



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : IA19/M2/2022

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Intelligence Artificielle (IA)

**Algorithme Génétique de tri non dominé combiné à une technique de
classification pour la prédiction du COVID-19**

Par :

RAMLA BOUCHAMI

Soutenu le 27/06/2022 devant le jury composé de :

Khaled REZEG

Professeur

Président

Sihem SLATNIA

MCA

Rapporteur

Mohamed RAMDANI

MCB

Examineur

Année universitaire 2021-2022

وقل صدق زكريا عظام

DEDICACES

A la clôture de mon cycle universitaire de graduation, je ne peux laisser cette occasion pour dédier ce fruit de plusieurs années à ceux qui ont contribués de loin ou de près à ce que je finalise ce rêve.

Tout d'abord, je remercie Dieu pour sa grâce et sa générosité et de m'avoir guidé vers ce chemin de savoir et de m'aider à l'achèvement de ce cycle.

Mes premières dédicaces vont à mon très cher **père Zoubir** et ma très chère **mère** à qui je leur dois beaucoup pour le soutien indéfectible, leur encouragement et leur présence pour que j'achève mon parcours d'étude dans de très bonne conditions.

Je ne dois pas oublier à dédier ce modeste travail à la mémoire de ma grande mère **Naziha**.

Des personnes méritent une dédicace nominative à savoir : mon oncle maternel **Mohamed Djellab** et mon oncle paternel **Mohamed Kouidri** que je considère comme un second père, je cite aussi mon oncle paternel **Mohamed Said**.

Je tiens aussi à dédier ce cher travail à mes frères qui m'ont fournis tout le confort moral très profond surtout dans les moments de charge de travail.

Je ne peux oublier une dédicace très spéciale à l'adorable et notre petite fleur de la famille Oumaima.

Mes dédicaces vont aussi à mes belles sœurs, mes nièces et mes neveux ainsi que mes deux grandes familles **Bouchami** et **Djellab**. Chaque personne et son nom

REMERCIEMENTS

Dans la vie, il faut reconnaître l'apport et l'assistance de ceux qui ont contribué à la réalisation de ce projet de fin d'étude.

En premier lieu, je tiens vivement à remercier ma directrice du projet et mon encadreur docteur **Sihem Slatnia** maître de conférences A au département d'informatique pour avoir accepté de m'encadrer et de me diriger avec ses conseils très judicieux et très constructifs.

J'adresse aussi mes remerciements à mon frère **Rabie** pour ses idées et les discussions très riches qui m'ont permis d'extraire des orientations constructives.

Je n'oublie pas de remercier mademoiselle **Imane Latrech** pour les discussions engagées envers ce projet.

J'exprime aussi mes remerciements aux membres du jury à savoir le professeur **Khaled Rezeg** pour sa présidence du jury ainsi que le docteur **Mohamed Ramdani** pour l'intérêt et les suggestions qui certainement me permettront de développer de nouvelles orientations de ce projet.

Table des matières

INTRODUCTION GENERALE	1
CHAPITRE I : Etat de l'art sur COVID-19.....	3
I.1 Introduction.....	3
I.2 La pneumonie et problème respiratoire.....	3
I.3 Les causes de la pneumonie.....	4
I.4 Pneumonie virale.....	5
I.4.1 Pneumonie bactérienne.....	5
I.4.2 Pneumonie atypique.....	5
I.4.3 Broncho-pneumonie.....	6
I.5 COVID-19	6
I.5.1 Les symptômes de COVID-19.....	7
I.5.2 COVID Syndrome respiratoire aigu sévère (SRAS)	8
I.5.3. Comportement de COVID.....	9
I.5.3.1 Le cas asymptomatique de COVID-19	9
I.5.3.2 Le cas présymptomatique de COVID-19	9
I.5.3.3 Cas bénins de COVID-19	10
I.6 Synthèse de travaux sur COVID-19	12
I.7 Synthèse comparative	16
I.8 Conclusion.....	18
CHAPITRE II : L'optimisation multi-objectif, l'algorithme NSGA-II	
II.1 Introduction.....	19
II.2 Classification des méthodes de résolution des problèmes combinatoire.....	19
II.2.1 Méthodes exactes.....	19
II.2.2 Méthodes approchées.....	19
II.3 Les algorithmes évolutionnaires.....	19
II.3.1 Les Algorithmes Génétiques.....	20
II.3.1.1 Concepts de l'AG.....	21
II.3.1.2 Opérateurs d'évolution.....	22
II.3.2 Optimisation multi-objectif	24
II.3.2.1 Définition	25

II.3.2.2 Les approches métaheuristiques.....	25
II.4 NSGA II	32
II.4.1 L’algorithme NSGA-II.....	33
II.4.2 Description détaillée de NSGA-II.....	35
II.4.2.1 Initialisation de population.....	35
II.4.2.2 Tri non dominé.....	36
II.4.2.3 Distance crowding	37
II.4.2.4 Sélection.....	38
II.4.3 NSGA III	39
II.5 La classification	42
II.5.1 Gradient Tree Boost	42
II.5.2 AdaBoost	43
II.5.3 XGBoost	44
II.6 Présentation de l’algorithme AdaBoost	45
II.7 Conclusion.....	46
CHAPITRE III : Modélisation d’un système d’aide a la décision pour la prédiction de	
Covid 19 basée sur NSGA II	47
III.1 Introduction	47
III.2 NSGA-II pour la prédiction de COVID-19 en utilisant les symptômes des patients.....	47
III.3 Conception globale	47
III.4 Conception détaillée.....	49
III.4.1 Phase d’apprentissage	49
III.4.2. Modélisation de l’individu.....	50
III.4.3 Sélection	50
III.4.4 Opérateurs génétique.....	52
III.4.4.1 Procédure de Croisement.....	52
III.4.4.2 Procédure de Mutation	54
III.5 Phase de concaténation	56
III.6 Phase de validation.....	56
III.7 Phase de calcul des fonctions d’objectives.....	57
III.8 Phase fast non dominated.....	58

III.9 Distance de crowding	60
III.10. Phase de Classification	63
III.11 Phase de test.....	63
III.11 Conclusion	66
CHAPITRE IV Etude expérimentale et validation.....	67
IV.1 Introduction	67
IV.2 Présentation des outils et environnements	67
IV.3 Implémentation	68
IV.3.1 Base de données utilisées	69
IV.3.2 Génération de la population	70
IV.3.3. Détection et résolution des problèmes	71
IV.3.4 Calcul des fonctions de fitness	72
IV.3.5 Structure stockage.....	76
IV.3.5.1 Stockage de la valeur fitness	76
IV.3.5.2 Stockage des fronts.....	77
IV.4 Conclusion.....	78
CONCLUSION GENERALE.....	79
ANNEXE.....	80
REFERENCES.....	88

Liste des figures

Figure I.1 Effet de la pneumonie	4
Figure I.2 Schéma d'une pneumonie	5
Figure I.3 Schéma de la pneumonie	6
Figure I.4 Modélisation du virus	7
Figure II.1 Concepts de l'AG	21
Figure II.2 Phase de croisement forme simple	23
Figure II.3 Phase de croisement forme évoluée	23
Figure II.4 Phase de mutation	23
Figure II.5 Principe de l'algorithme génétique.....	24
Figure II.6 Différentes techniques metaheuristique.....	26
Figure II.7 Formes les plus courantes de surfaces de compromis dans le cas de deux objectifs.....	27
Figure II.8 Illustration des notions de dominance	27
Figure II.9 : Illustration du Front de Pareto	29
Figure II.10 Distance de crowding	37
Figure II.11 Schéma de fonctionnement de NSGA-II	39
Figure II.12 Modèle de NSGA III	41
Figure II.13 Gradient Boost	43
Figure II.14 AdaBoost (a)	43
Figure II.14 AdaBoost (b)	44
Figure II.15 XGBoost	44
Figure II.16 structure AdaBoost	46
Figure III.1 Architecture globale de système	48
Figure III.2 Représentation générale des résultats de NSGA-II.....	49
Figure III.3 Le système NSGA-II.....	51
Figure III.4 Phase de concaténation.....	56
Figure III.5 Organigramme FAST Non Dominated.....	58
Figure III.6 Distance de crowding.....	60
Figure III.7 Classification avec Adaboost.....	65

Figure III.8 principe fonctionnel du système.....	66
Figure IV.1 Interface d'accueil de l'application.....	68
Figure IV.2 Interface d'inscription dans l'application	68
Figure IV.3 Interface de saisie des symptômes.....	69
Figure IV.4 Interface Résultat sous forme PDF.....	69
Figure IV.5 La base de données initiale	69
Figure IV.6 La base de données en format binaire	70
Figure IV.7 Représentation des pourcentages des symptômes (toux, Fièvre, Mal de gorge, Essoufflement, Mal de tête).....	70
Figure IV.8 Représentation des pourcentages des symptômes (Corona résultat, Age_60_et_plus, Sexe, Indiction test).....	71
Figure IV.9 Population initiale	72
Figure IV.10 Représentation d'individu.....	72
Figure IV.11 Représentation significative de l'erreur.....	74
Figure IV.12 Structure du dictionnaire pour le stockage des fronts.....	76
Figure IV.13 Nuage	77
Figure IV.14 Affichage des fronts pour 4 exécutions (a)	77
Figure IV.14 Affichage des fronts pour 4 exécutions (b)	78
Figure 1. Annexe. Représentation du front 0 pour l'itération 0.....	80
Figure 2. Annexe Représentation de front 0, front 1, front 2, pour l'itération 17.....	80
Figure 3. Annexe Représentation de front 0, front 1, front 2, pour l'itération 19.....	81
Figure 4. Annexe Représentation de front 0' Front Pareto' pour dernière itération.....	81
Figure 5. Annexe Représentation de front 0, 1, 2 pour l'itération 0.....	82
Figure 6. Annexe Représentation de front 0, 1, 2 pour l'itération 1.....	82
Figure 7. Annexe Représentation de front 0, 1, 2 pour l'itération 2.....	83
Figure 8. Annexe Représentation de front 0, 1, 2 pour l'itération n.....	83
Figure 9. Annexe Représentation de front Pareto (pour itération n) 2	84
Figure 10. Annexe Représentation de front 0, 1, 2, 3 pour l'itération 0	84
Figure 11. Annexe Représentation de front 0, 1, 2, 3 pour l'itération 1.....	85
Figure 12. Annexe Représentation de front 0, 1, 2, 3 pour l'itération 2	85
Figure 13. Annexe Représentation de front 0, 1, 2 pour l'itération n.....	86
Figure 14. Annexe Représentation de front Pareto pour l'itération n.....	86

Liste des équations

Equation IV.1 Calcul de F1	72
Equation IV.2 Calcul de F2	72
Equation IV.3 Cette formule de calcul de F1 ayant un problème	72
Equation IV.4 Cette formule de calcul de F2 ayant un problème	74

Liste des tableaux

Tableau I.1 Critères OMS des formes cliniques de la COVID-19	12
Tableau I.2 synthèse comparative des travaux	17
Tableau III.1 Individu (Forme binaire, Taille=nombre des symptômes).....	50

الملخص

في المجال الطبي ، أصبح تحليل البيانات قضية ذات أهمية كبيرة ، من الناحيتين الاقتصادية والعملية ، وكذلك في البحث بعد المساهمة الكبيرة جداً لهذه العملية لدعم القرار في المجال الطبي. خبير طبي.

تتعلق هذه المسألة أيضاً باختيار الاستراتيجيات التي سيتم تطبيقها في حالة الصعوبة العملية للتنبؤ. اكتسب هذا المفهوم الأخير زخماً مع ظهور جائحة COVID-19 ، والذي فرض توجهاً نحو استخدام تقنيات التحسين متعددة الأهداف. يستخدم مشروعنا خوارزمية NSGA-II ، والتي أثبتت فعاليتها في العديد من مجالات البحث ، كنموذج مناسب جداً لما يسمى "التعلم الآلي". يعتمد نظامنا على خطوة اختيار الميزة تليها خطوة أخرى تهدف إلى تصنيف التنبؤ بالمرضى المصابين بـ COVID-19. للخطوة الأولى ، نستخدم خوارزمية NSGA-II لتحديد أفضل مجموعة فرعية من الميزات التي تحل التعارض بين الوظائف الموضوعية للتعلم.

بالنسبة للمرحلة الثانية التي تتعلق بالتعلم ؛ يستخدم مصنف Adaboost للتنبؤ بالفئة المقابلة.

يتم تنفيذ مهمة أخرى بواسطة نظامنا ، إنها مهمة الاختبار التي تهدف إلى رؤية اتساق النموذج الذي تم إنشاؤه..

الكلمات الدالة

دعم القرار ، التصنيف ، البيانات ، COVID-19 ، الوظيفة الموضوعية

Abstract

In the medical field, data analysis has become an issue of great importance, both economically and practically, and also in research following the very considerable contribution of this process for decision support in the medical field. medical expert.

This issue also concerns the choice of strategies to be applied in the event of practical difficulty of prediction. This last concept gained momentum with the appearance of the COVID-19 pandemic, which imposed an orientation towards the use of multi-objective optimization techniques. Our project uses the NSGA-II algorithm, which has proven to be effective in several research areas, as a very suitable model for what is called "machine learning". Our system is based on a feature selection step followed by another which aims at classification for the prediction of patients with COVID-19. For the first step we use the NSGA-II algorithm to select the best subset of features that resolve the conflict between objective functions for learning.

For the second stage which concerns learning; the Adaboost classifier is used to predict the corresponding class.

Another task is performed by our system, it is the test task whose purpose is to see the consistency of the generated model.

Keywords : decision support, classification, COVID-19, objective function

Résumé

Dans le domaine médical, l'analyse de données est devenu un enjeu de grande importance que ce soit sur le plan économique que sur le volet pratique et aussi en recherche suite à l'apport très considérable de ce processus pour laide à la décision à l'expert médical.

Cet enjeu concerne aussi le choix des stratégies à appliquées en cas de difficulté pratique de prédiction. Ce dernier concept a pris de l'ampleur avec l'apparition de la pandémie COVID-19 ce qui a imposé une orientation vers l'utilisation des techniques d'optimisation multi-objectif. Notre projet exploite l'algorithme NSGA-II qui a fait preuve d'efficacité dans plusieurs domaines de recherche comme une modèle très adéquat pour ce qu'on appelle « machine learning ». Notre système est basé sur une étape de sélection des caractéristiques suivie d'une autre qui a pour but la classification pour la prédiction de malade atteint de COVID-19. Pour la première étape nous utilisons l'algorithme NSGA-II pour sélectionner le meilleur sous ensemble des caractéristiques qui résolvent le conflit entre les fonctions objectives pour l'apprentissage.

Pour la seconde étape qui porte sur l'apprentissage ; le classifieur Adaboost est utilisé pour prédire la classe correspondante.

Une autre tache est réalisée par notre système, c'est la tache de test dont le but est de voir la consistance du modèle généré.

Mots clés : aide à la décision, classification, COVID-19, fonction objective, optimisation

ANNEXE DE RESULTATS

Echantillon de différents résultats obtenus avec plusieurs exécutions pour une seule itération pour la population initiale.

Exécution 1

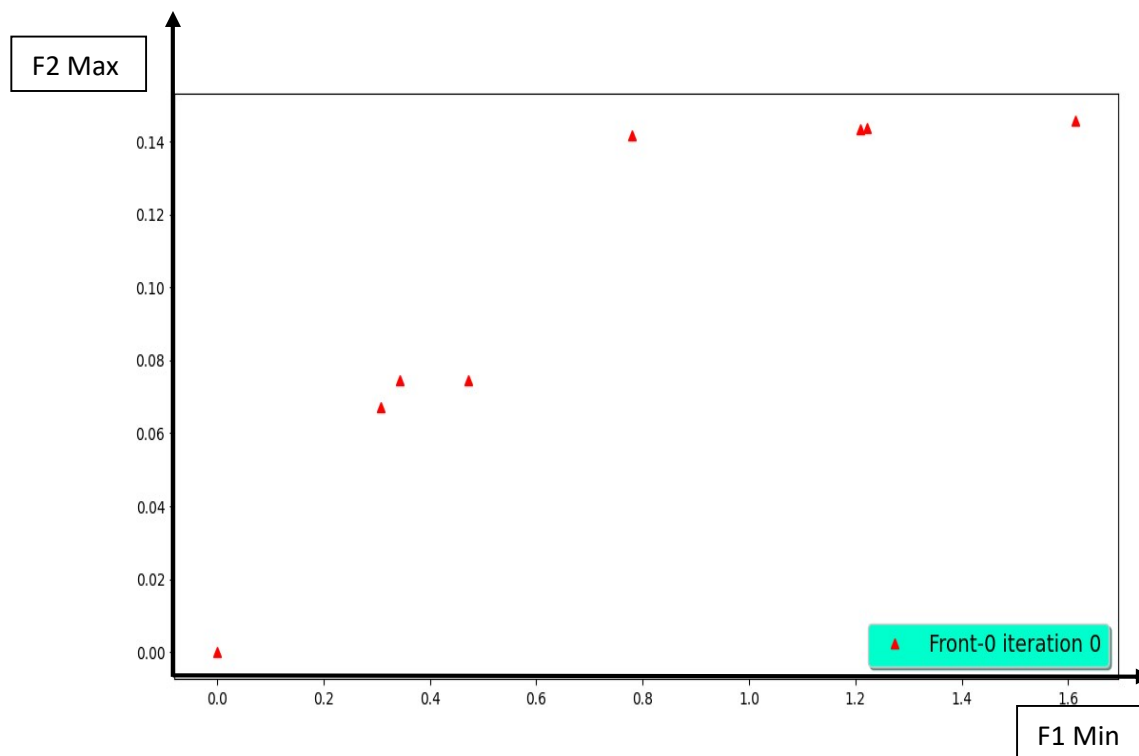


Figure 1. Annexe. Représentation du front 0 pour l'itération 0

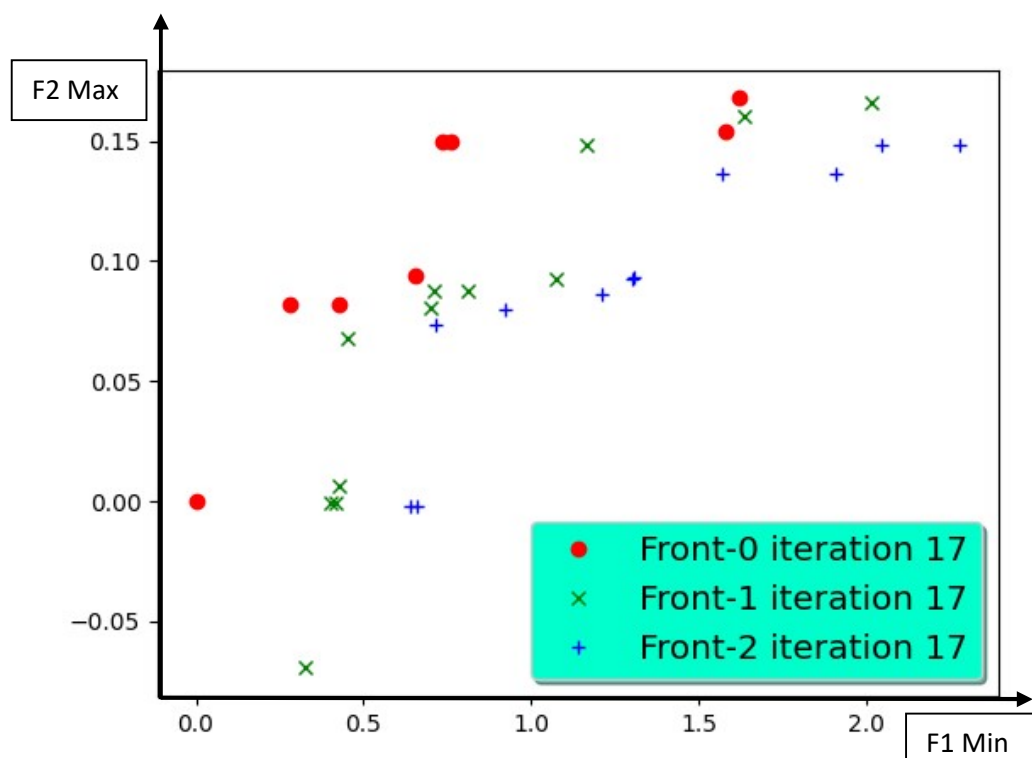


Figure 2. Annexe Représentation de front 0, front 1, front 2, pour

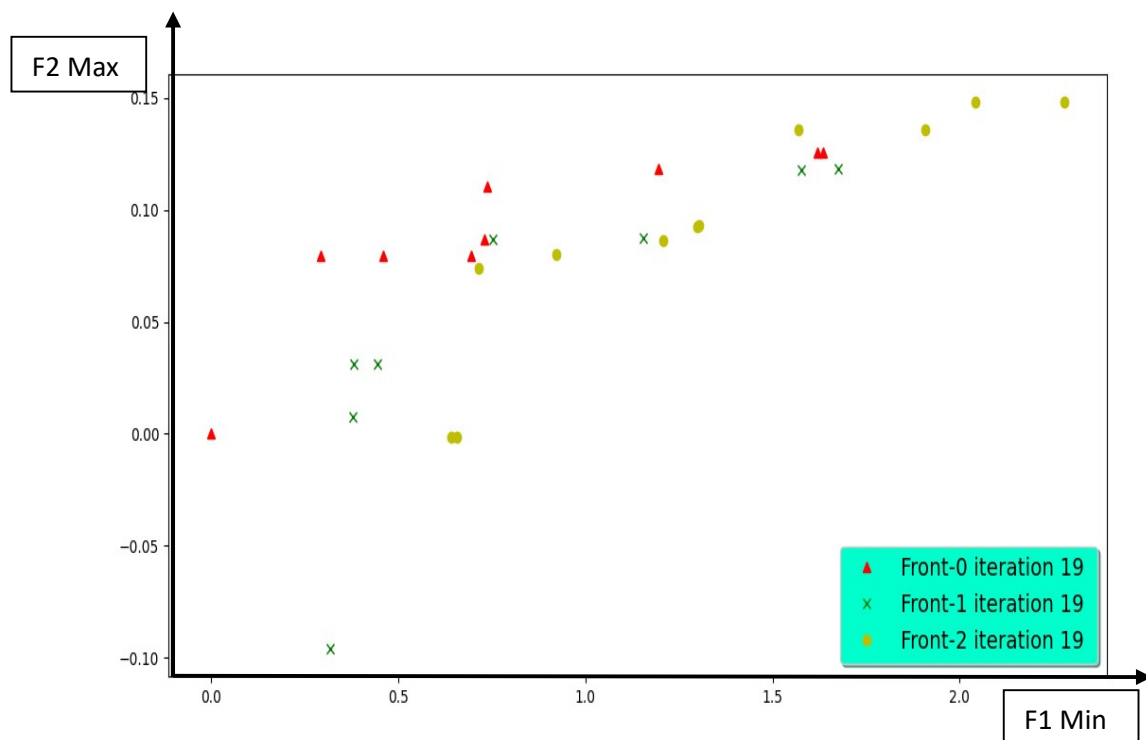


Figure 3. Annexe Représentation de front 0, front 1, front 2, pour l'itération 19

