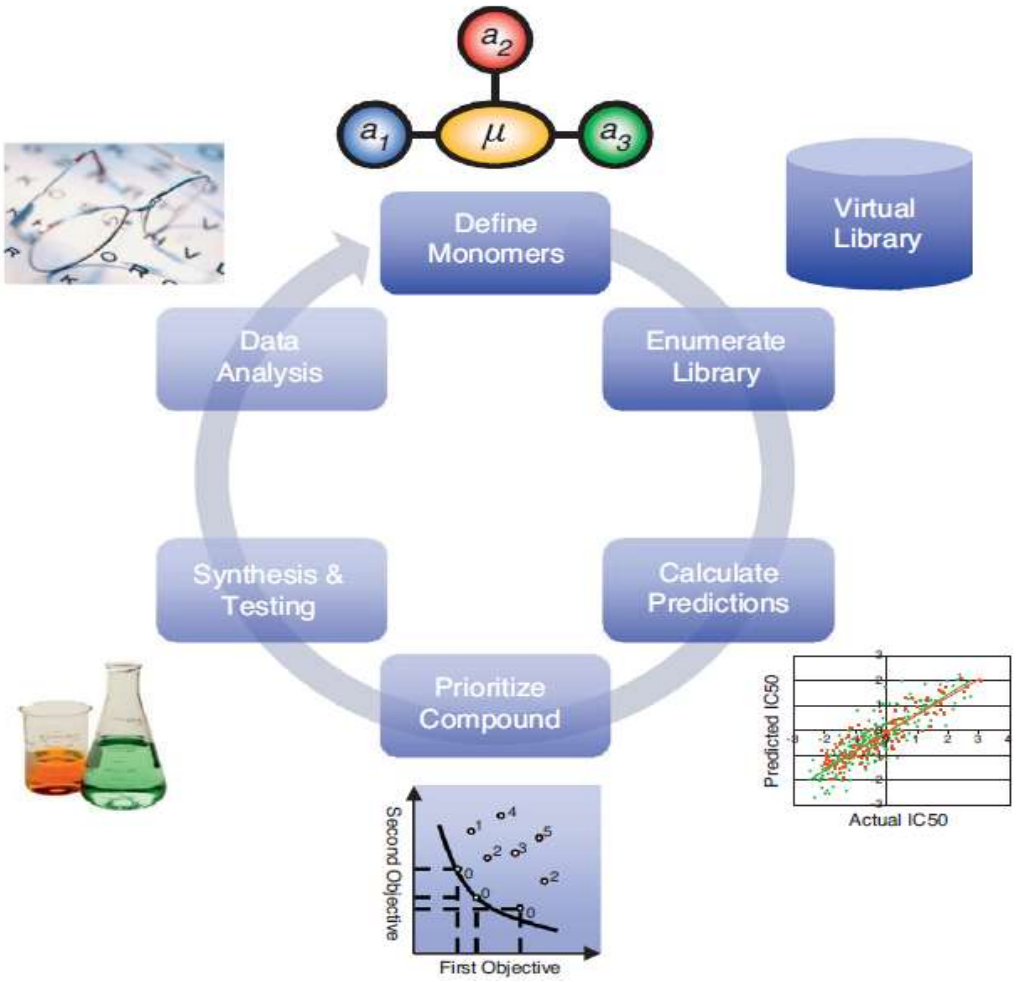


Figure 3.2: Workflow schématique de la conception de la librairie [67].

Un problème fréquent est la génération possible de trop grand nombre des produits virtuels qui répondent aux objectifs imposés. Les méthodes d'échantillonnage et les méthodes physico-chimiques peuvent être utilisées pour réduire de façon appropriée le nombre. Les méthodes multi-objectif sont souvent utilisées pour la conception de librairie pour équilibrer les nombreux d'objectifs avec la majorité des approches de la somme pondérée.

Pour créer des librairies satisfaisant de nombreux objectifs (la diversité, le drug-likeness, etc.), plusieurs expériences ont été réalisées en utilisant des algorithmes d'optimisation dans la conception de la librairie :

- ✓ Le programme SELECT [86] est la première application qui se base sur les techniques des algorithmes évolutionnaires pour construire une librairie générale comportant des composés ayant tous les propriétés "drug-like" et sont diversifiée. La diversité de la librairie a été déterminée en créant tout d'abord le fingerprint moléculaire de chaque molécule chimique (vecteur de bits) et en sommant ensuite les différences entre toutes les paires de leurs vecteurs.

- ✓ Une autre méthode basée Pareto a été introduite avec MoSELECT [87]. Cette méthode traite simultanément plusieurs objectifs tels que la diversité, les propriétés physico-chimiques et la facilité de synthèse.
- ✓ Plus récemment, Sharma et al ont proposé une méthode basée sur l'algorithme de recuit déterministe qui tient en compte simultanément les deux objectifs de la diversité et de la représentativité des composés dans la librairie obtenue [88].
- ✓ Nicolaou et al. ont donné une extension de l'algorithme de la conception De novo MEGA, appelé MEGALib, qui a été conçu pour le problème de conception de la librairie [89]. Cette méthode suit le même principe utilisé dans l'algorithme MEGA, sauf que le type de fragment et l'ensemble de règles chimiques pour la synthèse moléculaire sont différents.
- ✓ Meinl et al. proposent la méthode de Maximum-Score Diversity Selection (MSDS) qui utilise le concept de «l'érosion» pour calculer de manière itérative un score composite pour chaque molécule individuelle qui est, à son tour, utilisé pour déterminer la prochaine sélection de composés [90].
- ✓ Wager et al. présentent une nouvelle méthode d'optimisation multi-paramètre appelée "Central Nervous System MultiParameter Optimization" (CNS-MPO) qui utilise six paramètres physico-chimiques pour aider à la conception de librairie pour les recherches concernant le système nerveux central (Central Nervous System, CNS) [91]. Cette méthode construit un score de désirabilité pour chacune des six propriétés physico-chimiques (poids moléculaire, nombre de donneur de liaisons hydrogène, etc.) et un score composite en utilisant la somme pondérée.

7. Les différents outils en chemoinformatique :

La chemoinformatique propose de bons outils pour diminuer le coût et le temps de plusieurs étapes du processus de développement du médicament. Ces outils sont un ensemble de librairies comprenant des codes sources de divers algorithmes qui permettent aux chémoinformaticiens de développer leurs propres applications telles que, la recherche des similitudes structurelles, le criblage virtuel, QSAR, la manipulation de différents formats de fichiers, calcul des fingerprints moléculaires, etc. Cependant, le développement des outils chemoinformatique open-source a été initié depuis plus d'une décennie et plusieurs outils très intéressants sont aujourd'hui opérationnels entre autres : CDK [12], RDKit [97] et Open Babel [98]. Le CDK est écrit en java et trouve ses origines dans les universités tandis que OpenBabel et RDKit sont écrits en C++ développés par des entreprises respectivement OpenEye et Rational Discovery qui sont développés par la suite sous des licences Open Source. En général, tous ces outils partagent les mêmes fonctionnalités de base; bien que les détails d'implémentation et les modèles chimiques peuvent être différents. Toutefois, en raison de leur développement et histoire indépendants, chacun a ses propres fonctionnalités ainsi, chaque outil prend en charge plusieurs formats de fichiers et peut calculer différents fingerprints et descripteurs.

Le tableau 3.2 ci-dessous présente un résumé des différents outils connus en chemoinformatique:

Sr. No.	Name	Link	Programming language (wrapper, if any)	Operating system(s)	License
1	Chemistry Development Kit (CDK)	http://sourceforge.net/projects/cdk/	Java	Platform independent	LGPL
2	RDKit	http://www.rdkit.org/	C++ (Python, Java and C# wrapper)	Mac, Windows, and Linux	BSD
3	Open Babel	http://openbabel.org/	C++ (Java, .NET platform, Perl, Python, and Ruby wrapper)	Windows, Mac OS X, Linux	GPL
4	Cinfony	https://code.google.com/p/cinfony/ https://github.com/cinfony/cinfony	Python, Jython	Platform independent	BSD
5	Small Molecule Subgraph Detector (SMSD)	http://www.ebi.ac.uk/thornton-srv/software/SMSD/	Java	Platform independent	Creative Common (CC)
6	Biochemical Algorithms Library (BALL)	http://www.ball-project.org	C++	Windows, Linux, and Mac OS X	LPGL and GPL
7	Indigo	http://www.ggasoftware.com/opensource/indigo	C++ (Python, Java, and C# wrappers available)	Windows, Linux, and Mac OS X	GPL
8	jCompoundMapper	http://jcompoundmapper.sourceforge.net/	Java	Platform independent	LPGL
9	chemf	https://github.com/stefan-hoeck/chemf http://www.scala-lang.org/	Scala (runs on Java platform)	Platform independent	GPL
10	Cheminformatics in Python (ChemoPy)	https://code.google.com/p/pychem/downloads/list	Python	Linux and Windows	–
11	ChemmineR	http://www.bioconductor.org/packages/release/bioc/html/ChemmineR.html	Statistical programming environment "R"	Windows and Mac OS X	Artistic 2.0
12	Compound-Protein Interaction with R (Rcpi)	http://bioconductor.org/packages/release/bioc/html/Rcpi.html	Statistical programming environment "R"	Windows and Mac OS X	Artistic 2.0
13	Chemkit	http://wiki.chemkit.org/Main_Page	C++	Windows, Mac, and Linux	BSD

Tableau 3.2: Les outils de la chemoinformatique [99].

Le tableau présenté ci-dessus comporte les treize outils (kits) utilisés en chemoinformatique, nous présentons seulement les principales caractéristiques des trois premiers.

7.1. CDK :

Il s'agit d'une bibliothèque open source développée en Java pour la chemoinformatique et la bioinformatique [12]. Celle-ci peut être utilisée pour la génération de structure, recherche de structure (sous-structure ou structure exacte), édition et génération de molécules en 2D/3D, calcul de champs de force, préparation des empreintes moléculaire, calcul de descripteurs QSAR, génération des fragments, support de nombreux formats de fichiers (SMILES, formats MDL, etc.), détection de ligands similaires, détection des sites actifs des protéines, identification des métabolites, etc.

7.2. RDKit :

L'outil RDKit a été développé pour la recherche et la découverte rationnelle durant la période 2000- 2006, pour construire des modèles prédictifs pour des propriétés pharmacodynamiques (activités biologiques) et pharmacocinétiques (ADME/Tox) [97]. Ce kit offre diverses fonctionnalités telles que, les différents formats chimiques d'E/S (notamment SMILES/SMARTS, SDF, MOL2, PDB); recherche de sous-structure, de MCSS et de similitude; réactions chimiques; calcul de descripteurs moléculaires; etc.

7.3. Open Babel :

C'est un outil chimique open source offrant une librairie multiplateforme. Il est construit pour supporter l'inter-conversion entre les différents formats de fichiers, ainsi que d'autres fonctionnalités (représentation 2D, recherche de similarité, etc.) utilisées en chemoinformatique, la modélisation moléculaire et la chimie computationnelle [98].

8. Principales fonctionnalités des outils de la chemoinformatique :

Le but fondamental des logiciels de la chemoinformatique (souvent appelé «cheminformatics toolkit») est de fournir un langage spécifique au domaine adapté à la représentation de molécules [99]. En effet, ces kits de développement logiciels qui permettent de développer une application chimique dans un environnement de programmation spécifique offre l'ensemble de fonctions suivantes :

- ✓ Lire et enregistrer les structures dans divers formats de fichiers chimiques (SMILES, mol, etc.)
- ✓ Diviser la molécule en plusieurs fragments (fragmentation).
- ✓ Construction de molécules à partir de leurs fragments.
- ✓ Déterminer si une structure est une sous-structure d'une autre.
- ✓ Déterminer si deux molécules ont des structures semblables (recherche de la similarité).
- ✓ Identification de la sous-structure maximale commune dans un ensemble des structures (Maximal Common Sub-Structure, MCSS).
- ✓ Appliquer des réactions sur les structures moléculaires.
- ✓ Générer les empreintes (fingerprints) moléculaires.

9. Workflows de la chemoinformatique :

Les systèmes de workflow dans divers domaines, à savoir la chemoinformatique permettent aux utilisateurs de définir, gérer et exécuter successivement des processus pour accomplir l'ensemble des tâches récurrentes. Ils sont largement utilisées pour de multiples applications de la chemoinformatique telles que le criblage virtuel, le docking moléculaire, la conception de novo de médicaments, l'étude QSAR, etc. En effet, chaque processus de workflow peut être réalisé en utilisant une interface graphique dans laquelle différents types de noeuds ou de composants de logiciel sont reliés par des arcs ou des tubes [99]. Les nœuds sont définis par trois paramètres, à savoir, (1) les métadonnées d'entrée, (2) les algorithmes, les paramètres définis par

l'utilisateur ou les règles, et (3) les métadonnées de sortie. Les plus connues des systèmes de workflow open-source dans le domaine de la chemoinformatique sont Taverna, KNIME, et Cancer Grid (une extension de Taverna).

9.1. Taverna :

Taverna est une open source et le domaine du système de gestion de flux de travail indépendant, c'est une suite d'outils utilisés pour concevoir et exécuter des workflows scientifiques et aider dans les expérimentations in silico. Taverna a été créé par l'équipe de myGrid. Les outils de Taverna comprennent le Workbench (application client de bureau), l'outil de ligne de commande (pour une exécution rapide des flux de travail à partir d'un terminal), le serveur (pour l'exécution à distance des flux de travail) et le plugin Player (interface Web pour la présentation des flux de travail pour l'exécution à distance) [100].

Pour étendre les fonctionnalités de Taverna dans le domaine de la chemoinformatique, l'outil CDK a été combiné au système de workflow Taverna.

Ce plug-in CDK-Taverna [101] fournit 164 différents noeuds appelés "travailleurs". CDK-Taverna a le potentiel de devenir une alternative libre aux outils existants de workflow propriétaires.

Il existe aussi des centaines d'autres extensions pour le système Taverna, y compris CaGrid (Cancer Grid) [102] et BioDICE –Taverna (Biological Data Interactive Clustering Explorer- Taverna)[103].

9.2. KNIME (Konstanz Information Miner) :

Le Konstanz information Miner est un environnement modulaire qui permet un montage facile et visuel avec une exécution interactive d'un pipeline de données. Il est conçu comme un enseignement, la recherche et la collaboration plate-forme, ce qui permet une intégration facile de nouveaux algorithmes, la manipulation des données ou des méthodes de visualisation en tant que nouveaux modules ou nœuds [104].

KNIME a été développé par une équipe d'ingénieurs à l'Université de Constance (Allemagne) ; sa combinaison avec le CDK (KNIME-CDK [105]) permet d'offrir une large gamme de fonctionnalités et d'applications dans le domaine de la chemoinformatique (convertir des formats moléculaires, générer des fingerprints, etc.).

10. Les bases de donné de la chemoinformatique :

Une base de données est une collection de référentiel systématiquement organisé ou structuré d'informations indexées qui permet une récupération, une mise à jour, une analyse et une sortie de données faciles. Les bases de données de chemoinformatique facilement accessibles incluent les bases de données des structures chimiques, des protéines, des modèles QSAR, des médicaments, des cibles biologiques, des molécules bioactives, etc.

Les bases de données de chemoinformatique largement utilisées, leurs liens officiels et une brève description de chaque base de données sont illustrées dans le tableau 3.3 :

N°	Bases de données	Liens officiels	Brève description
1	QSAR DataBank	http://qsardb.org/repository/	QSAR DataBank
2	VAMMPIRE	http://vammpire.pharmchem.uni-frankfurt. de/vammpire/	Conception et optimisation de médicaments basés sur la structure
3	PubChem3D	https://pubchem.ncbi.nlm.nih.gov/	Référentiel ouvert pour les petites molécules et leur activité biologique
4	MMsINC	http://mms.dsfarm.unipd.it/MMsINC	Base de données des structures chimiques
5	CREDO	http://marid.bioc.cam.ac.uk/credo	Base de données des interactions protéine ligand
6	ChemBank	http://chembank.broadinstitute.org/	Base de données de petites molécules
7	DrugBank	http://www.drugbank.ca/	Informations sur la cible biologique et les médicaments
8	ChemDB	http://cdb.ics.uci.edu	Base de données des petites molécules
9	ChemMine	http://chemminedb.ucr.edu/	Base de données d'extraction de composés pour chimie du génome
10	National Cancer Institute (NCI) 3D database	http://www.cancer.gov/cancertopics/pdq/cancerdatabase	Base de données des médicaments Anticancéreux
11	ZINC	http://zinc.docking.org	Base de données gratuite des petites molécules disponibles dans le commerce
12	ChEMBL	https://www.ebi.ac.uk/chembl/	Molécules bioactives avec des propriétés "drug-like"
13	Therapeutic Target Database (TTD)	http://xin.cz3.nus.edu.sg/group/ttd/ttd.asp	Base de données des médicaments
14	PharmGKB	http://www.pharmgkb.org/	Ressources de connaissances pharmacogénomique
15	STITCH	http://stitch.embl.de/	Ressources pour explorer les interactions connues et prévues des produits chimiques et des protéines
16	SuperTarget	http://bioinf-apache.charite.de/supertarget_v2/	Base de données des médicaments et des protéines
17	ChemSpider	http://www.chemspider.com/About.aspx	Base de données des structures chimiques

Tableau 3.3 : Liste de base de données connue en chemoinformatique.

11. Conclusion :

Dans ce chapitre, nous avons présenté les approches d'optimisations multi-objectifs proposées dans la littérature pour résoudre les problèmes du domaine de la conception des médicaments notamment, le problème de docking, le problème de DND et le problème de conception de la librairie, ainsi que, nous avons abordé les différents outils et les bases de données de la chemoinformatique.

Dans le chapitre suivant, nous allons présenter les détails de notre travail après la présentation de l'algorithme choisie qui est NSGA-II et aussi l'outil CDK.

CHAPITRE 04 :

Contribution à la conception De Novo des médicaments

Au sommaire de ce chapitre :

1. Introduction.
2. L'idée de base.
3. Formulation de problème.
4. Description générale de l'architecture de notre outil.
5. Les étapes détaillées de notre travail.
6. Description de la dynamique de NSGAII-DND.
7. Le pseudo-code de l'algorithme NSGAII-DND.
8. Conclusion.

1. Introduction :

Traditionnellement, la plupart des méthodes de conception De novo de médicaments (De novo drug design) se concentrent particulièrement sur la conception de molécules qui satisfont un seul objectif (mono-objectif), soit la prédiction de l'affinité de liaison à une protéine connue ou, la similitude à un ligand et ignorent la nature multi-objectif de la découverte de médicament nécessaire à la prédiction des propriétés pour produire un composé "drug-like".

Récemment et comme vu au chapitre précédent de nombreuses nouvelles méthodes de conception de nouveaux médicaments sont apparues dans la littérature suite à l'introduction d'optimiseur multi-objectif à l'instar de MEGA, PLD, MOEA, MOLig, et d'autres qui tentent de synthétiser de nouvelles molécules "drug-like" satisfaisant plusieurs objectifs dans le but d'identifier celles qui ont le potentiel de devenir les médicaments de demain.

Dans cette lignée, on présente dans ce chapitre un nouvel outil pour la conception De novo de médicaments essentiellement basé sur l'algorithme multi-objectif NSGA-II. De ce fait nous faisons référence à cet outil par *NSGAII-DND* comme acronyme de «Non-dominated Sorting Genetic Algorithm for Do Novo Drug Design». Dans un souci de clarté, nous commençons tout d'abord par formuler le problème à résoudre ensuite une description générale suivie par des étapes détaillées de notre travail.

2. L'idée de base :

L'idée est d'utiliser un algorithme génétique multi-objectif afin de réaliser la conception de novo des nouveaux médicaments à partir d'une base de données contenant des fragments acides et amines. La solution (molécule) n'est pas forcement la meilleure solution qui existe. Au lieu de cela, la molécule doit être la meilleure solution optimale qui répond aux objectifs de conceptions ;

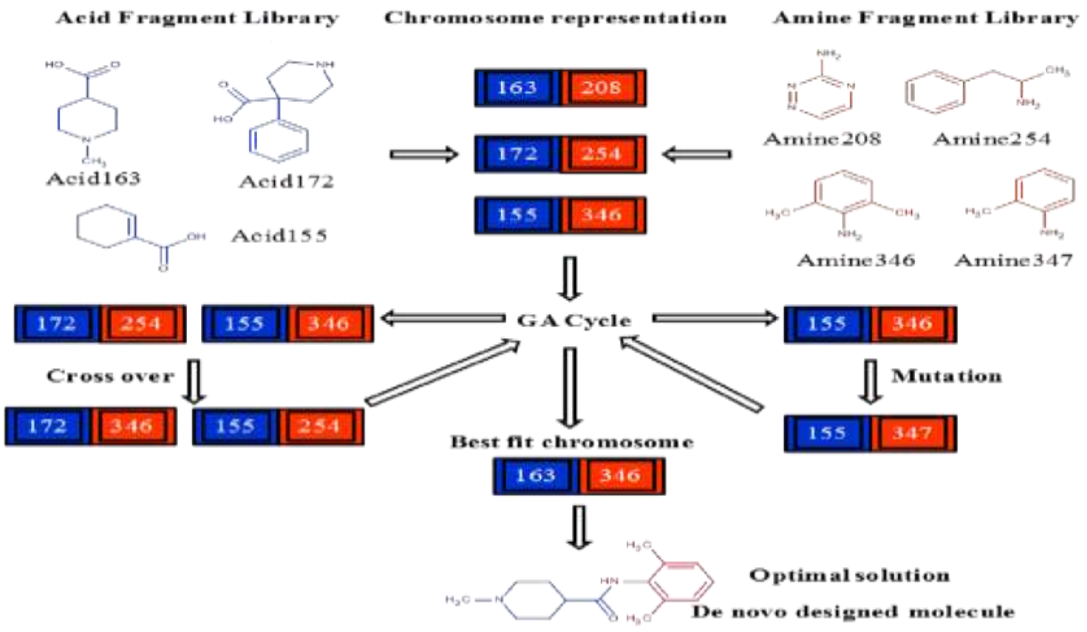


Figure 4.1 : L'idée générale de notre travail.

Notre étude est basée sur 2 objectifs, Premièrement il faut vérifier la biodisponibilité de la molécule conçue défini par la règle de 5 de Lipinski, puis la similarité de cette molécule a une molécule de référence connue calculé par le coefficient de Tanimoto.

3. Formulation de problème :

Etant donnés :

- ✓ k librairies de fragments de molécules où chaque librairie $L_{i,\ i=1,\dots,k}$ peut être vue comme un ensemble fini de fragments de molécule :

$$L_i = \{f_{i1}, f_{i2}, \dots, f_{il}\}$$ où l est la taille de la librairie.

- ✓ un ensemble F de m fonctions objectif représentant des propriétés physico-chimiques désirées :

$$F = \{F_1, F_2, \dots, F_m\}$$

Le problème à résoudre consiste à concevoir une molécule composite Mnew à partir de k fragments appartenant chacun à une des k librairies et qui optimise l'ensemble des m fonctions objectifs.

Matématiquement, notre problème consiste de trouver le k-uplet $(f^*_{1x1}, f^*_{2x2}, \dots, f^*_{kxk})$ de l'ensemble $\{L_1 \times L_2 \times \dots \times L_k\}$ qui répond aux exigences précédentes.

Le nombre de combinaisons possibles permettant de générer une molécule est de l'ordre de $l_1.l_2 \dots . l_k$, soit dans le pire des cas l^k où l est la taille maximale des librairies de fragments.

Il est bien clair que le problème à résoudre a une nature combinatoire et la tâche à résoudre consiste à effectuer une recherche dans l'espace croisé des indexes des librairies. Le nombre de molécules qu'il est possible de construire croit rapidement avec la taille et le nombre de librairies. Aussi une recherche exhaustive ne saurait être envisageable ainsi qu'une méthode d'optimisation exacte sachant qu'une telle complexité est accentuée par le coût de l'évaluation de fonctions objectives qui est souvent prohibitif. Une optimisation approchée s'impose donc comme l'alternative de choix pour identifier une molécule vérifiant au mieux les propriétés requises en un temps raisonnable. Notre choix s'est porté sur l'algorithme NSGA-II comme métaheuristique à utiliser pour accomplir cette tâche d'optimisation.

4. Description générale de l'architecture de notre outil :

L'objectif principal de notre travail consiste à offrir un support pour concevoir de nouvelles molécules et offrir les meilleures chances de sélectionner des composés "drug-like" en vue de poursuivre leur développement préclinique. Un tel outil doit offrir la possibilité de rechercher dans l'espace des fragments la molécule adéquate, de composer une molécule à partir de fragments et d'évaluer les propriétés d'une molécule conçue. Pour ce faire, nous avons développé *NSGAII-DND* un nouvel outil d'aide à la conception De novo de médicaments. Cet outil fonctionne en se basant sur

cinq composants principaux comme illustré sur la figure 4.2 et qui sont: l'algorithme multi-objectif NSGA-II qui représente le noyau de notre outil, les librairies de fragments, un évaluateur de similarité, un évaluateur de bio-disponibilité et enfin, l'outil open source CDK (Chemistry Development Kit).

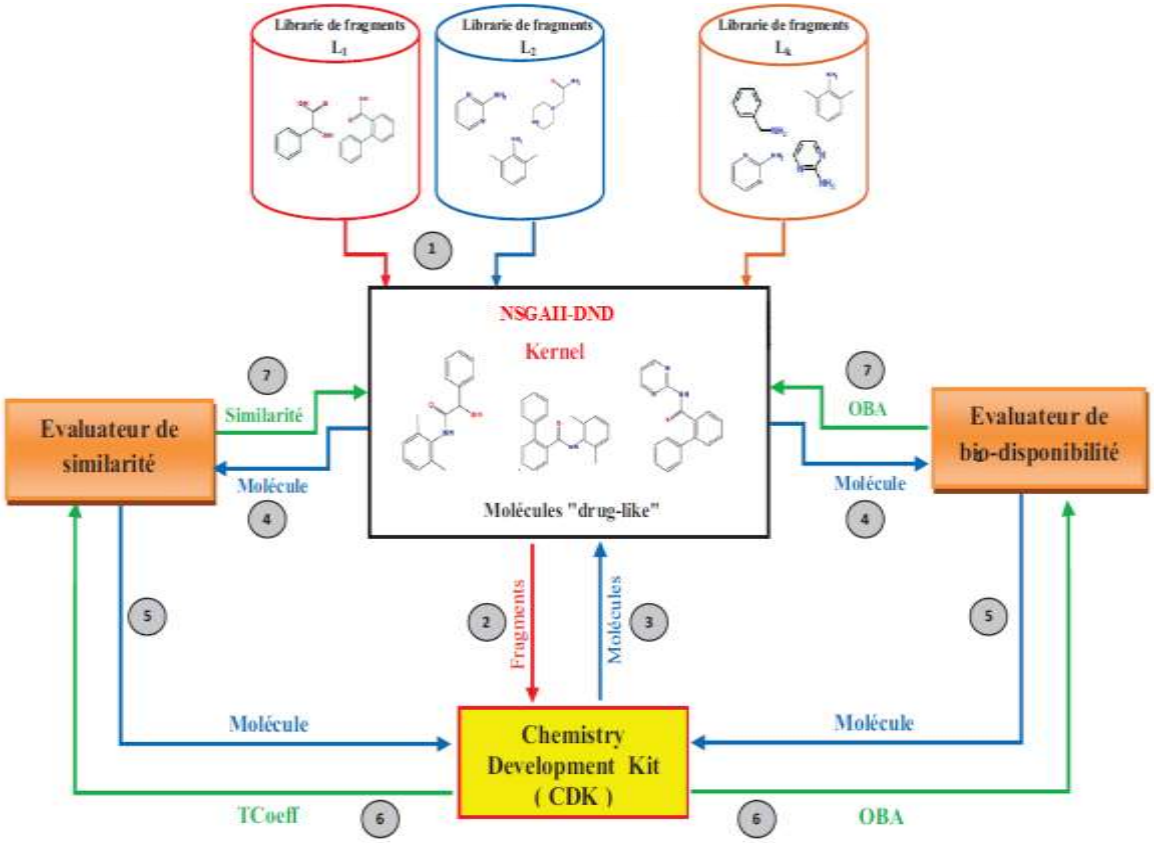


Figure 4.2: Modélisation de l'approche NSGAII-DND.

4.1. L'algorithme NSGA-II:

Dans notre travail, on va appliquer l'algorithme NSGA-II pour la conception de novo des nouveaux médicaments, son principe a été mentionné précédemment.

4.1.1. Définition :

NSGA-II est un algorithme génétique basé sur le principe de la non-domination qui pallie les inconvénients de NSGA tels que le manque d'élitisme et de la nécessité d'un choix à priori de la valeur du paramètre de partage optimal. Elle introduit la notion de rang de non-domination pour évaluer la force de chaque individu. Des procédures d'élitisme, visant à conserver les meilleures solutions de génération en génération, contribuent à accélérer la vitesse de convergence et les performances générales des algorithmes génétiques. La diversité est également garantie à l'aide d'une méthode de "crowding" ne nécessitant pas de paramètre supplémentaire [53].

4.1.2. Principe détaillé de NSGA-II :

4.1.2.1. Classification des individus :

Dans un premier temps, avant de procéder à la sélection, on affecte à chaque individu de la population P un rang « rank » (en utilisant le critère du rang de Pareto). Tous les individus non-dominés de même rang sont classés dans une catégorie (front).

- ✓ **Critère du rang de Pareto :** Avec ce critère de comparaison, le premier front F_1 correspond à l'ensemble des solutions non-dominées de P. En considérant le reste des individus dans P après avoir enlevé ceux de F_1, et après avoir réalisé un nouveau tri de dominance, nous obtenons le deuxième front F2 constitué des individus non-dominés de l'ensemble ($P \setminus F_1$). Cette procédure est répétée jusqu'à ce que tous les individus de P soient attribués à un front [120].

Le pseudo code de la procédure de classification des individus suivant le rang de Pareto est représenté dans l'algorithme suivant [113] :

Pseudo code 4.1 : Frontière de pareto

```
Début
   P = population ()     /*Ensemble total de la population*/
   i = 1         /*i est le compteur des fronts de Pareto, initialiser par 1 */
   Répéter jusqu'à P =Ø
       Fi = chercher-ind-non-dominés(P)      /*Trouver Fi, l'ensemble des
                                              Individus non dominés correspondant au
                                              front de Pareto i*/
       P = P \ Fi      /*Supprimer les ind-non-dominés à partir de population
                                    globale       P*/
       i = i + 1       /*Incrémenter le compteur de front*/
  Fin Répéter
Fin
```

La figure 4.3 suivante présente un exemple de classification en fronts de Pareto :

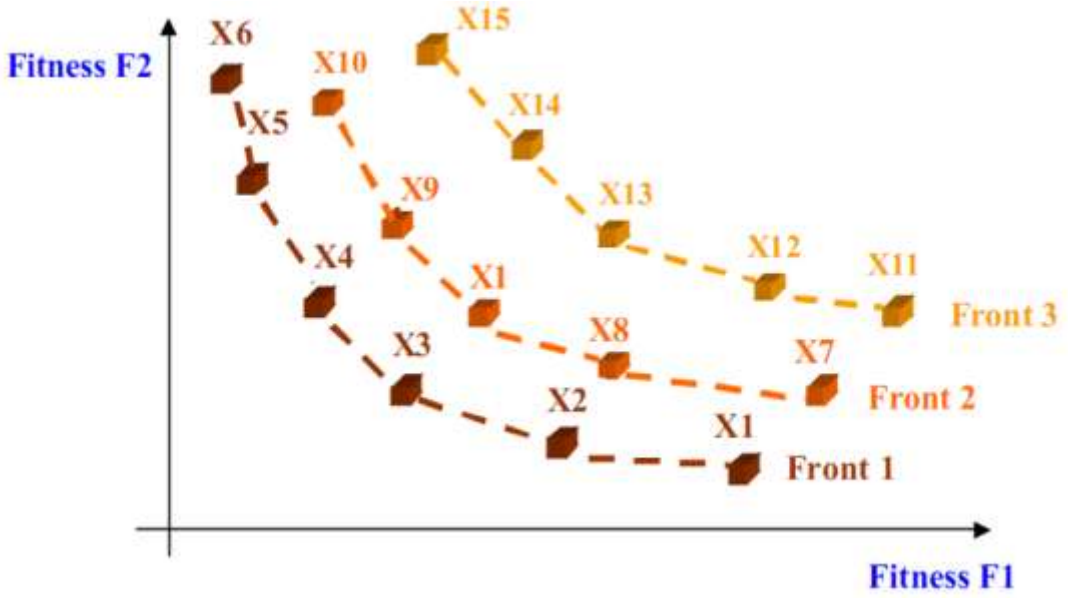


Figure 4.3: Classification des individus en plusieurs fronts de Pareto [113].

4.1.2.2. Préservation de la diversité :

La technique "crowding distance" est utilisée par le NSGA-II pour préserver la diversité des solutions sur le front de Pareto. Cette technique s'applique sur le dernier front de Pareto pour compléter la taille de la population parents pour la génération suivante.

✓ **Calcul de la distance de surpeuplement :**

La distance de surpeuplement nommé crowding distance d'une solution i (ou d'un individu) di correspond au semi-périmètre du cuboïde dont les vertex sont les voisins les plus proches de l'individu i, elle se calcule en fonction du périmètre formé par les points les plus proches de i sur chaque objectif. La figure 4.4 montre une représentation à deux dimensions associée à la solution i.

Le calcul de la distance de surpeuplement nécessite, avant tout, le tri des solutions selon chaque objectif, dans un ordre ascendant. Ensuite, pour chaque objectif, les individus possédant les valeurs limites (la plus petite et la plus grande valeur de fonction objectif) se voient associés une distance infinie ∞ . Pour les autres solutions intermédiaires, on calcule la crowding distance qui est égale à la différence normalisée des valeurs des fonctions objectifs de deux solutions adjacentes. Ce calcul est réalisé pour chaque fonction objectif. La crowding distance d'une solution est calculée en sommant les distances correspondantes à chaque objectif.

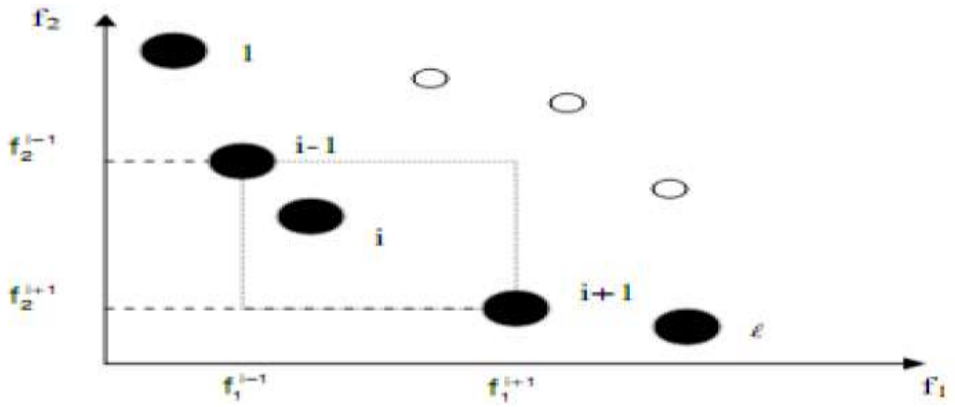


Figure 4.4: Principe de Crowding distance [120].

L'algorithme ci-dessous montre la procédure de calcul de la distance de surpeuplement de toutes les solutions non dominées de l'ensemble I (I est le front trié). Dans cet algorithme, f_m^{max} et f_m^{min} représentent les valeurs maximale et minimale de la $m^{\text{ème}}$ fonction objectif. Entre deux solutions de rangs différents, on préfère la solution avec le plus petit rang (ou le plus petit front). Pour deux solutions qui appartiennent au même front, on préfère l'individu possédant la plus grande valeur de crowding distance [113].

Pseudo code 4.2 : Procédure crowding distance [114].

Début
Initialiser $l = \|F\|$
Pour chaque $z^{(i)} \in F$ faire
Initialiser $d_i = 0$;
Pour chaque fonction d'objectif f_j faire
Trouver le vecteur d'indice $I^{(j)} = Trier(f_j, <)$;
Pour chaque fonction d'objectif f_j faire
Assigner $d_1^m = d_l^m = \infty$;
Pour chaque $z^{(i)}$ $i = 2 \ldots (l-1)$ faire
Calculer

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{max} - f_m^{min}}$$

Fin

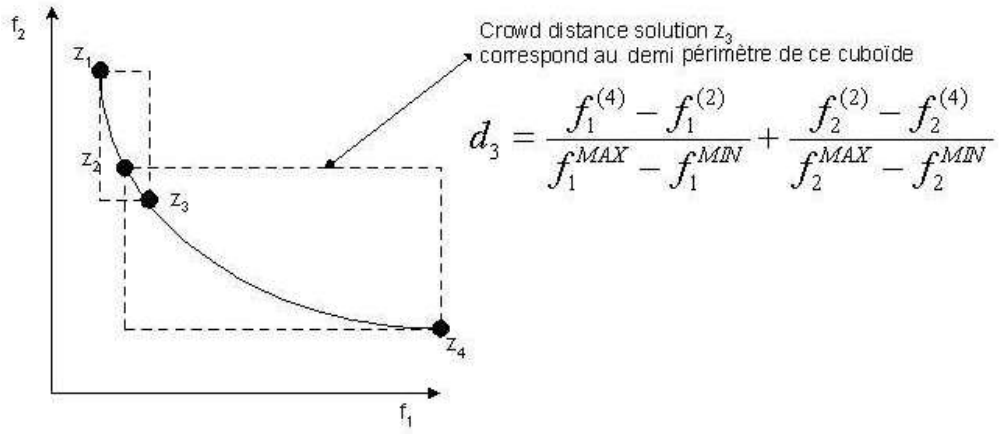


Figure 4.5 : Cas hypothétique de l'assignation de la Crowded distance [114].

4.1.2.3. L'élitisme :

L'élitisme était d'inclure le meilleur individu de la population P_t dans la population P_{t+1} dans le but de prévenir sa perte due aux effets d'échantillonnage ou aux perturbations des opérateurs. Cette stratégie, qui peut être étendue en copiant les b meilleures solutions à la génération suivante, est appelé élitisme. Donc, l'élitisme consiste à maintenir une population externe, qui permet d'archiver le meilleur ensemble des points non dominés découverts jusqu'ici. Cet ensemble est mis à jour continuellement pendant la recherche, et participe avec une certaine probabilité à l'étape de sélection. Cette méthode permet ainsi une intensification de la recherche. En optimisation évolutionnaire multi-objectifs, l'élitisme joue un rôle important [46].

4.1.2.4. Boucle principale de l'algorithme NSGA-II :

Initialement, il y a la création aléatoire d'une population parent P_0. Chaque individu de cette population est affecté à un front de Pareto adéquat. On applique les opérateurs génétiques (sélection, croisement et mutation) pour générer la population enfant Q_0 à partir de la population P_0.

Pour une génération t, les deux populations P_t et Q_t sont ensuite réunies pour former la population mixte R_t. Cette population est triée selon le critère du rang de Pareto pour former les fronts successifs. Par la suite, la nouvelle population P_{t+1} est créée et est remplie au fur et à mesure avec les différents fronts successifs. Comme la taille de la population R_t est supérieur à N, les fronts successifs ne peuvent pas intégrer en totalité la nouvelle population qui doit être de taille N. Ces fronts seront tout simplement éliminés.

Cependant, la taille du dernier front considéré peut être supérieure aux nombres de cases vides à remplir dans la nouvelle population. Dans ce cas, le critère de surpeuplement 'crowding distance' sera utilisé autour des individus pour assurer la diversité dans la population pour choisir parmi les solutions du dernier front, celles qui vont intégrer la nouvelle population afin de favoriser les solutions dans les régions les moins peuplées du front considéré dans le but d'améliorer la diversité des solutions.

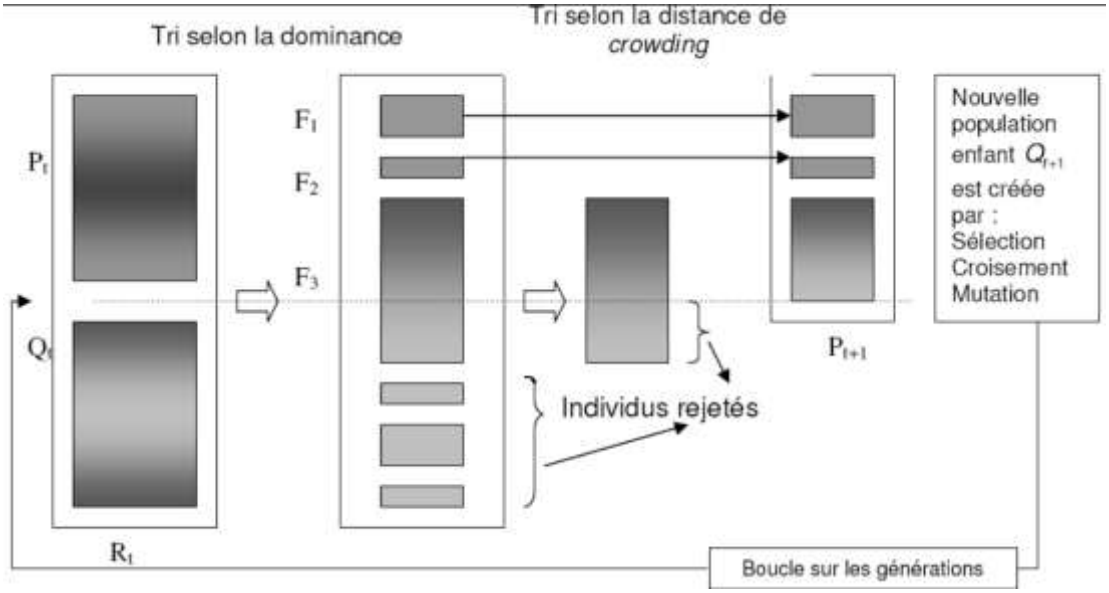


Figure 4.6: Schéma de l'évolution de l'algorithme NSGA –II [120].