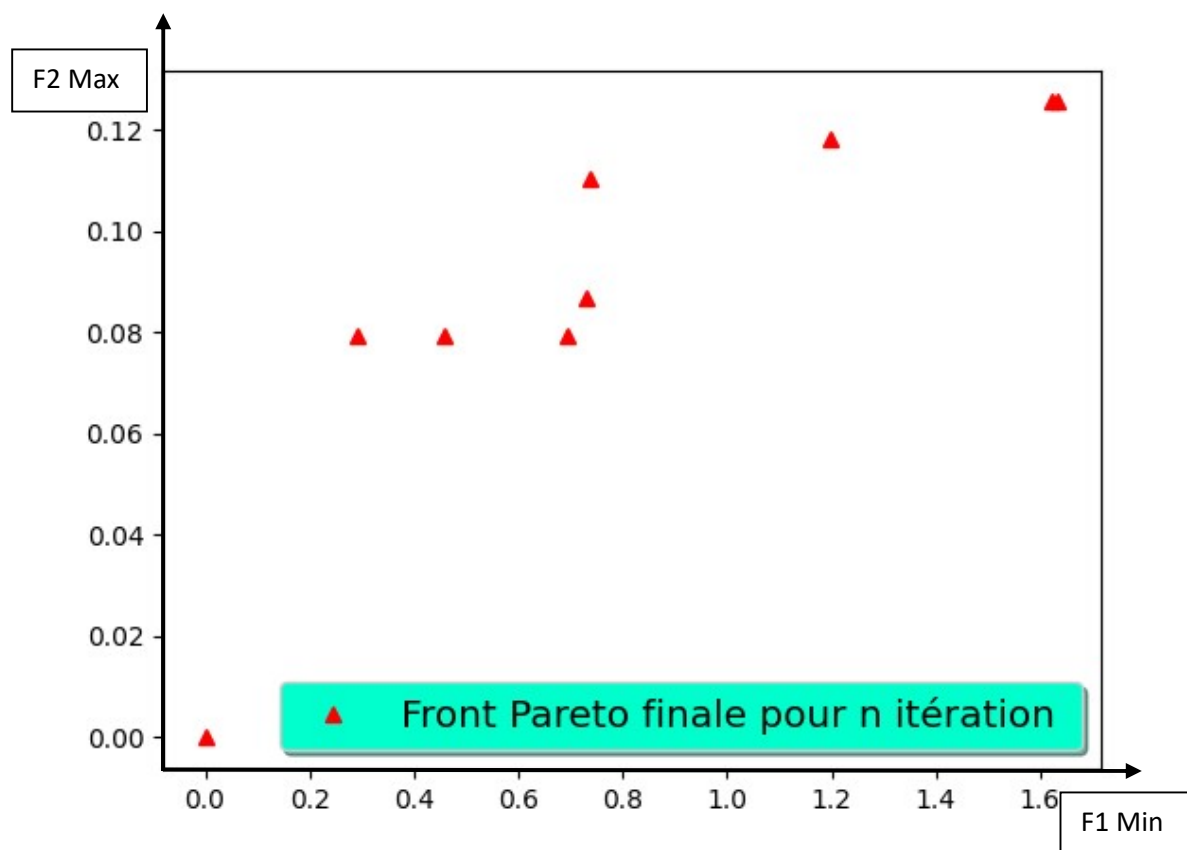


Figure 4. Annexe Représentation de front 0' Front Pareto' pour dernier

Exécution 2

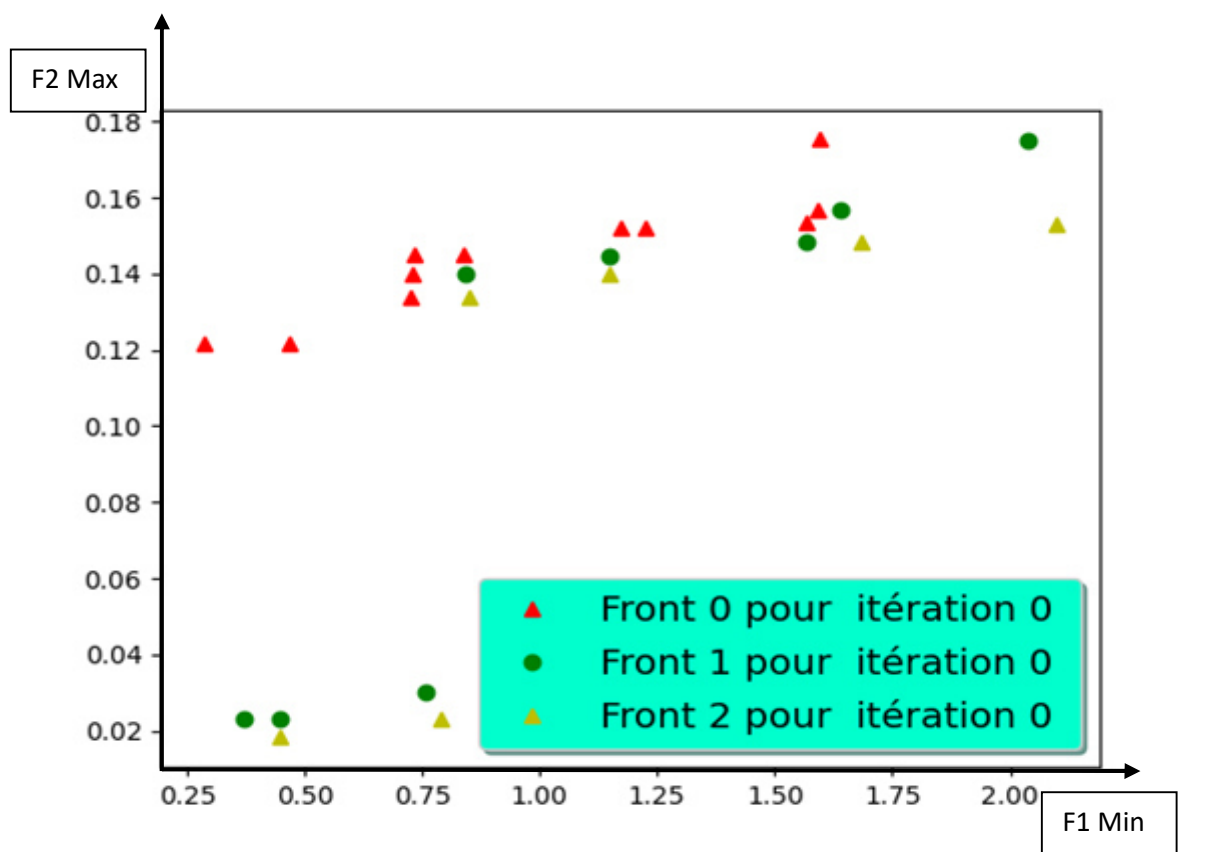


Figure 5. Annexe Représentation de front 0, 1, 2 pour l'itération 0

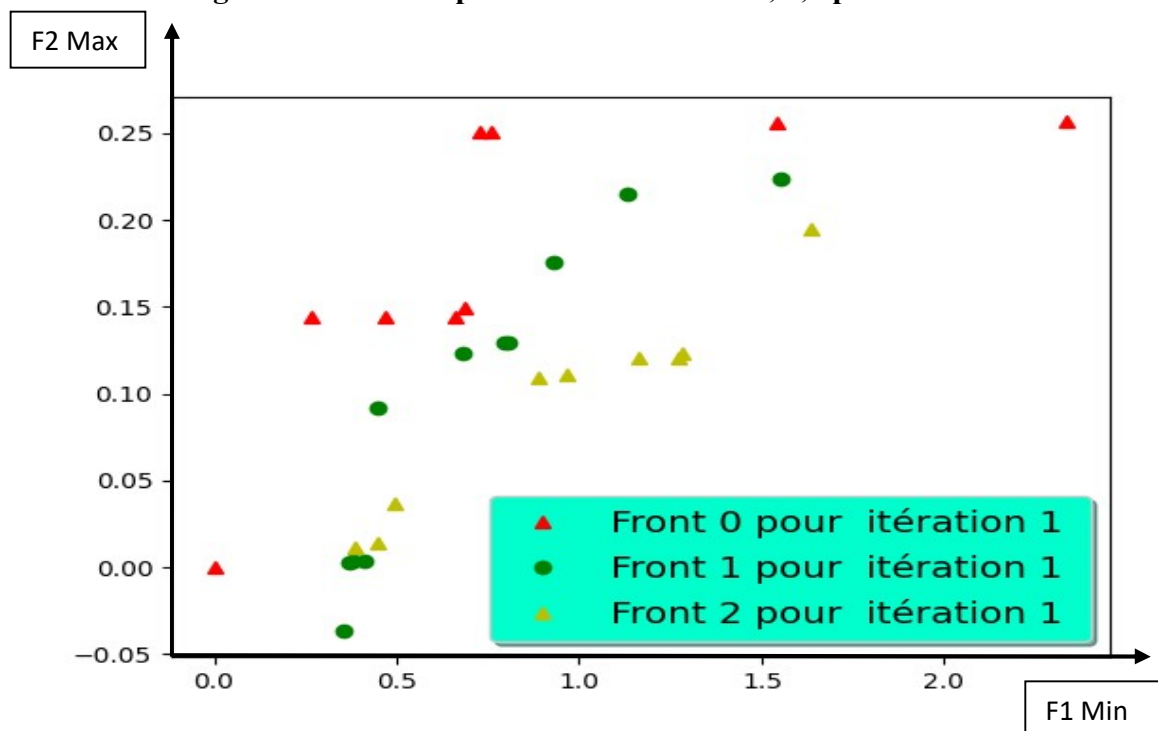


Figure 6. Annexe Représentation de front 0, 1, 2 pour l'itération 1

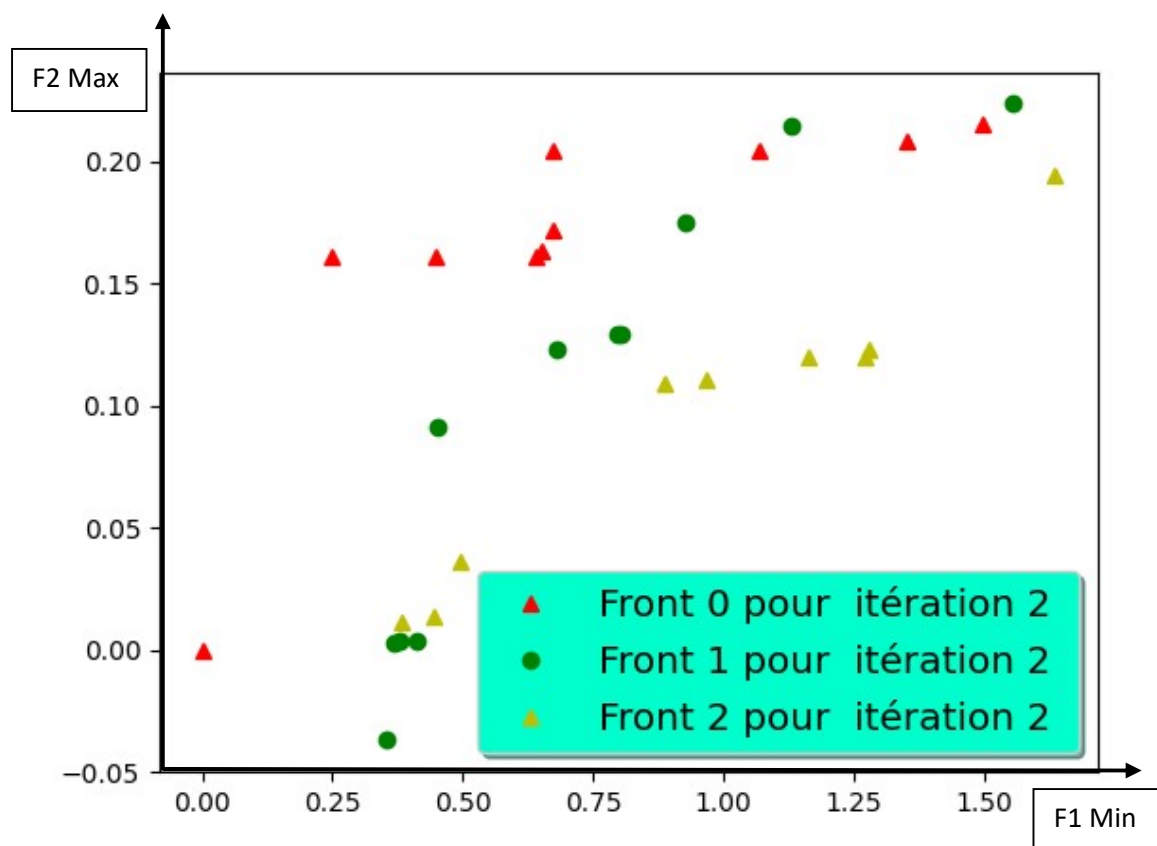


Figure 7. Annexe Représentation de front 0,1,2 pour l'itération 2

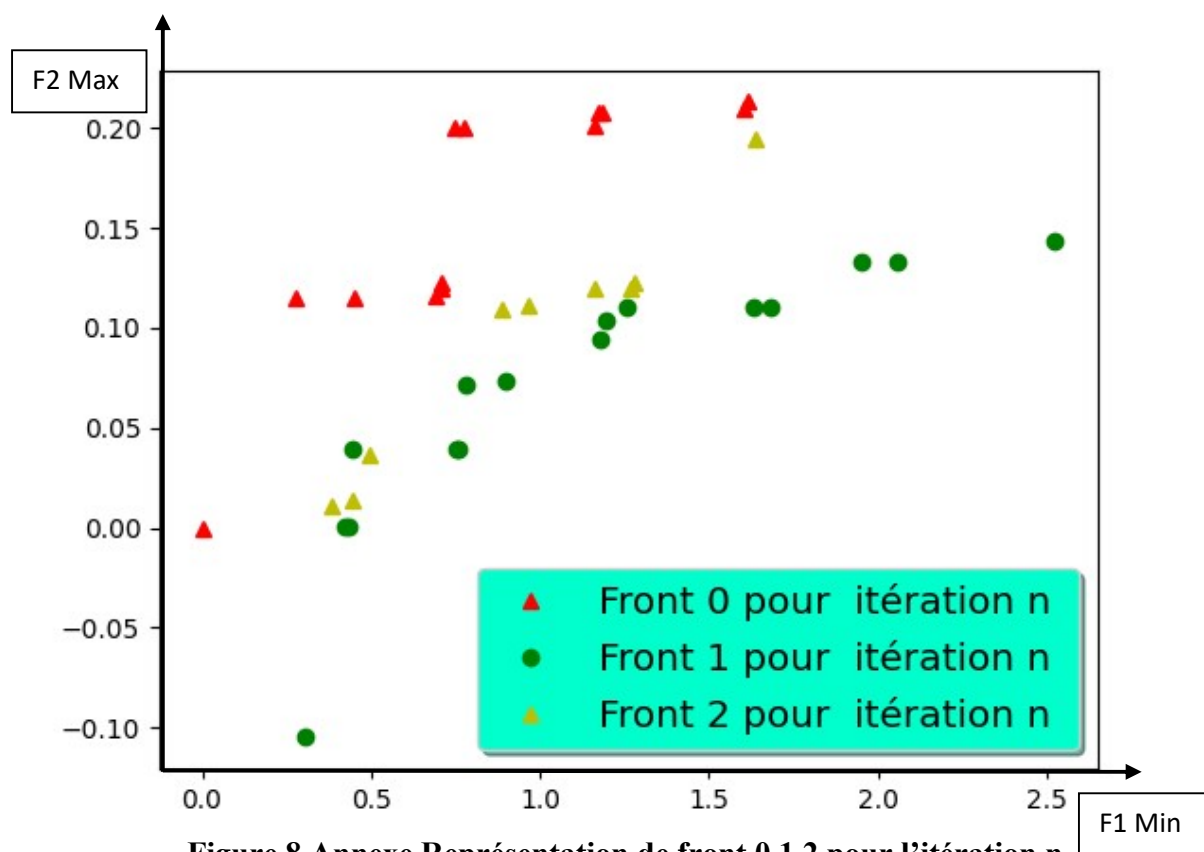


Figure 8. Annexe Représentation de front 0,1,2 pour l'itération n

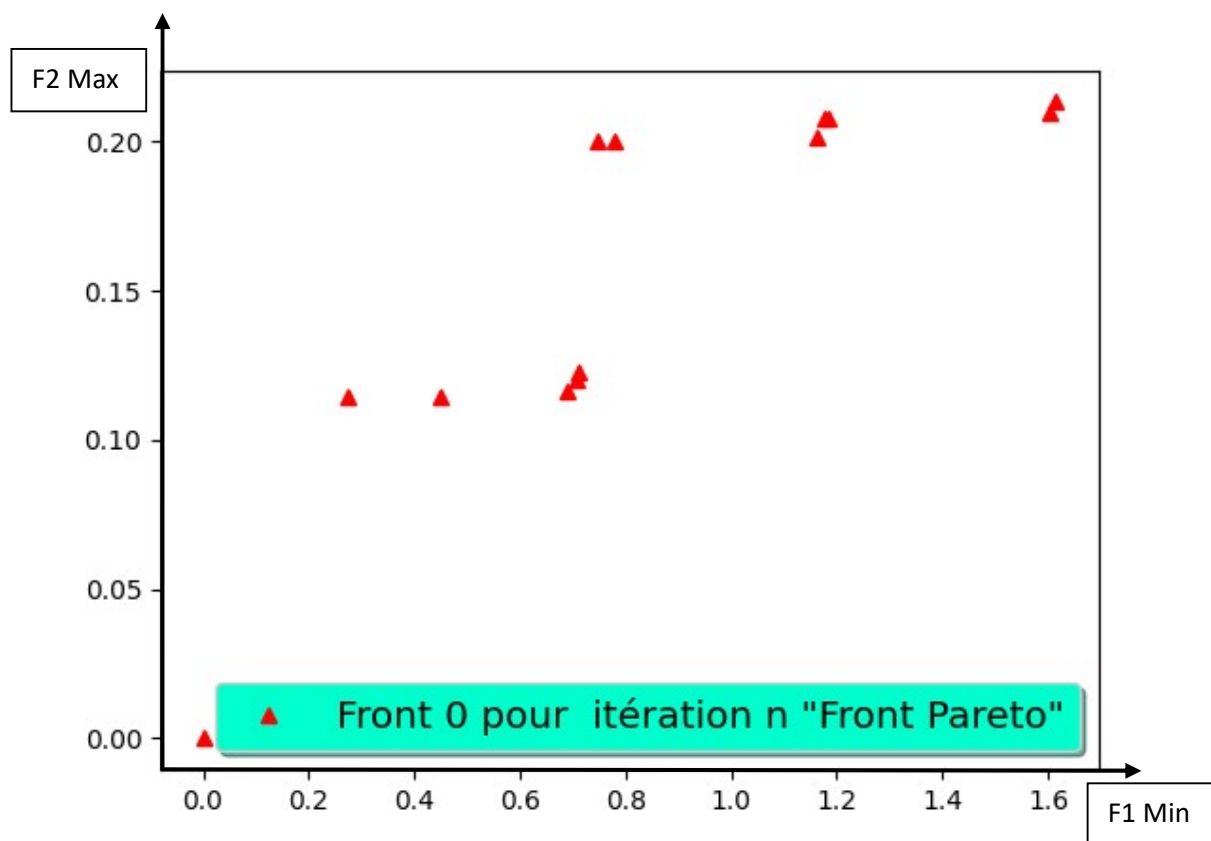


Figure 9. Annexe Représentation de front Pareto (pour itération n) 2

Exécution 3

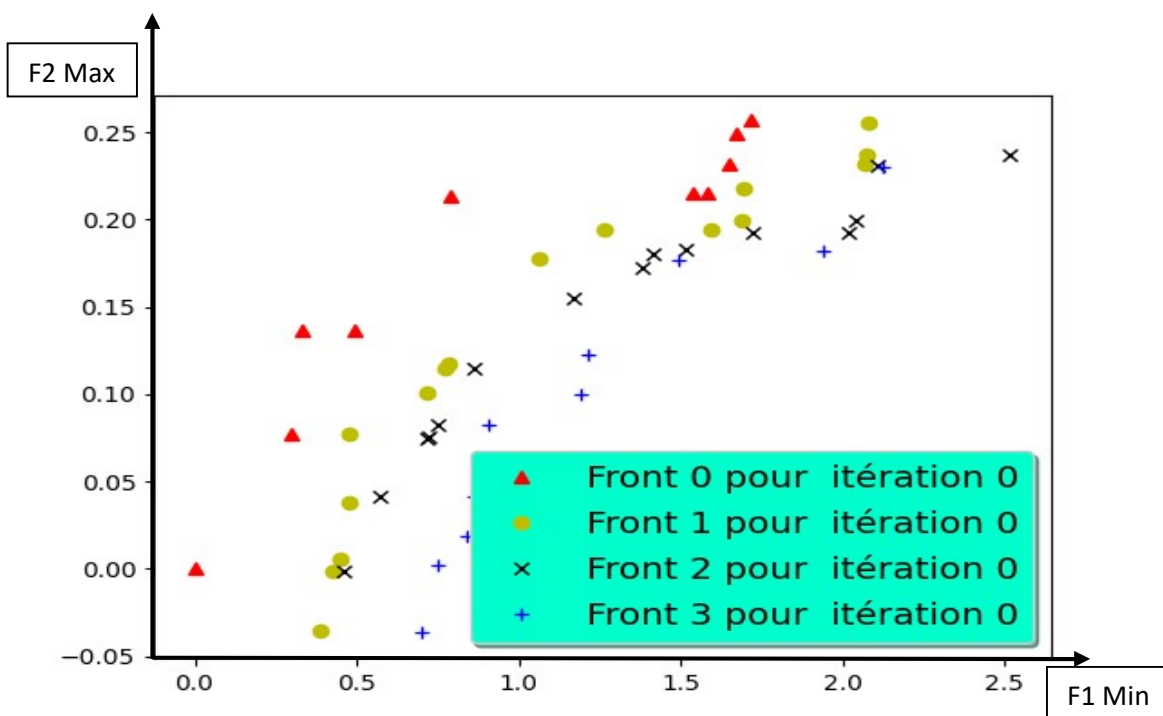


Figure 10. Annexe Représentation de front 0, 1, 2, 3 pour l'itération 0

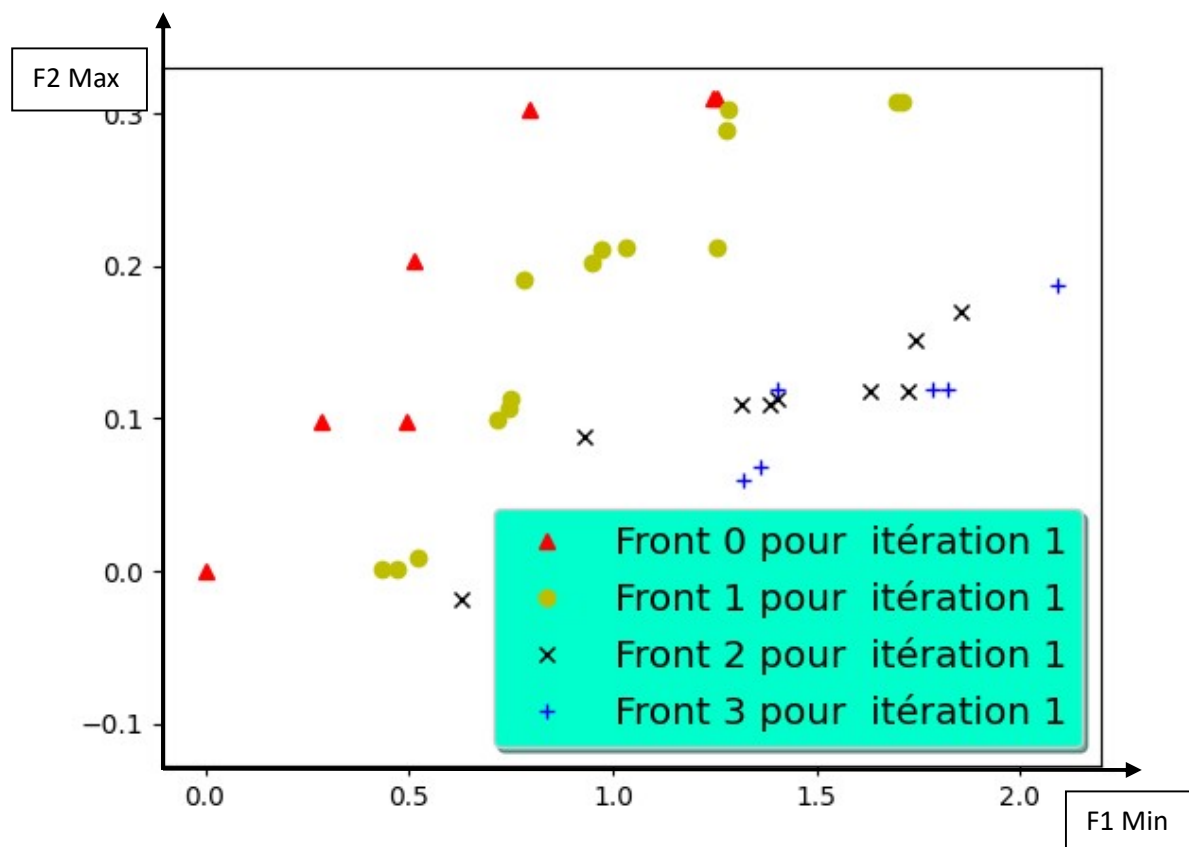


Figure 11. Annexe Représentation de front 0,1,2,3 pour l'itération 1

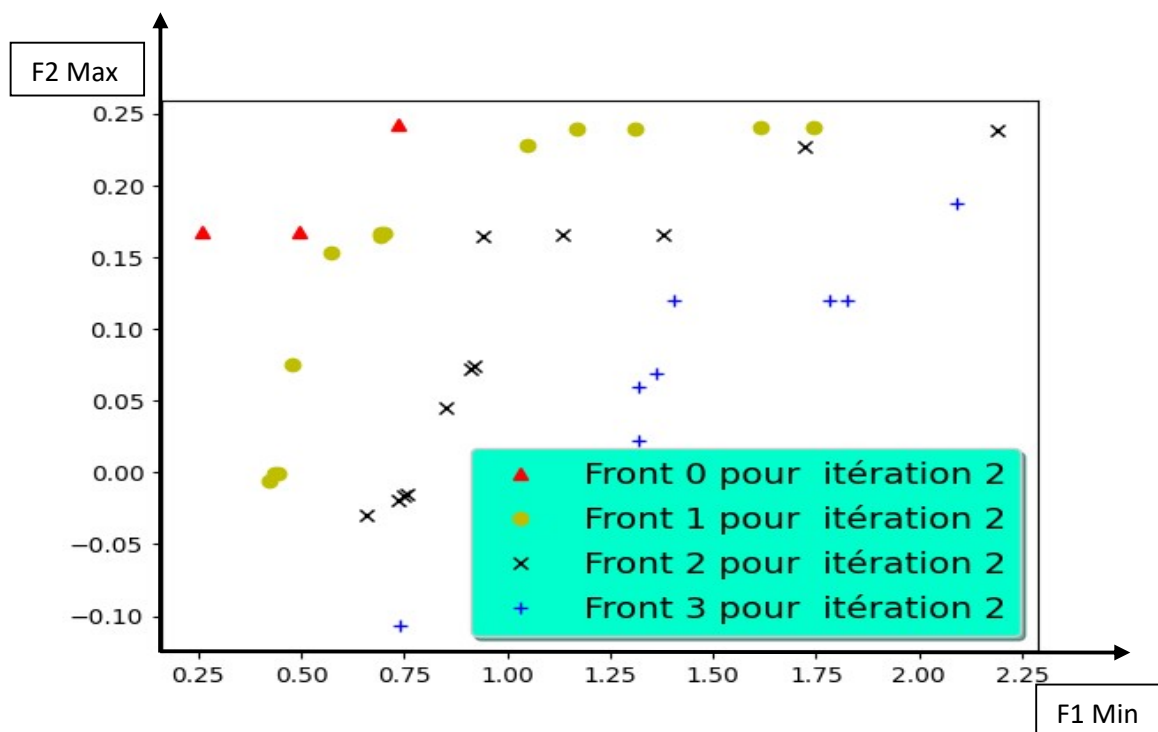


Figure 12. Annexe Représentation de front 0,1,2,3 pour l'itération 2

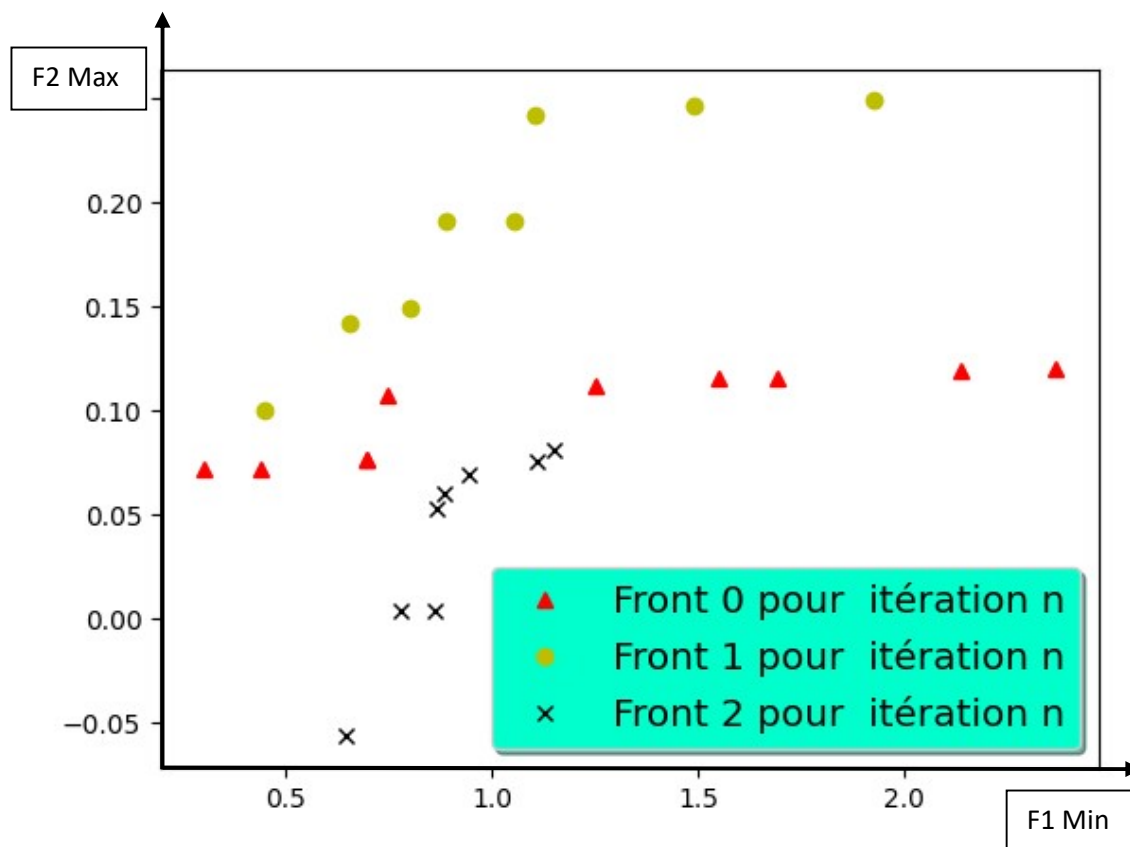


Figure 13Annexe Représentation de front 0,1,2 pour l'itération n

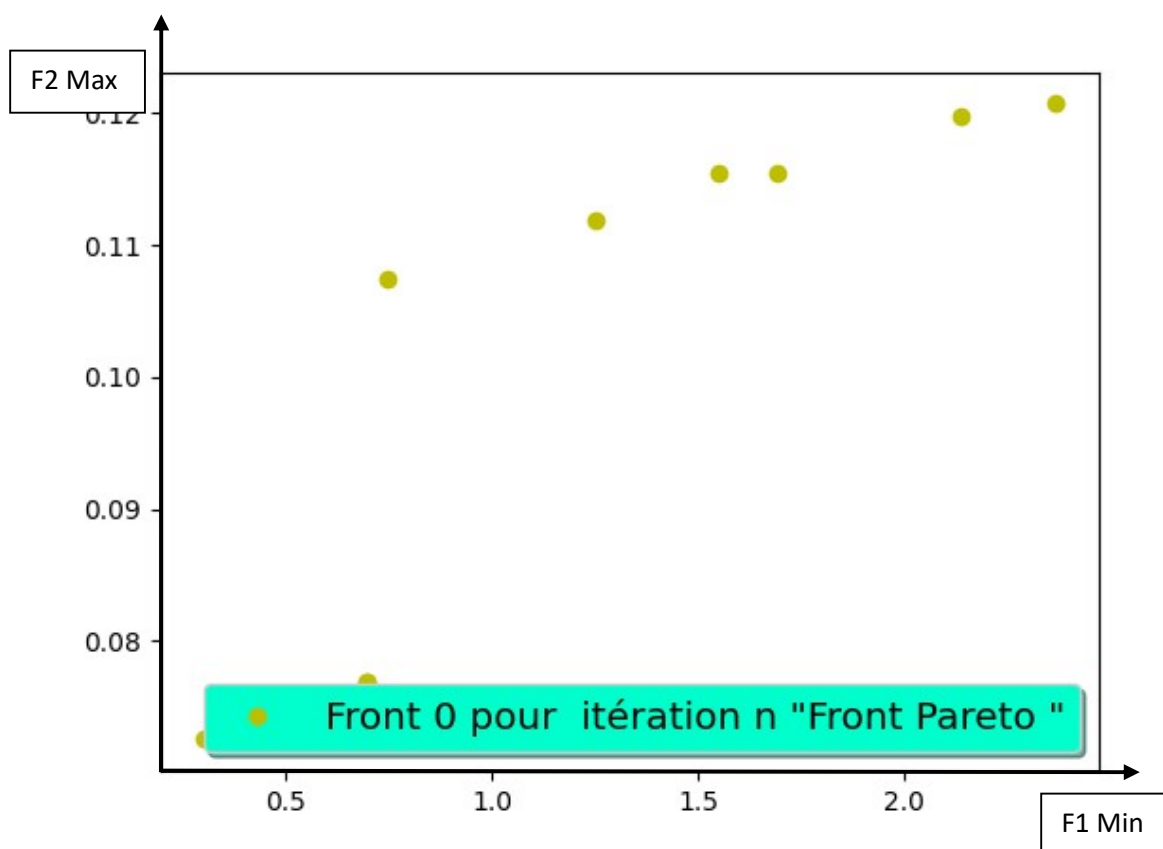


Figure 14. Annexe Représentation de front Pareto pour l'itération n



Acceptance Notification

-Presentation Only Without Publication-

- **Paper ID:** MT3002-A
- **Paper Title:** Data analytics for COVID-19 : state of the art and perspectives
- **Authors:** Ramla Bouchami and Sihem Slatnia

To whom it may concern,

Congratulations!