Pseudo code 4.3 : Algorithme NSGA-II [120]

1) $t \leftarrow 0, P_0 \leftarrow random(),\ |P_0| = N$
2) Repeat
3) Générer une nouvelle population enfant (Q_t) par sélection, croisement et mutation
4) $R_t \leftarrow P_t \cup Q_T$
5) $F \leftarrow Tri - dominance(R_t)\ //F = (F_1, F_2, F_3 \dots)$
6) $P_{t+1} \leftarrow \emptyset, i \leftarrow 1;$
7) while $|P_{t+1}| + |F_i| < N$ do
8) $P_{t+1} \leftarrow P_{t+1} \cup F_i,\ i \leftarrow i + 1$
9) end while
10) $P_{t+1} \leftarrow Tri - surpeuplement(P_{t+1})$ // Ordonner l'ensemble F_i selon les «distances de surpeuplement » (crowding distances) et inclure les $N - |P_{t+1}|$ solutions ayant les valeurs de distance les plus grandes dans la population P_{t+1}.
11) $t \leftarrow t + 1$ // (Incrémenter le compteur des générations)

Until Critère d'arrêt rencontré

4.1.2.5. Les opérateurs génétiques du NSGA-II :

a. Opérateur de sélection :

Après affectation des rangs aux individus en appliquant le critère de Pareto, on calcul la crowding distance des individus de la population parents. La sélection des individus parents pour faire le croisement est exécutée en utilisant l'opérateur de comparaison crowded ($\prec_n$) [113].

✓ **Opérateur de comparaison Crowded :**

L'opérateur de comparaison crowded ($\prec_n$) guide le processus de la sélection avec la répartition uniforme des solutions de Pareto. Un individu de la population a deux attributs :

- rang de non domination (i_{rank}).
- distance crowding ($i_{distance}$)

Soit les deux individus i et j,

$$i \prec_n j \text{ si } i_{rank} < j_{rank} \text{ ou } (i_{rank} = j_{rank}) \text{ et } (i_{distance} > j_{distance})$$

Avec cette relation pour la comparaison de deux solutions non dominées appartenant à deux fronts de Pareto, on préfère la solution appartenant au front de Pareto d'ordre le plus faible. Sinon, dans le cas où les deux solutions appartenant au même front de Pareto, on choisit la solution qui a la distance crowded la plus élevée.

b. Les opérateurs de croisement et de mutation :

Dans NSGA-II, nous appliquons les opérateurs de croisement et de mutation pour générer une nouvelle population Q_t qui sera combinées avec la population courant P_t.

4.1.3. Organigramme de l'algorithme NSGA-II :

La figure 4.7 représente l'organigramme de l'algorithme NSGA-II :

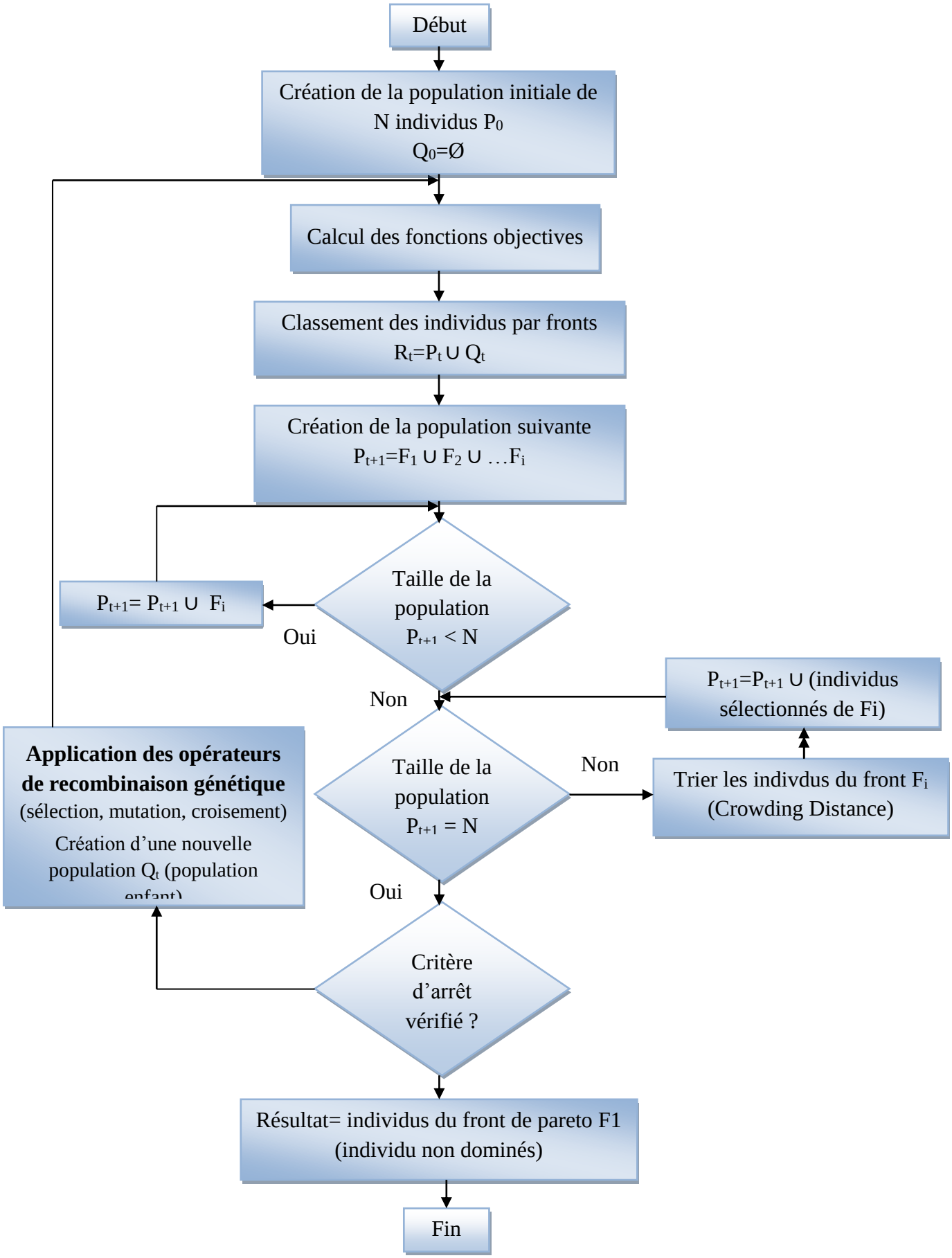


Figure 4.7: Organigramme de l'algorithme NSGA-II [113].

4.1.4. Pourquoi on a choisi NSGA-II ?

NSGA-II est apparue pour améliorer la version NSGA-I et résoudre toutes ses critiques: complexité, non élitiste et utilisation de sharing [40]. Donc NSGA-II a permis de :

- ✓ réduire la complexité de l'algorithme.
- ✓ créer une méthode plus élitiste pour obtenir des meilleurs résultats.
- ✓ NSGA-II est une méthode basée sur la dominance de Pareto donc il n y a pas de perte des solutions non dominées.

4.2. Les librairies de fragment :

Cette partie de l'outil représente le répertoire des données utilisées. Chaque librairie comporte un ensemble fini de fragments de molécule.

- ✓ Nous avons trois librairies de fragments, la première librairie contient 28 acides, la deuxième librairie contient 108 amines, les acides et les amines sont des fragments extrait à partir des médicaments connus, la troisième librairie contient deux molécules ce sont les molécules de références : "Furano_pyrimidine" qui est une molécule anti-cancer, et "Lidocaïne" est un anesthésique local le plus couramment utilisé en dentisterie. Ces molécules de références sont pour calculer la similarité des molécules.

4.3. Les évaluateurs :

Le but de notre outil *NSGAII-DND* est de concevoir des nouvelles entités chimiques qui doivent satisfaire simultanément deux objectifs d'une importance pharmaceutique qui sont: drug-likness (biodisponibilité) de la molécule conçue et sa similitude au médicament de référence. Le premier objectif qui est basé sur l'évaluation de la biodisponibilité orale (en anglais Oral Bio-Availability, OBA) nécessaire pour la sélection de composés "drug-like", se fait grâce aux " règles des 5 de Lipinski" au niveau du composant "Evaluateur de bio-diponibilité". Tandis que, le second objectif est calculé au niveau du composant "Evaluateur de similarité" en utilisant des indicateurs de similarité tels que "le coefficient de Tanimoto (TCoeff)" pour mesurer la similarité entre une nouvelle molécule conçue et une molécule de référence. Ces évaluateurs peuvent être vus comme une interface entre le composant *NSGAII-DND* et le CDK. Ils reçoivent du composant *NSGAII-DND* une molécule conçue et la transmettent au CDK pour calculer ses valeurs en OBA et TCoeff qui seront transmises à leur tour du CDK au composant *NSGAII-DND* via les évaluateurs.

4.4. L'outil CDK :

4.4.1. Définition:

Le Chemistry Development Kit (CDK) est une bibliothèque open source java pour la Chemo et la bioinformatique. Le CDK a été créé le 27-29 Septembre 2000 à l'Université de Notre Dame par Christoph Steinbeck, Egon Willighagen et Dan Gezelter, les développeurs de Jmol et JChemPaint à l'époque pour fournir une base de code commune. La première publication du code source a été faite le 11 mai 2011.

Depuis plus de 75 personnes ont contribué au projet, [106] conduisant à un ensemble riche de fonctionnalités. Cependant, il a été considéré jusqu'à présent comme étant l'un des projets open source les plus actifs de la chemoinformatique qui est largement supporté par la communauté scientifique.

4.4.2. Les logiciels «JChemPaint» et «Jmol» :

Actuellement, le CDK est utilisé comme source pour un certain nombre des logiciels notamment JChemPaint, Jmol, SENECA, JOELib et d'autres outils chimiques [107].

- ✓ **JChemPaint :** [108] est un logiciel open source écrit en Java qui est utilisé comme un afficheur et éditeur de structures chimiques en 2D. Ce logiciel peut être intégré en tant que composant dans des applications basées CDK. JChemPaint est actuellement en cours d'élaboration dans le cadre du kit de développement de la chimie, et une application de JChemPaint SWT basée est en cours d'élaboration, dans le cadre de Bioclipse [109].
- ✓ **Jmol :** [110] est un outil de visualisation en 3D de molécules open source pour les étudiants, les éducateurs et les chercheurs en chimie et en biochimie. Il est multi-plateformes, fonctionnant sous Windows, Mac OS X et les systèmes Linux/Unix. Cet outil peut être utilisé comme visualiseur 3D pour les projets basés CDK.

4.4.3. Principales fonctionnalités de la librairie CDK :

L'objectif principal de l'outil CDK est d'être une librairie de classes Java conçue pour soutenir les logiciels Java à résoudre les problèmes chimiques. Cette librairie CDK est organisée en arborescence des packages descendant d'une même racine "org.openscience.cdk" qui est un groupe fonctionnel de classes et d'interfaces permettant d'appliquer les méthodes de chimioinformatique, y compris: manipulation des structures chimiques 2D et 3D, création des empreintes moléculaires (fingerprints), gestion des fichiers d'entrée/sortie, manipulation des formats de fichiers SMILES, etc. Ces principales fonctionnalités seront détaillées dans les sous sections suivantes :

4.4.3.1. Classes de base :

Les classes de base contenues dans le package racine de CDK "org.openscience.cdk" sont toutes des représentations construisant l'ensemble des concepts chimiques de base tels que: les atomes, les liaisons, les molécules, etc. La figure 4.8 représente un diagramme UML expliquant la hiérarchie d'héritage et les dépendances entre les classes fondamentales de CDK (ChemObject, Element, AtomContainer, ElectronContainer, Atom, Bond, AtomType, Isotope, etc.).

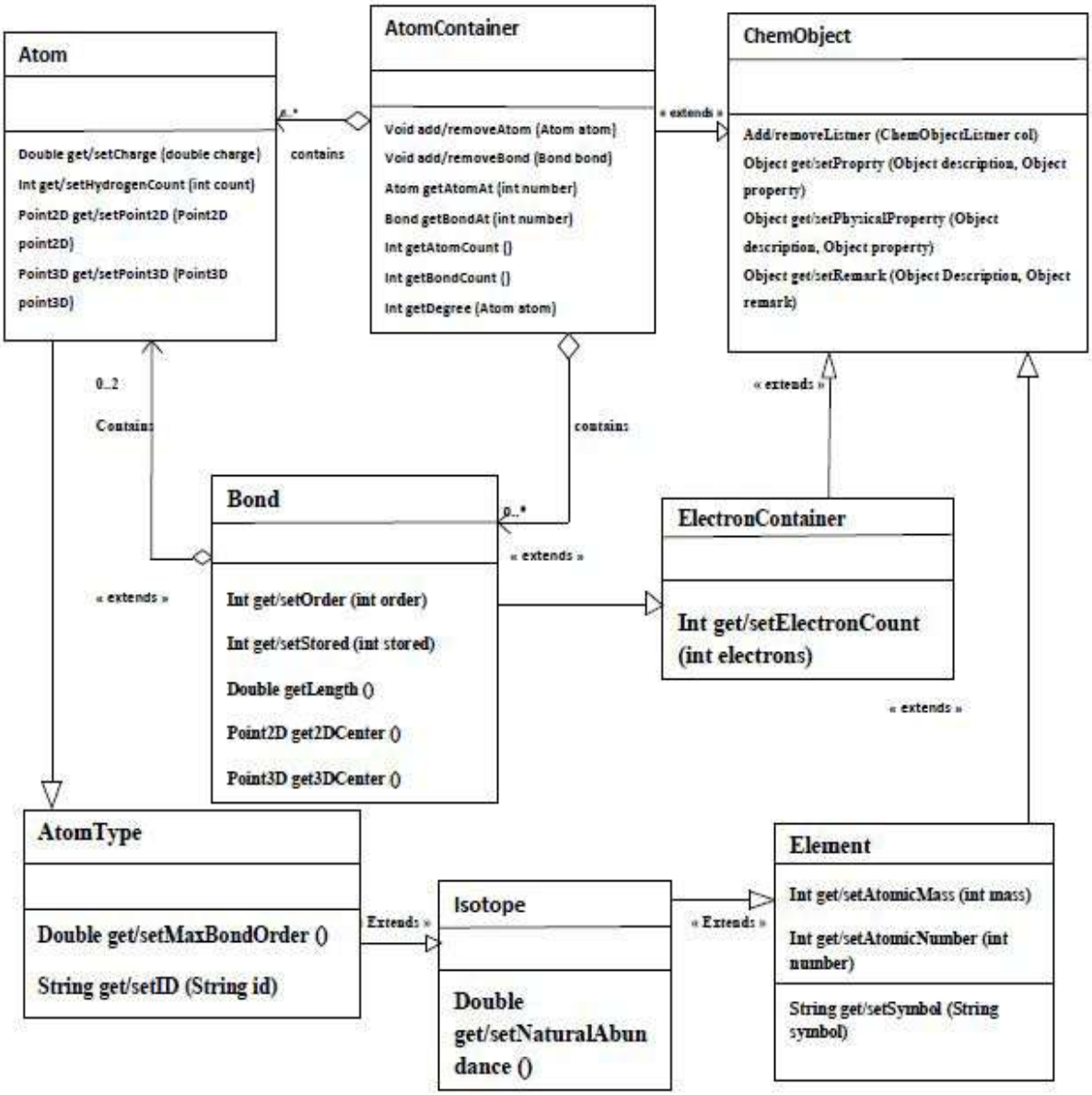


Figure 4.8: Diagramme UML expliquant la hiérarchie d'héritage et les dépendances entre les classes fondamentales du CDK [12].

A partir de ce diagramme UML qui illustre seulement la relation entre les classes de base, nous observons que la super classe ChemObject fournit des méthodes pour stocker des propriétés complexes pour tout objet CDK dérivé.

Une molécule est construite en créant une instance de la classe Molecule et l'ajout successivement d'instances de classe Atom et classe Bond au conteneur Molecule. Un exemple de pseudo-code de la création de la molécule "Ethene" est donné comme suit :

Pseudo-code 4.4 : Création de la molécule "Ethene"

```
Molecule mol = new Molecule ();
mol.addAtom (new Atom("C")); //atom(0)
mol.addAtom (new Atom("C")); //atom(1)
mol.addAtom (new Atom("H")); //atom(2)
mol.addAtom (new Atom("H")); //atom(3)
mol.addAtom (new Atom("H")); //atom(4)
mol.addAtom (new Atom("H")); //atom(5)
mol.addBond (0,1,2);
mol.addBond (0,2,1);
mol.addBond (0,3,1);
mol.addBond (1,4,1);
mol.addBond (1,5,1);
```

Les valeurs 1 et 2 en gras indiquent respectivement que la liaison entre les deux atomes est simple ou double. Notons que la classe Molecule n'est pas considérée comme une classe de base qui possède les méthodes addBond(Bond) et AddAtom(Atom). Elle est donc une sous-classe de la classe de base AtomContainer.

4.4.3.2. Manipulation des structures chimiques 2D et 3D :

La possibilité d'afficher et de manipuler des structures chimiques 2D et 3D est l'une des fonctionnalités les plus importantes de tout programme lié à la chimioinformatique. Cela inclut la capacité de générer des coordonnées de ces structures chimiques. Les classes pour la manipulation de la structure graphique 2D et 3D travaillent en haut d'une ChemModel. La classe Renderer2D et la classe Renderer3D produisent respectivement des structures 2D et 3D comparables à celles produites par la majorité des logiciels disponibles dans le marché. Le fonctionnement da la classe Renderer2D nécessite l'utilisation de l'applet CDK-JChemPaint, qui est un éditeur graphique et un visualisateur de structures chimiques 2D. Une application simple de celui-ci est représentée sur la figure 4.9 suivante:

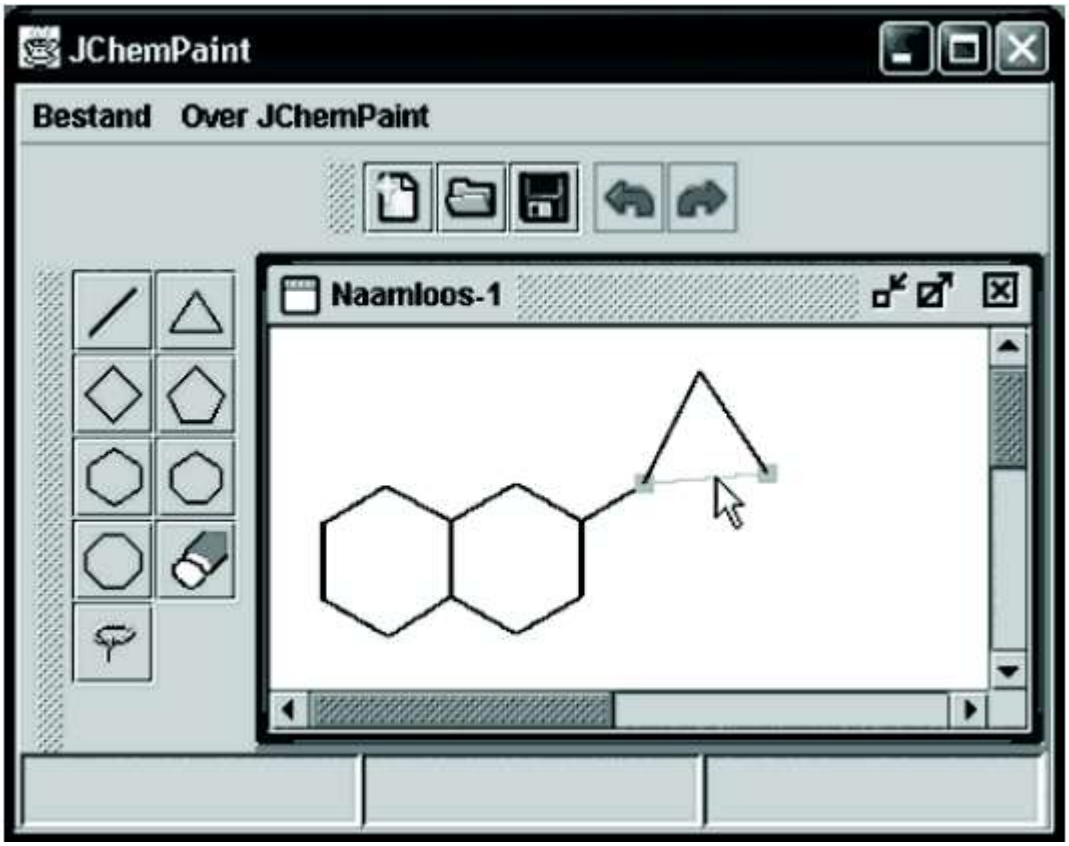


Figure 4.9: Visualisation de structures chimiques 2D [12].

4.4.3.3. Fichiers d'entrée/sortie:

Les fichiers d'entrée (Input) et de sorties (Output) sont généralisés dans les CDK. Toutes les classes IO mettent en œuvre soit l'interface ChemObjectReader ou l'interface ChemObjectWriter [34]. Le CDK supporte un certain nombre de formats de fichiers XYZ, MDL-sdf, MDL-mol, APB, et CML.

L'exemple suivant représente le pseudo code de la lecture d'une molécule "Lidocaïne" située dans le répertoire C.

Pseudo code 4.5: Lecture de la molécule "Lidocaïne".

```
File filename = new File (C:/ Lidocaïne.sdf);
IteratingMDLReader sdfreader =new IteratingMDLReader (new
FileInputStream(filename), DefaultChemObjectBuilder.getInstance());
IMolecule lidocaine= (IMolecule) sdfreader.next();
```

4.4.3.4. Smiles :

Le code smiles (Simplified Molecular Line Entry Specification) fournit des représentations de chaîne de constitutions moléculaires. Grâce à sa compacité et sa simplicité d'utilisation, il est aujourd'hui largement utilisé comme un format d'échange pour représenter les structures moléculaires.

Un exemple de pseudo code correspondant à la génération du format smile pour la molécule "Lidocaïne".

Pseudo code 4.6: Génération du format SMILES pour la molécule "Lidocaïne".

```
String  smile = " ";
SmilesGenerator sm =  new SmilesGenerator();
smile = sm.createSMILES(lidocaine);
```

4.4.3.5. Fingerprints :

Le fingerprint est aujourd'hui un outil indispensable pour vérifier la similitude entre les structures chimiques. Ils sont également un moyen pour éliminer les molécules avec de mauvaises propriétés, ainsi pour rechercher de telles sous-structures dans une base de données. Il existe généralement deux types de base de fingerprints : Clés de structure (Structure-Key) et empreintes avec hashage (Hashed Fingerprints). Dans le CDK, pour calculer le fingerprint haché, la classe Fingerprinter produit le fingerprint de Daylight [111] qui consiste à énumérer toutes les sous-structures linéaires ayant une longueur entre 2 et 7 atomes dans un vecteur binaire. La taille typique pour ce fingerprint haché est de longueur 512 ou 1024 bits, encodant donc uniquement la présence d'une sous-structure. La génération du fingerprint de Daylight pour la molécule "Lidocaïne" est donné par le pseudo code suivant :

Pseudo code 4.7 : Génération du fingerprint de Daylight pour la molécule "Lidocaïne" :

```
IFingerprinter  fingerprinter = new Fingerprinter();
IBitFingerprint fingerprint =fingerprinter.getBitFingerprint(Lidocaïne);
fingerprint.size();  //retourne 1024 bits
fingerprint.length();     //retourne l' ensemble de bits(la longueur réelle du
                                fingerprint)
```

Cependant, la classe PubchemFingerprinter est utilisée pour générer le fingerprint de type clé CACTVS (PUBCHEM_CACTVS_SUBSKEYS) [112] qui est représenté par un vecteur de taille 881 où, chaque bit du vecteur indique la présence ou l'absence de certaines espèces chimiques moléculaires. La génération du fingerprint de CACTVS pour la même molécule "Lidocaïne" est donné par le pseudo code suivant:

Pseudo code 4.8: Génération du fingerprint de CACTVS pour la molécule "Lidocaïne":

```
PubchemFingerprinter fingerprinter = new PubchemFingerprinter();
BitSet fingerprint = fingerprinter.getFingerprint(Lidocaïne);
fingerprint.size();    // retourne 881 bits
fingerprint.length();   // retourne la longueur réelle du fingerprint
```

5. Les étapes détaillées de notre travail :

5.1. Combinaison acide/amine :

La formule générale d'un acide est comme suite : R-COOH
La formule générale d'une amine est comme suite : R'-NH2

L'idée est de trouver l'emplacement des atomes OH dans l'acide correspondent et l'emplacement de l'atome H dans le fragment amine on utilisons le package SMARTS de CDK, puis de trouver les bonds (les liaisons) entre l'atome C et les atomes O,H dans l'acide et la liaisons entre N et H dans l'amine, puis d'enlever les atomes O et H à partir d'acide et H à partir d'amine avec les liaisons de chaque atome, puis de sauvegarder l'emplacement ou nous enlevons, et enfin de créer une liaison entre l'atome C d'acide et l'atome N d'amine, comme l'indique dans l'équation chimique suivante :

$$\mathbf{R\text{-}COOH + R'\text{-}NH2 \longrightarrow R\text{-}OCNH\text{-}R'+H2O}$$

5.2. Génération de la population initiale P_0 :

La population initiale est constituée d'un ensemble de N individus, chaque chromosome est représenté sous la forme d'un vecteur de nombres entiers, la valeur représentative de leur identité dans les librairies de fragments d'acide et amine. A la fin de la conception actuelle, seuls deux chromosomes de gènes sont utilisés. Chaque chromosome représente une molécule de médicament possible, comme cela peut être synthétisé à partir des deux fragments (un acide et une amine) qui composent les différents gènes. Ici, le premier gène représente le fragment acide et le deuxième gène représentent le fragment d'amine. La banque de fragments utilisés dans la conception des chromosomes se compose de 28 acides et 108 amines extraits de médicaments connus [13].ces fragments sont téléchargé à partir du site officielle de la base de donné moléculaire Chemspider (http://www.chemspider.com/).

Le pseudo-code suivant décrit bien le mécanisme de génération de la population initiale:

Pseudo code 4.9 : La population initiale.

```
idAmine : entier ;
idAcide : entier ;
individu: vecteur d'entier ;
Pour i allant de 1 à taille N faire
    idAmine= choisir aléatoirement l'id d'amine par la formule
        (math.random ()*taille de Library amine);
    idAcide= choisir aléatoirement l'id d'acide par la formule
        (math.random ()*taille de Library acide) ;
     individu= individu [idAmine] [idAcide] ;
Fin pour
```

Voilà un schéma qui représente la structure des individus sous forme d'un vecteur d'entier :

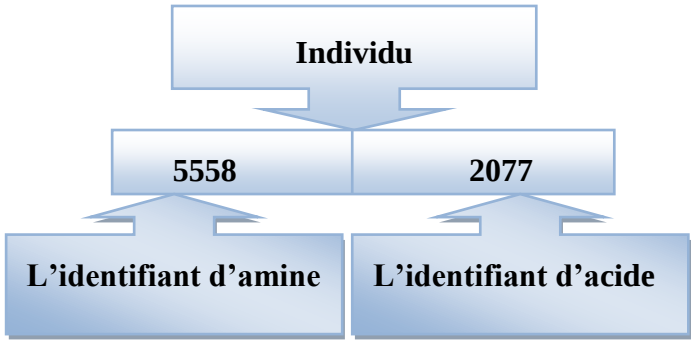


Figure 4.10: La représentation des individus en nombre entier.

5.3. Création de la population enfant Q_0 :

La population Q_0 de N enfants est créée à partir de la population parent P_0 en utilisant les opérateurs génétiques (sélection, croisement, mutation).

Pseudo code 4.10 : Le pseudo code suivant décrit comment trouver la population Q_0:

```
Début
    Pt : la population initiale
    St= Sélection (Pt) ;
    Ct=Croisement (St) ;
    Rt=Mutation (Ct) ;
    Retourne Rt ;
Fin
```

5.4. Combinaison des deux populations P_0 et Q_0 :

Selon l'algorithme NSGA-II, les deux populations P_0 et Q_0 sont combinées pour former la population mixte R_0 de taille 2N.

5.5. Les opérateurs génétiques :

5.5.1. L'opérateur de sélection :

La sélection dans l'algorithme NSGA-II se fait par l'opérateur de comparaison crowded qui est mentionné précédemment.

Pseudo code 4.11 : La sélection

```
Individu ind1, ind2;
Int ind1rank, ind2rank;
Float ind1distance, ind2distance ;
If((ind1rank < ind2rank) ou (ind1rank = ind2rank) et (ind1distance > ind2distance))
    Retourne ind1 ;
Else
    Retourne ind2 ;
```

5.5.2. L'opérateur de croisement :

Dans notre cas, nous utilisons le croisement à un point, le schéma suivant illustre comment le faire :

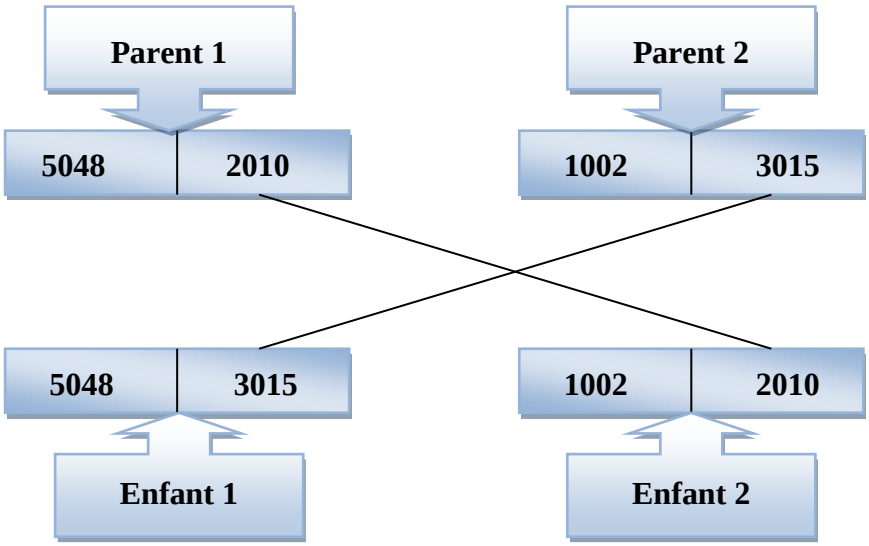


Figure 4.11: Le croisement des individus.

5.5.3. L'opérateur de mutation :

Dans notre cas, l'opérateur de mutation se fait comme illustre le schéma ci-dessous:

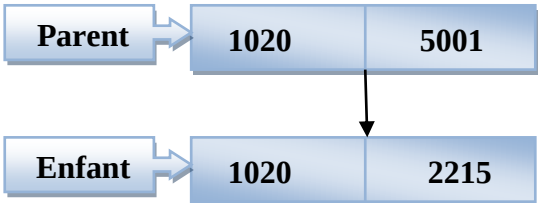


Figure 4.12 : La mutation d'un individu.

5.6. Evaluation des individus (la fonction objectif)

L'objectif de notre approche proposée est de trouver et concevoir une nouvelle molécule "drug-like" qui dispose d'une très bonne biodisponibilité par voie orale et qui est également similaire à une molécule de référence connue. Par conséquence, ce problème est typiquement un problème multi-objectif qui consiste à maximiser deux fonctions objectifs. La première fonction objectif "OBA" (Oral Bio-Availability) se focalise sur la maximisation du score de la biodisponibilité orale et la deuxième fonction "TCoeff" (Tanimoto Coefficient) consiste à maximiser la valeur du coefficient de similitude. On propose d'adopter une méthode d'optimisation par agrégation en transformant un problème multi-objectif en un problème mono-objectif selon la formule ci-dessous :

$$F(y) = \sum_{i=1}^{m} a_i f_i\,(y)$$

Ou les a_i sont les poids associé a chaque objectif telle que $a_i \geq 0$ et $\sum_{i=0}^{N} a_i = 1$, y est le vecteur de paramètre de la fonction fi la molécule synthétisé, m le nombre d'objectif (m=2), pour notre travaille la fonction objectif est comme suite :

$$F(y) = a_1 * OBA(y) + a_2 * TCoeff(y, Mref)$$

Avec, OBA et TCoeff sont deux fonctions d'évaluation de la molécule conçue (solution y) ; et Mref est la molécule de référence ;

5.6.1. La fonction "OBA" (drug-likness) :

La biodisponibilité d'un médicament correspond à la quantité du principe actif qui atteint la circulation sanguine. Elle dépend directement de la voie d'administration et de la forme du médicament. En effet, une molécule qui dispose d'une très bonne biodisponibilité par voie orale(OBA) est un bon candidat pour être un médicament ; elle est alors appelée "drug-like". L'évaluation de l'OBA se fait grâce à la "règle de 5 de Lipinski " qui a été formulée par Christopher A. Lipinski en 1997, basée sur l'observation que les médicaments administrés par voie orale plus sont relativement petites et modérément lipophiles molécules [115, 116]. Cette règle contient 4 paramètres concernant les propriétés qu'un médicament peut avoir :

- ✓ Le nombre d'hydrogène donneur <= 5.
- ✓ Le nombre d'hydrogène accepteur <= 10.
- ✓ La masse moléculaire <= 500.
- ✓ Le coefficient de partage entre l'eau et un solvant organique logP <= 5.

Notez que tous les nombres sont multiples de cinq, qui est l'origine du nom de la règle.

Ces différents paramètres (descripteurs) utilisés pour définir la règle des 5 sont calculés par l'outil open source CDK que nous avons déjà décrit précédemment dans la classe RuleOfFiveDescriptor.

Cependant, grâce à cette librairie CDK écrite en Java, nous pouvons calculer la fonction OBA et maximiser son score en définissant les 5 conditions suivantes:

- ✓ Lorsque tous les paramètres de Lipinski sont violés, score OBA = 0
- ✓ Lorsque un seul paramètre de Lipinski est satisfait, score OBA = 0.25
- ✓ Lorsque deux paramètres de Lipinski sont satisfaits, score OBA = 0.5
- ✓ Lorsque trois paramètres de Lipinski sont satisfaits, score OBA = 0.75
- ✓ Lorsque tous les paramètres de Lipinski sont satisfaits, score OBA = 1

Notons que les composés chimiques vérifiant toutes les propriétés de Lipinski ont tendance à avoir un taux d'attrition plus faible au cours des essais cliniques et ont donc une plus grande chance de devenir un jour des médicaments autorisés sur le marché.

5.6.2. La fonction "TCoeff" :

Pour concevoir des nouvelles molécules dites "drug-like", il faut d'abord évaluer la biodisponibilité orale des molécules candidates conçues et vérifier également la similarité structurelle de celles-ci avec une molécule de référence connue (un

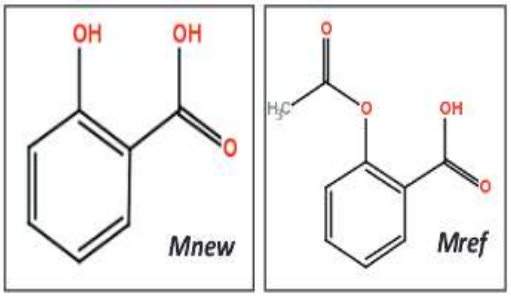
médicament choisi comme référence). Pour vérifier cette similitude, nous devons calculer la fonction "TCoeff"(Tanimoto coefficient) à l'aide de l'outil CDK qui consiste à créer deux fingerprints 2D de type "CACTVS" (PUBCHEM_CACTVS_SUBSKEYS), l'un pour la molécule conçue Mnew et l'autre pour la molécule de référence Mref. Ces fingerprints sont utilisés par PubChem [117] et ils sont représentés par des vecteurs binaires de taille 881. Chaque bit du vecteur indique la présence ou l'absence de certaines espèces chimiques moléculaires. Une description détaillée de la liste complète des bits de fingerprints structurels CACTVS peut être obtenue à partir du site [117].

La valeur de la fonction de similitude TCoeff est toujours comprise entre 0 et 1. La fonction de similitude "TCoeff" pour les deux molécules Mnew et Mref est donnée par la formule suivante:

$$TCoeff(M_{new}, M_{ref}) = \left(\sum_{i=1}^{881} x_{i,M_{new}} * x_{i,M_{ref}}\right) / \left(\sum_{i=1}^{881} x_{i,M_{new}}^2 + \sum_{i=1}^{881} x_{i,M_{ref}}^2 - \sum_{i=1}^{881} x_{i,M_{new}} * x_{i,M_{ref}}\right)$$

Où, $x_{i,M_{new}}$ et $x_{i,M_{ref}}$sont respectivement les i^{ième} bits de la signature (fingerprint) de la molécule conçue M_{new} et de la molécule de référence M_{ref}.

On suppose que la taille de chaque fingerprint est égale à 8, un exemple de ces deux composés et leurs fingerprints ainsi que, le calcul du coefficient de Tanimoto TCoeff(M_{new}, M_{ref}) est illustré par la figure 4.13 ci-dessous.



Fingerprint de *Mnew*	1	1	0	1	1	0	1	0
Fingerprint de *Mref*	1	1	0	1	0	0	0	0

$$TCoeff(Mnew, Mref) = \left(\sum_{i=1}^{8} x_{i,Mnew} * x_{i,Mref}\right) / \left(\sum_{i=1}^{8} x_{i,Mnew}^2 + \sum_{i=1}^{8} x_{i,Mref}^2 - \sum_{i=1}^{8} x_{i,Mnew} * x_{i,Mref}\right) = \frac{3}{5+3-3} = 0.6 \text{ où } 60\%$$

Figure 4.13: Exemple de calcul du coefficient de Tanimoto.

5.7. Affichage de la solution optimale :

L'affichage ce fait par les packages de jar guhautils, la structure optimale est affiché dans un panel.

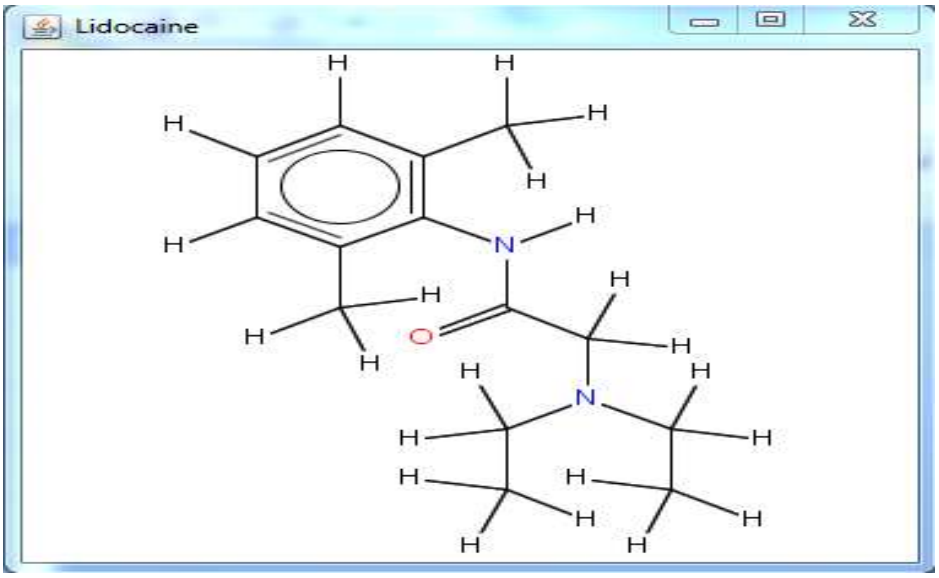


Figure 4.14 : Affichage des structures 2D par guhautils.

6. Description de la dynamique de NSGAII-DND :

La dynamique globale de l'algorithme *NSGAII-DND* proposé pour le problème de DND peut être décrite par l'organigramme donné par la figure 4.15 suivante :

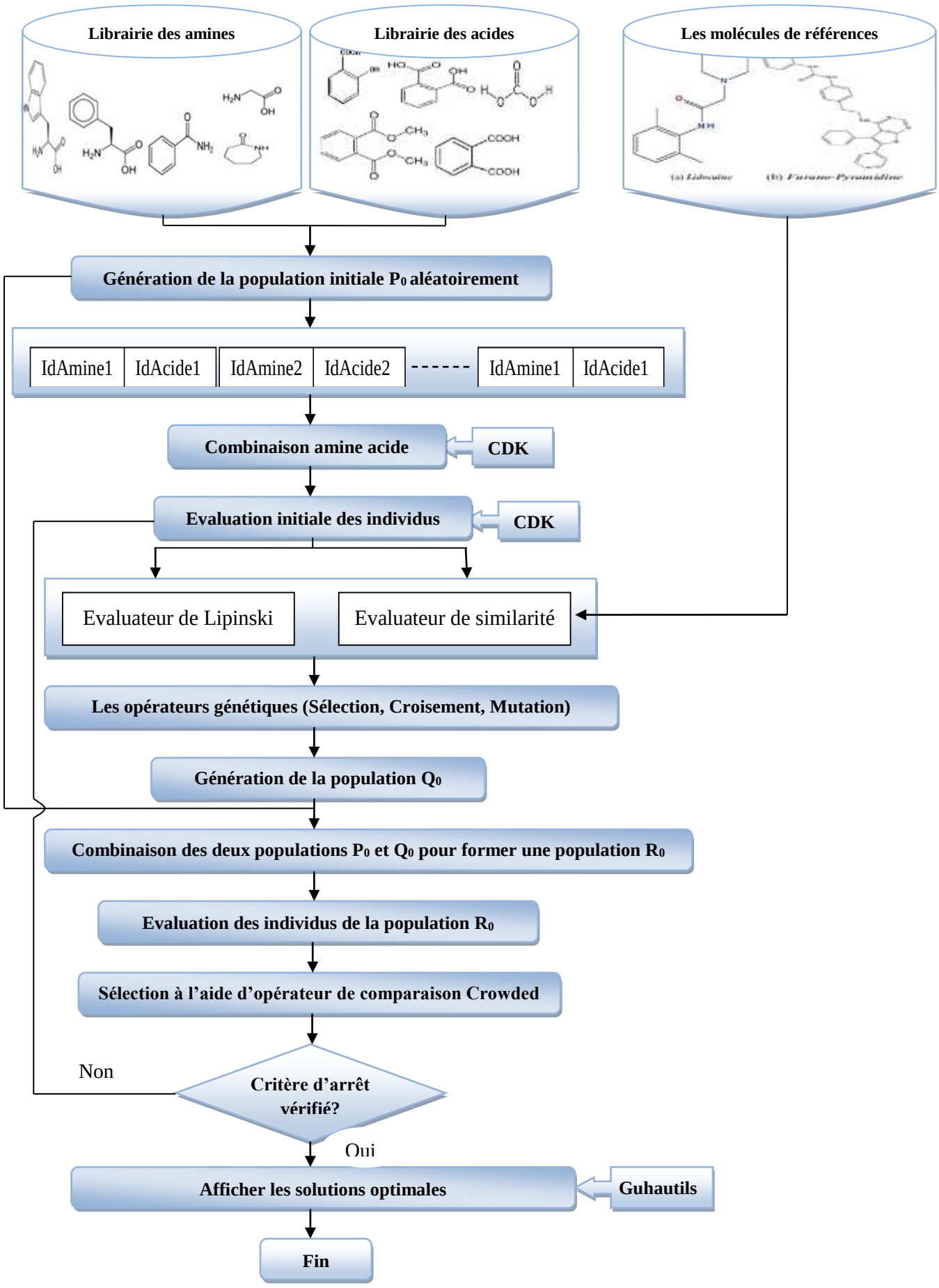


Figure 4.15 : Architecture générale de l'algorithme NSGAII-DND.