Based on the reviewer's comment and recommendation, we are glad to inform that abstract identified above is accepted for presentation only without publication on the conference.

The 11th International Conference on Information Communication and Applications (ICICA 2022) will be held at Chongqing University of Posts and Telecommunications, China during June 24-26, 2022. This year, ICICA 2022 is sponsored by Chongqing University of Posts and Telecommunications, China, and co-sponsored by Swinburne University of Technology, Australia and University of Wollongong, Australia.

On behalf of the organizing committee, we are cordially inviting the authors to attend the conference and present the paper at ICICA 2022 at Chongqing University of Posts and Telecommunications, China during June 24-26, 2022. Please complete the registration procedure before **March 7, 2022**. See below the registration procedure please.

We're looking forward to meeting you at Chongqing University of Posts and Telecommunications, China!

Yours sincerely
ICICA 2022 Organizing Committees



Email: icica@iacsit.org

Tel: +86-18302820449

WeChat: [iacsit2009](#)

I.1 Introduction

L'informatique est omniprésente dans plusieurs domaines de la médecine suite à l'apport très apprécié par les experts surtout en matière d'aide à la décision qualitative ainsi que le temps très réduit dans le processus de la prise de décision.

Les enjeux de l'informatique médicale sont très importants surtout avec l'apparition de la pandémie du COVID-19 ou l'exploitation de l'approche d'aide à la décision prédictive est devenue très importante.

Dans ce chapitre, nous dressons un état de l'art sur la **pneumonie** comme une maladie respiratoire, par la suite nous décrivons le covid-19 suivi d'une étude comparative de quelques travaux d'aide à la décision pour le Covid-19 exploitant des techniques de classification.

I.2 La pneumonie et problème respiratoire

La pneumonie est une infection des poumons causée le plus souvent par un virus ou une bactérie.

L'infection touche plus précisément les alvéoles pulmonaires, ces minuscules sacs en forme de ballons situés à l'extrémité des bronchioles (Figure I.1). Elle touche généralement un seul des 5 lobes du poumon (3 lobes dans le poumon droit et 2 dans le gauche), d'où le terme pneumonie lobaire. Lorsque la pneumonie atteint aussi les bronches, on l'appelle bronchopneumonie [1][2].

La pneumonie désigne en fait des infections très diverses qui peuvent être sans gravité ou mettre la vie en danger. Ainsi, si une pneumonie apparaît en complication d'une maladie grave ou chez une personne âgée, elle peut être mortelle. Dans de très rares cas, une pneumonie peut aussi être mortelle chez une personne en bonne santé [2].

La maladie se manifeste généralement par une toux souvent accompagnée d'expectorations, d'essoufflement, de fièvre et des frissons. La pneumonie s'attrape le plus souvent comme une grippe ou un rhume, en inhalant des particules contaminées.

Dans certains cas, elle survient après une autre infection respiratoire, comme une grippe ou une bronchite, qui « dégénère » et s'installe dans les alvéoles des poumons. Certains des symptômes peuvent durer plusieurs semaines. Elle est habituellement peu contagieuse.

Le terme pneumopathie est parfois utilisé comme synonyme de pneumonie. En fait, dans l'usage médical, la pneumonie désigne une infection des poumons causée par un microbe (virus, bactérie), tandis que la pneumopathie est un terme plus large signifiant « maladie des poumons ». Il existe, par exemple, des formes de pneumopathies causées non pas par des microbes, mais par l'inhalation de produits irritants, de poussières ou l'usage prolongé de tabac. Leurs symptômes ressemblent à ceux de la pneumonie [2].



Figure I.1 Effet de la pneumonie [2]

I.3 Les causes de la pneumonie

La pneumonie peut avoir plusieurs origines. Les formes les plus habituelles sont causées par [3]:

- Des bactéries : c'est la cause la plus courante chez les adultes. Les bactéries responsables se trouvent en permanence dans l'organisme (gorge), mais ne se développent que lorsque la personne se trouve affaiblie par une autre maladie.
- Des virus : plus fréquente chez les enfants de moins de 5 ans, cette forme de pneumonie survient généralement après une autre maladie (rhume, grippe). L'infection normale des sinus et de la gorge « descend » vers les poumons, causant ainsi la pneumonie.

Il existe aussi d'autres causes, plus rares, de contracter cette maladie :

- Lorsqu'on inhale certains gaz ou produits dangereux, comme des vernis ou des diluants à peinture.
- Lorsqu'on respire un air imprégné de moisissures.
- Lorsqu'on développe une allergie à certains produits.
- Lorsqu'on aspire vers les poumons une petite quantité du liquide contenu dans l'estomac, principalement lors de vomissements.

I.4 Pneumonie virale

La pneumonie peut aussi être d'origine virale. Ce type de pneumonie touche en priorité les enfants. Différents virus peuvent être en cause, en particulier le virus respiratoire syncytial (noté VRS). La plupart des pneumonies virales sont **bénignes** et circonscrites à un seul lobe pulmonaire (**Figure I.2**) [4].

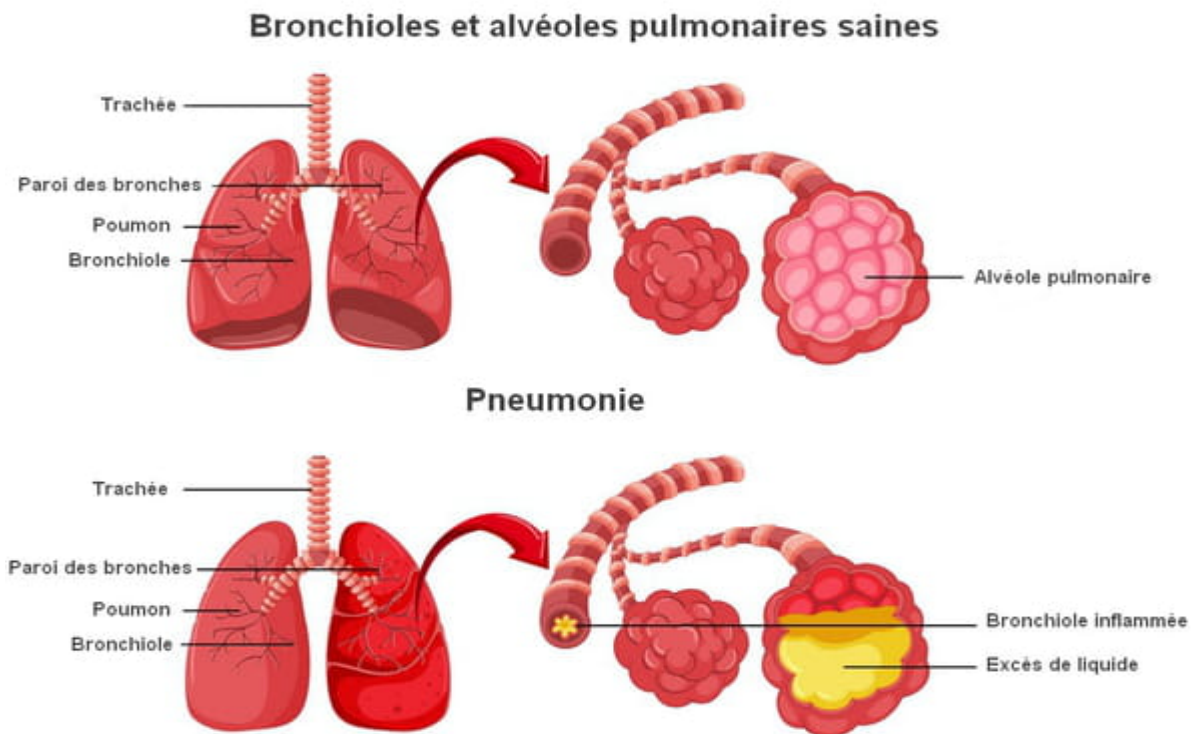


Figure I.2 Schéma d'une pneumonie [4]

I.4.1 Pneumonie bactérienne

Les pneumonies bactériennes représentent une grande partie des pneumonies (infections des poumons). Elles touchent généralement un seul lobe ou segment pulmonaire. Plusieurs bactéries peuvent être en cause, mais il s'agit le plus souvent d'un pneumocoque. Les pneumonies bactériennes peuvent être du type "**pneumonie franche lobaire aiguë**" (ne concernant qu'un seul lobe) ou de type "pneumonie atypique".

I.4.2 Pneumonie atypique

La pneumonie atypique est une **infection provoquée par des germes "atypiques"**, **souvent des bactéries** telles que Legionella pneumophila, la pneumonie Mycoplasma (souvent chez les adultes jeunes) ou la pneumonie Chlamydochila. Elles concernent principalement le patient jeune. Si les bronches sont aussi touchées, on parle alors de bronchopneumonie.

I.4.3 Broncho-pneumonie

La broncho-pneumonie, désigne l'association d'une inflammation au niveau des bronches, mais aussi au niveau du tissu pulmonaire.

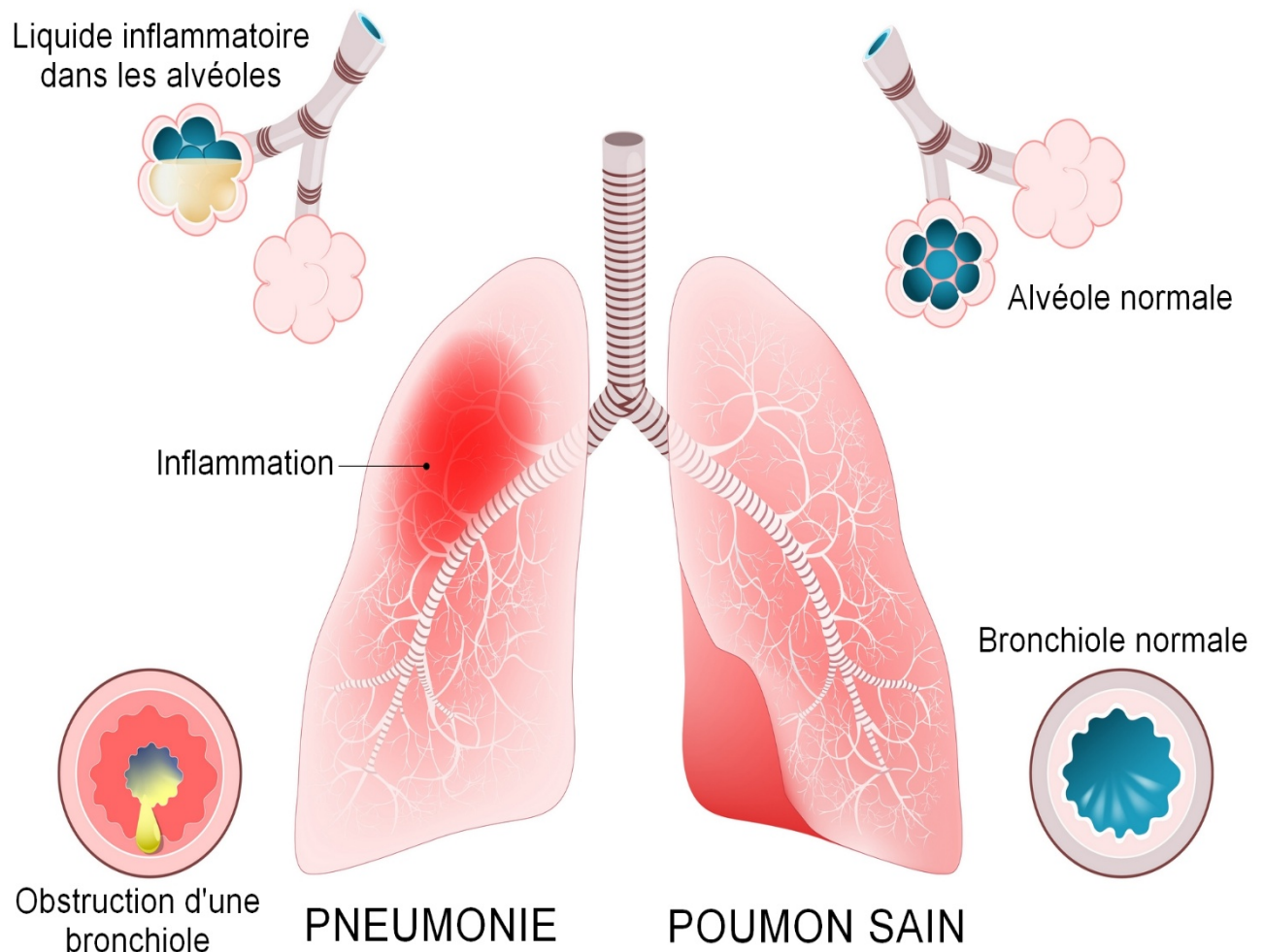


Figure I.3 Schéma de la pneumonie

I.5 COVID-19

Maladie à coronavirus 2019 ou COVID-19 est une maladie respiratoire qui peut se propager de personne à personne. Le virus qui cause le COVID-19 est un nouveau coronavirus identifié pour la première fois lors d'une enquête sur une épidémie de Wuhan, province du Hubei, Chine. Les premiers cas-patients ont déclaré avoir visité un grand marché de fruits de mer et d'animaux vivants à Wuhan.

Les coronavirus sont une grande famille de virus certains causent des maladies chez les gens, et d'autres n'infectent que les animaux. Certains coronavirus infectent alors les animaux se propager aux gens, puis propager la personne à personne telle que :

- Syndrome respiratoire du Moyen-Orient (MERS),
- Syndrome respiratoire aigu sévère (SRAS) [5],
- Maladie à coronavirus 2019 (COVID-19).

Les coronavirus courants incluent certains qui provoquent des maladies légères des voies respiratoires supérieures, comme les coronavirus ont une couronne le rhume.

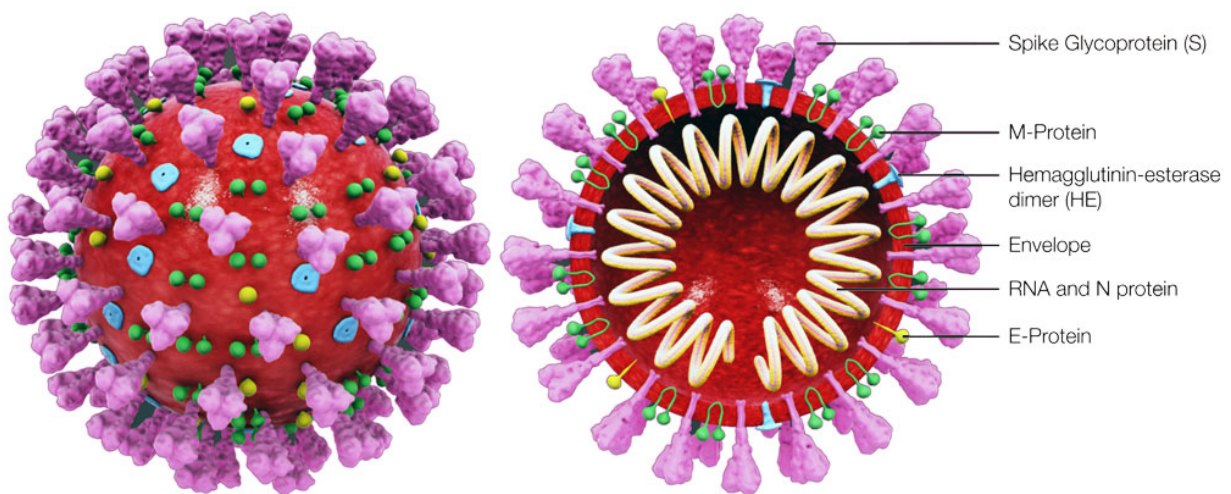


Figure I.4 Modélisation du virus [6]

I.5.1 Les symptômes de COVID-19

Dans la littérature, les symptômes du COVID-19 sont :

- Fièvre/frissons
- La toux
- Essoufflement/difficulté respiration
- Fatigue
- Douleurs musculaires/corporelles
- Maux de tête
- Maux de gorge
- Nouvelle perte de goût/odeur
- Congestion I nez qui coule
- La diarrhée
- Nausées ou vomissements

Les symptômes apparaissent 2 à 14 jours après l'exposition. La période d'incubation moyenne de COVID-19 est signalé à environ 5 jours. La plupart des cas rapportés ont impliqué une maladie respiratoire sévère nécessitant une hospitalisation, avec un taux de mortalité d'environ 35% ; cependant, au moins 21% des patients avaient des symptômes bénins ou aucun symptôme. Fièvre, frissons, myalgies et toux sont fréquents. Les symptômes gastro-intestinaux (p. ex., diarrhées, vomissements, douleurs abdominales) surviennent chez environ 1/3 des patients. Les manifestations peuvent être suffisamment graves pour nécessiter un traitement en USI, mais récemment, la proportion de ces cas a fortement diminué [6].

I.5.2 COVID Syndrome respiratoire aigu sévère (SRAS)

La maladie à coronavirus 2019 (COVID-19) est causée par le coronavirus 2 du syndrome respiratoire aigu sévère (SARS-CoV-2), un coronavirus nouvellement apparu, qui a été identifié pour la première fois à Wuhan, dans la province du Hubei (en Chine), en décembre 2019. Le SARS-CoV-2 est un virus à ARN simple brin, à polarité positive, qui est contagieux chez l'homme. Il s'agit du successeur du SARS-CoV-1, la souche à l'origine de la flambée épidémique de syndrome respiratoire aigu sévère (SRAS) de 2002-2004 [7].

SARS-CoV-2, virus à ARN est un Béta-Coronavirus très proche du virus du SARS, plus éloigné du virus du MERS. Ce virus peut infecter l'homme et l'animal. Chez l'homme, l'infection aux coronavirus se traduit le plus souvent par des rhumes et des infections respiratoires rarement graves (85% des cas). Des tableaux respiratoires sévères sont cependant possibles avec les virus SARS-CoV et MERS-CoV. L'infection à SARS-CoV-2 peut donner des formes sévères et mortelles [7].

Les études cliniques et virologiques qui ont porté sur des échantillons biologiques prélevés à intervalles réguliers chez des cas confirmés montrent que l'excrétion du SARS-CoV-2 est maximale dans les voies respiratoires hautes (nez et gorge) au début de la maladie, dans les trois jours qui suivent l'apparition des symptômes.

Une étude portant sur 77 paires de transmission entre infecteur et infecté a trouvé la plus forte charge virale dans des prélèvements de gorge effectués au moment de l'apparition des symptômes. Cela laisse supposer que le pic d'infectiosité est concomitant à l'apparition des symptômes ou la précède, avec une transmission des infections estimée à 44 % (intervalle de confiance [IC] à 95 % : 30-57 %) chez le cas indicateur pendant la phase présymptomatique.

La période d'incubation de la COVID-19, c'est-à-dire le temps écoulé entre l'exposition au virus (l'infection) et l'apparition des symptômes, est en moyenne de cinq à sept jours, mais peut aller jusqu'à 14 jours.

Durant cette période, également appelée « période présymptomatique », certaines personnes infectées peuvent être contagieuses pendant les un à trois jours qui précèdent l'apparition des symptômes. Il faut souligner que la transmission présymptomatique nécessite elle aussi une propagation du virus par le biais de gouttelettes infectieuses ou par contact direct ou indirect avec les liquides biologiques d'une personne infectée. Un cas asymptomatique est une personne infectée par le SARS-CoV-2 qui ne présente aucun symptôme. Chez les patients symptomatiques, la durée de l'excrétion du virus infectieux a été estimée à huit jours à compter de l'apparition des symptômes [7].

I.5.3. Comportement de COVID

La COVID-19 est associée à des manifestations mentales et neurologiques, notamment anxiété, dépression, troubles du sommeil, céphalées, vertiges, troubles de l'odorat ou du goût, myalgies, délire/encéphalopathie, agitation, accident vasculaire cérébral, lésion cérébrale hypoxique-ischémique, convulsions, coma, méningo-encéphalite et syndrome de Guillain-Barré.

L'anxiété et la dépression semblent fréquentes chez les personnes hospitalisées atteintes de COVID-19. Une étude de cohorte menée auprès de patients hospitalisés à Wuhan, en Chine, a ainsi révélé que plus de 34 % des personnes ressentaient des symptômes anxieux et 28 %, des symptômes dépressifs [7].

I.5.3.1 Le cas asymptomatique de COVID-19

Les cas asymptomatiques de COVID-19 sont des personnes infectées par le SRAS-COV-2 mais qui ne présentent pas de symptômes de COVID-19. Ces personnes ont été testées positives au SARS-COV-2, mais n'ont pas développé de symptômes tels que fièvre, toux ou difficultés à respirer. Elles peuvent être dans le déni de leur état d'infection ou mettre en doute la validité des méthodes de test. Elles ont donc tendance à refuser ou à ignorer tout conseil en matière de respect des mesures de santé publique visant à prévenir la transmission de la maladie à d'autres membres de leur famille, à leurs amis et à la communauté [8].

I.5.3.2 Le cas présymptomatique de COVID-19

Les cas présymptomatiques de COVID-19 sont des personnes infectées qui peuvent transmettre le virus à d'autres personnes bien avant de présenter les signes et symptômes de la COVID-19.

Ces personnes sont susceptibles de transmettre la COVID-19 quelques jours (2 à 3) avant l'apparition des signes et symptômes, et cette période est appelée phase présymptomatique ou latente.

Les cas présymptomatiques développent plus tard des signes et symptômes de la COVID-19, mais pourraient avoir déjà transmis la maladie à d'autres personnes sans le savoir [8].

I.5.3.3 Cas bénins de COVID-19

Certains cas de COVID-19 peuvent présenter des signes et symptômes bénins, mais sont susceptibles d'infecter d'autres personnes. L'attitude généralement observée qui consiste à attendre l'apparition de symptômes allant d'un niveau modéré à sévère avant de prendre des mesures préventives constitue un danger pour la santé des autres. La COVID-19 peut être transmise par des patients présentant ou non des symptômes [8].

Dans de nombreux cas, des manifestations neurologiques ont été signalées même en l'absence de symptômes respiratoires. Plus de 80 % des patients COVID-19 inclus dans une cohorte de patients hospitalisés aux États-Unis d'Amérique ont présenté des symptômes neurologiques pendant leur maladie.

Ces manifestations étaient associées à un risque quatre fois plus important de COVID-19 sévère dans cette cohorte (38). Une série de cas observationnelle française a établi que 65 % des personnes atteintes de COVID-19 hospitalisées en unité de soins intensifs (USI) présentaient des signes de syndrome confusionnel (ou de délire) et 69 % montraient des signes d'agitation (39). Le délire, en particulier, a été associé à un risque accru de mortalité chez les personnes atteintes de COVID-19.

Par ailleurs, la COVID-19 a été associée à des cas de maladie vasculaire cérébrale aiguë (y compris d'accident vasculaire cérébral ischémique et hémorragique), signalés dans plusieurs séries de cas et/ou cohortes en Chine, en France, aux Pays-Bas, au Royaume-Uni et aux États-Unis d'Amérique [9].

Des cas de syndrome de Guillain-Barré et de méningoencéphalite ont également été décrits chez des personnes atteintes de COVID-19.

L'OMS a défini que les formes cliniques de COVID-19 correspondent aux paramètres présents au **Tableau I.1** [10].

Maladie bénigne	Patients symptomatiques répondant à la définition du cas de COVID-19, exempts de signes de pneumonie virale ou d'hypoxie
Forme modérée	
Pneumonie	<p>Adulte présentant des signes cliniques de pneumonie (fièvre, toux, dyspnée, respiration rapide), mais aucun signe de pneumonie sévère, y compris $SpO_2 \geq 90\%$ en air ambiant</p> <p>Observation : le seuil de saturation en oxygène de 90 % comme critère d'une forme sévère de la COVID-19 était arbitraire et doit être interprété avec précaution. Par exemple, les médecins doivent s'en remettre à leur jugement propre pour déterminer si un taux de saturation en oxygène faible est un signe de sévérité ou s'il s'agit d'une caractéristique normale dans le cas d'un patient atteint d'une maladie pulmonaire chronique. De même, une saturation située entre 90–94 % en air ambiant est anormale (chez un patient ayant des poumons sains) et peut être un signe précoce de sévérité, si l'on observe une diminution de ce taux. De manière générale, en cas de doute, l'OMS suggère d'appliquer le principe de précaution en considérant qu'il s'agit d'une forme sévère de la maladie.</p>
Maladie sévère	
Pneumonie sévère	<p>Adulte présentant des signes cliniques de pneumonie (fièvre, toux, dyspnée, respiration rapide) plus l'un des signes ou symptômes suivants : fréquence respiratoire > 30 respirations/min ; détresse respiratoire sévère ; ou $SpO_2 < 90\%$ en air ambiant. Bien que le diagnostic puisse reposer sur l'examen clinique, l'imagerie thoracique (radiographie, tomodensitométrie, échographie) peut le faciliter et permettre d'identifier ou d'écarter des complications pulmonaires</p>
État critique	
Syndrome de détresse respiratoire aiguë	Apparition dans la semaine suivant un accident clinique connu (à savoir, une pneumonie) ou la survenue ou l'aggravation de

(SDRA)	<p>symptômes respiratoires Imagerie thoracique (radiographie, tomodensitométrie ou échographie pulmonaire) : opacités bilatérales ne pouvant entièrement s'expliquer par la présence d'une surcharge volémique, d'une atélectasie lobaire ou pulmonaire, ou de nodules</p> <p>Origine des infiltrats pulmonaires : insuffisance respiratoire ne pouvant entièrement s'expliquer par une insuffisance cardiaque ou une surcharge hydrique. En l'absence de facteurs de risque, une évaluation objective est nécessaire (par exemple, une échocardiographie) pour exclure une origine hydrostatique des infiltrats/de l'œdème</p>
--------	---

Tableau I.1 Critères OMS [11] des formes cliniques de la COVID-19.

I.6 Synthèse de travaux sur COVID-19

Plusieurs travaux ont été développés pour l'aide à la décision du COVID-19. Dans ce paragraphe nous dressons une étude sous forme de synthèse de certains travaux comme références.

Les auteurs dans ce travail [12] visent à identifier l'efficacité de l'utilisation OGA-ELM pour détecter COVID-19 à l'aide d'images radiographiques pulmonaires. Dans le but de réduire la dimensionnalité d'un gradient orienté histogramme, ils ont utilisé l'analyse par composante principale. Les performances d'OGA-ELM sont évaluées sur un ensemble de données de référence contenant 188 radiographies pulmonaires avec deux classes : une classe saine et une autre classe infectée par le COVID-19. Les résultats expérimentaux montrent que l'OGA-ELM atteint une précision de 100,00 % avec un temps de calcul très rapide. Cela démontre que OGA-ELM est une méthode efficace pour la détection de COVID-19 en utilisant des images radiographiques du thorax.

Dans le travail [13], les auteurs ont utilisé l'algorithme génétique de tri non dominé (NSGA-II) pour sélectionner les fonctionnalités en trouvant les meilleurs compromis entre deux objectifs contradictoires : minimiser le nombre de caractéristiques et maximiser les poids des caractéristiques sélectionnées. Ensuite, une phase de classification est menée à l'aide d'un Classificateur AdaBoost. Le modèle proposé est évalué en utilisant deux jeux de données différents. Pour maximiser les résultats, ils ont appliqué une sélection naturelle des hyper-paramètres du classificateur en utilisant aussi l'algorithme génétique.

Les résultats obtenus prouvent l'efficacité de NSGA-II en tant que fonctionnalité algorithmique de sélection combinée avec le classificateur AdaBoost. Il présente des résultats de classification plus élevés qui ont surpassé les méthodes existantes.

Le travail [14] permet de présenter la classification de dix SARS-CoV-2 séquences de protéiques liées au Covid-19 utilisant la fréquence k-mer comme caractéristiques et considérant deux objectifs ; classification performances et sélection de fonctionnalités. La première série d'expériences a considéré les objectifs en même temps, quatre techniques ont été utilisées pour la sélection des fonctionnalités et douze méthodes d'apprentissage automatique bien connues, où trois sont basés sur un réseau de neurones pour la classification. La deuxième série d'expériences considérée comme une approche multi objective où ils ont testé une approche multi-objectifs bien connue à savoir l'Algorithme de Tri génétique non dominé II (NSGA-II), et les archives multidimensionnelles des élites phénotypiques (MAP-Elites), qui considèrent les conteneurs de qualité plus la diversité pour guider la recherche à travers des solutions d'élite. Les résultats expérimentaux montrent que ResNet et PCA est la meilleure combinaison en utilisant des objectifs uniques. Considérant que, pour la classification multiple, NSGA-II surpasse ME avec deux classificateurs sur trois, tandis que ME obtient des résultats compétitifs apportant plus d'ensemble diversifié de solutions.

Les auteurs [15] dans ce travail de recherche proposent un système de détection du COVID-19 à l'aide de réseaux de neurones à convolution profonde et sélection de fonctionnalités basée sur un algorithme différentiel binaire à partir d'Images radiographiques.