7. Le pseudo-code de l'algorithme NSGAII-DND :

Le pseudo-code de notre algorithme *NSGAII-DND* proposé pour résoudre le problème de la conception De novo dans le contexte de l'optimisation multi-objectif, est résumé par les étapes suivantes :

Début:

Etape1: Génération de la population initiale P_0.

idAmine : entier ;
idAcide : entier ;
individu: vecteur d'entier ;
Pour i allant de 1 à taille N faire
    idAmine= choisir aléatoirement l'id d'amine par la formule
      (math.random ()*taille de Library amine);
    idAcide= choisir aléatoirement l'id d'acide par la formule
      (math.random ()*taille de Library acide) ;
    individu= individu [idAmine] [idAcide] ;
Fin pour

Etape2 : Evaluation initiale des individus parents.

Fitness : réel;
a_1, a_2: réel; // les coefficients de somme pondérée
OBA : réel ; // la valeur de la fonction OBA
TCoeff : réel ; // la valeur de la fonction TCoeff
Fitness= a_1*OBA+a_2*TCoeff;
Retourne Fitness;

Etape3 : Les opérateurs génétiques (sélection, croisement et mutation) sur la population initiale P_0.

Etape4 : Génération de la population Q_0.

P_0 : la population initiale
S_0= Sélection (P_0) ;
C_0=Croisement (S_0) ;
R_0=Mutation (C_0) ;
Retourne R_0 ;

Etape5 : Combinaison des deux populations P_0 et Q_0.

P_0 : la population initial
Q_0 : la population enfant.
$R_0 = P_0 \cup Q_0$

Etape6: Evaluation de tous les individus (calculer la valeur de Fitness).

Etape7 : La sélection à l'aide d'opérateur de comparaison Crowded.

ind1, ind2 : Individu;
$i\text{nd1}_{rank}$, $i\text{nd2}_{rank}$:entier;
$\text{ind1}_{distance}$, $\text{ind2}_{distance}$: réel ;
If($i\text{nd1}_{rank} < \text{ind2}_{rank}$ ou ($i\text{nd1}_{rank} = \text{ind2}_{rank}$) et ($\text{ind1}_{distance} > \text{ind2}_{distance}$))

```
    Retourne ind1 ;
Else
    Retourne ind2 ;
EndIf ;
```

Etape 8 : Affichage de la solution optimale.

```
IAtomContainer molecule ;
Molecule= Misc.get2Dcoords (molecule) ;
Renderer2DPanel rendererPanel=new Renderer2DPanel (molecule, 500, 500);
Retourne rendererPanel;
```

8. Conclusion :

Dans ce chapitre, nous avons décrit notre outil *NSGAII-DND* proposé pour résoudre le problème de conception De novo de médicaments dans le contexte d'optimisation multi-objectif. Ce nouvel outil a donc pour but de construire des nouvelles molécules en combinant des fragments appartenant à 2 types de librairies : acides et amines et offrir les meilleures chances de sélectionner des composés ayant des propriétés "drug-like". Notre outil proposé fonctionne en se basant sur cinq composants principaux et en considérant les deux objectifs OBA et Tcoeff pour évaluer la fitness globale comme une agrégation des deux objectifs.

Dans le prochain chapitre, nous allons présenter les résultats obtenus de notre outil.

CHAPITRE 05 :

Les résultats obtenus et étude comparative

Au sommaire de ce chapitre :

1. Introduction.
2. Les données.
3. La configuration des paramètres.
4. Les résultats expérimentaux et étude comparative.
5. Conclusion.

1. Introduction :

Dans ce chapitre, nous présentons les résultats que nous avons obtenus en implémentant l'outil proposé dont le noyau est l'algorithme *NSGAII-DND*. Nous décrivons dans la première section les données utilisées pour réaliser nos tests.

Ensuite, dans la deuxième section nous abordons la configuration des paramètres à ajuster en effectuant plusieurs tests afin de déterminer leur influence sur la qualité des résultats et la performance de l'algorithme proposé.

La dernière section est divisée en trois parties : dans les deux premières parties, nous commençons par présenter les résultats obtenus par l'approche proposée (*NSGAII-DND*) dans notre mémoire, ainsi que ceux obtenus par l'algorithme MOGA_DND proposé dans [31] pour résoudre le même problème, et la troisième partie est consacrée à l'étude comparative de nos résultats avec ceux obtenus à partir de MOGA_DND. Pour réaliser notre prototype, nous avons utilisé la librairie CDK qui nous offre la possibilité de calculer toutes les valeurs des fonctions objectifs et de visualiser les molécules.

2. Les données :

Nos expériences ont été réalisées moyennant un ensemble de données qui est constitué d'une part d'une librairie de fragments moléculaires utilisée pour construire des nouveaux composés chimiques "drug-like" et d'autre part d'une banque de données de référence représentant des molécules de référence.

2.1. Les librairies des fragments :

Dans le cadre de ce travail nous avons utilisé une librairie de 28 fragments de type Acide et une autre librairie de 108 fragments de type Amine. Tous les fragments sont téléchargés au format ".mol".

Les nouvelles molécules "drug-like" sont générées par une concaténation d'un fragment acide avec un fragment amine. Autrement dit, pendant l'étape de l'évaluation de fitness qui se fait grâce au l'outil open source CDK [14], des réactions chimiques se produisent entre des acides carboxyliques (R-COOH) et des amines primaires (R-NH2) ou secondaires(R-NH-R') pour former des nouvelles molécules de type amide (ROCNHR'). Seules les molécules "drug-like" qui satisfont tous les objectifs (OBA et TCoeff) simultanément sont gardés et enregistrées ensuite dans le format de fichier ".mol" ainsi que dans le format d'image ".png". En outre, les molécules "drug-like" peuvent être visualisées en 2D à l'aide de l'applet java "guhautils" [118]. La figure 5.1 ci-dessous présente la structure 2D d'une nouvelle molécule visualisée par cette applet (guhautils.jar) lors de l'implémentation de notre approche.

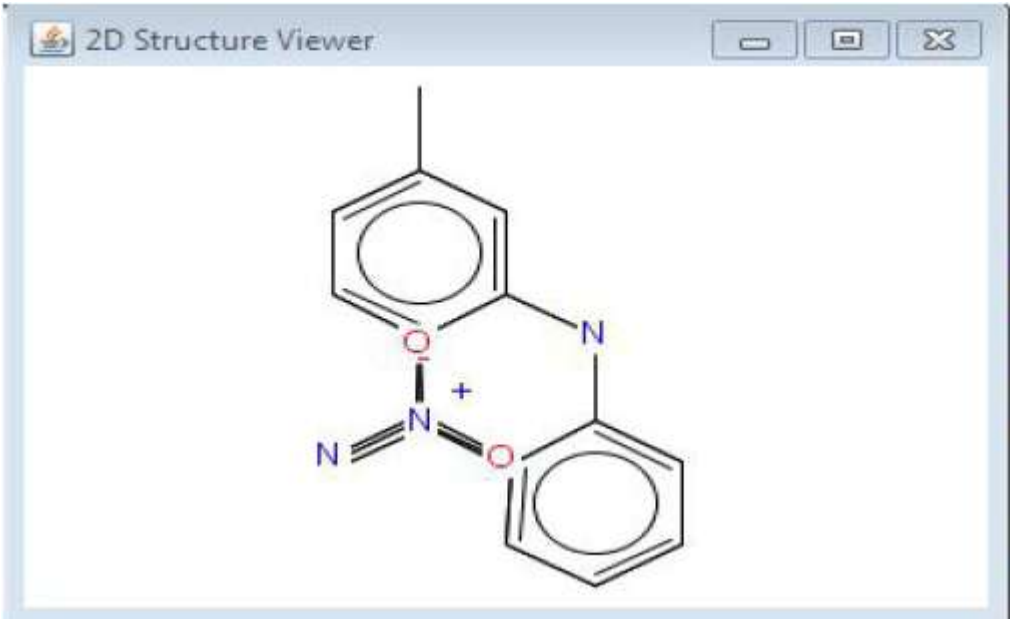


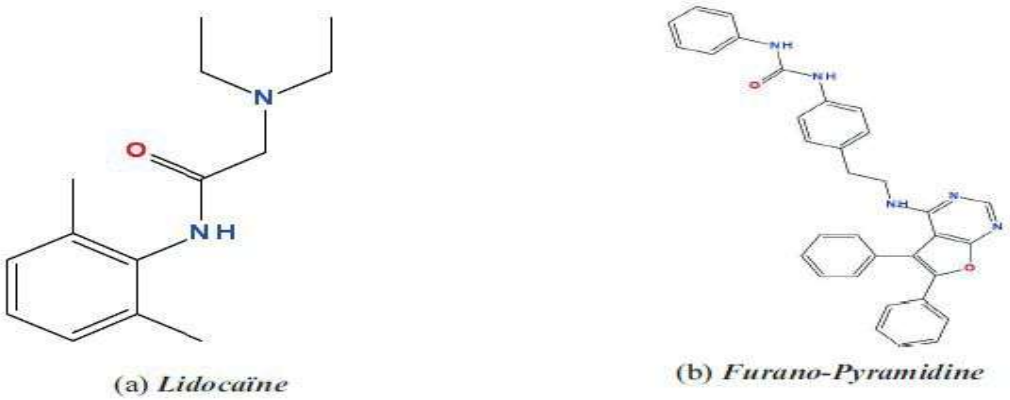
Figure 5.1: Visualisation 2D d'une nouvelle molécule dessinée à l'aide de l'applet guhautils.

2.2. Les molécules de référence :

Pour mesurer l'efficacité et la performance de notre outil et la qualité des solutions qu'il fournit pour résoudre le problème de DND, nous avons effectué deux cas de tests :

- ✓ **Dans le premier cas :** la conception des molécules a été basée sur la molécule de référence de type Amide qui s'appelle "Lidocaïne". Ce médicament est un anesthésique local le plus couramment utilisé en dentisterie.
- ✓ **Dans le deuxième cas :** la conception a été basé sur la molécule de référence "Furano_pyrimidine", c'est une molécule anti-cancer expérimental déclarée dans l'année 2010.

Les deux molécules sont téléchargées en format .mol à partir de la base de données de la chemoinfomatique Chemspider [119], et voilà une figure qui représente la structure 2D de ces deux molécules :



(a) *Lidocaïne* (b) *Furano-Pyramidine*

Figure 5.2: La structure 2D des 2 molécules de référence

3. La configuration des paramètres :

Comme tous les algorithmes évolutionnaire, l'algorithme NSGA-II nécessite la configuration de certains paramètres comme la taille de la population et le nombre d'itération, Ces paramètres permettent d'aider l'algorithme à converger d'une manière efficace et rapide vers la bonne solution. En effet, pour déterminer les bonnes valeurs des paramètres nous avons effectué plusieurs tests basés sur l'amélioration du résultat de la fonction objectif globale (maximisation de la somme pondérée de deux fonction OBA et TCoeff décrites dans le chapitre précédent) associée à chaque molécule conçue. Nous proposons pour cela une interface (figure 5.3) qui permet de saisir tous les informations nécessaire pour l'exécution de l'algorithme proposé, durant les expérimentations fait, l'influence de ces paramètres sur le fonctionnement d'outil proposé a été étudiée.

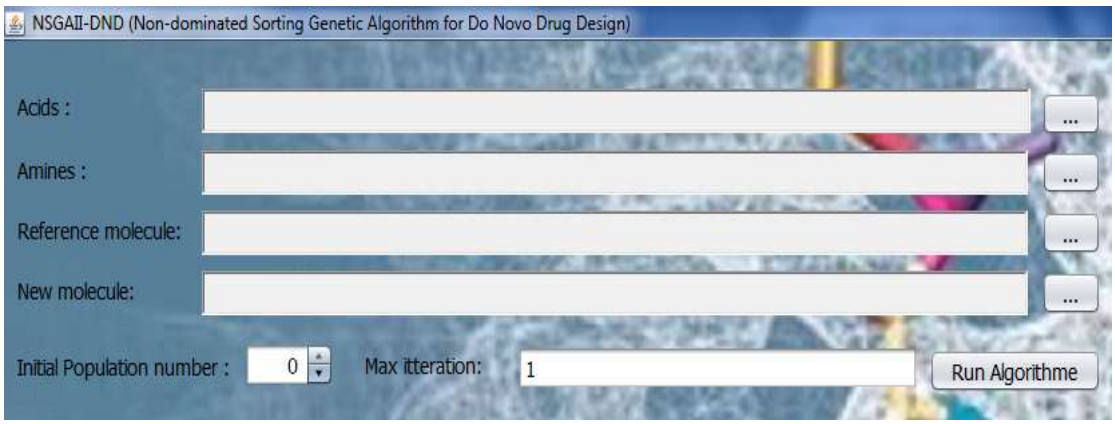


Figure 5.3 : L'interface pour la configuration des paramètres de l'algorithme.

Les champs acids, amines, et Reference molecule représentent le chemin d'emplacement des Library de fragments acide, amine et des molécules de référence, le champ new molecule représente l'emplacement pour sauvegarder la molécule de référence, le champ initial Population number représente la taille de la population et le champ Max itteration le nombre de génération :

Figure 5.4 : Interface pour ajouter l'emplacement de la Library acide.

3.1. L'effet de la taille de population P :

Un des paramètre influant est la taille de la population P. nous avons effectué un certain nombre d'essais avec un nombre de variable de P à savoir P=10, 25, 50, 100 et 150. Notons que le nombre d'itération est fixé à 50, le facteur de mutation a 0.5 et de croisement à 0.5 tout au long des tests. Après plusieurs exécutions de l'algorithme avec ces différents valeurs, les résultats obtenus sont représentés sur le tableau 5.1 et la figure 5.5 suivants :

P/ Exécution	P=10	P=25	P=50	P=100	P=150
Exécution 1	0.9261	0.7562	0.9410	0.9433	0.9435
Exécution 2	0.9184	0.9202	0.8398	0.9422	0.9439
Exécution 3	0.7054	0.9215	0.9404	0.9397	0.9405
Exécution 4	0.9165	0.6860	0.9389	0.9408	0.9449

Tableau 5.1 : Les valeurs de la fonction de fitness obtenue pour différentes taille de population.

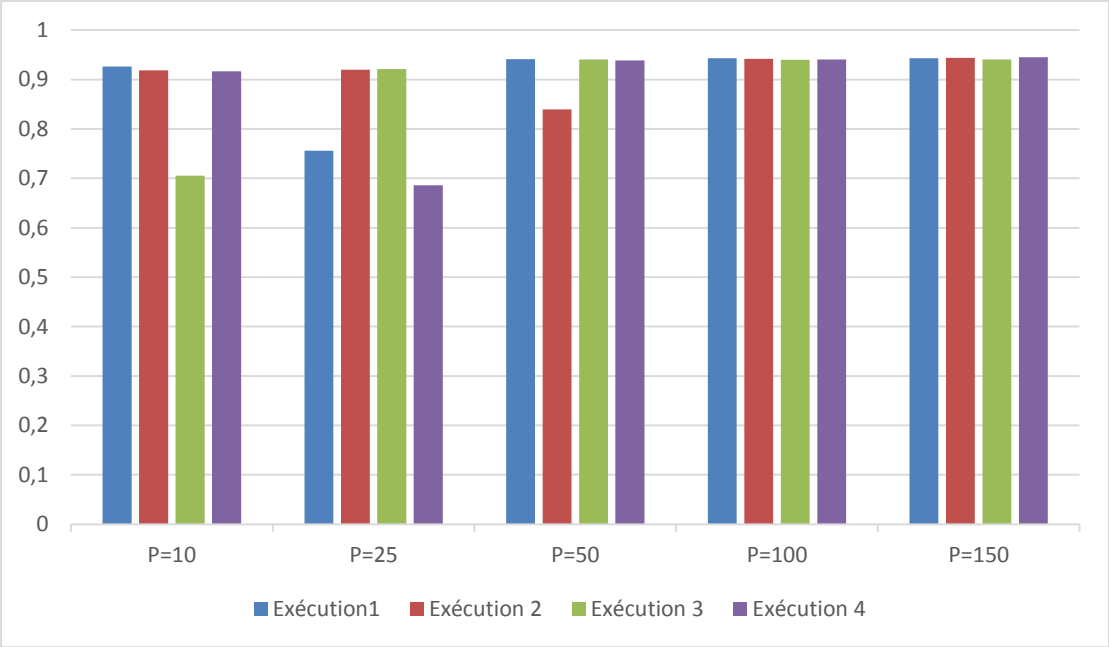


Figure 5.5 : Evaluation de la fonction de fitness pour différentes tailles de population P.

Les résultats obtenus montrent que pour les populations de taille de 10 à 50, les valeurs de la fitness n'est pas optimum, alors que pour une population de 100 et 150 individus la valeur de fitness converge vers la valeur optimum, la valeur optimum est entre 0.9435 et 0.9449. Il en ressort donc que les résultats sont dépendants de la taille de population P et l'augmentation de celle-ci permet d'améliorer la convergence de l'algorithme vers une solution optimale (la meilleure molécule).

3.2. L'effet de nombre d'itération NItér :

Dans le même contexte de problème, nous avons effectué des expériences pour observer l'influence de nombre d'itération NItér. Pour se faire, nous avons utilisé le paramètre fixé à 50 pour la taille de population, 0.5 pour le facteur de croisement et 0.5 pour le facteur de mutation. Nos tests étant effectués pour un nombre d'itération égal à 50, 100 et 150 dont chacun s'exécute 4 fois. Le tableau 5.2 et la figure 5.6 représentent l'évaluation de 4 réplications de la fitness sur ces trois différentes itérations :

NItér/ Exécution	NItér=50	NItér=100	NItér=150
Exécution 1	0.9415	0.9396	0.9420
Exécution 2	0.7589	0.8996	0.9465
Exécution 3	0.8012	0.9445	0.9447
Exécution 4	0.9127	0.7719	0.9433

Tableau 5.2 : Les valeurs de la fonction de fitness obtenue pour chaque nombre d'itération.

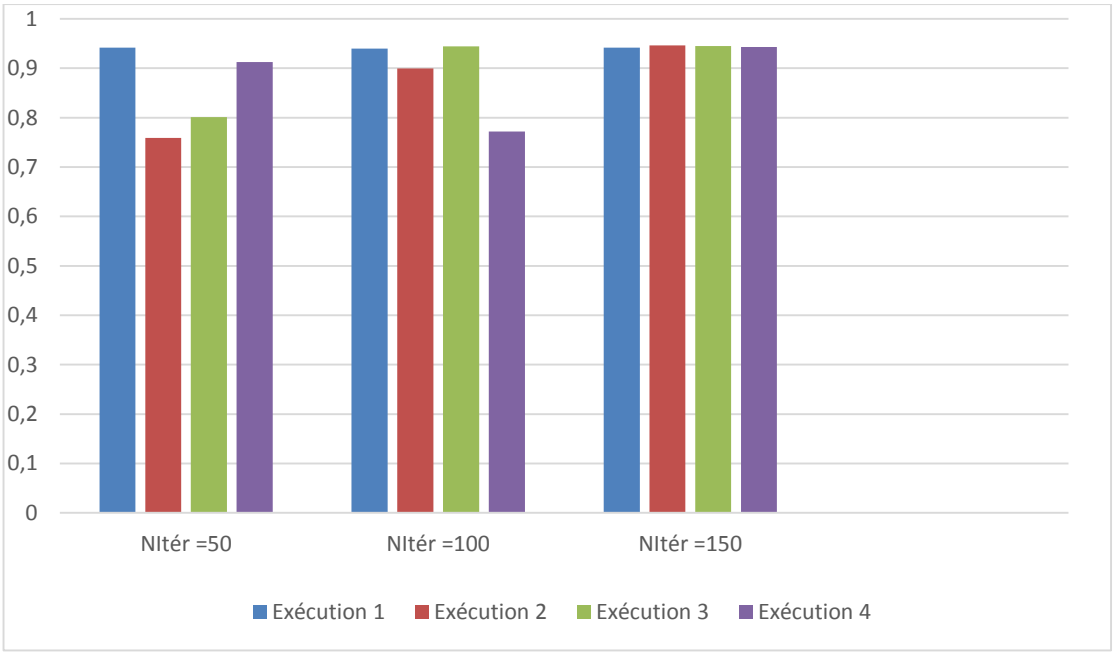


Figure 5.6 : Evaluation de la fonction de fitness pour différentes nombres d'itération.

Selon ce graphique, nous observons qu'il existe une différence nette entre les résultats obtenus avec le nombre d'itérations 50 et 100. Alors que pour le nombre d'itération 150, la valeur de fitness converge vers la valeur optimum et les autres nombre d'itérations (NItér=50 et NItér=100) restent loin de refléter l'optimum, la valeur optimum est 0.9465. On constate donc, que plus le nombre d'itération est élevé

plus on se rapproche du meilleur résultat. Par conséquence, on conclut que le nombre d'itération influe sur la qualité des résultats obtenus.

En conclusion, les résultats expérimentaux sur le problème d'optimisation de DND montrent bien l'existence d'une relation entre les résultats trouvés et les paramètres (taille de la population et le nombre d'itération) de l'algorithme *NSGAII-DND* proposé. On peut conclure que l'algorithme donne des bonne résultats avec :

- ✓ Une taille de population >=50.
- ✓ Un nombre d'itération>= 100.

Ces paramètre seront utilisés dans la section suivante afin d'évaluer et valider les performances de l'algorithme *NSGAII-DND*.

4. Les résultats expérimentaux et l'étude comparative :

Dans cette section, nous allons commencer d'appliquer notre algorithme *NSGAII-DND*, présenter les résultats obtenus et comparer avec un autre algorithme MOGA_DND [31] qui est réalisée par H. Bouanane et A. Benloucif, afin de l'obtention de diplôme de master, centre universitaire de Mila 2015/2016. Les deux algorithmes utilisent la même base de données (les fragments acides et amines), calculent les mêmes objectifs (la biodisponibilité orale OBA et la similarité TCoeff) et sont implémenté en java.

L'objectif est d'étudier le comportement de notre algorithme, afin de justifier son efficacité et sa performance. Pour cela, on va tester l'algorithme en utilisant les deux types de molécules de référence mentionnées ci-dessus à savoir la molécule de référence "Lidocaïne" et la molécule "Furano_pyrimidine" et comparer avec les résultats obtenus dans [31].

4.1. Résultats obtenus par l'algorithme NSGAII-DND :

Notre algorithme est implémenté en java et testé sous Windows 7, sur un processeur core i4, cadencé à 2.20 GHz, avec 4 Go de RAM, pour évaluer la performance de l'algorithme proposé nous effectuons les deux cas de teste suivant :

4.1.1. Cas de test 1 (Lidocaïne) :

Dans le premier cas de test qui se base sur la molécule de référence "Lidocaïne", le comportement de l'algorithme proposé a été suivi en enregistrant les meilleures valeurs de la fonction objectif, ses valeurs vont de 0 à 1. La fonction fitness correspondant à la somme pondérée des deux objectifs OBA et TCoeff, la première fonction objectif OBA est important, car elle mesure sa capacité à générer des molécules satisfaisant la règle de cinq de Lipinski. Ses valeurs vont de 0 à 1. Plus la valeur obtenue de la fonction OBA est élevée, plus la biodisponibilité orale (drug-likeness) de la molécule conçue est excellente. La deuxième fonction objectif TCoeff est une mesure de la similarité structurelle entre la molécule de référence et la nouvelle molécule conçue. Comme la fonction OBA, elle varie de 0 à 1 et les valeurs plus élevées indiquent une très grande similitude moléculaire. La capture écran ci-

dessus donne un exemple sur l'interface de notre outil qui visualise la structure 2D et les valeurs de la fonction de fitness obtenu :

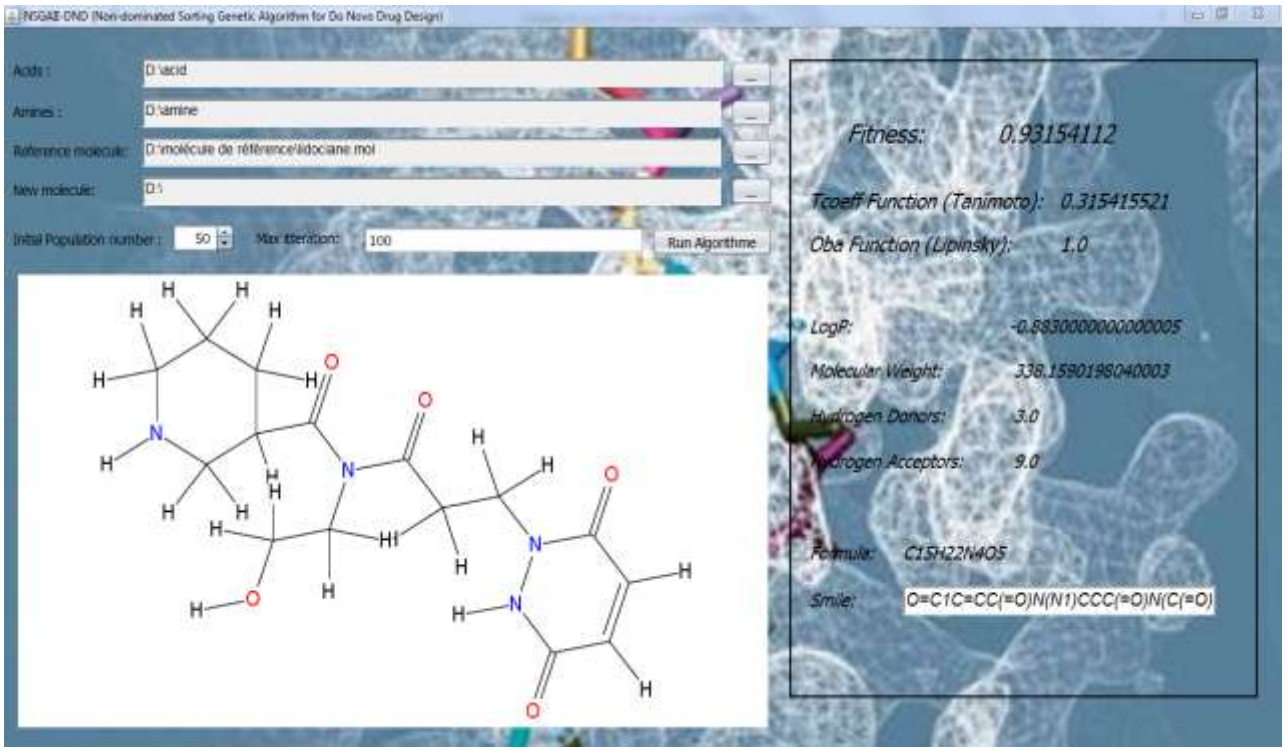


Figure 5.7 : L'interface qui affiche les meilleures solutions.

Cependant, après plusieurs exécutions de l'algorithme pour la molécule "Lidocaïne", des nouvelles et différentes molécules ont été produites pendant les exécutions. Les résultats obtenus sont présentés dans le tableau 5.3 ci-dessous :

New molécule	Fitness	Fonction OBA	Fonction TCoeff
A	0.9324	1.0	0.3246
B	0.9315	1.0	0.3154
C	0.9255	1.0	0.2554
D	0.932	1.0	0.3160
E	0.9322	1.0	0.3203
F	0.9292	1.0	0.2922

Tableau 5.3 : Les résultats obtenus pour le cas de test 1 (Lidocaïne) par l'algorithme NSGAII-DND.

Ces six nouvelles molécules "drug-like" similaires à la molécule de référence "Lidocaïne" sont illustrées dans la figure 5.8 suivante :

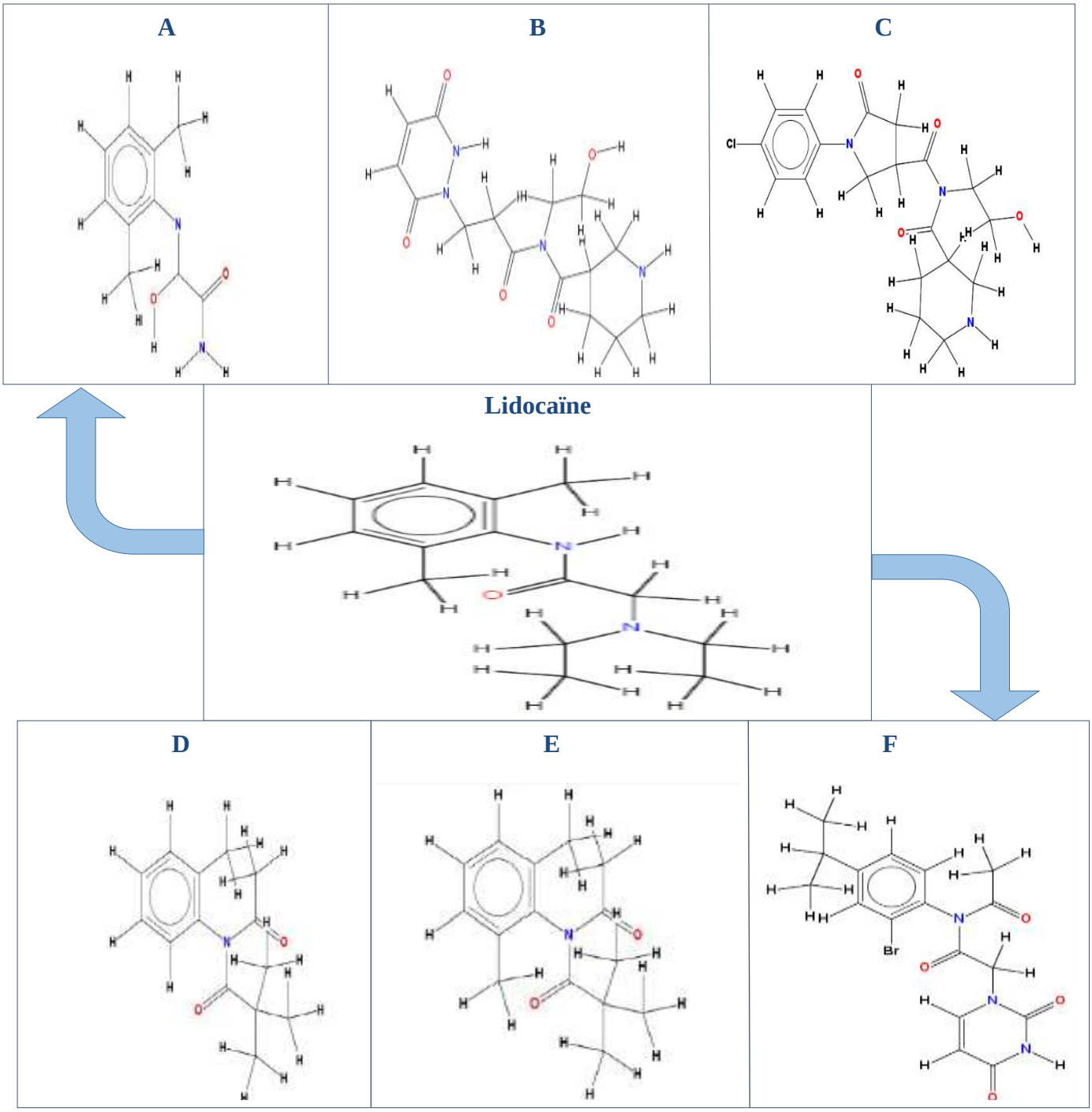


Figure 5.8 : Les structures 2D des meilleures molécules obtenues par l'algorithme NSGAII-DND liées à la molécule de référence "Lidocaïne".

Vu les résultats du tableau 5.3, l'analyse des six molécules en utilisant les règles de biodisponibilité (OBA) de Lipinski montre que tous les composés sont "drug-like". Ainsi que les résultats obtenus par la fonction TCoeff sont loin de 1 (l'optimum= 0.3246) cela est du à la structure des fragments amines et acides choisis qui sont différent à la structure de "Lidocaïne".

En général, les résultats sont proche les uns des autres et proche de l'optimum qui est la molécule A (Fitness= 0.9324, OBA=1.0, TCoeff=0.3246).

4.1.2. Cas de teste 2 (Furano_pyrimidine) :

Dans ce deuxième cas de test, on se focalise sur la molécule de référence "Furano_pyrimidine". Comme le premier cas, l'algorithme *NSGAII-DND* fonctionne en utilisant la fonction de fitness issue de la somme des deux fonctions objectis OBA et TCoeff dans le contexte d'optimisation multi-objectif.

L'interface qui visualise le meilleur résultat obtenu par apport à la molécule de référence "Furano_pyrimidine" se présente dans la figure 5.9 :

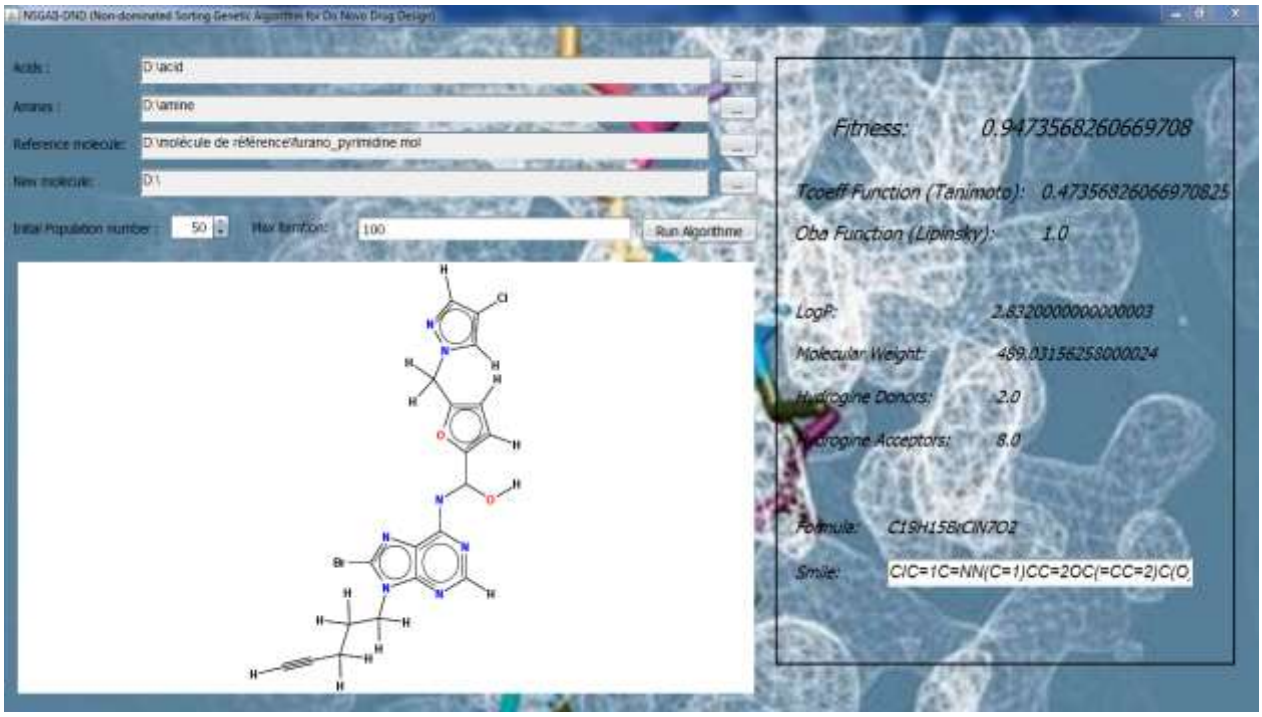


Figure 5.9 : L'interface qui affiche les meilleures solutions par apport à la molécule de référence "Furano_pyrimidine".

Les résultats obtenus après un certain nombre d'exécution sont présenté dans le tableau ci-dessus :

New Molécule	Fitness	Fonction OBA	Fonction TCoeff
A	0.9437	1.0	0.4379
B	0.9442	1.0	0.4422
C	0.94	1.0	0.4001
D	0.9473	1.0	0.4735
E	0.9431	1.0	0.4315
F	0.9425	1.0	0.4250

Tableau 5.4 : Les résultats obtenu pour le cas d'étude 2 (Furano_pyrimidine) par l'algorithme NSGAII-DND.

À partir de tableau, On peut voir que le meilleur résultat obtenu attient la valeur maximale de fitness (fitness= 0.9473, Tanimoto= 0.4735, OBA=1.0) pour la molécule de référence "Furano_pyrimidine", la valeur de la fonction TCoeff est presque moyenne, ainsi que la valeur de la fonction OBA est la plus part de temps vérifié OBA=1.0. Rarement, il ne sera pas vérifié lorsque certain règles de Lipinski ne seront pas vérifiés. Les structures 2D des meilleurs résultats est représenté dans la figure 5.10 :

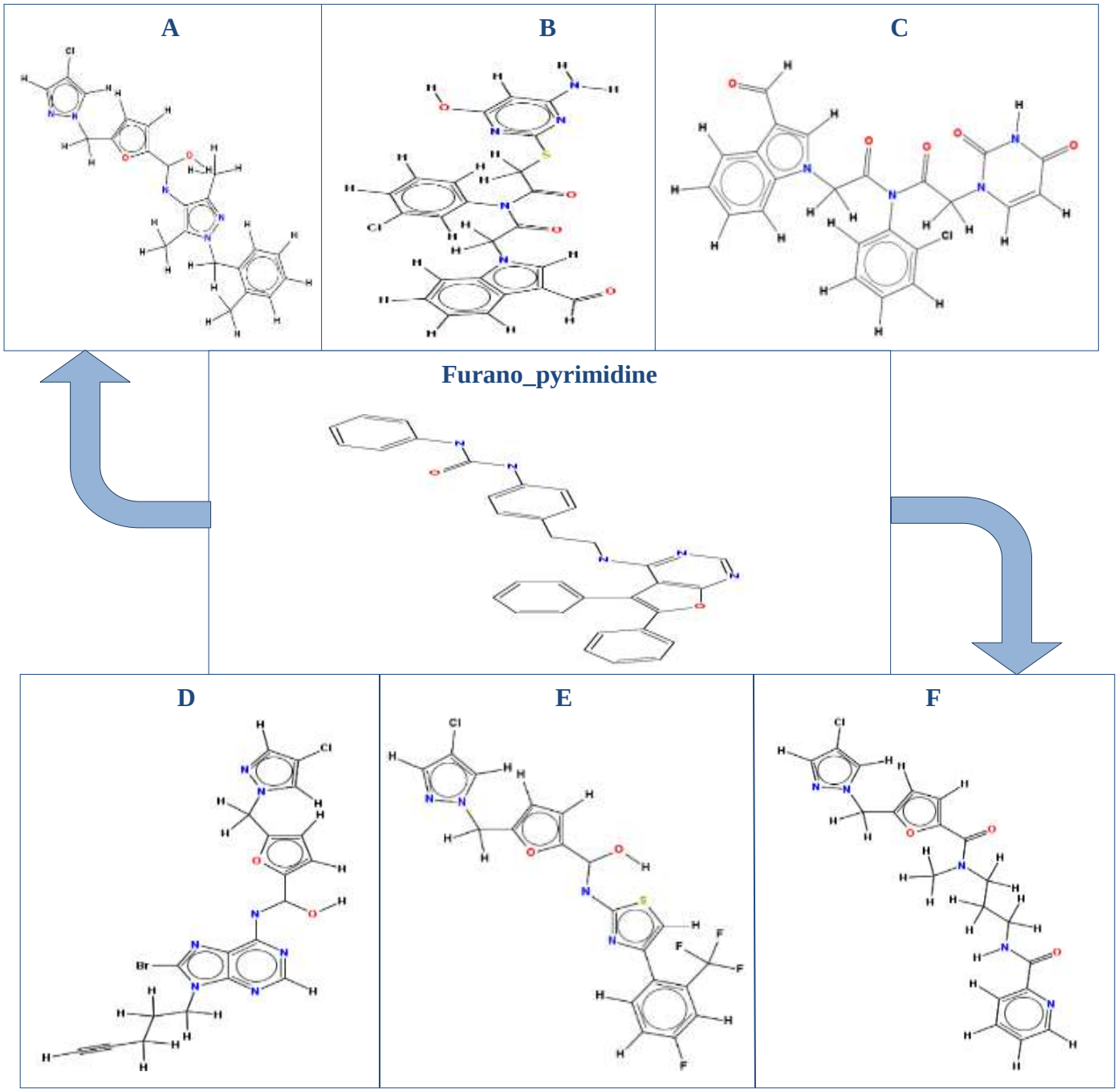


Figure 5.10 : Les structures 2D des meilleures molécules obtenues par l'algorithme NSGAII-DND liées à la molécule de référence " Furano_pyrimidine".