En conséquence, le travail [15] permet de présenter une approche hybride basée sur des réseaux de neurones à convolution profonde qui sont des outils très efficaces pour la classification d'images. Les vecteurs de caractéristiques ont été extraits des images à l'aide d'un réseau de neurones à convolution profonde, et l'algorithme métaheuristique différentiel binaire a été utilisé pour sélectionner les caractéristiques les plus précieuses. Le classificateur SVM a ensuite reçu ces fonctionnalités optimisées. Pour la validation de cette proposition conceptuelle, une base de données contenant des images de trois catégories, dont le COVID-19, la pneumonie et une catégorie saine, comprenant 1092 échantillons de rayons X, a été utilisée. La méthode proposée a atteint une précision de 99,43 %, une sensibilité de 99,16 % et une spécificité de 99,57 %. Les résultats indiquent que la méthode proposée a surpassé les études récentes sur la détection du COVID-19 à l'aide d'images radiographiques.

Les auteurs [16] partent du fait le défi critique est la détection rapide du coronavirus sur des personnes infectées. Les techniques disponibles utilisées sont la mesure de la température corporelle, ainsi qu'avec une analyse par prélèvement nasal antérieur.

Cependant, les prélèvements nasaux et les tests de laboratoire sont complexes, intrusifs, et nécessitent de nombreuses ressources. De plus, le manque de kits de test pour répondre aux cas de dépassement est aussi une limite majeure.

Le défi actuel consiste à développer une technologie permettant de manière non intrusive de détecter les patients suspects de coronavirus grâce à des techniques d'intelligence artificielle (IA) telles que l'apprentissage en profondeur (DL). Un autre défi pour mener des recherches dans ce domaine est la difficulté de l'obtention de l'ensemble de données en raison d'un nombre limité de patients ayant donné leur consentement pour participer à l'étude. Étude de recherche. En ce qui concerne l'efficacité de l'IA dans les systèmes de santé, c'est un grand défi pour les chercheurs pour développer un algorithme d'IA qui peut aider les professionnels de la santé et les responsables gouvernementaux à identifier et isoler automatiquement les personnes présentant des symptômes de coronavirus. Ainsi, ils proposent une nouvelle méthode CoVIRN et (modèle COVID Inception-ResNet), qui utilise les radiographies pulmonaires pour diagnostiquer automatiquement les patients COVID-19.

L'algorithme proposé a une conception différente des blocs résiduels qui répondent aux informations en utilisant des cartes de caractéristiques de différentes profondeurs à différentes échelles, avec les différentes couches. Les caractéristiques sont concaténées à chaque bloc de classification proposé, en utilisant la couche de regroupement moyen et les entités concaténées sont transmises à la couche entièrement connectée.

Les blocs d'apprentissage en profondeur proposés efficaces utilisaient différentes techniques de régularisation pour minimiser les rajustements en raison du petit ensemble de données COVID-19. Les caractéristiques multi-échelles sont extraites à différents niveaux du modèle d'apprentissage en profondeur proposé, puis intégré dans divers modèles d'apprentissage automatique pour valider la combinaison de modèles d'apprentissage en profondeur et d'apprentissage automatique.

La proposition du modèle CoVIR-Net a atteint une précision de 95,7 % et l'extracteur de caractéristiques CoVIR-Net avec une forêt aléatoire le classificateur qui a produit une précision de 97,29 %, ce qui est le plus élevé, par rapport à l'état de l'art existant des méthodes d'apprentissage en profondeur. Le modèle proposé serait une solution automatique pour l'évaluation et la classification du COVID-19.

Dans leur travail, les auteurs [17] proposent CityCOVID qui est un modèle détaillé basé sur des agents qui représente les comportements et les interactions sociales de 2,7 millions d'habitants de Chicago alors qu'ils se déplacent et colocalisés dans 1,2 million d'endroits distincts, y compris les ménages, les écoles, les lieux de travail et les hôpitaux, tel que déterminé par les programmes d'activités horaires individuels et les comportements dynamiques tels que l'isolement en raison de l'apparition des symptômes.

La propagation de la maladie est intégrée à chaque agent qui suit les transitions entre les états possibles de la maladie COVID-19, en fonction de attributs d'agents hétérogènes par l'exposition de la colocation et effets des comportements de protection des individus sur transmissibilité.

Tout au long de l'épidémie de COVID-19, les résultats du modèle CityCOVID ont été fournis à la ville, au comité et à l'État qui sont des parties prenantes en réponse à l'évolution des priorités décisionnelles, tout en intégrant les informations émergentes sur le SRAS-CoV-2 épidémiologie.

Les auteurs tentent d'intégrer leur modèle de simulation épidémiologique haute performance à l'apprentissage automatique à grande échelle afin de développer une plate-forme analytique généralisable, flexible et performante pour la planification et la réponse aux crises.

Dans cette recherche [18] les auteurs visent à mettre en évidence l'impact de modèles d'apprentissage profond et d'apprentissage automatique dans la détection du COVID-19 à partir d'images médicales. Cela est réalisé en procédant à une revue de l'état de l'art des approches proposées par les travaux récents en ce champ.

L'objectif principal de cette étude est les développements récents de la classification et de la segmentation approches de détection COVID-19 basée sur l'image. L'étude passe en revue 140 articles de recherche publiés dans différentes bases de données de recherche académique.

Ces articles ont été examinés et filtrés en fonction de critères spécifiques, pour acquérir des informations prudentes sur la détection COVID-19 basée sur l'image. Les méthodes discutées dans ce travail incluent différents types de modalité d'imagerie, principalement Radiographies et tomodensitogrammes. Ces modalités sont également utilisées pour les tâches de classification et de segmentation.

Cet examen vise à catégoriser et à discuter des différentes architectures d'apprentissage en profondeur et d'apprentissage automatique employés pour ces tâches, en fonction de la modalité d'imagerie utilisée.

Il fait également allusion à d'autres profondeurs possibles architectures d'apprentissage et d'apprentissage automatique qui peuvent être proposées pour de meilleurs résultats face au COVID-19 détection. Parallèlement à cela, un aperçu détaillé des tendances émergentes et des percées dans l'artificiel. La détection du COVID-19 basée sur le renseignement a également été discutée.

Les auteurs donnent comme synthèse sur le travail réalisé en précisant les défis techniques et non techniques auxquels sont confrontés les chercheurs et illustre les avantages de la détection COVID-19 basée sur l'image avec les techniques de l'intelligence artificielle.

Dans ce travail [19] les auteurs essayent de classer les rapports cliniques textuels en quatre classes en utilisant des algorithmes d'apprentissage automatique classiques et d'ensemble.

L'ingénierie des fonctionnalités a été réalisée à l'aide de techniques telles que Termfréquence/fréquence de document inverse (TF/IDF), Sac de mots (BOW) et la longueur du rapport. Ces fonctionnalités étaient fournies à l'apprentissage automatique traditionnel et d'ensembles classificateurs.

L'utilisation de la régression logistique et naïve multinomiale Bayes a montré de meilleurs résultats que les autres algorithmes de ML ayant une précision de test de 96,2 %.

I.7 Synthèse comparative

Dans le tableau suivant, nous dressons une synthèse comparative des travaux développés dans cette section.

Auteurs/	objectif	approche	Data base	Resultats
[12]	détecter COVID-19 à l'aide d'images radiographiques pulmonaires	OGA-ELM	188 radiographies pulmonaires avec deux classes : une saine et une infectée par le COVID-19.	100,00 % avec un temps calcul très rapide
[13]	sélectionner les fonctionnalités en trouvant les meilleurs compromis entre deux objectifs contradictoires : minimiser le nombre de caractéristiques et maximiser les poids des caractéristiques sélectionnées	l'algorithme génétique non dominé de tri (NSGA-II)	- données issues de la bio-informatique pour le diabète et l'obésité - de données issues de biopuces	résultats de classification plus élevés
[14]	la classification de dix SARS-CoV-2 séquences de protéines liées au Covid-19	la fréquence k-mer comme caractéristiques et considérant deux objectifs ; classification performances et sélection de fonctionnalités		Les résultats expérimentaux montrent que ResNet et PCA est la meilleure combinaison en utilisant des objectifs uniques

[15]	Proposent un système de détection du COVID-19	Réseaux de neurones à convolution profonde et sélection de fonctionnalités basée sur un algorithme différentiel binaire à partir d'Images radiographiques.	Pour la validation de cette proposition conceptuelle, une base de données contenant des images de trois catégories, dont le COVID-19, la pneumonie et une catégorie saine, comprenant 1092 échantillons de rayons X, a été utilisée.	une précision de 99,43 %, une sensibilité de 99,16 % et une spécificité de 99,57 %.
[16]	détection rapide du coronavirus sur des personnes infectées	Apprentissage profond CoVIRNet (modèle COVID Inception-ResNet),	les radiographies pulmonaires pour diagnostiquer automatiquement les patients COVID-19	une précision de 95,7 % et l'extracteur de caractéristiques CoVIR-Net avec une forêt aléatoire le classificateur qui a produit une précision de 97,29 %
[17]	CityCOVID est un modèle détaillé basé sur des agents qui représente les comportements et les interactions sociales	simulation épidémiologique haute performance à l'apprentissage automatique à base d'agents	les interactions sociales de 2,7 millions d'habitants	
[18]	L'objectif principal de cette étude est les développements récents de la classification et de la segmentation approches de détection COVID-19 basée sur l'image.	mise en évidence l'impact de modèles d'apprentissage profond et d'apprentissage automatique dans la détection du COVID-19	L'étude passe en revue 140 articles de recherche publiés dans différentes bases de données de recherche académique	
[19]	Classifier les rapports cliniques textuels en quatre classes	en utilisant des algorithmes d'apprentissage automatique classiques et d'ensemble. à l'aide de techniques telles que Term fréquence/fréquence de document inverse (TF/IDF), Sac de mots (BOW) et la longueur du rapport		ML en ayant une précision de test de 96,2 %.

Tableau I.2 synthèse comparative des travaux

I.8 Conclusion

Dans ce chapitre nous avons dressé un état de l'art sur la maladie pulmonaire dite la « pulmonie ». Cette maladie qui est devenue un enjeu de recherche et d'investissement économique suite à l'apparition de la pandémie COVID-19.

Nous avons présenté ses différentes formes ainsi que les symptômes qui apparaissent sur le malade atteint de la pulmonie.

Une partie de ce chapitre contient aussi une description concerne le COVID-19 comme une forme de maladie qui touche plus spécialement les poumons des personnes et pour laquelle de grands travaux de recherche sont en cours.

Dans le chapitre suivant, nous allons décrire un état de l'art sur l'optimisation multi-objectif et l'algorithme NSGA-II

II.1 Introduction

Face à un problème d'optimisation combinatoire donné, la question à laquelle on doit répondre est la résolution du problème. Ce dernier possède généralement un nombre énorme de solutions réalisables. La résolution la plus évidente est de lister toutes les combinaisons possibles afin de trouver celles qui sont valides et meilleures. On appelle ce type d'algorithme : Algorithme Exhaustif.

Ces algorithmes sont très gourmands en termes de complexité. Ils passent d'algorithmes polynomiaux à non-polynomiaux avec l'augmentation de la dimension du problème à traiter.

L'objectif de ce chapitre est de faire une synthèse bibliographique sur les principaux concepts de l'optimisation multi-objectifs.

II.2 Classification des méthodes de résolution des problèmes combinatoire

Les méthodes de l'optimisation combinatoire peuvent être classées en deux grandes familles de classes : les méthodes exactes et les méthodes approchées [20].

II.2.1 Méthodes exactes

Les méthodes exactes (appelées aussi complètes) produisent une solution optimale pour une instance de problème d'optimisation donné. Elles se reposent généralement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions. Elles sont utilisées pour trouver au moins une solution optimale d'un problème.

Les algorithmes exacts les plus connus sont la méthode de séparation et évaluation, la programmation dynamique, et la programmation linéaire. L'inconvénient majeur de ses méthodes est l'explosion combinatoire : Le nombre de combinaisons augmente avec l'augmentation de la dimension du problème.

L'efficacité de ces algorithmes n'est prometteuse que pour les instances de problèmes de petites tailles [21].

II.2.2 Méthodes approchées

Contrairement aux méthodes exactes, les méthodes approchées ne fournissent pas forcément une solution optimale, mais seulement une bonne solution (de qualité raisonnable) dans un temps raisonnable [22].

II.3 Les algorithmes évolutionnaires

Le terme algorithme évolutif (AE) désigne une classe de méthodes d'optimisation Stochastique qui simuler le processus d'évolution naturelle.

Les origines des EA remontent à la fin années 1950, et depuis les années 1970, plusieurs méthodologies évolutives ont été proposées, principalement algorithmes génétiques, programmation évolutive et stratégies d'évolution [23].

Toutes ces approches opèrent sur un ensemble de solutions candidates. En utilisant de fortes simplifications, cet ensemble est ensuite modifié par les deux principes de base de l'évolution : la sélection et la variation. La sélection représente la compétition pour les ressources entre les êtres vivants. Certains sont meilleurs que d'autres et plus susceptibles de survivre et de reproduire leur information génétique. Dans les algorithmes évolutionnaires, naturel la sélection est simulée par un processus de sélection stochastique.

Chaque solution a une chance de se reproduire un certain nombre de fois, en fonction de leur qualité. Ainsi, la qualité est évaluée en évaluant les individus et en leur attribuant une valeur scalaire de remise en forme. L'autre principe, la variation, imite la capacité naturelle de créer de « nouveaux » êtres vivants par recombinaison et mutation. Bien que les principes sous-jacents soient simples, ces algorithmes se sont avérés être un mécanisme de recherche général, robuste et puissant. De plus, les AE semblent être particulièrement adaptés à l'optimisation multi-objectifs car ils sont capables de capturer plusieurs solutions pareto-optimales en une seule simulation et peut exploiter similitudes de solutions par recombinaison. L'application d'algorithmes évolutifs (EA) dans l'optimisation multi-objectifs est actuellement suscitée un intérêt croissant de la part de chercheurs d'horizons divers [24].

II.3.1 Les Algorithmes Génétiques

Les Algorithmes Génétiques (AG) sont des algorithmes de recherche heuristiques adaptatifs qui appartiennent à la plus grande partie des algorithmes évolutionnaires. Les algorithmes génétiques sont basés sur les idées de sélection naturelle et de génétique. Il s'agit d'une exploitation intelligente de la recherche aléatoire fournie avec des données historiques pour diriger la recherche dans la région de meilleures performances dans l'espace de solution. Ils sont couramment utilisés pour générer des solutions de haute qualité pour les problèmes d'optimisation et les problèmes de recherche.

Les algorithmes génétiques simulent le processus de sélection naturelle, ce qui signifie que les espèces capables de s'adapter aux changements de leur environnement sont capables de survivre, de se reproduire et de passer à la génération suivante. En termes simples, ils simulent la « survie du plus apte » parmi les individus de génération consécutive pour résoudre un problème.

Chaque génération est constituée d'une population d'individus et chaque individu représente un point dans l'espace de recherche et une solution possible.

Chaque individu est représenté par une chaîne de caractères/entiers/flottants/bits. Cette chaîne est analogue au chromosome [25].

Les algorithmes génétiques sont basés sur une analogie avec la structure génétique et le comportement des chromosomes de la population.

Voici le fondement des AG basées sur cette analogie.

- L'individu dans la population est en compétition pour les ressources et le partenaire
- Les individus qui réussissent (les plus aptes) s'accouplent pour créer plus de descendants que les autres
- Les gènes du parent "le plus apte" se propagent tout au long de la génération, c'est-à-dire que parfois les parents créent une progéniture meilleure que l'un ou l'autre des parents.
- Ainsi, chaque génération successive est plus adaptée à leur environnement.

L'algorithme de ce processus est :

1. Initialiser aléatoirement les populations p
2. Déterminer l'aptitude de la population
3. Jusqu'à convergence répéter :
 - 3.1. Sélectionner les parents dans la population
 - 3.2. Crossover (croisement) et générer une nouvelle population
 - 3.4. Effectuer une mutation sur une nouvelle population
- 3 5. Calculer la valeur de la fonction fitness pour la nouvelle population

III.3.1.1 Concepts de l'AG

Ce paradigme, associé avec la terminologie de la génétique, nous permet d'exploiter les algorithmes génétiques : Nous retrouvons les notions de Population, d'Individu, de Chromosome et de Gène [26].

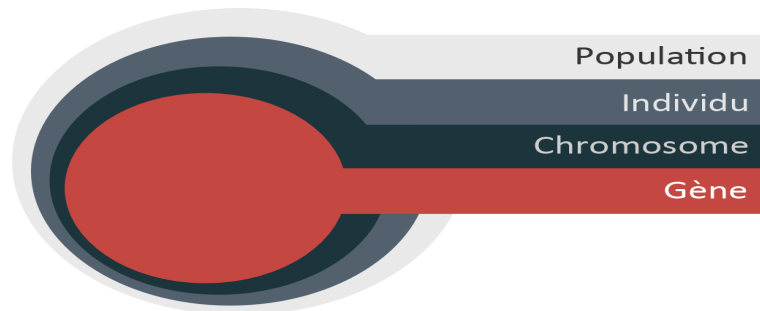


Figure II.1 Concepts de l'AG

- **La population** est l'ensemble des solutions envisageables.
- **L'individu** représente une solution.
- **Le Chromosome** est une composante de la solution.
- **Le Gène** est une caractéristique, une particularité.

Avec ces notions, nous obtenons trois opérateurs d'évolution.

II.3.1.2 Opérateurs d'évolution

Il y a trois opérateurs d'évolution dans les algorithmes génétiques [27]:

- **La sélection** : Choix des individus les mieux adaptés.
- **Le croisement** : Mélange par la reproduction des particularités des individus choisis.
- **La mutation** : Altération aléatoire des particularités d'un individu.

❖ Sélection

La sélection consiste à choisir les individus les mieux adaptés afin d'avoir une population de solution la plus proche de converger vers l'optimum global. Cet opérateur est l'application du *principe d'adaptation* de la théorie de Darwin. Il existe plusieurs techniques de sélection. Voici les principales utilisées :

- **Sélection par rang** : Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation.
- **Probabilité de sélection proportionnelle à l'adaptation** : Technique de la roulette ou roue de la fortune, pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème.
- **Sélection par tournoi** : Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation.
- **Sélection uniforme** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation.
- **Sélection par élitisme** : cette méthode de sélection permet de mettre en avant les meilleurs individus de la population. Ce sont donc les individus les plus prometteurs qui vont participer à l'amélioration de notre population. Cette méthode a l'avantage de permettre une convergence (plus) rapide des solutions, mais au détriment de la diversité des individus. Une autre possibilité relevant aussi du domaine de l'élitisme, pour s'assurer que les meilleurs individus feront effectivement partie de la prochaine génération, est tout simplement de les sauvegarder pour pouvoir les rajouter à coup sûr dans la population suivante.

❖ Croisement

Le croisement, ou enjambement, *crossing-over*, est le résultat obtenu lorsque deux chromosomes partagent leurs particularités. Celui-ci permet le brassage génétique de la population et l'application du *principe d'hérédité* de la théorie de Darwin. Il existe deux méthodes de croisement : simple ou double enjambement.

- Le simple enjambement consiste à fusionner les particularités de deux individus à partir d'un pivot, afin d'obtenir un ou deux enfants :

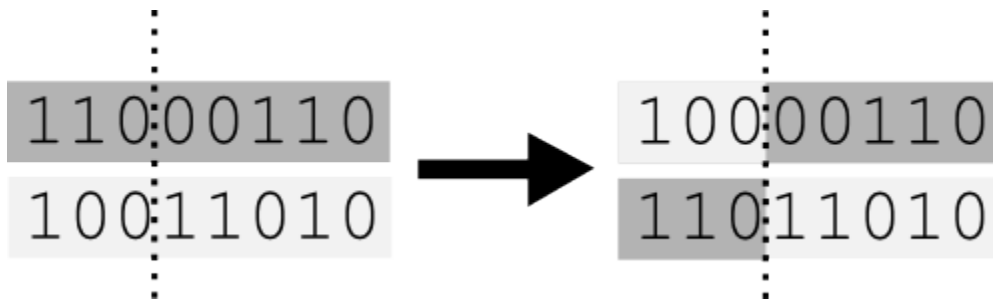


Figure II.2 Phase de croisement forme simple

- Le double enjambement repose sur le même principe, sauf qu'il y a deux pivots :

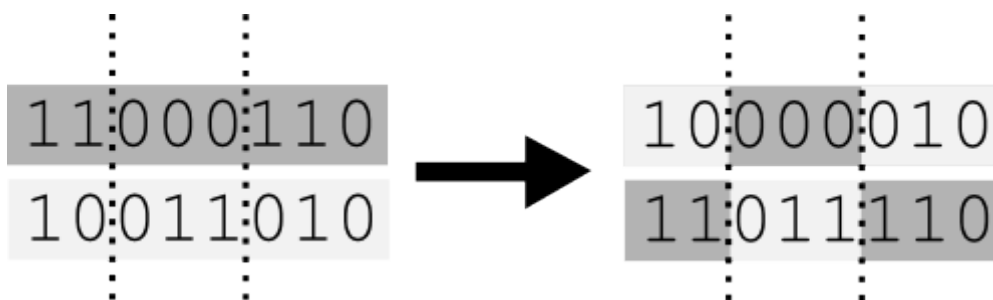


Figure II.3 Phase de croisement forme évoluée

On réalise ensuite une mutation sur les enfants obtenues lors du croisement.

❖ Mutation

La mutation consiste à altérer un gène dans un chromosome selon un facteur de mutation. Ce facteur est la probabilité qu'une mutation soit effectuée sur un individu.

Cet opérateur est l'application du *principe de variation* de la théorie de Darwin et permet, par la même occasion, d'éviter une convergence prématurée de l'algorithme vers un extremum local. Voici un exemple de mutation sur un individu ayant un seul chromosome :

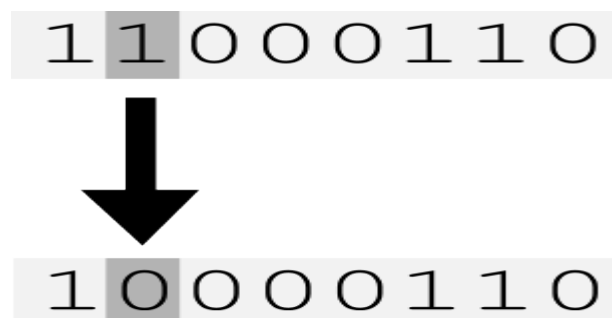


Figure II.4 Phase de mutation

Avec ces trois opérateurs d'évolution, nous pouvons appliquer les algorithmes génétiques

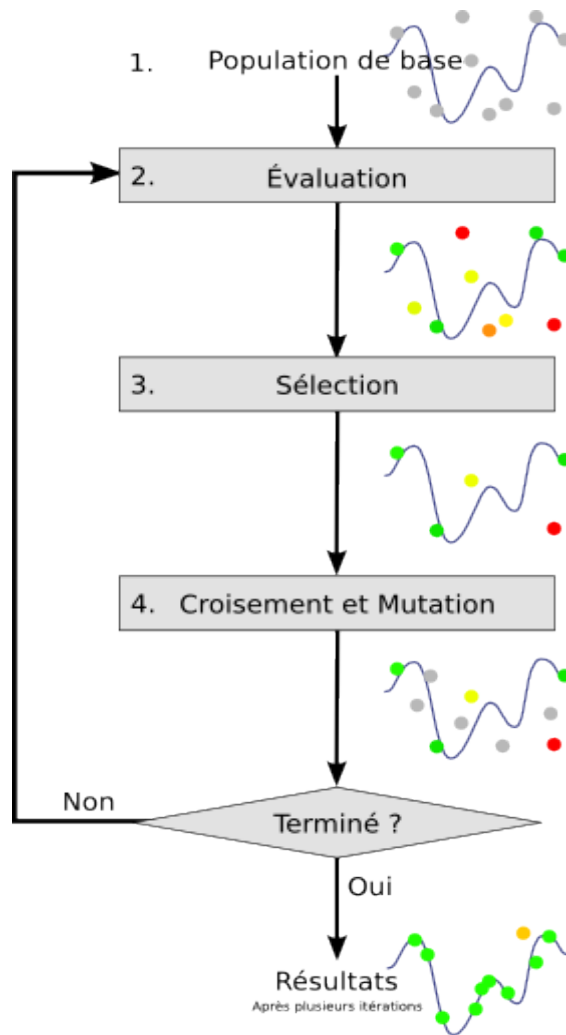


Figure II.5 Principe de l'algorithme génétique

II.3.2 Optimisation multi-objectif

L'optimisation multi-objectifs, également appelée optimisation multi-critères ou optimisation vectorielle, a été définie comme mathématique problèmes d'optimisation, qui trouve un vecteur de variables de décision qui satisfait les contraintes et donne des valeurs acceptables à toutes les fonctions objectives.

Dans ces problèmes, il y a plus d'une fonction objective ou fonction de coût (un vecteur d'objectifs) à optimiser (minimiser ou maximiser) simultanément, parfois des problèmes exigent une combinaison entre maximiser une fonction et minimiser une autre.

Ces objectifs sont souvent en conflit les uns avec les autres, donc une amélioration de l'un d'eux en détériorera un autre.

Par conséquent, il n'existe pas de solution optimale unique par rapport à toutes les fonctions objectives. Au lieu de cela, il existe un ensemble de solutions optimales, appelées Pareto optima ou solutions ou front de Pareto [28].

Au fil des années, un nombre important de techniques ont été produites par un nombre important de chercheurs opérationnels pour traiter problème d'optimisation objectif.

Cependant, il était relativement nouveau jusqu'à ce que la recherche a réalisé le potentiel des algorithmes évolutionnaires (EA) dans cette zone [29].

II.3.2.1 Définition

D'un point de vue formel, le problème d'optimisation multi-objectif (POMO), sous sa forme la plus générique, est posé comme suit [30]:

$$\text{minimiser } F(x) = [F_1(x), F_2(x), \dots, F_k(x)]^T$$

$$\text{Sujet aux contraintes : } g_j(x) \leq 0, j = 1, 2, 3, \dots, m$$

$$h_l(x) = 0, l = 1, 2, 3, \dots, e$$

$$a_i \leq x_i \leq b_i$$

Où k est le nombre de fonctions « objectif », m est le nombre de contraintes d'inégalité et e est le nombre de contraintes d'égalité. $x \in E^n$ est le vecteur de variables de décision, où n est le nombre de variables x_i .

Ces variables sont limitées, parfois, dans une plage de valeurs $[a_i b_i]$. $(x) \in E^k$ est le vecteur de fonctions « objectif » tel que $F(x) : E^n \rightarrow E^1$; $F_i(x)$ est appelée aussi objectif, critère, fonction de coût ou fonction de valeur.

Les k fonctions objectifs peuvent être linéaires ou non-linéaires, continues ou discrètes, et le vecteur de variables de décision peut également être continu ou discret.

L'espace de solutions faisables ou possibles X , appelé aussi espace de décision ou ensemble de contraintes, est défini comme suit [31]:

$$\{x | g_j(x) \leq 0, j = 1, 2, 3, \dots, m \text{ et } h_l(x) = 0, l = 1, 2, 3, \dots, e \text{ et } a_i \leq x_i \leq b_i\}$$

II.3.2.2 Les approches métaheuristiques

Une méta-heuristique est une méthode générique pour la résolution de problèmes combinatoires NP-difficiles.

La résolution de ces problèmes nécessite l'examen d'un très grand nombre (exponentiel) de combinaisons.

Les méta-heuristiques contournent le problème de l'explosion combinatoire en n'explorant délibérément qu'une partie de l'espace des combinaisons.