4.2. Résultats obtenus par l'algorithme MOGA_DND :

Comme l'algorithme *NSGAII-DND*, l'algorithme génétique MOGA_DND proposé dans [31] a été soumis également à deux études de cas, tout en gardant les mêmes paramètres.

4.2.1. Cas de test 1 : (Lidocaïne)

Dans cette première étude qui concerne la molécule "Lidocaïne", le comportement de l'algorithme MOGA_DND a été aussi suivi en évaluant la fonction fitness qui est la somme pondérés des deux fonctions objectifs : la foction OBA (les règles de Lipinski) et la fonction TCoeff (Le coefficient de Tanimoto). Les résultats obtenus par cet algorithme sont présentés dans le tableau 5.5 suivant :

Molécules	Fitness	Fonction OBA	Fonction TCoeff
A	0.93541	1.0	0.3541
B	0.92378	1.0	0.2778
C	0.93246	1.0	0.3246
D	0.93203	1.0	0.3203
E	0.931603	1.0	0.31603
F	0.9245	1.0	0.2615

Tableau 5.5 : Les résultats obtenus pour le cas de test 1(*Lidocaïne*) par l'algorithme MOGA_DND.

A partir de tableau 5.5, l'optimum est la molécule A (Fitness=0.93541, OBA=1.0, TCoeff=0.3541), tous les autres résultats vérifient les règles de Lipinski OBA=1.0, et les résultats de fonction TCoeff sont proche à l'optimum.

Les structures 2D des meilleurs résultats de l'algorithme MOGA_DND est dans la figure 5.11 :

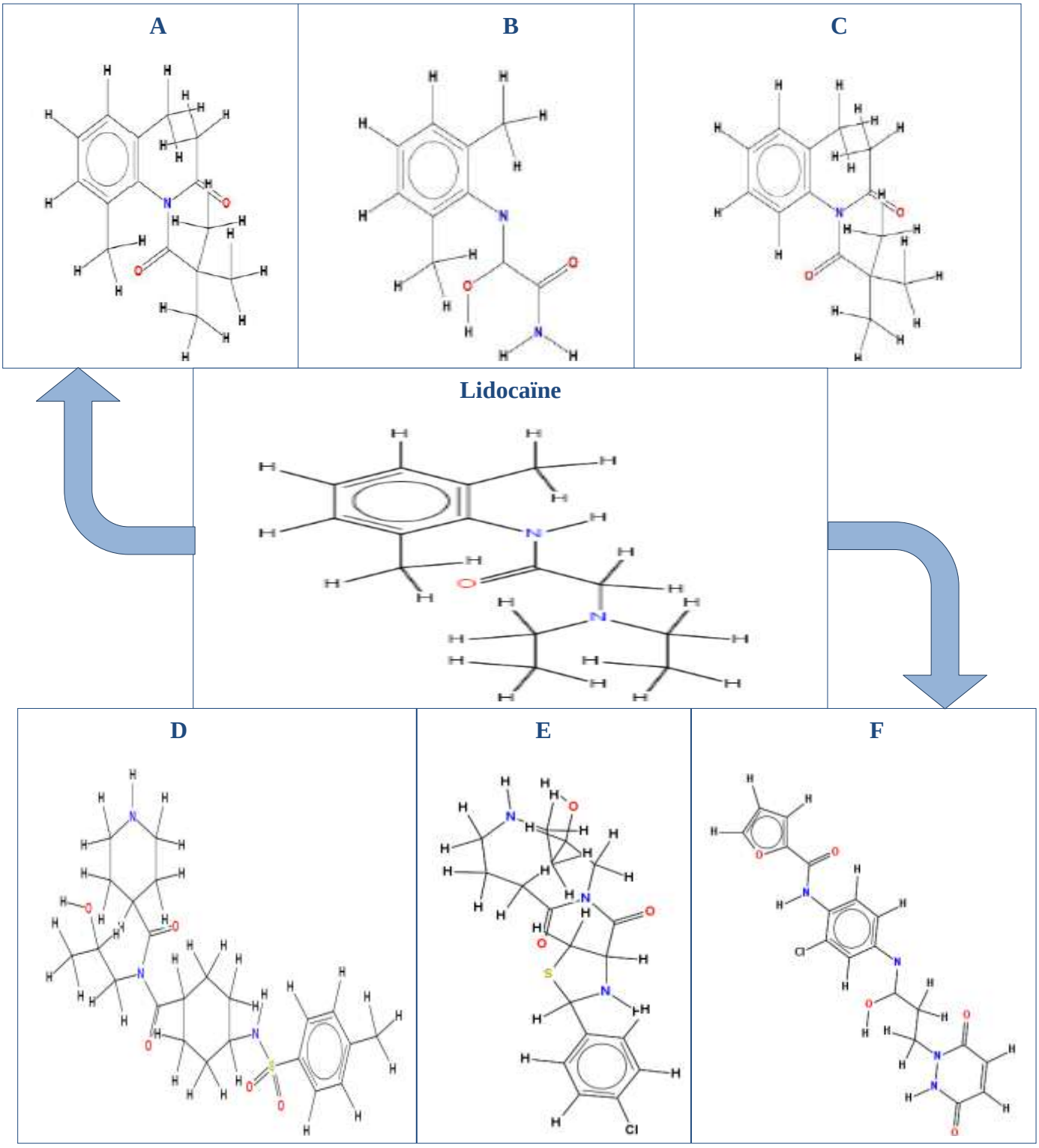


Figure 5.11 : Les structures 2D des meilleures molécules obtenues par l'algorithme MOGA_DND liées à la molécule de référence "Lidocaïne".

4.2.2. Cas de teste 2(Furano_pyrimidine) :

Dans ce deuxième cas de test qui est basé sur la molécule de référence "Furano_pyrimidine", l'évolution des fonctions objectifs par l'algorithme MOGA_DND est illustrée dans le tableau 5.6 suivant :

Molécules	Fitness	Fonction OBA	Fonction TCoeff
A	0.9403	1.0	0.4031
B	0.9394	1.0	0.3949
C	0.9415	1.0	0.4158
D	0.9353	1.0	0.3833
E	0.9358	1.0	0.3835

Tableau 5.6 : Les résultats obtenu pour le cas d'étude 2 (Furano_pyrimidine) par l'algorithme MOGA_DND.

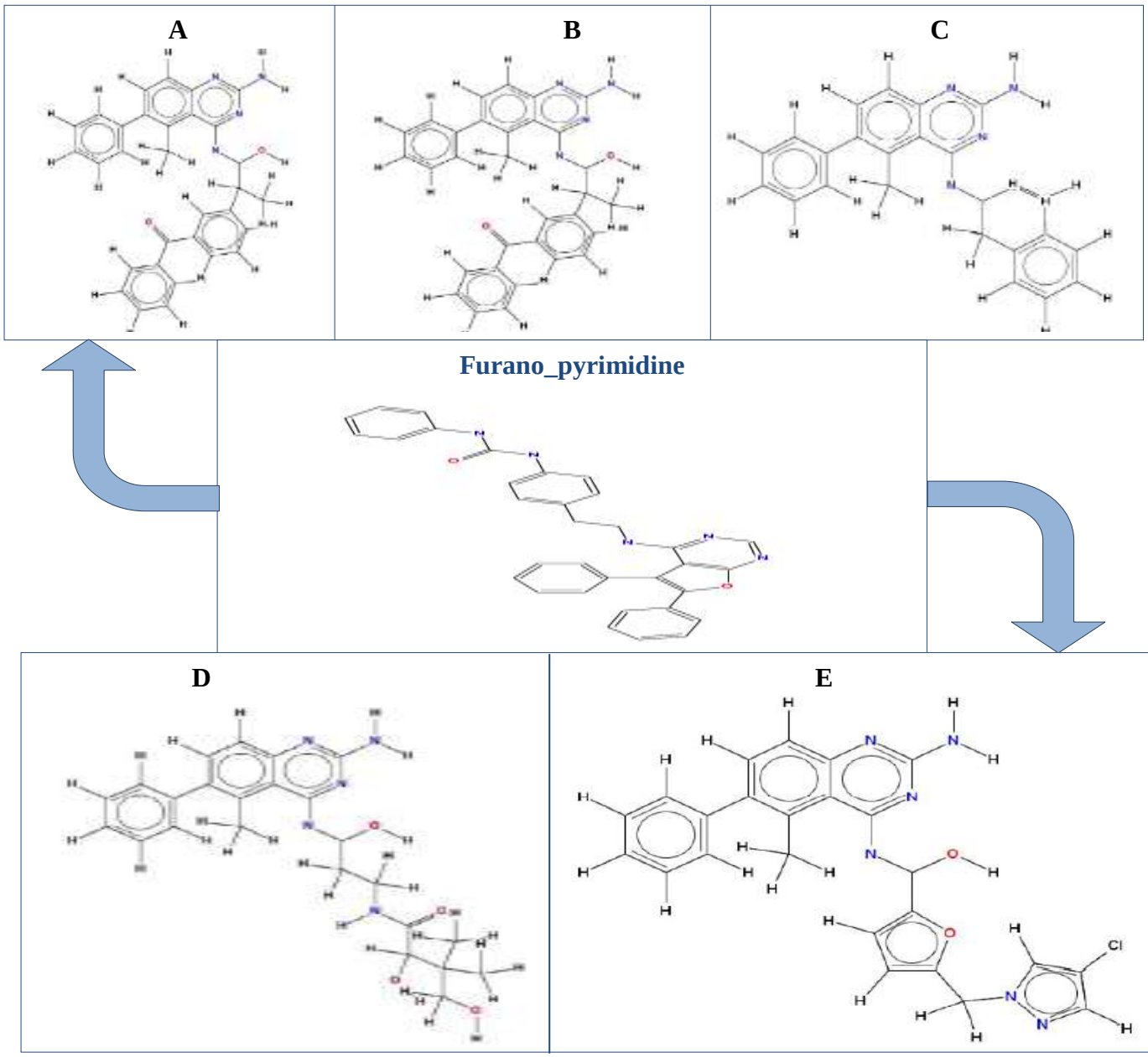


Figure 5.12 : Les structures 2D des meilleures molécules obtenues par l'algorithme MOGA_DND liées à la molécule de référence "Furano_pyrimidine".

4.3. Etude comparative :

4.3.1. Evaluation de Fitness :

Pour procéder à cette étude comparative, plusieurs exécutions de l'algorithme proposé *NSGAII-DND* et l'algorithme *MOGA_DND* ont été effectuées et les valeurs de fintness obtenus sont été enregistrées, ainsi que les valeurs des deux fonctions objectifs : la fonction OBA qui mesure la biodisponibilité orale et la fonction TCoeff qui mesure la similarité avec une molécule de référence, dans notre cas on prend deux molécules de références "Lidocaïne" et " Furano_pyrimidine".

4.3.1.1. Cas de test 1 (Lidocaïne) :

Dans le cas de test 1, pour la molécule de référence "Lidocaïne", les valeurs de Fitness des deux algorithmes : l'algorithme *NSGAII-DND* et l'algorithme MOGA_DND sont représentés dans la figure 5.13 suivante :

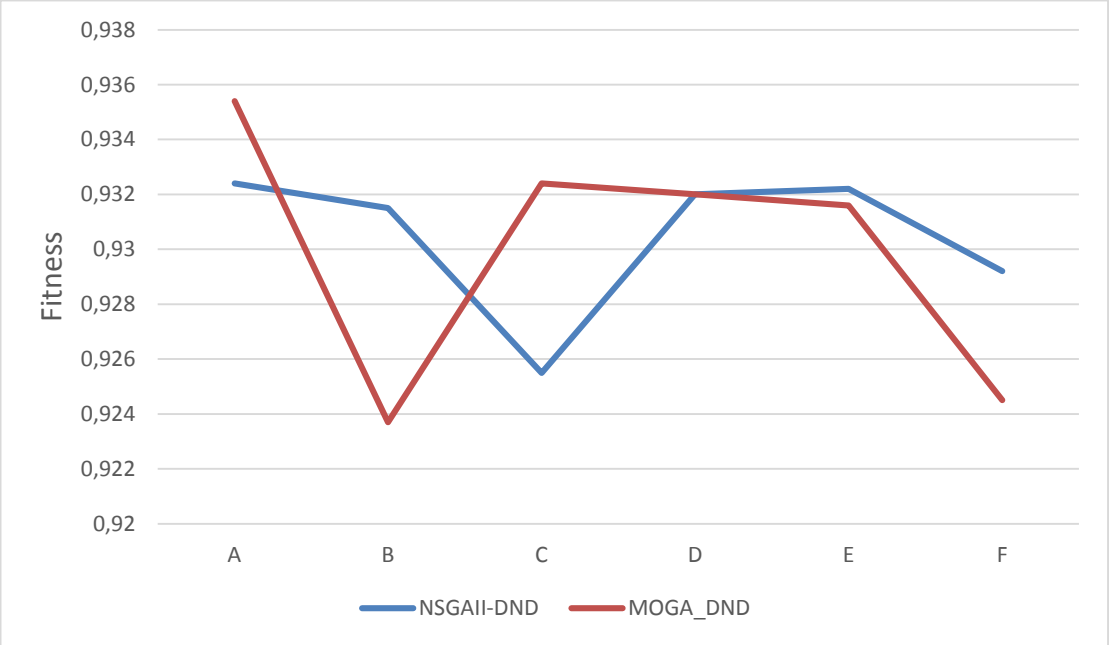


Figure 5.13 : Comparaison des résultats de fitness pour les deux algorithmes : NSGAII-DND et MOGA_DND liées à la molécule de référence "Lidocaïne".

A partir de ce graphique, on remarque que le meilleur résultat est obtenu par l'algorithme MOGA_DND (Fitness=0.9354). Dans les autres résultats, l'algorithme *NSGAII-DND* est meilleur que l'algorithme MOGA_DND dans la stabilité des solutions tel que la différence entre l'optimum et le mauvais résultat est 0.0069 (Fitness de molécule A – Fitness de molécule C = 0.9324 - 0.9255 = 0.0069) pour l'algorithme *NSGAII-DND*, ainsi que ceux pour l'algorithme MOGA_DND est 0.0117 (Fitness de molécule A –Fitness de molécule B = 0.9354 - 0.9237 = 0.0117).

Généralement, il existe des similitudes entre les résultats des deux algorithmes pour la molécule de référence "Lidocaïne".

4.3.1.2. Cas de teste 2 (Furano_pyrimidine) :

Dans le deuxième cas de test, pour la molécule de référence "Furano_pyrimidine ", les valeurs de Fitness des deux algorithmes sont illustrées dans la figure 5.14 suivante :

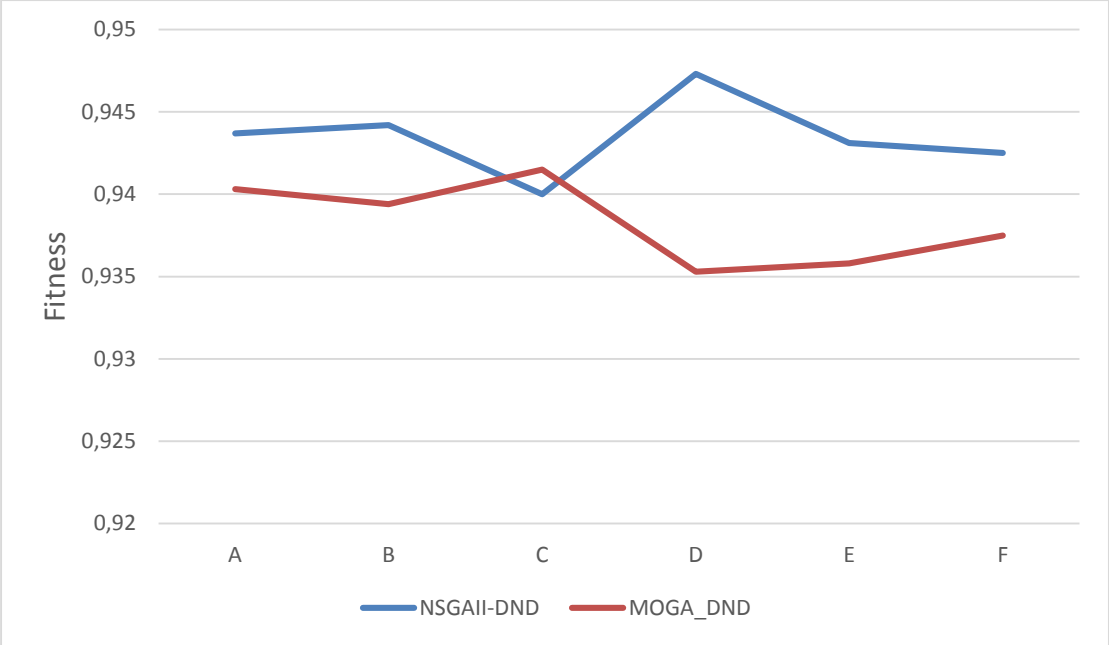


Figure 5.14 : Comparaison des résultats de fitness pour les deux algorithmes : NSGAII-DND et MOGA_DND liées à la molécule de référence "Furano_pyrimidine ".

Selon ce graphique, on remarque que l'algorithme *NSGAII-DND* est meilleur que l'algorithme MOGA_DND dans la plus part des résultats, et le meilleur résultat est obtenu par l'algorithme *NSGAII-DND* Fitness D=0.9473.

En général, l'algorithme *NSGAII-DND* est mieux que l'algorithme MOGA_DND dans la plus parte des résultats surtout pour la molécule de référence "Furano_pyrimidine". Donc, il y a une amélioration dans les résultats, cette amélioration est attendue car l'algorithme NSGAII est basée sur la notion d'élitisme qui garde les meilleurs solutions et les combine avec la population courante au contraire de l'algorithme non élitiste MOGA.

4.3.2. Le temps d'exécution des algorithmes NSGAII-DND et MOGA_DND :

Les deux algorithmes ont presque le même temps d'exécution. Pour l'algorithme *NSGAII-DND*, le temps d'exécution moyenne est environ 15 secondes, et pour l'algorithme MOGA_DND, le temps d'exécution moyenne est 12 secondes.

On remarque aussi que le temps d'exécution pour la molécule de référence "Furano_pyrimidine" (ayant un poids moléculaire élevé) est supérieur que le temps d'exécution pour la molécule de référence "Lidocaïne" (ayant un faible poids moléculaire). Il ressort de ces résultats que la taille des molécules de références influe grandement sur le temps de calcul.

5. Conclusion :

Dans ce chapitre, plusieurs expérimentations ont été conduites dans le but d'évaluer l'approche proposée, nous avons présenté les résultats obtenus par notre algorithme *NSGAII-DND* et l'algorithme MOGA_DND [31], et comparer ces résultats.

L'algorithme *NSGAII-DND* a amélioré les résultats de fitness mais pas le temps d'exécution que l'algorithme MOGA_DND. Dans la future, on peut améliorer de plus les résultats et optimiser le temps d'exécution par d'autre méthode d'optimisation.

Conclusion :

Conclusion générale

Dans ce travail, on a développé un nouvel outil pour la conception de novo des médicaments qu'on a désigné par *NSGAII-DND*. Cet outil comporte un noyau qui consiste en un optimiseur multi-objectif basé sur l'algorithme évolutionnaire NSGAII.

L'algorithme *NSGAII-DND* permet de trouver la bonne combinaison des fragments et donnant lieu à des nouvelles molécules "drug-like", à l'aide des deux évaluateurs : un évaluateur de similarité à des molécules de références et un évaluateur de biodisponibilité pour l'évaluation des molécules produites sur la base des fonctions objectifs choisies à savoir OBA et TCoeff en utilisons un schéma d'agrégation.

L'outil *NSGAII-DND* donnez-nous des bons résultats, surtout pour l'évaluateur de biodisponibilité orale OBA, Cela signifie des molécules "drug-like". Pour le deuxième évaluateur qui est l'évaluateur de similarité à des molécules de références, les résultats sont presque moyens, Cela est dû à la structure des fragments acides et amines qui sont dissemblable aux molécules de références.

Par ailleurs, afin de mesurer l'efficacité et la performance de l'outil *NSGAII-DND* et la qualité des solutions qu'il fournit pour résoudre le problème de DND, nous avons conduit une étude comparative des résultats obtenus par notre approche avec ceux obtenus par l'algorithme MOGA_DND [31] en effectuant deux cas de test, le premier cas consiste à produire des molécules similaires à la molécules de référence "Lidocaine", alors que le deuxième cas, considère la molécule "Furano_pyrimidine" comme molécule de référence.

Une amélioration des résultats a été marquée aussi lors d'exécution de l'outil proposé par apport à l'outil MOGA_DND [31] dans les deux cas de test, mais le temps d'exécution reste le même pour les deux algorithmes.

Les travaux future oriente vers l'utilisation de parallélisassions afin d'améliorer le temps d'exécution, d'améliorer la valeur de similarité soit par la recherche des bons fragments afin d'améliorer la diversification ou par l'amélioration de la qualité des molécules de références, et d'appliquer des algorithmes plus efficace et innovant pour optimiser de plus le temps d'exécution.