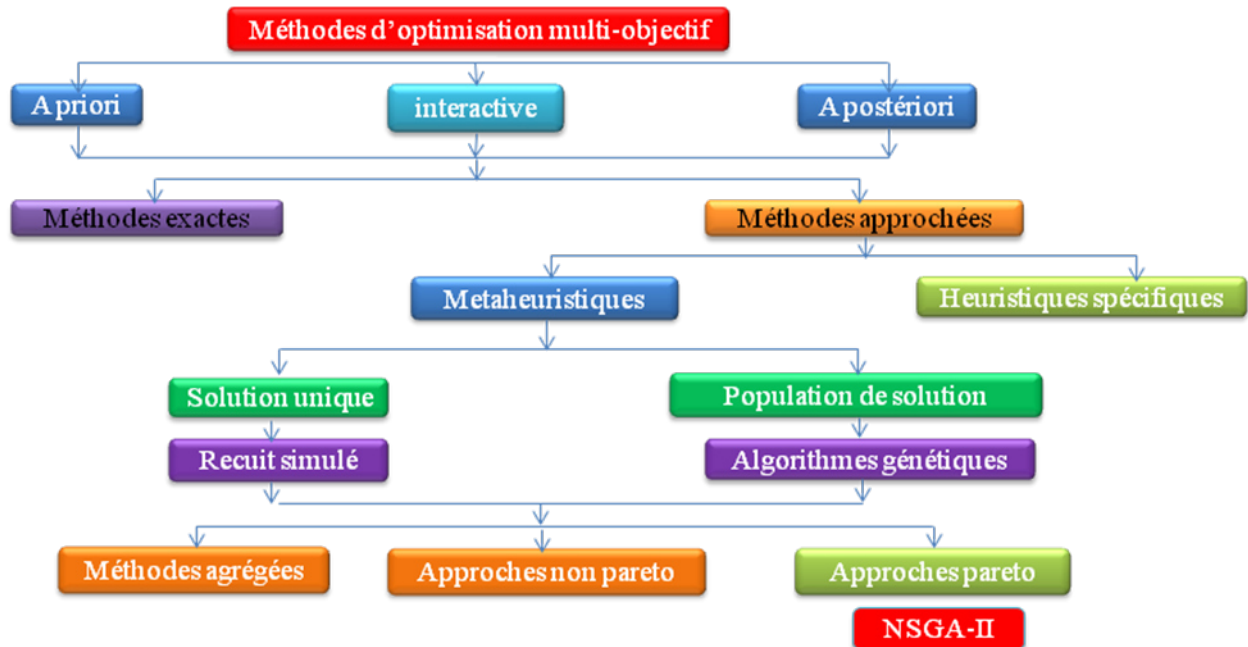


Figure II.6 Différentes techniques metaheuristique [32]

Contrairement à l'optimisation mono-objectif, l'optimisation multi-objectif est incapable de trouver une solution globale. Elle se base sur le principe de dominance pour rechercher un ensemble de solutions dites points non-dominés.

On dit que le vecteur x domine un vecteur y si et seulement si x est au moins aussi bon que y sur tous les objectifs et x est strictement meilleur que y sur au moins un objectif.

D'un point de vue formel, ceci s'exprime comme suit [33] :

$$(x) \leq F_i(y) \forall i \in \{1, 2, \dots, k\}$$

$$\text{et } \exists j \in \{1, 2, \dots, k\} \text{ tel que } F_j(x) < F_j(y)$$

Notons que pour toute paire (x, y) de l'espace de solutions réalisables, une et une seule des affirmations suivantes est vraie : x domine y , x est dominé par y ou x et y sont équivalentes (c-à-d. incomparables) au sens de dominance (voir la figure **Figure. 2.7**).

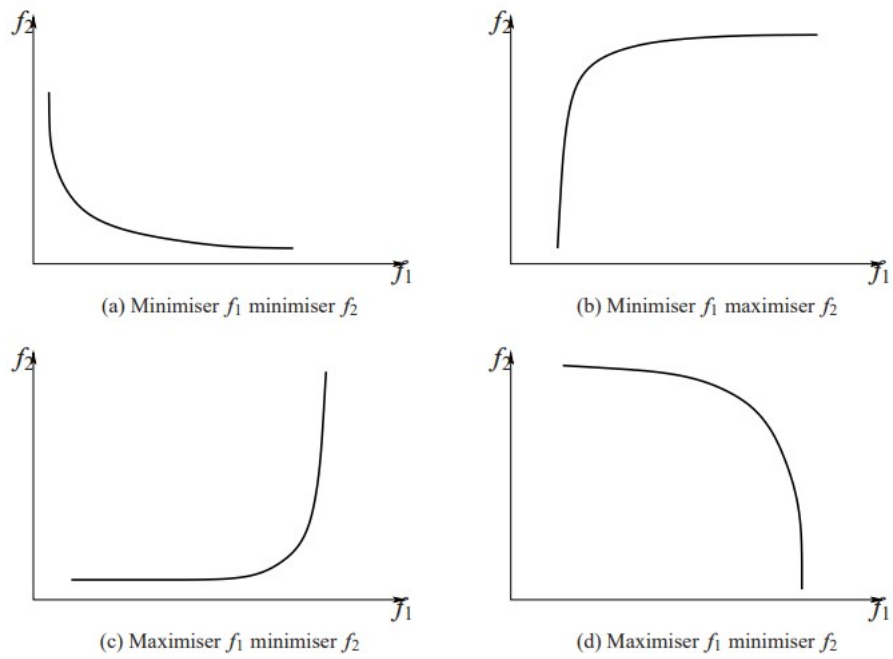


Figure II.7 Formes les plus courantes de surfaces de compromis dans le cas de deux objectifs [34]

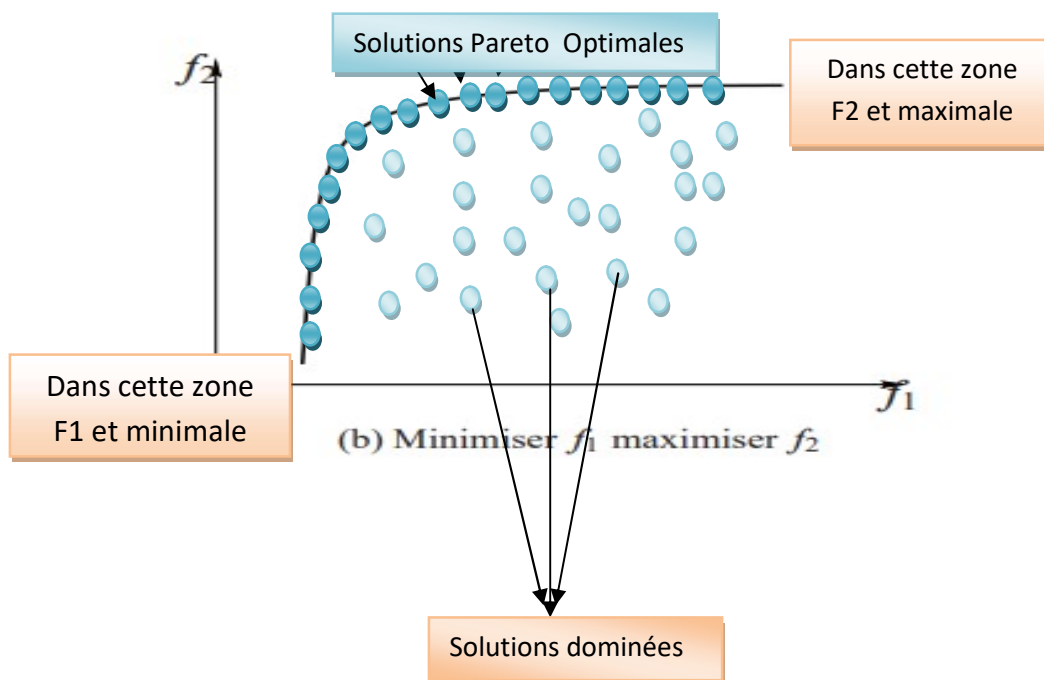


Figure II.8 Illustration des notions de dominance

L'optimalité de Pareto et front de Pareto peuvent être interpréter comme suit :

- Un point $x^* \in X$ est dit un optimum de Pareto (ou un point efficace global, non-dominé ou point non-inférieur) pour un POMO si et seulement s'il n'existe pas un $x \in X$ qui le domine (voir la figure **Figure II.8**).
- Toutes les solutions du front de Pareto sont des solutions sur le contour de l'espace des critères faisables. En d'autres termes, le front de Pareto ou la frontière Pareto est construite avec les solutions x^* qui dominant toutes les autres mais ne se dominant pas entre elles [34].

Les méthodes d'optimisation classiques sont séparées en deux catégories : recherche directe méthodes et méthodes basées sur les gradients. Dans la méthode de recherche directe, l'optimisation de problème peut être résolue en utilisant uniquement la fonction objective et les contraintes, tandis que dans la méthode basée sur le gradient, la première et la seconde dérivée de la fonction objective et des contraintes pourraient être utilisées. La première méthode prend trop de temps pour atteindre la solution à traiter ; et le défaut de la deuxième méthode est de résoudre les discontinuités les fonctions. Aussi, dans certaines méthodes classiques, la convergence de la solution optimisée dépend de la solution initiale choisie et la plupart des algorithmes préfèrent trouver la solution optimale. L'efficacité de cet algorithme vis-à-vis de divers problèmes d'optimisation est différente, et normalement, pour les problèmes avec discret frontières, manque leur efficacité.

L'algorithme évolutionnaire n'utilise pas la différenciation, et il est classé comme la méthode directe et leur application l'amplitude augmente [34].

Dans cette catégorie de méthodes de résolution, il existe :

- La classification ;
- L'approche scalaire ;
- L'approche Pareto,
- L'approche non scalaire et non Pareto.

Dans la littérature plusieurs Algorithmes évolutionnaire sont utilisés pour la résolution de problèmes multi-objectifs. C'est ainsi que durant le 19ème Siècle, Vilfredo Pareto [35], 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 objectif sans détériorer au moins un des autres objectifs. Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs contradictoires. Ces approches sont de type à posteriori. La notion de dominance au sens de Pareto repose sur les étapes suivantes :

Pour illustrer le concept de dominance, voici son principe : nous avons deux points A et B, on dit que A domine B quand la condition suivante est vérifiée avec **F1 : Fonction d'objectif 1 (Min)** et **F2 : Fonction d'objectif 2 (Max)**

Tel que

$$\{X1=F1(A) \quad / \quad Y1=F2(A)$$

$$X2=F1(B) \quad / \quad Y2=F2(B)\}$$

Si et seulement si ($X1 \leq X2$ et $Y2 \leq Y1$) et ($X1 < X2$ ou $Y1 < Y2$)

La relation de dominance est une relation d'ordre partiel stricte transitive, non réflexive et non antisymétrique (DUPAS, 2004). Une solution est dite Pareto optimale si elle n'est dominée par aucune autre solution réalisable.

Le front de Pareto, appelé aussi surface de compromis, est composé de l'ensemble des solutions Pareto-optimales qui forment une frontière d'optimalité. Ces solutions ne peuvent pas être comparées entre elles car aucune n'est systématiquement meilleure que les autres sur tous les objectifs [36]. C'est au décideur d'évaluer le front de Pareto en termes de qualité, de convergence et de diversité des solutions. De nombreux indicateurs de performances, comme l'indicateur de l'hypervolume, sont proposés et étudiés plus en détail dans [37]. Le décideur intervient également dans la sélection de la solution finale.

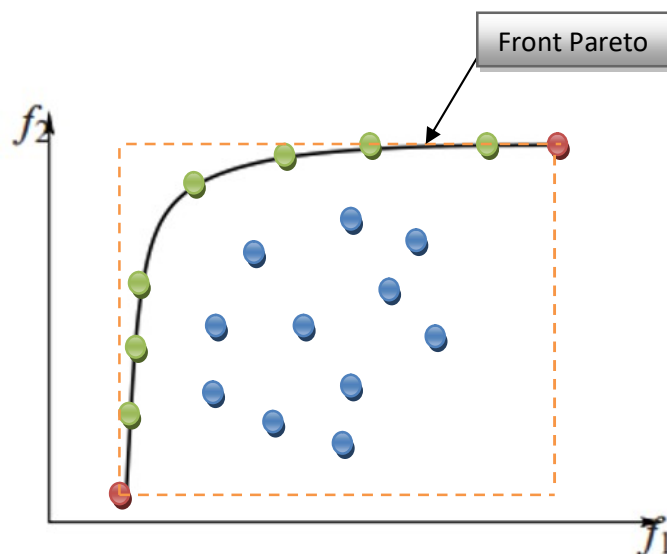


Figure II.9 : Illustration du Front de Pareto

Une des techniques utilisée pour ce type de problème s'appelle AG multi-objectifs (appelée MOGA Multi-Objectif Génétique Algorithmes). Elle a été introduite pour la première fois en 1993 [38].

Le MOGA a utilisé la classification non dominée d'une population GA. Un MOGA met explicitement l'accent sur les solutions non dominées et maintient simultanément la diversité des solutions non dominées.

Le MOGA diffère d'un GA tripartite standard dans la façon dont l'aptitude est attribuée à chaque solution dans la population. Les autres aspects de l'algorithme (la sélection universelle stochastique (SUS), un croisement en un seul point et une mutation au niveau du bit) sont les mêmes que l'AG classique [39].

L'algorithme génétique multi-objectifs (multi-objective génétique algorithme (MOGA) est l'une des nombreuses techniques d'optimisation de l'ingénierie, une méthode de recherche aléatoire guidée. Il convient à la résolution de problèmes liés à l'optimisation multi-objectifs avec la capacité d'explorer les diverses régions de l'espace de solution.

Ainsi, il est possible de rechercher un ensemble diversifié de solutions avec plus de variables qui peuvent être optimisées en même temps. Les solutions de MOGA sont illustrées à l'aide des fronts de Pareto. Un ensemble optimal de Pareto est un ensemble de solutions qui sont des solutions frontières non dominées.

Avec l'ensemble optimal de Pareto, les valeurs de la fonction objectif correspondante dans l'espace objectif sont appelées le front de Pareto. Les méthodes conventionnelles de résolution de problèmes multi-objectifs consistent en des recherches aléatoires, une programmation dynamique et des méthodes de gradient, tandis que les méthodes heuristiques modernes incluent le paradigme cognitif comme les réseaux de neurones artificiels, le recuit simulé et les approches lagrangiennes [40][41].

Dans de nombreuses applications réelles d'optimisations multi-objectives, une approximation du FP (Front de Pareto) est requise par un décideur pour sélectionner une solution préférée finale. La plupart des problèmes multi-Objectif (PMO) peuvent avoir vecteurs optimaux de Pareto nombreux ou même infinis.

Il est très temps-con-somme, sinon impossible, d'obtenir le FP complet. Sur le d'autre part, le décideur peut ne pas être intéressé à avoir un trop grand nombre de vecteurs optimaux de Pareto à traiter en raison d'un débordement d'informations. Par conséquent, de nombreux algorithmes d'optimisation consiste à trouver un nombre gérable de vecteurs optimaux de Pareto qui sont uniformément répartis le long du FP, et donc de bons représentants de l'ensemble du FP [42].

Quelques chercheurs se sont également efforcés d'approcher le FP en utilisant un modèle mathématique.

Il est bien connu qu'une solution optimale de Pareto à un PMO, dans des conditions douces, pourrait être une solution optimale d'un scalaire problème d'optimisation dans lequel l'objectif est une agrégation de tous les sous problèmes.

Par conséquent, l'approximation du FP peut être décomposée en un certain nombre de sous-problèmes d'optimisation d'objectif scalaire. C'est une idée de base derrière de nombreuses mathématiques traditionnelles méthodes de programmation pour approximer le PF. Nombreuses des méthodes pour construire des fonctions d'agrégation peuvent être trouvées dans la littérature. Les plus populaires parmi eux en- l'approche de la somme pondérée et l'approche de Tchebycheff [43]. Récemment, les méthodes d'intersection de frontières ont également attiré beaucoup d'attention.

Il n'y a pas de décomposition impliquée dans la majorité des algorithmes évolutifs multiobjectifs à la pointe de la technologie (MOEA : Multi Objective Evolutionnaire Algorithmes).

Ces algorithmes traitent un PMO comme un ensemble de contraintes à vérifier. Ils n'associent pas chaque solution individuelle à tout problème d'optimisation scalaire particulier.

Dans un objectif scalaire problème d'optimisation, toutes les solutions peuvent être comparées en fonction sur leurs valeurs de fonction objectif et la tâche d'un l'algorithme évolutif (EA) subjectif consiste souvent à trouver une seule solution optimale.

Dans les PMO, cependant, la domination ne définir un ordre complet parmi les solutions de l'objectif l'espace et les MOEA visent à produire un certain nombre d'optimums des solutions temporelles aussi diverses que possible pour représenter l'ensemble PF.

Par conséquent, les opérateurs de sélection conventionnels, qui ont été conçu à l'origine pour l'optimisation scalaire, ne peut pas être directement utilisé dans les MOEA sans décomposition.

On peut dire que s'il y a un schéma d'affectation de remise en forme pour l'affectation d'une solution individuelle une valeur de fitness relative pour refléter son utilité pour la sélection, puis les EA d'optimisation scalaire peuvent être facilement étendus pour traiter avec les MOP, bien que d'autres techniques telles que l'accouplement restriction, maintien de la diversité, quelques propriétés des POMs (Problème d'Optimisation Multi objectifs), et des populations externes peuvent également être nécessaires pour améliorer les performances de ces algorithmes étendus.

Pour cette raison, l'affectation de la condition physique a été un problème majeur dans Recherche MOEA.

Les stratégies d'affectation de fitness populaires inclure une affectation alternée de remise en forme basée sur des objectifs, comme l'algorithme génétique d'évaluation vectorielle (VEGA), et devoir de remise en forme basé sur la domination tel que Pareto archivé stratégie évolutive (PAES), force Pareto d'algorithme évolutif II (SPEA-II), et tri génétique non dominé algorithme II (NSGA-II) [44].

Nous notons ici que l'algorithme évolutif multi-objectif basé sur l'entropie avec un mécanisme d'élite amélioré (E-MOEA) améliore efficacement la convergence et la diversité des solutions définies dans les MOP. Dans cet algorithme, un mécanisme d'élite amélioré est appliqué pour guider la direction de l'évolution de la population. Plus précisément, il accélère la population pour approcher le vrai front de Pareto au début du processus d'évolution. Une stratégie basée sur l'entropie est aussi utilisée pour maintenir la diversité de la population lorsque la population est proche du front de Pareto [45].

D'autres méthodes n'utilisent aucun des deux principes énoncés précédemment. Elles possèdent un processus de recherche qui traite séparément les objectifs.

Par exemple, la méthode Vecteur Évaluâtes Génétique Algorithmes (VEGA) utilise le principe des algorithmes génétiques vus précédemment. La seule différence porte sur l'opérateur de sélection.

La population initiale est divisée en m sous-populations (m étant le nombre d'objectifs), chacune de ces sous-populations étant sélectionnée en fonction d'un des objectifs. La population est ensuite reconstruite et l'on applique les opérateurs génétiques de croisement et de mutation. Ainsi, seuls les individus répondant le mieux aux différents objectifs sont conservés [46].

II.4 NSGA II

Des changements importants ont été appliqués à la NSGA d'origine pour créer la NSGA-II. Le seul point commun entre les deux était l'adoption d'un tri non dominé. La NSGA-II dispose de l'archive Pareto P et de la population Q pour la recherche génétique, tout comme l'algorithme évolutif de force Pareto (SPEA). Cependant, il utilise les archives Pareto de manière plus agressive. Dans NSGA-II, le nombre d'individus dans l'archive P est N , et il est égal au nombre d'individus dans la population Q . En revanche, dans SPEA, le meilleur nombre d'individus dans l'archive P est généralement perçu comme unquatrième des individus de Q . Les opérateurs génétiques de croisement et de mutation sont effectués sur la population Q_t , et la sélection pour extraire la génération suivante a été appliquée sur l'union d'ensembles $P_t \cup Q_t$, créant P_{t+1} .

La sélection se compose de deux mécanismes : le « tri non dominé » et le « tri par distance d'encombrement ». Q_{t+1} a été généré à partir de P_{t+1} au moyen de la sélection, ce qu'on appelle la « sélection d'accouplement ».

Pendant ce temps, Zitzler [47] a amélioré son SPEA. SPEA2 a été proposé en 2001. SPEA2 était une amélioration par rapport à SPEA dans les domaines suivants : le mécanisme de calcul de fitness a été modifié pour assurer une convergence effective vers le front de Pareto, les archives P a le même nombre d'individus que la population Q , et une nouvelle méthode de réduction des individus dans l'archive (méthode de troncature d'archive) est utilisée. Le deuxième point mentionné ci-dessus est le même que le mécanisme utilisé dans NSGA-II.

De plus, SPEA2 et NSGA-II ont les points communs suivants entre eux : le croisement et la mutation sont effectués dans la recherche de la population Q archive pour la prochaine génération $pt+1$ est sélectionné parmi $P_t \cup Q_t$, et la prochaine population de recherche Q_{t+1} est sélectionné parmi P_{t+1} .

En d'autres termes, le schéma de l'algorithme est similaire à la fois dans NSGA-II et SPEA2, et les différences n'existent que dans la définition de l'aptitude, le mécanisme de réduction de l'archive et la sélection pour P_{t+1} . Aujourd'hui, ces deux algorithmes sont les principales méthodes de l'algorithme génétique multi-objectif. On dit aussi que la performance des deux est presque égale à [48].

NSGA-II est l'un des algorithmes d'optimisation multi-objectifs les plus populaires avec trois caractéristiques, approche de tri rapide non dominé, procédure d'estimation de distance encombrée rapide et opérateur de comparaison simple encombré. Deb et al. [49] a simulé plusieurs problèmes de test des précédentes études utilisant la technique d'optimisation NSGA-II, et il est affirmé que cette technique a surpassé PAES et SPEA en termes de recherche d'un ensemble diversifié de solutions.

II.4.1. L'algorithme NSGA II

Le schéma général de l'algorithme NSGA-II peut être grossièrement détaillé par les étapes suivantes.

Étape 1 : Initialisation du peuplement population

Initialisez la population en fonction de la plage du problème et de la contrainte.

Étape 2 : Tri non dominé

Processus de tri basé sur des critères de non domination de la population qui a été initialisée.

Étape 3 : Distance de Crowding

Une fois le tri terminé, la valeur de la distance d'entassement est attribuée par front. Les individus en La population est sélectionnée en fonction du rang et de la distance de surpeuplement.

Étape 4 : Sélection

La sélection des individus est effectuée à l'aide d'une sélection de tournoi binaire avec comparaison de foule opérateur.

Étape 5 : Opérateurs génétiques

GA réel codé utilisant un croisement binaire simulé et une mutation polynomiale.

Étape 6 : Recombinaison et sélection

L'algorithme de NSGA II est comme suit [50] :

```

1. Les entrées : L1 : liste d'individus ; // contient les ID des individus
2. Les sorties : L1 : liste d'individus avec un changement dans Rank,
3. Dominationset, Dominatedcountdes individus ;
4. Front : liste ;
5. Front = {} ; // Front de Pareto
6. Pour chaque individu
7.     Rank = 0, Dominationset = {}, Dominatedcount = 0 ;
8. Fin pour
9. Pour i = 1 à taille (L1)
10.     Pour j = i+1 à taille (L1)
11.         Si individu L1(i) domine individu L1(j) Alors
12.             Individu L1(i) .Dominationset = Dominationset + {L1(j)} ;
13.             Individu L1(j).Dominatedcount = Dominatedcount + 1 ;
14.         Fin si
15.         Si individu L1(j) domine individu L1(i) Alors
16.             Individu L1(i) .Dominatedcount = Dominatedcount + 1 ;
17.             Individu L1(j) .Dominationset = Dominationset + {L1(i)} ;
18.         Fin si
19.     Fin pour
20.         Si Individu i .Dominatedcount = 0 Alors
21.             Rank = 1 ; Front = Front + {L1(i)} ;
22.         Finsi
23.     Fin pour
24.     Rank-pop = individus ayant Rank égale 1 ;
25.     R = 2 ;
26.     Tant qu'il y a des individus avec Rank = 0 faire
27.         Pour chaque individu dans Rank-pop faire
28.             Pour chaque individu dans la liste Dominationsetfaire
29.                 Si individus ayant Dominatedcount = 0 et Rank = 0 alors Rank = R ;
30.                 Rank-pop = individus ayant Rank égale R ;
31.                 R = R + 1
32.             Finsi ;
33.         Finpour
34.     Finpour
35.     Fin tant que

```

La population de descendants et la population de la génération actuelle sont combinées et les individus de la génération suivante génération sont fixés par sélection. La nouvelle génération

est occupée par chaque front par la suite jusqu'à ce que la taille de la population dépasse la taille actuelle de la population [51].

La population est initialisée comme d'habitude. Une fois la population initialisée, la population est triée sur la base de la non-dominance dans chaque front.

Le premier front étant complètement non dominant dans la population actuelle et le deuxième front étant dominé par les individus du premier front seulement et le front continue ainsi.

Chaque les individus dans chaque front se voient attribuer des valeurs de classement (condition physique) ou basés sur le front dans auxquels ils appartiennent. Les individus du premier front reçoivent une valeur de fitness de **0** et les individus en seconde se voient attribuer une valeur de fitness de **1** et ainsi de suite.

En plus de la valeur de fitness, un nouveau paramètre appelé distance d'encombrement est calculé pour chaque individu. La distance d'encombrement est une mesure de la proximité d'un individu est à ses voisins.

Une grande distance de foule moyenne se traduira par une meilleure diversité dans la population. Les parents sont sélectionnés parmi la population en utilisant la sélection de tournois binaires en fonction du rang et de la distance d'entassement.

Un individu est sélectionné dans le rang est inférieur à l'autre ou si la distance d'encombrement est supérieure à l'autre 1.

La population génère des descendants à partir d'opérateurs de croisement et de mutation, qui seront discutés en détail dans une section ultérieure.

La population avec la population actuelle et les descendants actuels est à nouveau triée basé sur la non-dominance et seuls les N meilleurs individus sont sélectionnés, où N est la taille de la population.

La sélection est basée sur le rang et la distance d'encombrement sur le dernier front [52].

II.4.2 Description détaillée de NSGA-II

Dans cette section, nous allons décrire l'algorithme NSGA-II qui sera le fondement de notre contribution décrite dans le chapitre 3.

II.4.2.1 Initialisation de la population

La population est initialisée en fonction de l'étendue du problème et des contraintes éventuelles.

II.4.2.2 Tri non dominé

La population initialisée est triée sur la base de la non domination. L'algorithme de tri rapide est décrit ci-dessous [52]:

1. **Pour** chaque individu p de la population principale P , procédez comme suit
2. Initialiser $S_p = \emptyset$. Cet ensemble contiendrait tous les individus étant dominé par p .
3. Initialiser $n_p = 0$. Ce serait le nombre d'individus qui dominent p .
4. **Pour** chaque individu q dans P
5. **Si** p dominait q **Alors**
6. ajouter q à l'ensemble S_p c'est-à-dire $S_p = S_p \cup \{q\}$
7. **Sinon Si** q domine p **Alors**
8. Incrémenter le compteur de domination pour p c'est-à-dire $n_p = n_p + 1$
9. **Finsi**
10. **Si** $n_p = 0$ aucun individu ne domine p **Alors**
11. p appartient au premier front;
12. Définissez le rang de p individuel sur un, c'est-à-dire $r_p = 1$.
13. Mettez à jour le premier front défini en ajoutant p à l'avant ($F_1 = F_1 \cup \{p\}$)
14. **Finsi**
15. **Finsi**
16. **Finpour**
17. **Finpour**
18. Ceci est réalisé pour tous les individus de la population principale P .
19. Initialiser le compteur avant à 1
20. le suivi est effectué alors que le i ème front est non vide ($F_i \neq \emptyset$)
21. $Q = \emptyset$ L'ensemble pour ranger les individus pour le $(i + 1)$ ème front.
22. **Pour** chaque individu p devant F_i
23. **Pour** chaque individu q dans S_p (S_p est l'ensemble des individus dominés par p)
24. $n_q = n_q - 1$, décrémente le compte de domination pour l'individu q .
25. **Si** $n_q = 0$ **Alors** aucun des individus dans la suite des fronts domineraient q .
26. **Finsi**
27. **Finpour**
28. **Finpour**
29. Par conséquent, définissez $q_{\text{rank}} = i + 1$.
30. Mettre à jour l'ensemble Q avec l'individu q c'est-à-dire $Q = Q \cup q$.

Cet algorithme est meilleur que l'original NSGA car il utilise les informations sur l'ensemble qu'un individu domine (S_p) et le nombre d'individus qui dominent l'individu (n_p).

II.4.2.3 Distance de crowding

Une fois le tri non dominé terminé, la distance d'encombrement est attribuée. Étant donné que les individus sont sélectionnés en fonction du rang et de la distance d'encombrement, tous les individus de la population se voient attribuer une valeur de distance d'encombrement.

La distance d'encombrement est attribuée à l'avant et comparer la distance d'encombrement entre deux individus sur un front différent a moins de sens.

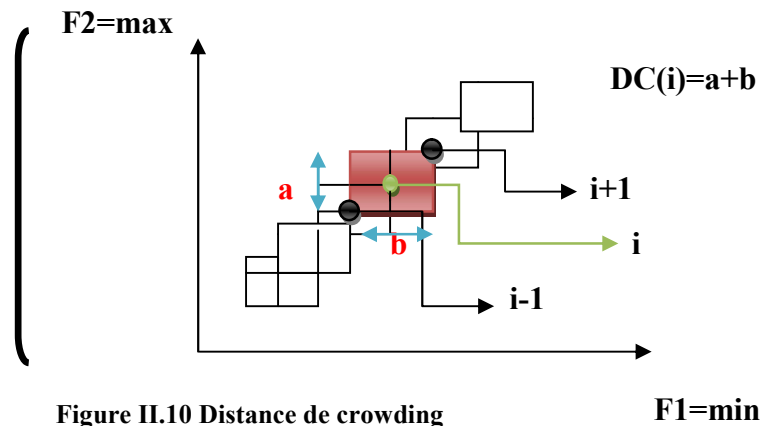


Figure II.10 Distance de crowding

F1=min

1. La distance de crowding est calculée comme ci-dessous :
2. **Pour** chaque front F_i , n est le nombre d'individus faire
3. initialiser la distance à zéro pour tous les individus soit $F_i(d_j) = 0$,
4. Où j correspond au $j^{\text{ième}}$ individu devant F_i .
5. **Fin pour**
6. **Pour** chaque fonction objective m **Faire**
7. Trier les individus devant F_i en fonction de l'objectif m c'est-à-dire $I = \text{trier}(F_i, m)$
8. Attribuez une distance infinie aux valeurs limites pour chaque individu
9. **Finpour**
10. dans F_i soit $I(d_1) = \infty$ et $I(d_n) = \infty$
11. **Pour** $k = 2$ à $(n - 1)$ **Faire**
12.
$$I(d_k) = I(d_k) + (|I(k + 1).m - I(k - 1).m|) / (f_m(\max) - f_m(\min))$$
13. **Finpour**
14. // $I(k).m$ est la valeur de la $m^{\text{ième}}$ fonction objectif de la $k^{\text{ième}}$ individu en //

L'idée de base derrière la distance de crowding est de trouver la distance euclidienne entre chaque individu dans un front en fonction de leurs objectifs m dans l'hyper espace à m dimensions. Les individus dans la frontière sont toujours sélectionnés car ils ont une affectation à distance infinie.

II.4.2.4 Sélection

Une fois les individus triés sur la base de la non-domination et avec une distance d'encombrement attribuée, la sélection est effectuée à l'aide d'un opérateur de comparaison d'encombrement.

NSGA-II contient un opérateur de sélection qui est basé sur un calcul de la distance de crowding. NSGA-II s'inspire d'une approche explorant un algorithme élitiste qui n'exploite pas d'archive externe pour stocker l'élite.

Pour gérer l'élitisme, NSGA-II assure qu'à chaque nouvelle génération les meilleurs individus rencontrés soient conservés.

Cet algorithme est l'un des algorithmes les plus efficaces pour trouver l'ensemble optimal de Pareto avec une excellente variété des solutions.

Chaque individu i de la population possède deux attributs : rang de non domination i_{rank} et distance crowding d_i .

Un opérateur de comparaison défini en fonction de ces deux attributs permet de guider le processus de la sélection avec la répartition uniforme des solutions Pareto. Soient deux individus i et j , on dit que i est meilleur que j si : $[(i_{\text{rank}} < j_{\text{rank}})]$ ou $[(i_{\text{rank}} = j_{\text{rank}}) \text{ et } (d_i > d_j)]$.

En se basant sur cette relation pour la comparaison de deux solutions non dominées appartenant à deux fronts Pareto, seule la solution appartenant au front Pareto d'ordre le plus faible est conservée.

Dans le cas où les deux solutions appartiennent au même front de Pareto (le dernier front pour compléter la taille de la population parent), la solution qui a la distance crowding la plus élevée est prise en compte.

Le tri de Crowding des points de dernier front pour compléter la taille N est pris dans l'ordre décroissant de leurs valeurs de distance Crowding, alors les points de la partie supérieure de la liste ordonnée sont choisis.

La distance de Crowding d_i du point i est une mesure de l'espace objectif autour de i en estimant le périmètre du cuboïde ou hypercube formé en utilisant les voisins les plus proches dans l'espace objectif.

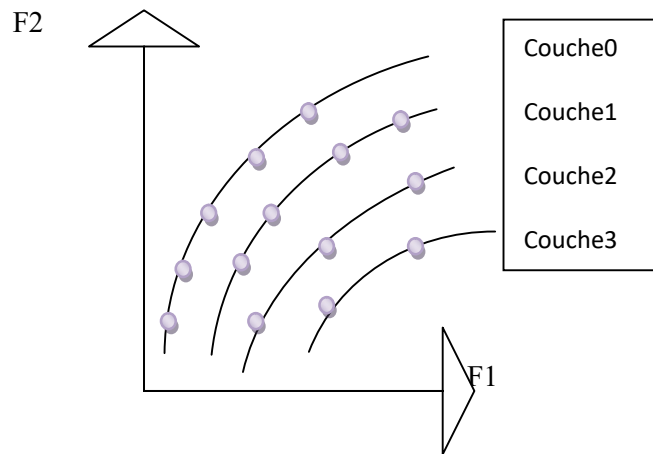


Figure II.11 Schéma de fonctionnement de NSGA-II

II.4.3 NSGA III

Les techniques évolutives sont capables de surmonter les difficultés associées aux méthodes classiques telles que l'exécution multiple. L'objectif de la technique d'optimisation multi-objectifs est non seulement d'orienter la recherche vers le front Pareto optimal mais aussi de préserver la diversité de la population dans l'ensemble des solutions non dominées.

La NSGA-III récemment proposée [53] est une technique puissante pour éliminer les inconvénients de NSGA-II tels que le manque de diversité uniforme et l'absence d'opérateur préservant la diversité latérale parmi les meilleures solutions actuelles non dominées.

NSGA-III commence avec une population aléatoire de taille N et un ensemble de prépoints de référence H spécifiés de dimension M sur un hyperplan unitaire ayant un vecteur normal de ceux couvrant l'ensemble de la RM^+ région. L'hyper-plan est placé de manière à ce qu'il coupe chaque axe objectif à un.

La technique de Das et Dennis [54] est utilisée pour placer $H = (M + p - 1)p$ référence points sur l'hyper-plan ayant $(p + 1)$ points le long de chaque frontière.

La population la taille N est choisie pour être le plus petit multiple de quatre supérieurs à H , avec l'idée que pour chaque point de référence, on s'attend à trouver un membre de la population.

À une génération t , les opérations suivantes sont effectuées. Premièrement, la population entière P_t est classée en différents niveaux de non-dominance, de la même manière que dans NSGA-II, suivant le principe du tri non dominé.

Une population de descendants Q_t est créée à partir de P_t en utilisant des opérateurs habituels de recombinaison et de mutation. Étant donné qu'un seul membre de la population est attendu à trouver pour chaque point de référence, aucune opération de sélection n'est nécessaire dans NSGA-III, car tout opérateur de sélection permettra d'établir une compétition entre différents points de référence.

Une population combinée $R_t = P_t \cup Q_t$ est alors formée. Par la suite, les points à partir du premier le front non dominé est sélectionné pour P_{t+1} une par une jusqu'à ce que toutes les solutions d'un front complet ne peut pas être inclus.

Cette procédure est également identique à celle de NSGA-II. Notons en finale avant qui n'a pas pu être complètement sélectionné comme FL.

En général, seules quelques solutions de FL ont besoin à sélectionner pour P_{t+1} à l'aide d'un opérateur de conservation de niche, que nous décrivons ensuite.

Premièrement, chaque membre de la population de P_{t+1} et FL est normalisée en utilisant la répartition actuelle de la population de sorte que tous les vecteurs objectifs et points de référence ont des valeurs correspondantes.

Par la suite, chaque membre de P_{t+1} et FL est associé à un point de référence spécifique en utilisant la plus courte perpendiculaire distance ($d()$) de chaque membre de la population avec une ligne de référence créée en joignant l'origine avec un point de référence fourni.

Ensuite, une stratégie de niche prudente est employée pour choisir ces membres associés aux points de référence les moins représentés dans P_{t+1} .

La stratégie de niche met l'accent sur la sélection d'un membre de la population pour autant de points de référence fournis que possible. Un membre de la population associé à une référence sous-représentée ou non représentée est immédiatement préféré.

Avec un accent continu pour souligner les individus non-dominés, l'ensemble du processus devrait alors trouver un membre de la population correspondant à chaque point de référence fourni proche du front Pareto-optimal, à condition que la variation génétique des opérateurs (recombinaison et mutation) sont capables de produire des solutions respectives.

L'utilisation d'un point de référence bien réparti assure un ensemble bien réparti de points de compromis à la fin.

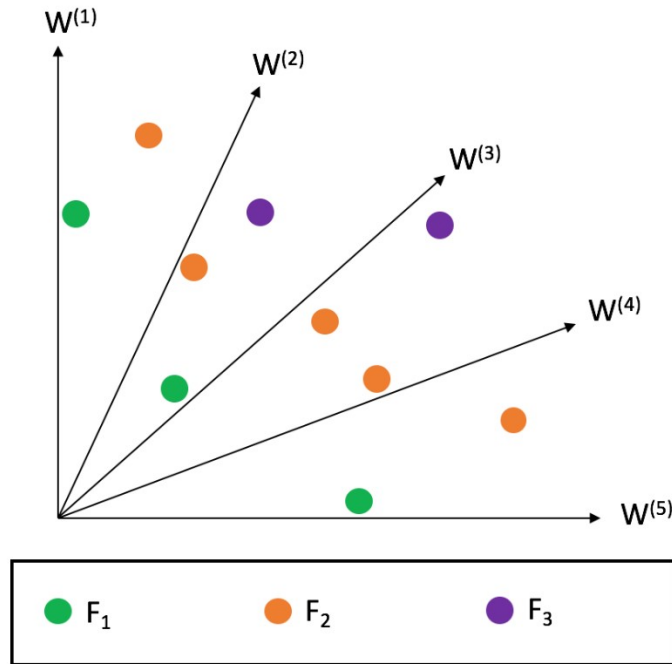


Figure II.12 Modèle de NSGA III

L'étude originale NSGA-III [55] s'est avérée efficace de 3 à 15 objectifs DTLZ et d'autres problèmes. Un aspect clé de la NSGA-III est qu'elle ne nécessite aucun paramètre supplémentaire.

La méthode a également été étendue pour gérer les contraintes sans introduire tout nouveau paramètre.

Cette étude a également introduit une approche de calcul rapide par laquelle l'ensemble de points de référence est mis à jour de manière adaptative à la volée en fonction de l'état d'association de chaque référence sur plusieurs générations. L'algorithme est décrit comme suit :

Entrée : H points de référence structurés Z_s ou points d'aspiration fournis Z_a , population parente P_t

Sortie : P_{t+1}

1 : $S_t = \emptyset, i = 1$

2: $Q_t = \text{Recombinaison} + \text{Mutation}(P_t)$

3 : $R_t = P_t \cup Q_t$

4: $(F_1, F_2, \dots) = \text{Non-dominé-sort}(R_t)$

5: **Répéter**

6 : $S_t = S_t \cup F_i$ et $i = i + 1$

7 : **jusqu'à** $|S_t| = N$

8 : Dernière façade à inclure : $F_1 = F_i$

9 : **Si** $|S_t| = N$ **Alors**

10 : $P_{t+1} = S_t$, pause

11 : **Sinon**

12 : $P_{t+1} = \bigcup_{j=1}^{l-1} F_j$

13 : Points à choisir parmi F_1 : $K = N - |P_{t+1}|$

14 : Normaliser les objectifs et créer l'ensemble de référence Z^r : Normaliser(f^n, S_t, Z^r, Z^s, Z^a)

15 : Associer chaque membre s de S_t à un point de référence : $[\pi(s), d(s)] = \text{Associer}(S_t, Z^r)$

% $\pi(s)$: point de référence le plus proche, d : distance entre s et $\pi(s)$

16 : Calculer le nombre de niches du point de référence $j \in Z^r$: $\rho_j = \sum_{s \in S_t} \mathbb{1}(\pi(s) = j) ? 1 : 0$

17 : choisissez K membres un à la fois de F_1 pour construire P_{t+1} :

Niching($K, \rho_j, \pi, d, Z^r, F_1, P_{t+1}$)

18 : **Finsi**

II.5 La classification

Pour la tâche de classification nous nous sommes intéressés à la famille Boosting et de ses algorithmes. De la famille dite de Boosting, on trouve des techniques et les plus utilisées dans le domaine de la prédiction sont :

- ❖ Gradient Tree Boosting
- ❖ AdaBoost (**Adaptive Boosting**)
- ❖ XGBoost

II.5.1 Gradient Tree Boost

XGBoost signifie «Extreme Gradient Boosting», où le terme «Gradient Boosting» provient de l'article Greedy Function Approximation: A Gradient Boosting Machine, par Friedman.

Les arbres à gradient renforcé existent depuis un certain temps et il existe de nombreux documents sur le sujet.

Ce didacticiel expliquera les arbres boostés de manière autonome et fondée sur des principes en utilisant les éléments de l'apprentissage supervisé. Nous pensons que cette explication est plus propre, plus formelle et motive la formulation du modèle utilisé dans XGBoost.

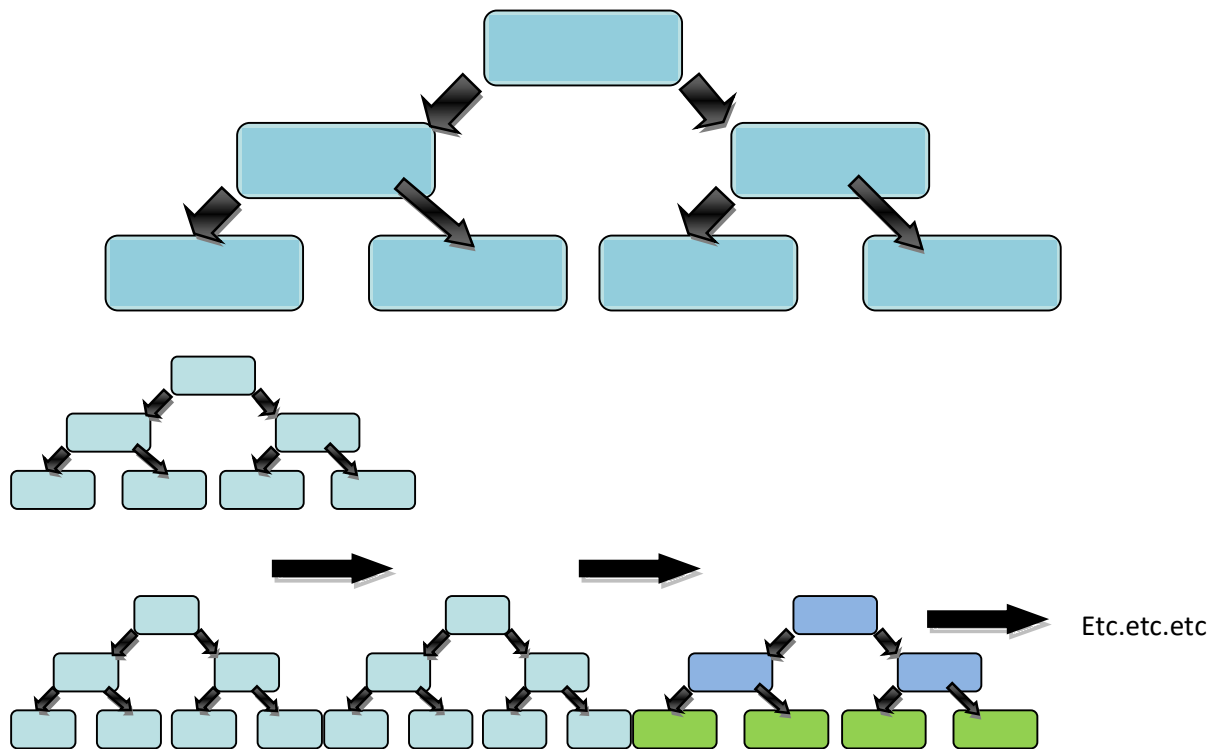


Figure II.13 Gradient Boost

II.5.2 AdaBoost

AdaBoost est une méthode d'apprentissage d'ensemble (également connue sous le nom de "méta-apprentissage") qui a été initialement créée pour augmenter l'efficacité des classificateurs binaires. AdaBoost utilise une approche itérative pour apprendre des erreurs des classificateurs faibles et les transformer en forts.

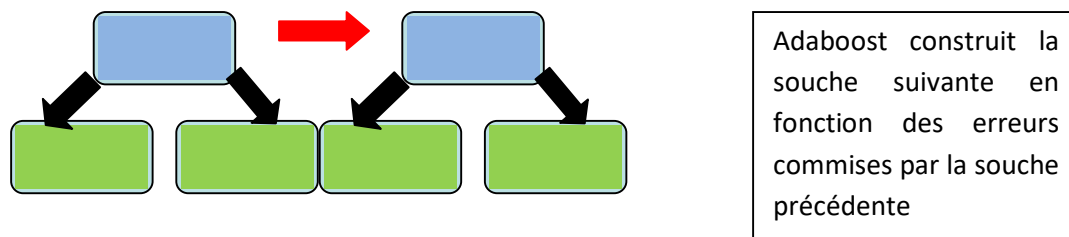


Figure II.14 AdaBoost (a)

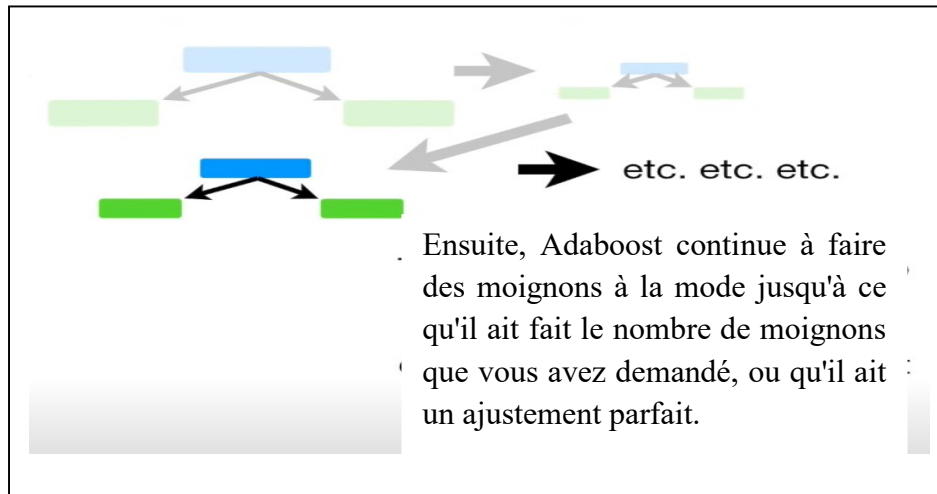


Figure II.14 AdaBoost (b)

II.5.3 XGBoost

XGBoost, qui signifie Extreme Gradient Boosting, est une bibliothèque d'apprentissage automatique évolutive et distribuée à arbre de décision boosté par gradient (GBDT). Il fournit une amplification d'arbre parallèle et est la principale bibliothèque d'apprentissage automatique pour les problèmes de régression, de classification et de classement. Il est essentiel pour comprendre XGBoost de comprendre d'abord les concepts et les algorithmes d'apprentissage automatique sur lesquels XGBoost s'appuie : apprentissage automatique supervisé, arbres de décision, apprentissage d'ensemble et amplification de gradient. L'apprentissage automatique supervisé utilise des algorithmes pour former un modèle afin de trouver des modèles dans un ensemble de données avec des étiquettes et des caractéristiques, puis utilise le modèle formé pour prédire les étiquettes sur les caractéristiques d'un nouvel ensemble de données.

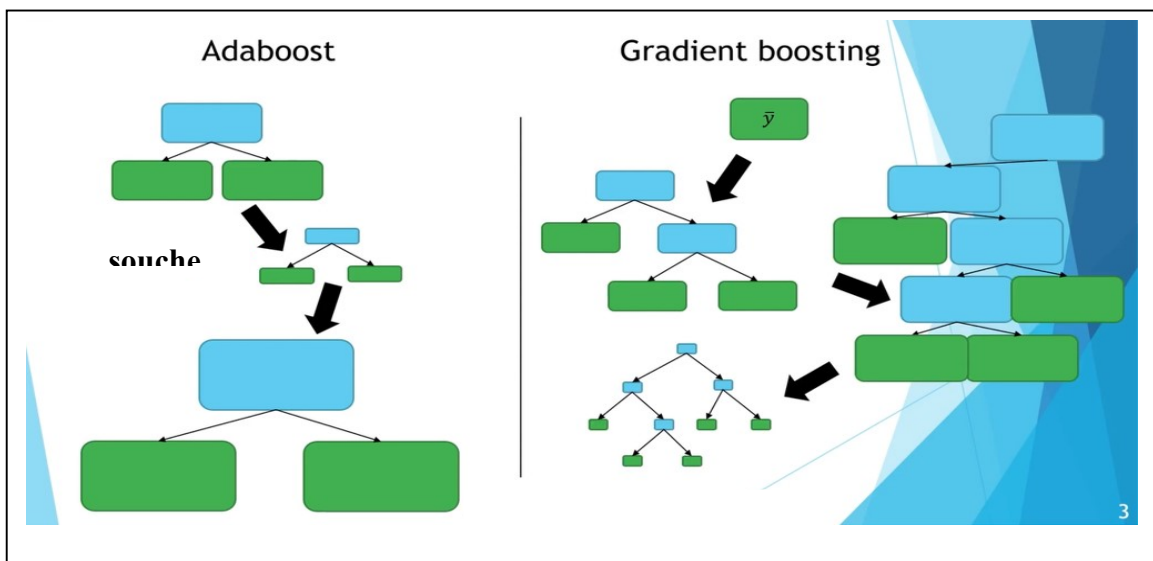


Figure II.15 XGBoost

Comme classifieur, nous avons utilisé dans notre projet celui appelé AdaBoost. Aujourd'hui, l'apprentissage automatique est la prémisse de grandes innovations et promet de continuer à permettre aux entreprises de prendre les meilleures décisions grâce à des prédictions précises. Mais que se passe-t-il lorsque la sensibilité aux erreurs de ces algorithmes est élevée et inexplicable ?

II.6 Présentation de l'algorithme AdaBoost

AdaBoost (Adaptive Boosting) est une technique de boost très populaire qui vise à combiner plusieurs classificateurs faibles pour créer un classificateur fort. L'article original d'AdaBoost a été rédigé par Yoav Freund et Robert Schapire [55].

Un seul classificateur peut ne pas être en mesure de prédire avec précision la classe d'un objet, mais lorsque nous regroupons plusieurs classificateurs faibles, chacun apprenant progressivement des objets mal classés des autres, nous pouvons construire un tel modèle fort. Le classificateur mentionné ici peut être n'importe lequel de vos classificateurs de base, des arbres de décision (souvent la valeur par défaut) à la régression logistique, etc.

Un classificateur faible est un classificateur qui fonctionne mieux que la supposition aléatoire, mais qui réussit toujours mal à désigner des classes aux objets. Par exemple, un classificateur faible peut prédire que toute personne âgée de plus de 40 ans ne pourrait pas courir un marathon, mais que les personnes en dessous de cet âge le pourraient.

Maintenant, vous pourriez obtenir une précision supérieure à 60 %, mais vous seriez toujours en train de mal classer de nombreux points de données. Plutôt que d'être un modèle en soi, AdaBoost peut être appliqué au-dessus de n'importe quel classificateur pour apprendre de ses lacunes et proposer un modèle plus précis. Il est généralement appelé le "meilleur classificateur prêt à l'emploi" pour cette raison [56].

Le principe de fonctionnement de l'algorithme de AdaBoost est comme suit [56] :

Étape 1 : Un classificateur faible (par exemple, une souche de décision) est créé au-dessus des données d'apprentissage basées sur les échantillons pondérés. Ici, les poids de chaque échantillon indiquent à quel point il est important d'être correctement classé. Initialement, pour la première souche, nous donnons à tous les échantillons des poids égaux.

Étape 2 : Nous créons une souche de décision pour chaque variable et voyons dans quelle mesure chaque souche classe les échantillons dans leurs classes cibles. Par exemple, dans le diagramme ci-dessous, nous vérifions l'âge, la malbouffe et l'exercice. Nous

examinerons le nombre d'échantillons correctement ou incorrectement classés comme aptes ou inaptes pour chaque souche individuelle.

Étape 3 : Plus de poids est attribué aux échantillons mal classés afin qu'ils soient correctement classés dans la souche de décision suivante. Le poids est également attribué à chaque classificateur en fonction de la précision du classificateur, ce qui signifie une grande précision = un poids élevé !

Étape 4 : réitérez à partir de l'étape 2 jusqu'à ce que tous les points de données aient été correctement classés ou que le niveau d'itération maximal ait été atteint.

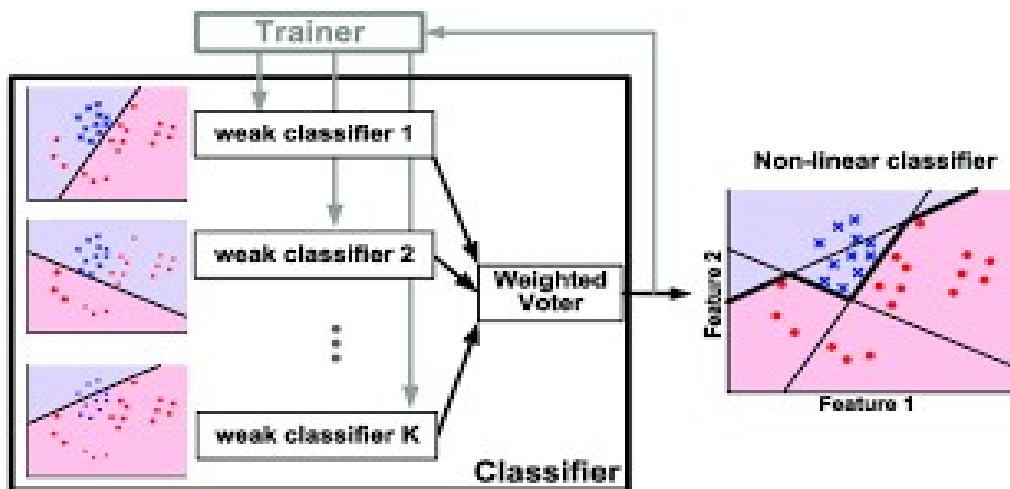


Figure II.16 structure AdaBoost

II.7 Conclusion

Dans ce chapitre nous avons dressé un étatde l'art sur les principaux concepts de l'optimisation combinatoire et les différentes approches de résolutions de problème nécessitant une technique d'optimisation. Nous avons explicité avec détails l'algorithme NSGA II car il a fait preuve de très bons résultats en matière de classification, il sera l'objet de notre principale contribution conceptuelle et expérimentale dans les chapitres suivants.

III.1 Introduction

L'analyse de données est devenue un grand enjeu scientifique, économique et surtout vital. Ce dernier est la conséquence de la pandémie du COVID-19 dont la maîtrise de son phénomène et la prise de décision sont une clé majeure pour la vie humaine à l'échelle mondiale.

A cet effet, plusieurs travaux ont vu le jour pour répondre aux besoins tant attendus par les décideurs. Ces travaux se placent dans une approche de « data science » ou analyse de données et dans le souci majeur porté sur l'extraction de la connaissance pour l'aide à la décision prédictive.

Dans ce chapitre, nous nous intéressons à l'utilisation de techniques de classification et d'analyse de données dite NSGA II pour sa spécification formelle et sa mise en œuvre sur des données concernant COVID-19.

III.2 NSGA-II pour la prédiction de COVID-19 en utilisant les symptômes des patients

COVID-19 est une pandémie qui a laissé et qui est encore virulente suite à son impact trop cruel dans le monde. La maîtrise de la donnée et son traitement via des algorithmes performants sont un besoin très demandé pour la maîtrise de l'acte médical préventif et d'aide à la décision. C'est ainsi que notre projet porte sur le développement d'un système d'aide à la décision appliqué sur COVID-19 en appliquant une technique de l'optimisation multi-objectif ; nous avons choisi NSGA-II.

III.3 Conception globale

Dans cette section, nous allons décrire notre proposition conceptuelle pour la résolution du problème de COVID-19 par **NSGA-II**.

Nous introduisons notre modèle de « machine learning » hybride qui est basé sur une phase de sélection des caractéristiques et une phase de classification pour la prédiction d'un malade atteint de COVID-19. Une vue globale est décrite dans la figure **Fig.III.1**.

La première phase utilise l'algorithme NSGA-II pour sélectionner le sous-ensemble optimal des caractéristiques qui satisfait le conflit des fonctions objectives pour l'apprentissage.

La seconde phase concerne l'apprentissage de notre modèle en utilisant le classifieur Adaboost pour prédire la classe correspondante.

Finalement, une phase d'évaluation est réalisée pour valoriser l'efficacité du modèle en utilisant des données de test.