Les références

[1] J. Gasteiger, T. Engel (Eds), "**Chemoinformatics: A Textbook**", John Wiley & Sons, 2004.

[2] C. Shekhar, "**In silico pharmacology: computer-aided methods could transform drug development**", Chem. Biol., vol. 15, pp. 413-414, 2008.

[3] R. Ashok Ra, "**A recent technology in drug discovery and development**", Int. J. Innovat. Drug. Discov., vol. 2, pp. 22-39, 2012.

[4] A. Oduguwa, A. Tiwari, R. Roy, C. Bessant, "**An overview of soft computing techniques used in the drug discovery process**", Applied Soft Computing Technologies: The Challenge of Complexity, Springer, Berlin, vol. 34, pp. 465-480, 2006.

[5] H.J. Huang, H.W. Yu, C.Y. Chen et al., "**Current developments of computer-aided drug design**", J. TAIWAN. INST. CHEM. E., vol. 41, pp. 623-635, 2010.

[6] R. Vasundhara Devi, S. Siva Sathya, and M. Selvaraj Coumar, "**Evolutionary algorithms for de novo drug design**", Appl. Soft. Comput., vol. 27, pp. 543-552, 2015.

[7] G. Schneider and U. Fechner, "**Computer-based de novo design of drug-like molecules**", J. Nat. Rev. Drug, vol. 4, pp. 649-663, August 2005.

[8] J.P. Hughes, S.B. Rees Kalindjian and K .L. Philpott, "**Principes of early drug discovery**", Br J . Pharmacol, vol. 162, pp. 1239-1249, 2011.

[9] G. Allaire, "**Analyse numérique et optimisation: une introduction à la modélisation**", Livre, novembre 2007.

[10] E.G. Talbi et al., "**New genetic approach for multicriteria optimization problems: Application to the flow-shop**", Evolutionary Multi-criterion Optimization (EMO)," vol. 1993, pp. 416-428, 2001.

[11] E.G. Talbi, "**Métaheuristiques pour l'optimisation combinatoire multi objectif: Etat de l'art**", CENT, vol. 33, pp. 98-757, 1999.

[12] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willig hagen, "**The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics**", J. Chem. Inf. Comput. Sci., vol. 43, pp. 493-500, 2003. [http://sourceforge.net/projects/cdk/]

[13]http://www.ecole.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO/conteneur.php?page=pageb&ban=1

[14] K. Brown, "**Annual Reports in Medicinal Chemistry**", 33, 375-384, 1998.

[15] G. Paris, "**Meeting of the American Chemical Society**", août 1999.

[16] J.Gasteiger, ange T. (Eds.), "**Chemoinformatics – A Textbook**", Wiley-VCH, Weinheim, 2003.

[17] A. Bhalerao et al, "**Chemoinformatics: The Application of Informatics Methods to Solve Chemical Problems**", Research Journal of Pharmaceutical, Biological and Chemical Sciences, vol.4, pp. 475-499, 2013.

[18]https://teststfff972.wordpress.com/2015/03/31/drug-design-technique-de-modelisation-moleculaire-in-silico-innovante-pour -la-recherche-therapeutique/

[19] L. HOFFER, "**Développement et validation du logiciel S4MPLE. Application au docking moléculaire et à l'optimisation de fragments assistée par ordinateur dans le cadre du Fragment-Based Drug Design**", Thèse pour obtenir le grade de Docteur de l'Université de Strasbourg, 2013.

[20] www.labunix.uqam.ca/~makarenv/BIF7002/Sirois_2008.pdf

[21] rxcinternational.com/different-types

[22] https://mpkb.org/home/patients/assessing_literature/in_vitro_studies

[23] A. Mahdadi, "**IDE-DD: Un outil d'aide à la conception De novo de médicaments en chimioinformatique à base d'évolution différentielle multiobjectif en nombres entiers**", Mémoire en vue de l'obtention du diplôme de Magister en Informatique, Université Constantine 2 - Abdelhamid Mehri, 2015.

[24] https://www.info2.uqam.ca

[25] LI et AP, "**Screening for human ADME/Tox drug properties in drug discovery**", Drug Discovery Totay, vol. 6, pp. 357-366, 2001.

[26] F. Louacheni, "**Développement d'un portail web pour le criblage virtuel sur la grille de calcul**", Mémoire de fin d'études pour l'obtention du diplôme de Master II Informatique, Institut de la Francophonie pour l'Informatique, 2014.

[27]https://www.google.dz/search?q=les+types+de+drug+design&tbm=isch&tbs=rimg:CRaN7lmvi8-0IjgYODF8X-g8XKuqKESmVTXsCoPPyVkkJuTqMk39-6g7eTPLATpuThUafV8CnENPrCfYbql4Nx8iwyoSCRg4MXxf6DxcEU_1sOkMiUZDQKhIJq6ooRKZVNewRFoXxqECFtUMqEgkKg8_1JWSQm5BEdJ4N4xUkBsyoSCeoyTf37qDt5EUK55ztxvv-CKhIJM8sBOm5OFRoRIXV27Q0qwfkqEgl9XwKcQ0-sJxEtzQpu0wq1_1ioSCdhuqXg3HyLDEewH_1wnDHqvT&tbo=u#imgrc=XI_V4lLvgh9YnM%3A

[28] A.Saadi, A. Cowe, "**Modélisation moléculaire 3D et criblage virtuel**", Séminaire interdisciplinaire de bioinformatique, février 2007.

[29] J.W. Davies, M. Glick and J.L. Jenkins, "**Streamlining lead discovery by aligning in silico and high-throughput screening**", Curr.Opin.Chem.Biol., vol. 10, pp. 343-351, 2006.

[30] http://lausanne.isb-sib.ch/~agrosdid/files/pharmd_thesis.pdf

[31] H. Bouanane, A. Benloucif, "**Approche pour la conception de médicaments assistée par ordinateur**", mémoire pour l'obtention de diplôme de master, Centre Universitaire Abd elhafid Boussouf Mila, 2016.

[32] http://fr.slideshare.net/prasanthperceptron/de-novo-drug-design

[33] C.Oksel, X.Z.Wang, "**Quantitative Structure-Activity Relationship (QSAR) models**", University of Leeds, A pan-European Infrastructure for Quality in Nanomaterials Safety Testing, 2013.

[34] Steinbeck, Christoph et al, "**Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics**", Current Pharmaceutical Design vol. 12, pp. 2111-2120, June 2006.

[35] V. Khanna, "**Molecular similarity and diversity analysis of bioactive small molecules using chemoinformatics approaches**", Doctoral thesis, Department of Chemistry and Biomolecular Sciences Macquarie University Sydney, Australia, 2011.

[36] P. Willett, J.M. Barnard and G.M. Downs, "**Chemical Similarity Searching**", J.Chem. Inf. Comput. Sci., vol. 38, pp. 983-996, 1998

[37] J.L. Faulon and A. Bender, "**Handbook of chemoinformatics algorithms**", Chapman & Hall/CRC mathematical and computational biology series, QD39.3.E46H357, p. 542, 2010.

[38] A. Varnek and A. Tropsha (Eds), "**Chemoinformatics Approaches to Virtual Screening**", Royal Society of Chemistry, 2008.

[39] M. Ben Amar, Léonard et Louis, "**Les psychotropes : pharmacologie et toxicomanie**", Livre, chapitre3, p58, 2002.

[40] S. Mahdi, "**Optimisation Multiobjectif Par Un nouveau Schéma De Coopération Méta/Exacte**", Mémoire de Magister, Université Mentouri de Constantine, 2005.

[41] Y. Collette et P. Siarry, "**Optimisation multiobjectif**", Editions Eyrolles, 2002.

[42] S. Meshoul, H. AlBaity and A. Kaban, "**On Extending Quantum Behaved Particle Swarm Optimization to MultiObjective Context**", In IEEE World Congress on Computational Intelligence June, 10-15, 2012 - Brisbane, Australia.

[43] A. Osyczka, "**Multicriteria optimization for engineering design**", In J. S. Gero, editor, Design Optimization, pp. 193-227, 1985.

[44] J. Dipama, "**Optimisation multiobjectif des systèmes énergétiques**", Thèse de doctorat, Université de Montréal, 2010.

[45] C. Dhaenens-Flipo , "**Optimisation Combinatoire Multi-Objectif: Apport des Methodes Cooperatives et Contribution a l'Extraction de Connaissances**" , Thèse pour obtenir le grade de Habilitation a Diriger des Recherches de l'U.S.T.L. Discipline : Informatique, université de Lille ,2005

[46] D.Francisci, "**Algorithmes évolutionnaires et optimisation multi-objectifs en data mining** ", Projet MECOSI, Rapport de recherche, université nice, mars 2002.

[47] Cohon, J. L. "**Multiobjective Programming and Planning**", New York : Academic Press, (1978)

[48] T. Gräbener, A. Berro, "**Optimisation multiobjectif discrète par propagation de contraintes**", IRIT — Université de Toulouse, 2008.

[49] N.Jozefowiez, "**Optimisation combinatoire multi-objectif : des méthodes aux problèmes, de la Terre à (presque) la Lune**", préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Institut National Polytechnique de Toulouse, 2013

[50]http://www.gauvainmarquet.fr/wpcontent/uploads/2014/04/rapport_projet_dolphi n. pdf.

[51] J.DRÉO, "**Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical**", Thèse de doctorat en sciences, Université Paris 12 – Val de Marne, 2004

[52] R. FabregatEnglish, "**Multi-Objective Optimization in Computer Networks Using Metaheuristics**", Book, ISBN: 0849380847,468 p, 2007.

[53] M. Hafid Zidani, "**Représentation de solution en optimisation continue, multi-objectif et applications**", thèse de doctorat en cotutelle, université mohamed 6 – Agdal, Rabat- Maroc, octobre 2013

[54] A. Berro, "**Optimisation multiobjectif et stratégies d'évolution en environnement dynamique**", thèse de doctorat, université de Toulouse, 2001.

[55] S.Luciano AVILA, "**Optimisation multiobjectif et analyse de sensibilité appliquées à la conception de dispositifs**", thèse pour obtenir le grade de DOCTEUR, L'ECOLE CENTRALE DE LYON, 2006

[56] http://ufrsciencestech.u-bourgogne.fr/m2bdia/OIA/Dipanda/AG_Master2.pdf

[57] S.Amédée, R.Francois-Gérard , "**Algorithmes génétiques**", TE de fin d'année Tutorat de Mr Philippe Audebaud, 2004.

[58] R. STryczek and B. PyTlAk, "**Multi-objective optimization with adjusted PSO method on example of cutting process of hardened 18crMo4 steel**", eksploatacja/ Niezawodnosc –Maintenance and reliability, vol. 16, pp. 236-245, 2014.

[59] Z. Michalewicz, "**Genetic Algorithms + Data Structures = Evolution Programs**", Springer, 1996.

[60] T. Bäck, "**Evolutionary algorithms in theory and practice**," Oxfor d University Press, 1995.

[61] A. Gherboudj, "**Méthodes de résolution de problèmes difficiles académiques**", Thèse de doctorat, Université de Constantine2, Algérie, 2013.

[62] http://khayyam.developpez.com/articles/algo/genetic/

[63] http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/node5.html

[64] V. Pareto, "**Cours d'économie politique**", Rouge, 1896.

[65] C. A. Nicolaou, N. Brown & C. S. Pattichis, "**Molecular optimization using computational multi-objective methods**", Current Opinion in Drug Discovery & Development 2007 10(3):316-324© The Thomson Corporation ISSN 1367-6733.

[66] Lusher, S.J. et al. "**A molecular informatics view on best practice in multi-parameter compound optimization**", Drug Discov. Today 16, 555–568, 2011.

[67] C.A. Nicolaou, N.Brown, "**Multi-objective optimization methods in drug design**", Drug Discovery Today: Technologies, vol.10, pp. 427-435, 2013

[68] S. Janson et al., "**Molecular docking with multi-objective Particle Swarm Optimization**", Applied Soft Computing vol. 8, pp. 666–675, 2008.

[69] G. Jones, P. Willett, R.C. Glen., A.R. Leach. and R.Taylor, "**Development and validation of a genetic algorithm for flexible docking**", J. Mol. Biol., vol. 267, pp. 727-748, 1997.

[70] K. et al., "**A fast elitist multi-objective genetic algorithm: NSGA-II**", IEEE Trans. Evol. Comput., vol. 6, pp. 182–197, 2000.

[71] J-C. Boisson, et al., "**Parallel multi-objective algorithms for the molecular docking problem**", arX@@iv:0811.0514 [q-bio.QM], 2008.

[72] E. Zitzler and S. Kunzli, "**Indicator-based selection in multiobjective search**", Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Springer, pp. 832-842, 2004.

[73] H.Li et al., "**An effective docking strategy for virtual screening based on multi-objective optimization algorithm**", BMC Bioinform, vol. 10, pp. 58, 2009.

[74] T.J. Ewing et al., "**DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases**", J. Comput. Aided Mol. Des., vol. 15, pp. 411-428, 2001.

[75] SJ. Weiner, PA. Kollman, DT. Nguyen and DA. Case, "**A New force filed for molecular mechanical simulation of nucleic acids and proteins**", JCompChem, vol. 7, pp. 230–252, 1986.

[76] Lennard-Jones and J. E. Cohesion, "**Proceedings of the Physical Society**", vol. 43, pp. 461-482, 1931.

[77] Janson S, Merkle D, "**A new multi-objective particle swarm optimization algorithm using clustering applied to automated docking**", In: Hybrid Metaheuristics Second International Workshop, HM 2005, Barcelona, Spain, August 29-30, 2005, Proceedings. Blesa MJ Blum C, Roli A, Sampels M et al (Eds), Springer-Verlag, Berlin, Germany, 128-141, 2005.

[78] Ray T, Liew KM, "**A swarm metaphor for multiobjective design optimization**", Engineering Optimization, 34(2):141-153, 2002.

[79] C.A. Nicolaou, et al., "**De novo drug design using multi-objective evolutionary graphs**", J. Chem. Inf. Model., vol. 49, pp. 295–307, 2009.

[80] C.A. Nicolaou et Al, "**Multi-Objective Optimization Methods in De Novo Drug Design**", Mini-Reviews in Medicinal Chemistry, vol. 12, p. 5, 2012.

[81] N. Brown et al. "**Graph-based genetic algorithm and its application to the multi-objective evolution of median molecules**", J. Chem. Inf. Model. vol. 44, pp. 1079-1087, 2004.

[82] J.W. Kruisselbrink et al, "**Enhancing search space diversity in multiobjective Evolutionary Drug Molecule Design using Niching**", ACM 978-1-60558-325-9/09/07, 2009.

[83] S. Ekins, J.D. Honeycuttand and J.T. Metz, "**Evolving molecules using multi-objective optimization: applying to ADME/Tox**", J. Drug. Discov. Today., vol. 15, pp. 11-12, June 2010.

[84] E. Van der Horst et al., "**Multi-objective evolutionary design of adenosine receptor ligands**", J. Chem. Inf. Model., vol. 52, pp. 1713-1721, 2012.

[85] S. Sengupta, S. Bandyopadhyay and S. Member, "**De novo design of potential RecA inhibitors using multi-objective optimization**", IEEE/ACM Trans. Comput. Biol. Bioinform, vol. 9, pp. 1139-1154, 2012.

[86] Gillet, Willett et al., "**Selecting combinatorial libraries to optimize diversity and physical properties**", 1999.

[87] V.J. Gillet et al., "**Combinatorial library design using a multiobjective genetic algorithm**", J. Chem.Inf. Comput.Sci., vol. 42, pp. 375-385, 2002.

[88] P. Sharma et al., "**A scalable approach to combinatorial library design**", In Chemical Library Design, Methods in Molecular Biology, Humana Press Inc, vol. 685, pp. 71-89, 2011.

[89] C.A. Nicolaou and C. Kannas, "**Molecular library design using multi-objective optimization methods**", In Chemical Library Design, Methods in Molecular Biology (Zhou, J., ed.), pp. 53-69, 2011.

[90] T. Meinl, et al., "**Maximum-score diversity selection for early drug discovery**", J. Chem. Inf. Comput. Sci., vol. 51, pp. 237-247, 2011.

[91] T.T. Wager et al., "**Moving beyond rules: the development of a central nervous system multi-parameter optimization (CNS MPO) approach to enable alignment of druglike properties**", ACS Chem. Neurosci., vol. 1, pp. 435–449, 2010.

[92] Nicolotti, O. et al. "**Multiobjective optimization in quantitative structure–activity relationships: deriving accurate and interpretable QSARs**", J. Med. Chem. 45, 5069–5080, 2002.

[93] Nicolotti, O. et al. "**Improving quantitative structure–activity relationships through multi-objective optimization**", J. Chem. Inf. Model. 49, 2290–2302, 2009.

[94] Manoharan, P. et al. "**Rationalizing fragment based drug discovery for BACE1: insights from FB-QSAR, FB-QSSR, multi-objective (MO-QSPR) and MIF studies**", J. Comput. Aided Mol. Des. 24, 843–864, 2010.

[95] Stockfisch, "**TP: Partially unified multiple property recursive partitioning (PUMP-RP): A new method for predicting and understanding drug selectivity**", J Chem Inf Comput Sci (2003) 43(5):1608-1613.

[96] Rao SN, Stockfisch, "**TP: Partially unified multiple property recursive partitioning (PUMP-RP) analyses of cyclooxygenase (COX) inhibitors**", J Chem Inf Comput Sci (2003) 43(5):1614-1622.

[97] G. Landrum, "**RDKit: cheminformatics and machine learning software. rdkit.org**", 2013.

[98] NM. O'boyle et al., "**Open Babel: an open chemical toolbox**", JCheminform, vol. 3, p. 33, 2011.

[99] P. Ambure et al., "**Recent Advances in the Open Access Cheminformatics Toolkits, Software Tools, Workflow Environments, and Databases**", Methods in Pharmacology and Toxicology , DOI 10.1007/7653, 2014.

[100] http://www.taverna.org.uk/

[101] T. Kuhn, EL.Willighagen, A. Zielesny and C. Steinbeck, "**CDK-Taverna: an open workflow environment for cheminformatics**", BMC Bioinformatics, vol. 11, p. 159, 2010.

[102] W. Tan, R. Madduri, A. Nenadic, S. Soiland-Reyes, D. Sulakhe, I. Foster and CA. Goble, "**CaGrid Workflow Toolkit: a taverna based workflow tool for cancer grid**", BMC Bioinformatics, vol. 11, p. 542, 2010.

[103] A. Fiannaca, M.La. Rosa, G.Di. Fatta, S. Gaglio, R. Rizzo and A. Urso, " **The BioDICE Taverna plugin for clustering and visualization of biological data: a workflow for molecular compounds exploration**", J. Cheminform, vol. 6, pp. 1–6, 2014.

[104] Miner,Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias K¨otter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel,ALTANA Chair for Bioinformatics and Information Mining, "**KNIME: The Konstanz Information**", Department of Computer and Information Science, University of Konstanz,

[105] S. Beisken, T. Meinl, B. Wiswedel, LF. ,de Figueiredo, MR. Berthold and C. Steinbeck, "**KNIME-CDK: workflow-driven cheminformatics**", BMC Bioinformatics, pp. 214-257, 2013

[106] http://ggasoftware.com/opensource/indigo

[107] http://cdk.sourceforge.net/old_web/intro.html

[108] http://fr.downloadv.com/download-JChemPaint-628348.htm

[109] O. Spjuth et al., "**Bioclipse: An open source workbench for chemo- and bioinformatics**", BMC Bioinformatics, pp. 8-59, 2007. http://www.bioclipse.net/.

[110] http://wiki.jmol.org/index.php/Jmol_Cdk_Integration

[111] C.James, D. Weininger and J. Delany, "**Daylight theory manual**", website, 2004. Available at http://www.daylight.com/ dayhtml/doc/theory/theory.toc.html.

[112] ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.txt.

[113] H.Benzater, "**Contrôle de la trajectoire du bras manipulateur PUMA560 par les algorithmes évolutionnaires multi-objectifs NSGA-II et NSBBO-II**", Thème de magister, Université des sciences et de la technologie d'Oran Mohamed Boudiaf usto-mb, 2015.

[114] C. Pascal, "**Optimisation multi-objective des problèmes combinatoires : application à la génération des horaires d'examens finaux** ", Thése pour l'obtention de la maitrise en génie, Université du Québec, Montréal, 2004.

[115] Lipinski CA, Lombardo F, Dominy BW, Feeney PJ, "**Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings**", Adv. Drug Deliv. Rev. 46 (1-3): 3–26, March 2000.

[116] Lipinski CA, "**Lead- and drug-like compounds: the rule-of-five revolution**", Drug Discovery Today: Technologies 1 (4): 337–341, December 2004.

[117] T. Cheng, Y. Wang, and SH. Bryant, "**PubChem as a public resource for drug discovery**", Drug. Discov. Today, vol. 15, pp. 1052-1057, December 2010. https://pubchem.ncbi.nlm.nih.gov/

[118] http://www.rguha.net/code/java/

[119] http://www.chemspider.com/about.aspx

[120] Deb et al, "**A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II**", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 2, APRIL 2002.