L'architecture fonctionnelle de notre système est comme suit :

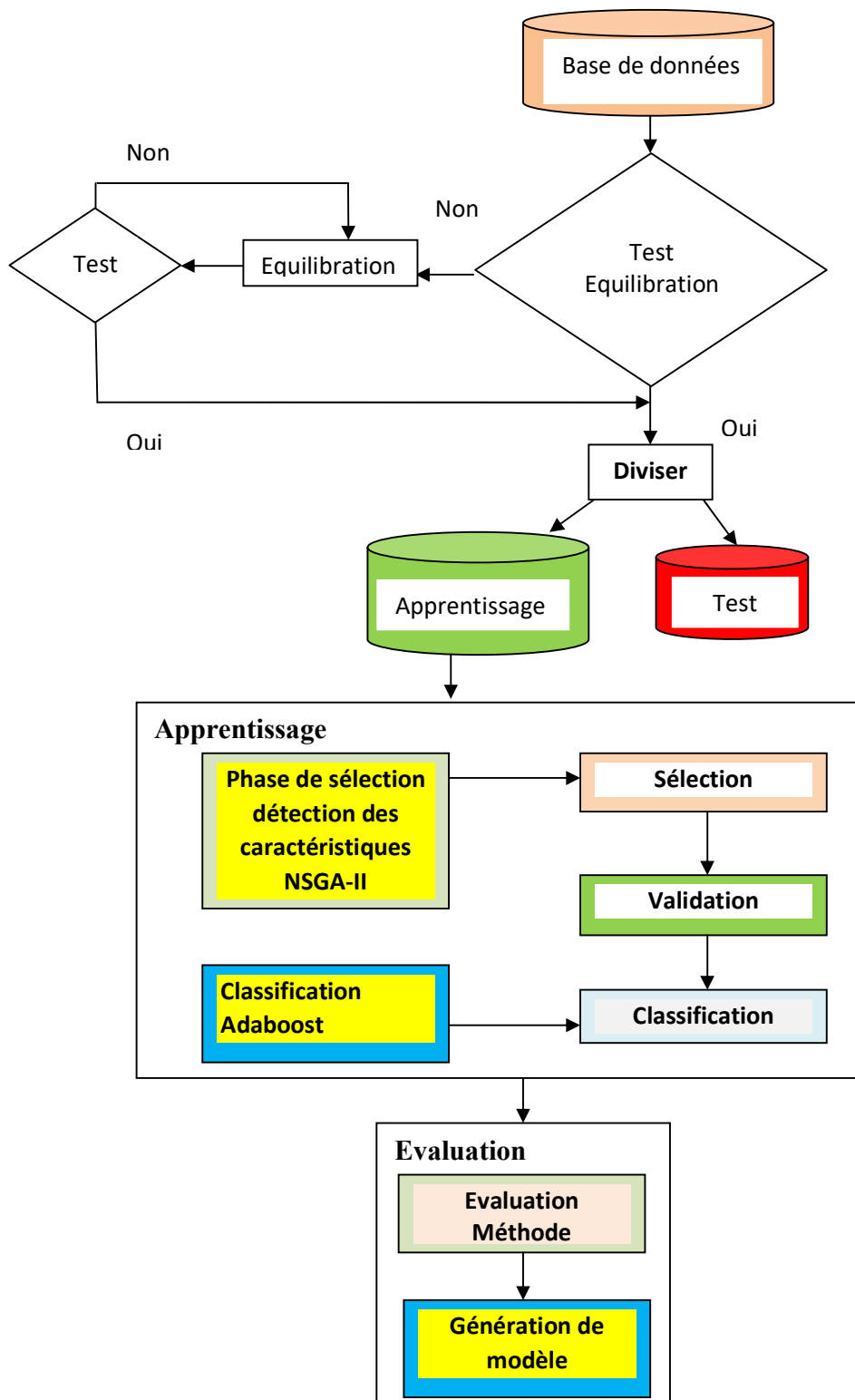


Figure III.1 Architecture globale de système

Comme présenté dans le schéma précédent, notre système se base sur la division de la base de données en deux sous bases, l'une pour l'apprentissage et l'autre pour le teste. Le système est composée essentiellement de trois phases qui sont ; la phase de détection des caractéristiques, la phase de classification et enfin la phase d'évaluation.

III.4 Conception détaillée

Dans cette section nous allons décrire de manière détaillée les différentes parties qui composent notre contribution.

III.4.1 Phase d'apprentissage

Dans cette section, nous allons explicite le principe d'apprentissage utilise dans notre système.

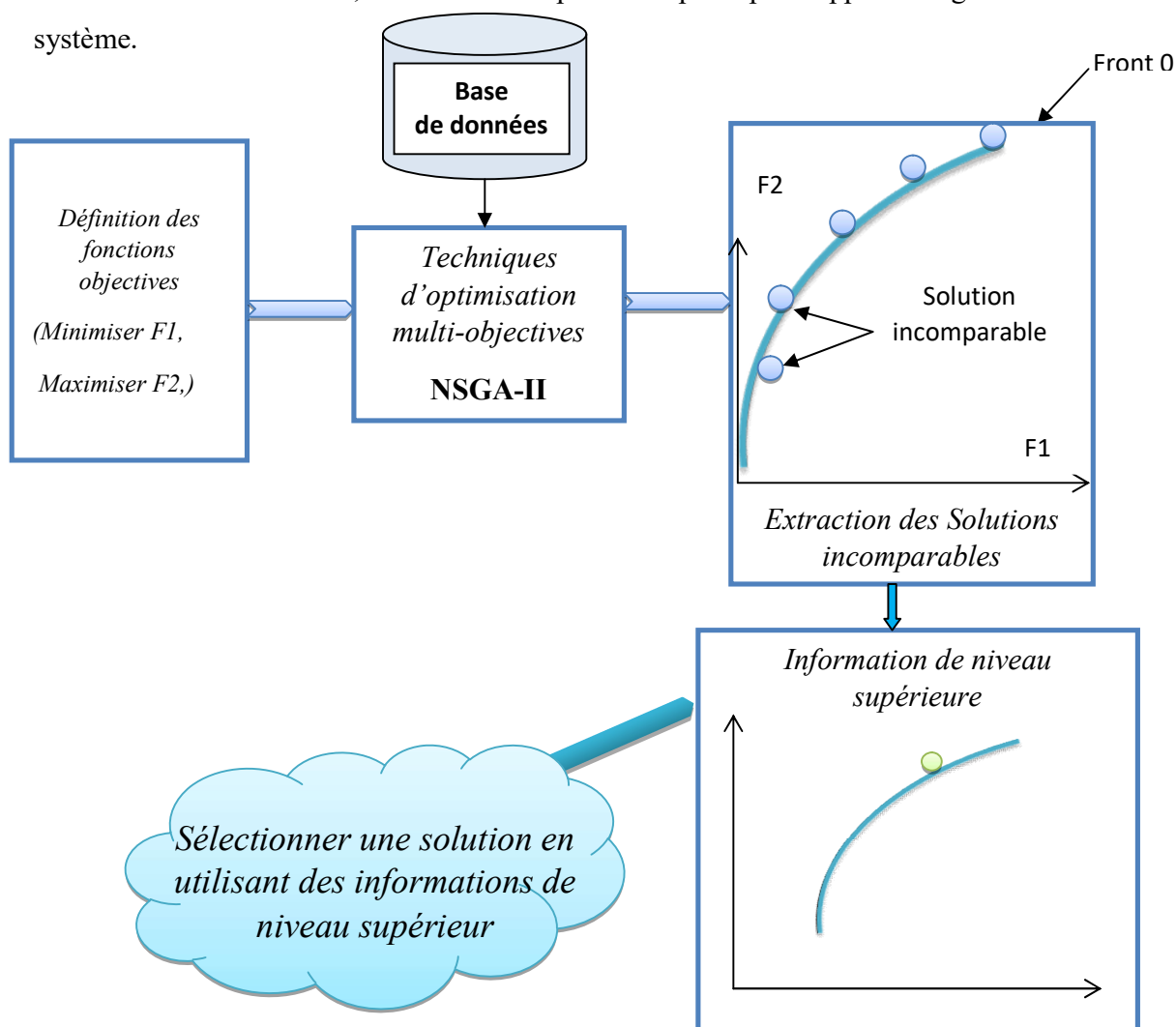
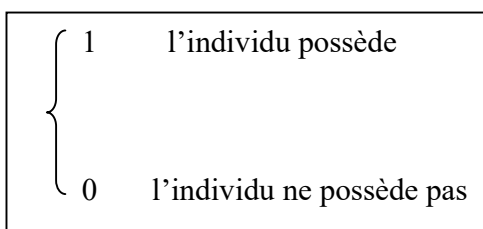


Figure III.2 Représentation générale des résultats de NSGA-II

III.4.2. Modélisation de l'individu

Une fois que la population est sélectionnée, nous avons utilisé une matrice comme une structure de représentation et de stockage. Dans notre expérimentation, nous avons choisi la taille de la population de 100 individus. Nous avons choisi une représentation binaire de l'individu sachant qu'il est formé par huit bits qui représentent le nombre de symptômes :



La liste des symptômes est :

toux, fièvre, mal de gorge, essoufflement, mal de tête, âge_60_et_plus, sexe, Indication-test

Symptômes

	toux	fièvre	mal de gorge	essoufflement	mal de tête	corona résultat	âge_60_et_plus	sexe	Indication-test
P									
a	1	1	1	1	0	0	0	0	0
t	1	1	0	0	0	0	0	1	1
i	0	1	1	0	1	0	1	0	0
e	0	1	1	1	1	1	1	0	0
n	1	0	0	0	1	1	1	1	0
t
s									

Tableau III.1 Individu (Forme binaire, Taille=nombre des symptômes).

III.4.3 Sélection

La phase de sélection est une étape cruciale dans le processus de l'algorithme génétique. Pour notre système, son fonctionnement est décrit par le diagramme suivant :

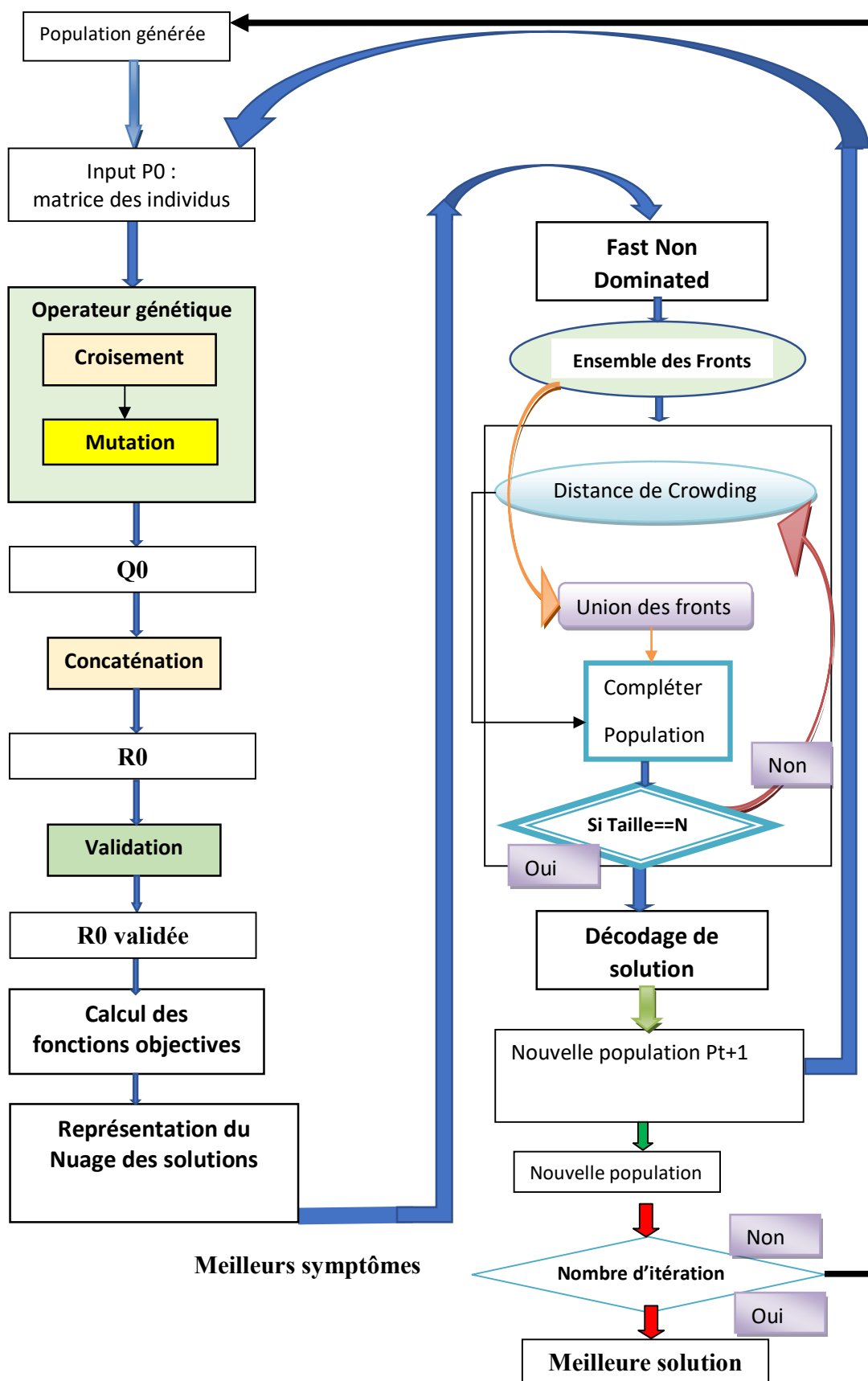


Figure III.3 le système NSGA-II

III.4.4 Opérateurs génétique

Dans ce paragraphe, nous allons explicités les opérateurs génétiques utilisés dans notre système.

III.4.4.1 Procédure de Croisement

Pour réaliser ce processus, l'opérateur de croisement permet d'exploité au mieux les individus parents en s'appuyant sur un certain nombre de gènes et appliquons sur cette base le croisement, comme suit:

Exemple : Extraire une sous matrice à partir P0 pour l'opérateur génétique

Test Covid-19

0	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1
1	1	0	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1

- On utilise 'random' pour le choix aléatoire sur les parents
- Puis on fixe le nombre des gènes qui choisit par exemple 'Nombres gènes = $k = 3$ ' telle que $k =$ de l'indice (0 jusqu'à 2)

Exemple appliqué :

- $K=3$
- Les parents après l'utilisation de 'random' en fait Croisement entre individu de l'indice 0 et de l'indice 2

K

←————→

0	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1
1	1	0	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1

Résultat après croisement :

1	1	0	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1
0	0	1	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1

(A)

La procédure de croisement est modélisée comme suit : Fonction croisement(MAT)

```

1  n = nombre des lignes
2  m = nombre des colonnes
3  I = aléatoire (0, n)
4  J = aléatoire (0, n)
5  Tant que (J == I) faire
6      J = aléatoire (0, n)
7  Tant Que
8  Individu1 = []
9  Individu2 = []
10 Pour t (0, m) Faire
11     individu1.ajouter (MAT[I][t])
12     individu2.ajouter (MAT[J][t])
13 FinPour
14 k = 5
15 Pour i (0, k) Faire
16     resul = individu1[i]
17     individu1[i] = individu2[i]
18     individu2[i] = resul
19 Fin Pour
20 MAT[I, :] = individu1
21 MAT[J, :] = individu2
22 retourner MAT
    
```

III.4.4.2 Procédure de Mutation

Pour réaliser cette tâche, on choisit aléatoirement deux points à partir de l'individu sélectionné et ensuite on fait une permutation à base de gène unique.

On applique la Mutation de manière aléatoire en utilisant « random » toujours sur la sortie de 'Croisement = (A)', résultat précédemment.

J=5
↓

1	1	0	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1
0	0	1	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1

J=la position dans l'individu aléatoirement

Par exemple, J=5 (résultat random)

J=5
↓

1	1	0	1	1	0	0	0	0
1	1	1	0	0	0	0	1	1
0	0	1	0	1	1	1	1	0
1	0	0	0	0	1	1	1	1

(B)-La population après les phases génétiques (croisement, mutation)

Après l'application de l'opérateur génétique, nous obtenons Q0 qui représente le résultat de la phase de mutation.

La procédure de mutation est spécifiée comme suit

Fonction mutation(X)

1. $w = \text{aléatoire}(0, 4)$
2. $n = \text{nombre des lignes}$
3. $m = \text{nombre des colonnes}$
4. $R = \text{aléatoire}(0, n)$
5. $S = \text{aléatoire}(0, n)$
6. **Tant Que** ($S == R$) **faire**
7. $S = \text{aléatoire}(0, n)$
8. **Fin Tant Que**
9. $\text{individu1} = []$
10. $\text{individu2} = []$
11. **Pour** $t(0, m)$ **Faire**
12. $\text{individu1.ajouter}(X[R][t])$
13. $\text{individu2.ajouter}(X[S][t])$
14. **Finpour**
15. **Si** $\text{individu1}[w] == 1$:
16. $\text{individu1}[w] = 0$
17. **Sinon**
18. $\text{individu1}[w] = 1$
19. **Finsi**
20. **Si** $\text{individu2}[w] == 1$:
21. $\text{individu2}[w] = 0$
22. **Sinon**
23. $\text{individu2}[w] = 1$
24. **Finsi**
25. $X[R, :] = \text{individu1}$
26. $X[S, :] = \text{individu2}$
27. Retourner X

III.5 Phase de concaténation

Dans cette phase on réalise la concaténation entre la population initiale et la population fils pour obtenir R0.

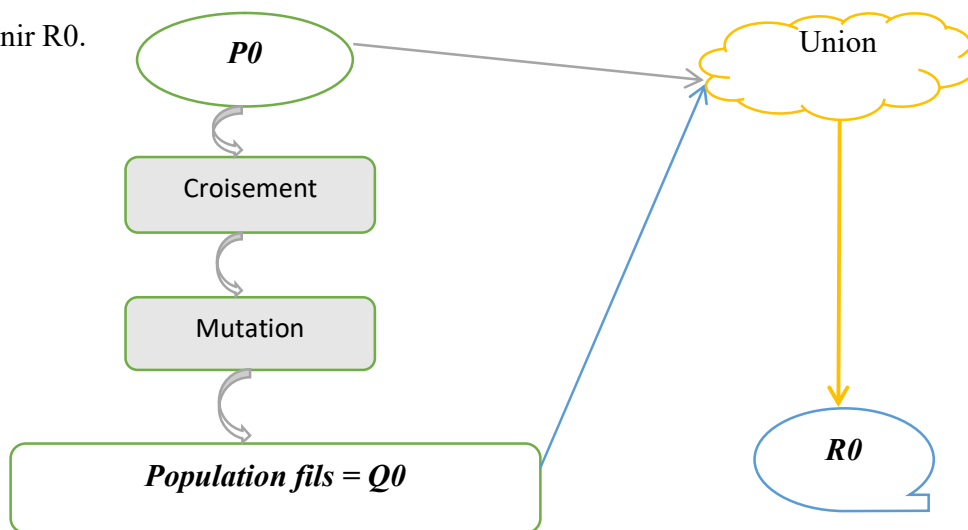


Figure III.4 Phase de concaténation

III.6 Phase de validation

Pour l'étape de validation, nous avons développés les algorithmes suivants :

1. Matrice sans duplication \leftarrow vide //R0 valider//
2. Matrice duplique \leftarrow vide
3. **Pour I de 0 Taille (R0) Faire**
 4. Si (R0 not in matrice-sans-duplication Alors
 5. Ajouter Matrice (matrice-sans-duplication, R0(i))
 6. Sinon
 7. Ajouter Matrice (matrice-duplication, R0(i))
 8. Finsi
9. **FinPour**
10. **Tant Que** (Taille matrice-sans-duplication < 200) **Faire**
11. **Croiser-dupliquer = croisement (matrice-dupliquer)**
12. **Muter-dupliquer = mutation (croiser-dupliquer)**
13. **Pour J = 0 à Taille (muter-dupliquer) Faire**
 14. Si (muter-duplique(j), matrice-sous-dupliquer (R0)) Alors
 15. Ajouter(Matrice-sans-dupliquer, muter)
 16. Finsi
17. **FinPour**
18. **Fin Tant Que**

III.7 Phase de calcul des fonctions d'objectives

Le principe de cette phase est décrit par les étapes suivantes :

- **Calcul des fonctions d'objectifs**

On calcule les fonctions d'objectifs (F_1, F_2) pour chaque individu, sachant que :

F_1 = minimiser le nombre des symptômes

F_2 = maximiser le poids de chaque symptôme

1 : est infecté par COVID-19

0 : Pas infecté par COVID-19

- **Extraire une sous matrice à partir R0 pour le calcul**

Colonne :

Test Covid-19

0	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1
1	1	0	0	1	0	1	1	0

Voici les règles de calcul utilisées dans la littérature :

$$F1(i) = \sum_{i=1}^n 1 / i$$

$$F2(i) = IV(i) \quad / i=1 \dots n$$

$$IV(i) = \sum (C\% - NC\%) * WOE(i)$$

$$WOE = \ln [(ci / C) / (nci / NC)]$$

/ C% : le pourcentage de COVID-19

/ NC% : le pourcentage de NonCOVID-19

C: le nombre total COVID-19.

NC : le nombre total de NON COVID-19.

ci : le nombre de COVID 19 avec symptôme i.

nci : le nombre de NON COVID 19 avec symptôme i.

Remarque : les valeurs de F2 Doit être dans l'intervalle [-1, +1]

III.8 Phase fast non dominated

Le processus de la phase de calcul appelée fast non dominated est décrit par l'Organigramme ci-dessous. Une ligne représente un individu et une colonne correspond à ses symptômes.

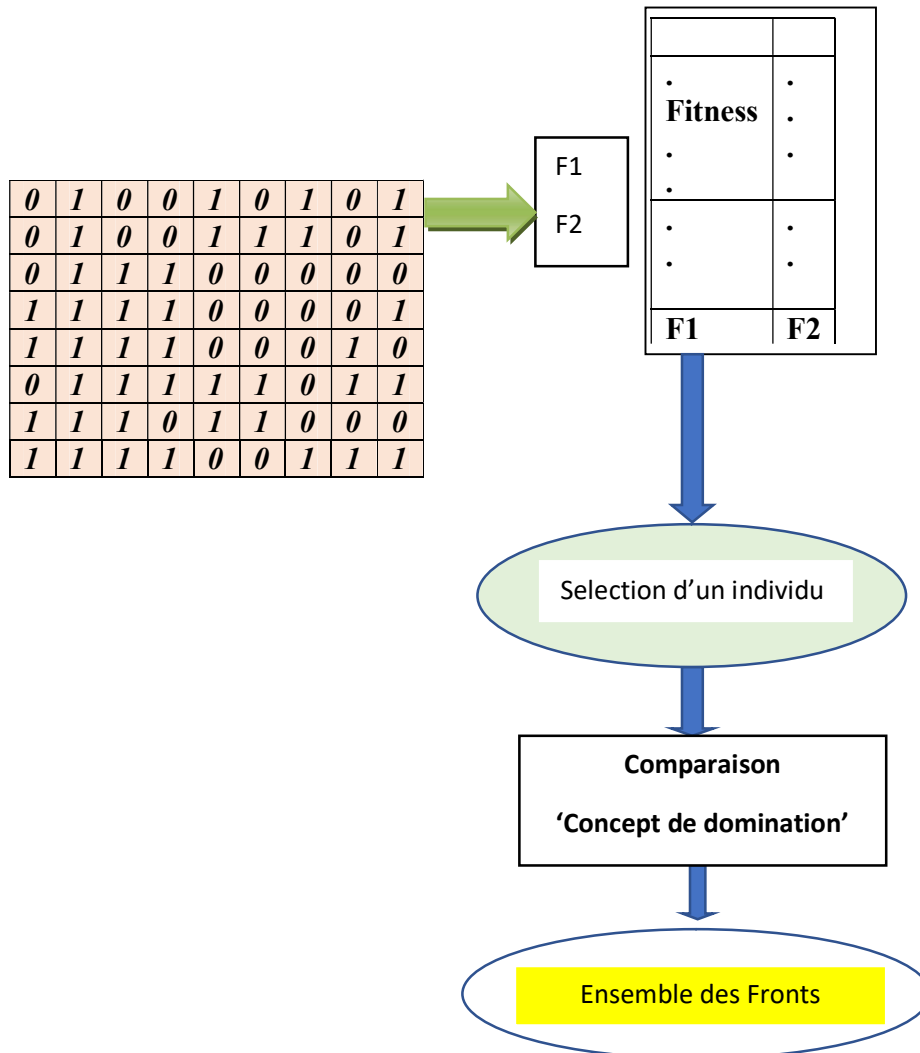


Figure III.5 Organigramme FAST Non Dominated

L'algorithme fast non dominated et la tracé des fronts est comme suit :

```

1. Fonction Fast_non_dominated(Mat):
2. groupe_lines_list=list()
3. fitness=Appel(fonction(fitness))
4. fitness = Trier (fitness) parpor le F1
5. excluded_points_indice_list=list()
6. Pour r de (0 a taille(fitness)) Faire
7.     Si (r not in excluded_points_indice_list) Alors
8.         point_name = Clé
9.         Ajouter(excluded_points_indice_list,r)
10.    Finsi
11.
12.    Pour j de (0 a taille(fitness)) Faire
13.    Si ((j # i) et (j not in excluded_points_indice_list)) Alors
14.    Si (fitness[i][0] <= fitness[j][0] and fitness[i][1] <= fitness[j][1]) Alors
15.    Si (i not in excluded_points_indice_list) Alors
16.        Ajouter(points_dictionnaire[point_name],fitness(j) )
17.        Ajouter(excluded_points_indice_list,(j))
18.    Finsi
19.    Finsi
20.    Finsi
21.    i = j,
22.    Si (i in excluded_points_indice_list) Alors
23.        Ajouter(points_dictionnaire[point_name],fitness(j) )
24.        Ajouter(excluded_points_indice_list,(j))
25.    Finsi
26.    i = j,
27.    i = 0,
28.    Pour key de 0 a taille(points_dictionnaire) Faire
29.        listx = list() ; listy =list()
30.        Pour i de taille(points_dictionnaire[key] ) Faire
31.            Ajouter(listx, i[0])
32.            Ajouter(listx, i[1])
33.            Ajouter(groupe_lines_list, ([listx, listy])
34.
35.        Finpour
36.    Finpour
37.    Pour elem de ((taille(groupe_lines_list )) Faire
38.        Dessiner(les fronts)
39.    Finpour
40.    retourner points_dictionnaire

```

III.9 Distance de crowding

Pour le traitement du processus du calcul de la distance de crowding, nous avons suivi les étapes suivantes :

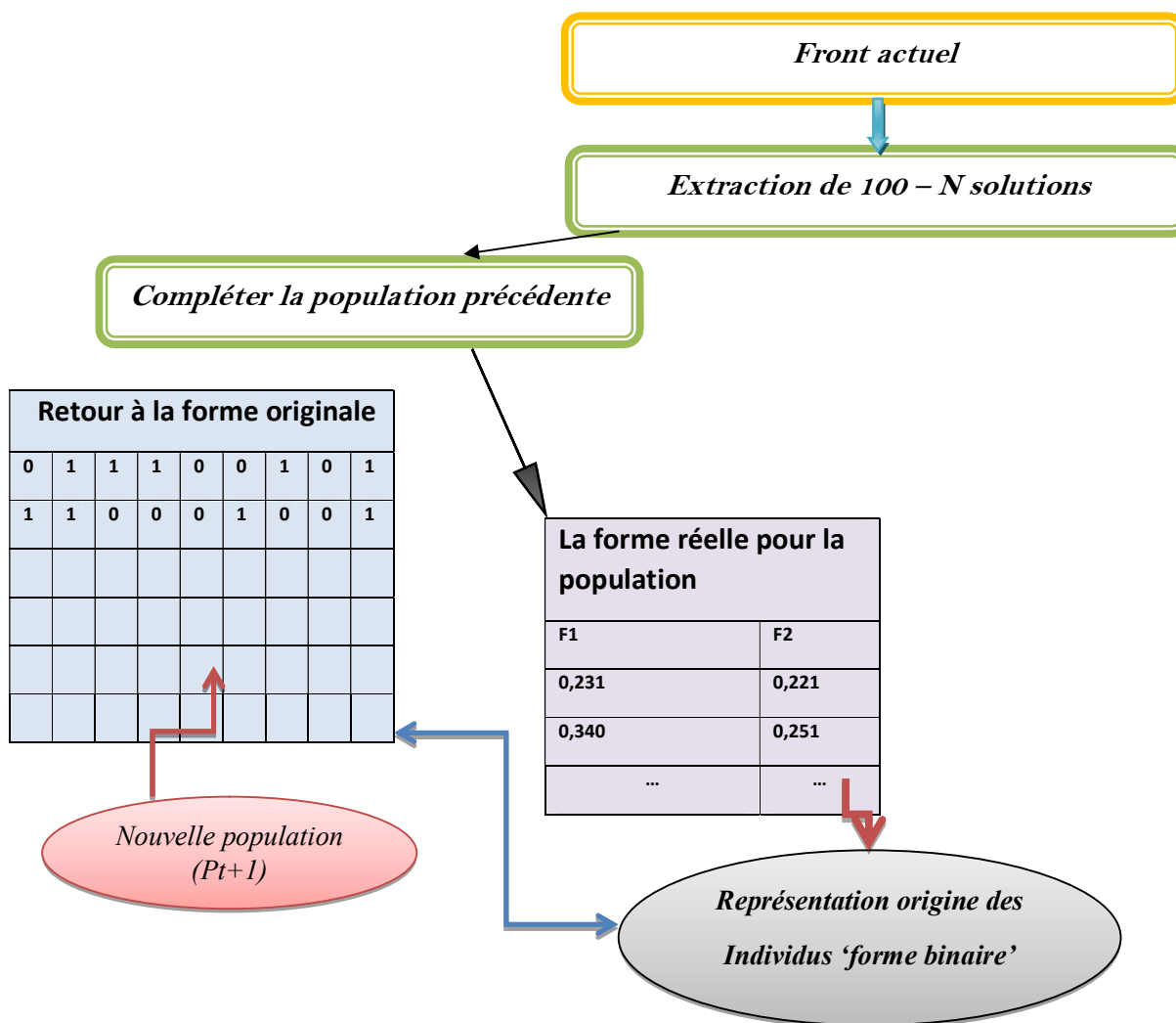


Figure III.6 Distance de crowding

Le principe de l'algorithme de calcul de la distance de crowding est une synthèse de questionnement ui sont comme suit :

- Le tri rapide non dominé nous aide à identifier les bonnes solutions, les solutions moyennes ou les mauvaises solutions, ces solutions sont triées sur les différents fronts.
- Donc, ce Tri Non Dominé va nous aider d'aller vers la Convergence.

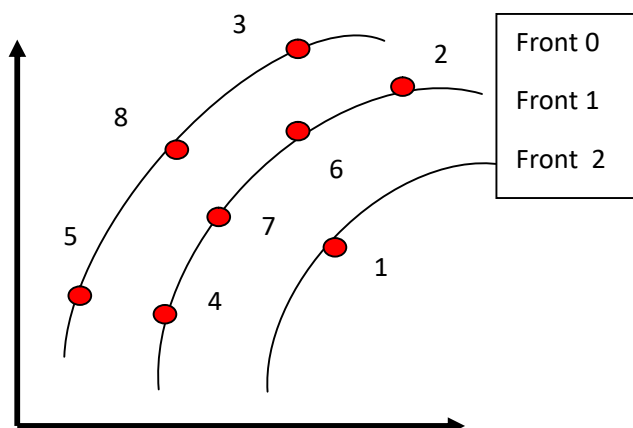
- Maintenant, la question à se poser est de montrer que nous pouvons préserver la diversité parmi les solutions? donc NSGAII apporte l'idée de distance d'encombrement par conséquent quelle est cette distance d'encombrement ?
- La distance d'encombrement mesure l'encombrement d'une solution par rapport à ses voisines situées sur le même front.

L'algorithme de distance de crowding est comme suit:

```

1.  $r = |F|$  //nombre de solution pour chaque front, appelé cardinalité du front//
2. Pour chaque front Faire
3.    $d_i = 0$ 
4.   Pour chaque m Faire // m fonction d'objectif//
5.      $F = \text{sort}(F, m)$  // trier chaque front par rapport à m//
6.      $d_1 = d_r = \infty$  // solutions extremes  $\rightarrow = \infty$ 
7.     Pour  $i = 1$  à  $r-1$  Faire
8.        $d_i = d_i + [(F_m(i+1) - F_m(i-1)) / (f_m^{\max} - f_m^{\min})]$ 
9.     FinPour
10.  FinPour
11. FinPour
    
```

Exemple d'appliquer la distance de crowding



Le front 0 possède les solutions : 5, 8, 3 c'est à dire que $r / F_0 = 3$, Initialement $d_5 = d_8 = d_3 = 0$

CHAPITRE III Modélisation d'un système d'aide à la décision pour la prédiction de Covid 19
basé sur NSGA II

Calculer distance de Crowding : « idée pour chaque front et ce dernier pour chaque solution » et chaque solution pour chaque fonction d'objet if (pour F_1, F_2) :

Pour chaque F_1				Pour chaque F_2			
- On tri les solutions par rapport F_1				- On tri les solutions par rapport F_2			
Avant	5	8	3	Avant	5	8	3
Avant	0.13	0.88	0.34	Avant	0.90	0.50	0.80
////	////	////	////	////	////	////	////
Après	5	3	8	Après	5	3	8
Après	0.13	0.34	0.88	Après	0.90	0.80	0.50
$i+1=8$ $i=3$ $i-1=5$							

$$df_1^i(3) = df_1 \text{ initiale} + [(f_1^{i+1}(8) - f_1^{i+1}(5)) / (f_1^{\max} - f_1^{\min})]$$

$$= 0 + (0.88 - 0.13) / (0.88 - 0.13) = 1$$

$$df_2^i(3) = df_1^i(3) + [(f_2(8) - f_2(5)) / (f_2^{\max} - f_2^{\min})]$$

$$= 1 + (0.5 - 0.9) / (0.9 - 0.5)$$

$$= 2$$

$$df_2(3) = DC(3) = 2$$

III.10. Phase de Classification

Pour réaliser la tâche de classification nous avons choisi AdaBoost car il génère deux classes pour chaque génération. L'algorithme AdaBoost est comme suit :

1. **Input:** A sequence of m instances
2. $S = \{(x_1, y_1) \dots (x_m, y_m)\}$ where $x_i \in X_k$ with
3. labels $y_i \in Y = \{0, 1\}$ from COVID-19 symptoms dataset
4. Weak learner: Decision Tree
5. T (number of iterations)
6. **Result:** The final classifier with hypothesis
7. $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$
8. **Initialization:** $D_1(i) = 1/m$
9. **For all** $i = 1, \dots, m$
10. **For** $t = 1$ to T
11. Call Weak learner using distribution D_t
12. Get Weak classifier and obtain hypothesis $h_t : X \rightarrow -1, +1$
13. Calculate the error rate $\epsilon_t = \sum_{i=1}^m D_t(i)[h_t(x_i) \neq y_i]$ of h_t
14. **If** $\epsilon_t > 0.5$ then $T = t - 1$ and abort loop
15. set $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
16. Update $D_{t+1}(i) = D_t(i) \exp(\alpha_t y_i h_t(i) / Z_t)$ (Z is a normalization factor)

III.11 Phase de test

Notre base générée est testée en utilisant les principaux symptômes qui influent sur COVID-19. Comme exemple de choix sur les symptômes de base utilisée à savoir : mal de gorge, essoufflement, mal de tête ainsi que l'indication-test.

toux	fièvre	mal de gorge	essoufflement	mal de tête	âge_60_et plus	sexe	Indication-test	corona résultat
0	0	1	1	1	0	0	1	1 :COVID :Vrai

Ci-dessous notre base générée organisée en correspondance avec nos choix prédéfini pour tester avec les symptômes de base si le patient possède COVID ou Non.

Procédure pour l'amélioration de la base initiale que nous a généré la base binaire ci-dessous, elle représente une exécution est comme suit :

```

1. Fonction test_covide_result(list_best_solution,list_candidat)
2. sambtomp_0_candidat=list_candidat[0]
3. sambtomp_1_candidat=list_candidat[1]
4. sambtomp_2_candidat=list_candidat[3]
5. sambtomp_3_candidat=list_candidat[5]
6. sambtomp_4_candidat=list_candidat[6]
7. sambtomp_5_candidat=list_candidat[7]
8. sambtomp_0_best_solution=list_best_solution[0]
9. sambtomp_1_best_solution=list_best_solution[1]
10. sambtomp_2_best_solution=list_best_solution[3]
11. sambtomp_3_best_solution=list_best_solution[6]
12. sambtomp_4_best_solution=list_best_solution[7]
13. sambtomp_5_best_solution=list_best_solution[8]
14. Iprimer(list_candidat,list_best_solution)
15. Si (sambtomp_0_candidat == sambtomp_0_best_solution ET sambtomp_1_candidat
16. == sambtomp_1_best_solution ET sambtomp_2_candidat ==
17. sambtomp_2_best_solution ET sambtomp_3_candidat == sambtomp_3_best_solution
18. ET sambtomp_4_candidat == sambtomp_4_best_solution ET sambtomp_5_candidat
19. == sambtomp_5_best_solution) Alors
20.     Iprimer("infecte par covid")
21.     retourner True
22. Sinon
23.     ecrire("n'infect pas par le covid")
24.     retourner False
25. Finsi
Fonction reseult_covide_test (list,list_best_solution):
26.     list_of_index=[]
27. Pour index in range(0,list.shape[0]) Faire
28.     list_of_index=list[index,:]
29.     result_verification=Appel fonction(test_covide_result)
30. Finpour
31. Si (result_verification == False) Alors
32.     Si (list[index][5] 1) Alors
33.         list[index][5] = 0
34.     Sinon
35.         Si (list[index][5] == 0) Alors
36.             list[index][5] =
37.         Finsi
38.     Finsi
39. Finsi
40.     retourner list

```

CHAPITRE III Modélisation d'un système d'aide a la décision pour la prédiction de Covid 19
basé sur NSGA II

<i>toux</i>	<i>fièvre</i>	<i>nal de gorge</i>	<i>essoufflement</i>	<i>mal de tête</i>	<i>âge_60_ et plus</i>	<i>sexe</i>	<i>Indication-test</i>	<i>corona résultat</i>
0	1	0	0	1	0	0	0	Faux
1	1	1	1	1	0	1	1	Vrai
0	1	0	1	0	1	0	1	faux
1	0	1	1	1	0	1	1	Vrai
1	1	0	1	1	0	0	1	Faux
1	0	1	0	0	1	1	1	Faux
1	0	1	1	0	1	0	1	Vrai
0	1	1	1	1	0	0	1	Vrai

A la fin cette base après l'étape de test de symptômes passe à la phase de classification en utilisant Adaboost.

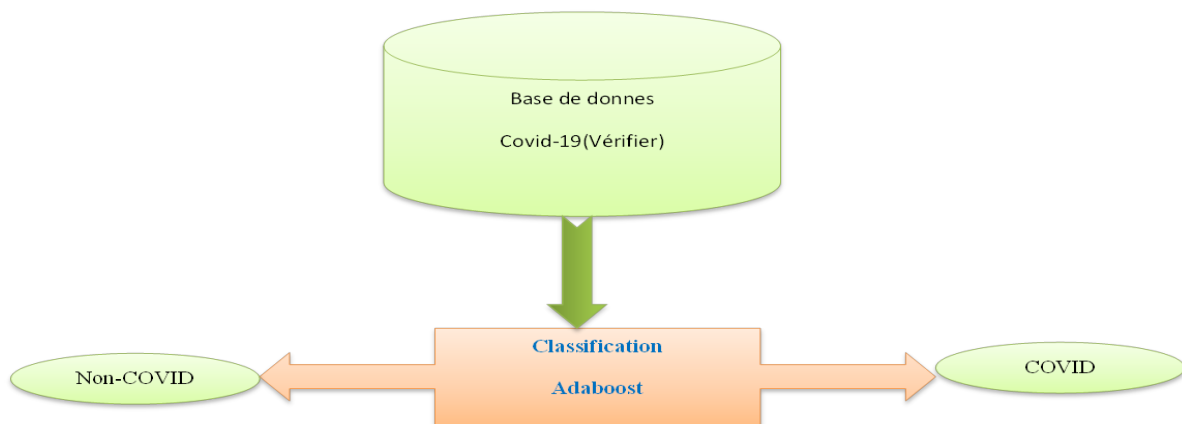


Figure III.7 Classification avec Adaboost

Le principe fonctionnel du système est synthétisé sur la figure suivante :

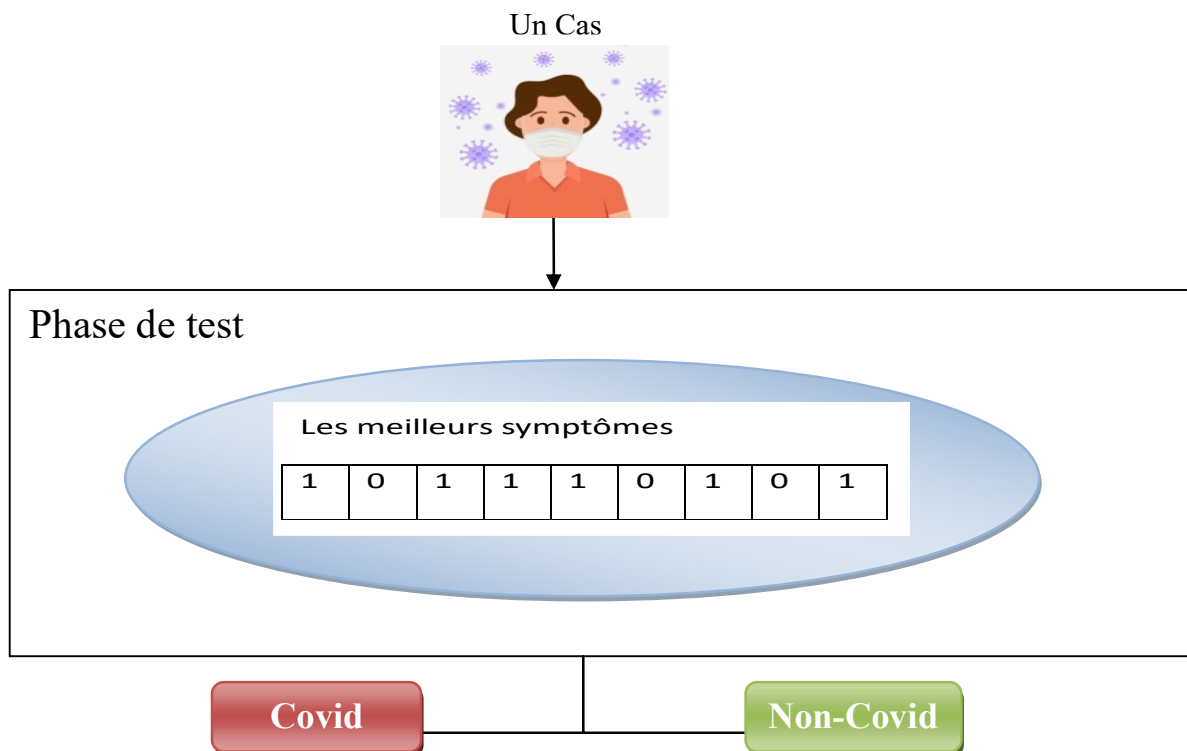


Figure III.8 principe fonctionnel du système

III.12 Conclusion

Dans ce chapitre nous avons présenté notre système pour l'aide à la décision dans le domaine médical appliqué au COVID-19. Le modèle décisionnel utilisé dans notre approche repose sur l'algorithme NSGA-II. Ce dernier est utilisé pour la détection des principales caractéristiques (symptômes) concernant COVID-19.

Dans le chapitre suivant, nous dresserons notre validation des propositions conceptuelles.

IV.1 Introduction

Toute proposition conceptuelle nécessite une validation expérimentale pour mettre en valeur les contributions théoriques et assurer des interprétations qualitatives et ouvrant des perspectives.

C'est ainsi que l'objectif de ce chapitre consiste en mettre en clair des résultats obtenus à travers des expériences. Ce chapitre contient en premier lieu la présentation des différents outils et environnements utilisés pour la réalisation des expérimentations, ensuite nous présentons quelques résultats significatifs.

IV.2 Présentation des outils et environnements

Dans cette section, nous présentons les principaux langages, environnement et outils utilisés dans notre partie expérimentale.

Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique.

Ses structures de données intégrées de haut niveau, combinées au typage dynamique et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de collage pour connecter des composants existants entre eux.

La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de maintenance du programme.

Python prend en charge les modules et les packages, ce qui encourage la modularité du programme et la réutilisation du code. L'interpréteur Python et la vaste bibliothèque standard sont disponibles gratuitement sous forme source ou binaire pour toutes les principales plateformes et peuvent être librement distribués.



IV.3 Implémentation

Nous décrivons dans cette section les aspects représentant les structures de données utilisées, les résultats pratiques obtenus à travers des expérimentations.



Figure IV.1 Interface d'accueil de l'application



Figure IV.2 Interface d'inscription dans l'application



Figure IV.3 Interface de saisie des symptômes

Résultat Covid-19

Test utilisé : COVID-19

Fiche de rendu de résultat

Nom du patient :Bouchami.....

Prénom du patient :Ramla.....

Numéro de téléphone: 0734879822

E-mail : boucharamla@gmail.com

' Bonjour, Vous avez effectué ce jour un Test Rapide de détection de Covid-19.'

Résultat est :...Négative.....

Figure IV.4 Interface Résultat sous forme PDF

IV.3.1 Base de données utilisées [

Ci-dessous la base de données initiale :

	A	B	C	D	E	F	G	H	I	J
1	test_date,cough,fever,sore_throat,shortness_of_breath,head_ache,corona_result,age_60_and_above,gender,test_indication									
2	2020-11-12,0,0,0,0,0,negative,No,male,Other									
3	2020-11-12,0,1,0,0,0,negative,No,male,Other									
4	2020-11-12,0,0,0,0,0,negative,Yes,female,Other									
5	2020-11-12,0,0,0,0,0,negative,No,male,Other									
6	2020-11-12,0,1,0,0,0,negative,No,male>Contact with confirmed									

Figure IV.5 La base de données initiale

La base de données normalisée ci-dessous est obtenue suite à notre choix qui repose sur la représentation binaire en donnant à un symptôme et les autres propriétés la valeur 1 s'il apparaît chez un malade et 0 sinon. Nous signalons que ce choix est adopté par les auteurs de l'article de base que nous avons exploité.

	A	B	C	D	E	F	G	H	I
1	cough,fever	sore_throat,shortness_of_breath,head_ache,corona_result,age_60_and_above,gender,test_indication							
2	0,0,0,0,0,0,0,1,0								
3	1,0,1,0,0,0,0,0,1,0								
4	2,0,0,0,0,0,0,1,0,0								
5	3,0,0,0,0,0,0,0,1,0								
6	4,0,1,0,0,0,0,0,1,1								

Figure IV.6 La base de données en format binaire Figure IV.5 La base de données initiale

IV.3.2 Génération de la population

La population de notre système ayant une forme matricielle est extraite de la base précédente sur laquelle nous avons détecté une anomalie qui est un déséquilibre dans tous les symptômes et surtout le nombre de porteur de covid et des non Covid. Nous montrons quelques images de la base de données qui était déséquilibrée et comment nous avons utilisé l'une des méthodes existantes pour équilibrer cette base de données, nous avons utilisé 'SMOTE' comme méthode d'équilibrage.

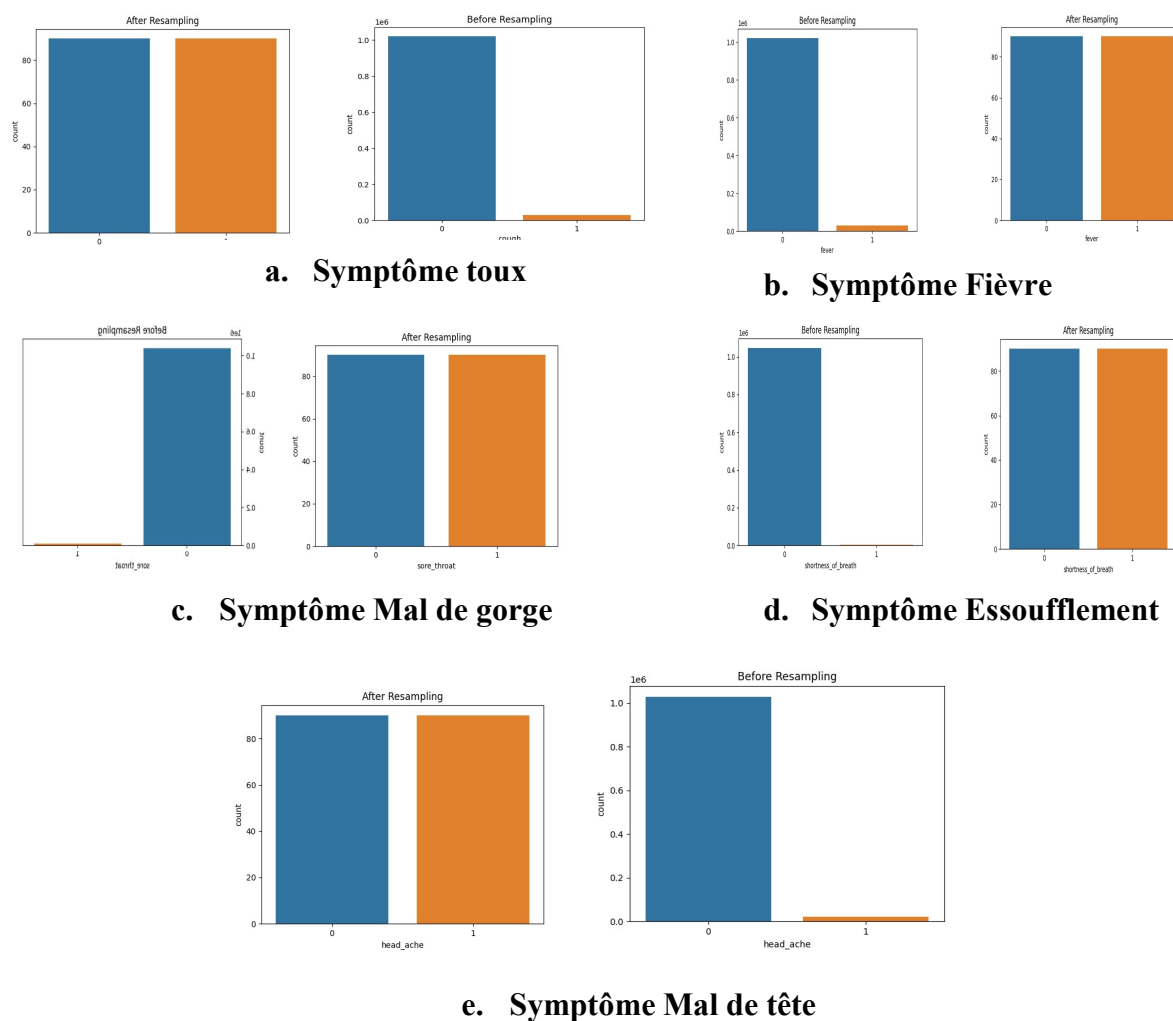


Figure IV.7 Représentation des pourcentages des symptômes (toux, Fièvre, Mal de gorge, Essoufflement, Mal de tête)

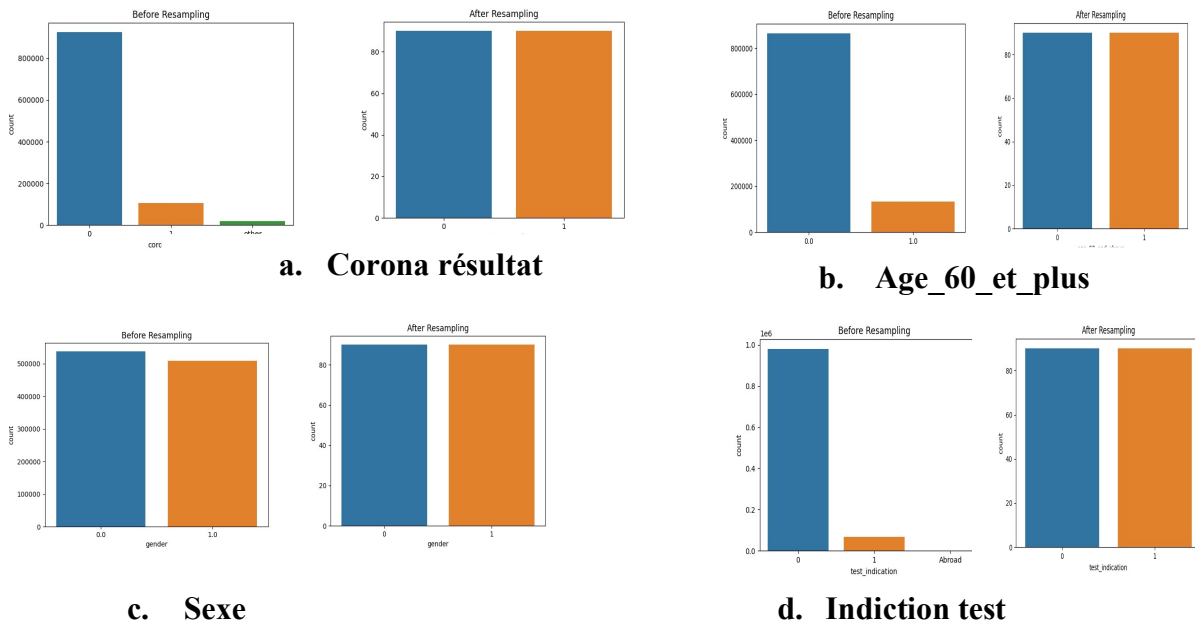


Figure IV.8 Représentation des pourcentages des symptômes (Corona résultat, Age_60_et_plus, Sexe , Indication test)

IV.3.3. Détection et résolution des problèmes

- **Problème : Déséquilibre de la base de données**

Nous constatons que SMOTE n'a pas équilibré les données selon la théorie et ce n'est pas toujours efficace. La technique SMOTE nous a montré que la base de données était déséquilibrée.

- **Solution : Équilibrage de la base de données**

Pour résoudre ce problème, nous avons pris un échantillon de la première base non pondérée et nous avons choisi un groupe qui comprend environ 2000 cas Covid et cas non Covid pour créer notre base de données.

Nous signalons que la solution très acceptable et parfois idéale serait de générer une nouvelle base de données. Ici le choix de l'ensemble de solutions est P0 : Population P0 de taille 100 individus, et la taille de l'individu est égale aux nombres de symptômes qui est 9 tel que la représentation de l'individu est une forme binaire. Ci-dessous un exemple de représentation.

```
P0 = np.array(
[[0, 0, 0, 0, 0, 0, 0, 1, 0], [0, 1, 0, 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 0, 1, 0, 0], [0, 1, 0, 0, 0, 0, 0, 0, 1, 1],
[0, 0, 0, 0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0, 1, 1, 0], [1, 0, 0, 0, 0, 0, 1, 0, 0], [0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 1, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0, 1, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 1, 1], [1, 0, 0, 0, 1, 0, 0, 1, 0], [0, 1, 1, 0, 1, 0, 0, 1, 0], [1, 1, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 1, 0, 1], [1, 0, 1, 0, 0, 0, 1, 0, 1], [1, 0, 0, 1, 0, 0, 0, 1, 0], [1, 1, 0, 0, 0, 0, 0, 1, 0],
[1, 1, 0, 1, 0, 0, 0, 1, 0], [1, 1, 0, 0, 0, 1, 0, 1, 0], [0, 1, 0, 1, 0, 0, 1, 0, 0], [0, 1, 1, 0, 0, 0, 0, 1, 1],
[0, 0, 1, 0, 0, 0, 1, 0, 1], [1, 0, 0, 0, 0, 1, 1, 1, 0], [1, 0, 0, 1, 0, 0, 1, 0, 0], [0, 0, 1, 1, 0, 0, 0, 0, 0],
[1, 0, 0, 1, 0, 0, 0, 1, 1], [1, 0, 1, 0, 1, 0, 0, 0, 0], [1, 0, 0, 1, 0, 1, 0, 0, 0], [0, 0, 1, 0, 0, 1, 0, 0, 1],
[1, 0, 0, 1, 0, 1, 0, 1, 1], [1, 1, 0, 0, 1, 1, 0, 0, 0], [0, 1, 0, 0, 1, 1, 0, 1, 1], [1, 1, 0, 0, 0, 1, 0, 1, 1],
[0, 0, 0, 0, 1, 0, 0, 1], [1, 0, 0, 0, 0, 1, 0, 0, 1], [0, 0, 0, 1, 0, 1, 0, 1, 0], [1, 0, 1, 0, 0, 1, 0, 0, 0],
[1, 0, 0, 1, 0, 1, 1, 0, 0], [1, 1, 0, 0, 0, 1, 0, 1, 1], [1, 0, 0, 0, 1, 1, 0, 0, 1], [1, 0, 0, 0, 0, 1, 0, 0, 0],
[1, 1, 0, 0, 1, 1, 1, 0, 0], [1, 0, 0, 1, 1, 1, 0, 0, 1], [0, 0, 1, 1, 1, 0, 1, 0, 0], [1, 0, 1, 0, 1, 0, 0, 0, 1],
[0, 1, 0, 1, 0, 0, 0, 1, 0], [1, 0, 0, 1, 1, 0, 0, 0, 1], [0, 0, 1, 0, 0, 0, 1, 1, 0], [0, 1, 1, 0, 1, 0, 0, 1, 0],
[0, 1, 0, 0, 1, 0, 0, 0, 0], [1, 1, 0, 0, 1, 0, 0, 1, 1], [1, 0, 1, 0, 0, 0, 1, 1, 1], [1, 0, 0, 0, 0, 1, 1, 1],
[0, 0, 1, 1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 1, 1, 0], [1, 1, 0, 1, 0, 0, 0, 1, 1], [1, 1, 0, 1, 0, 1, 0, 1, 0],
[1, 0, 1, 1, 0, 0, 1, 1, 0], [0, 0, 0, 0, 0, 0, 1, 1], [1, 0, 1, 0, 0, 0, 1, 1, 1], [0, 1, 0, 0, 0, 1, 1, 1, 1],
[0, 1, 0, 1, 0, 0, 1, 1, 0], [1, 0, 1, 1, 0, 0, 1, 0, 1, 1], [0, 0, 1, 0, 0, 1, 0, 1, 1], [1, 0, 0, 0, 0, 1, 0, 1, 1],
[1, 0, 0, 1, 0, 1, 1, 1, 1], [1, 0, 0, 0, 0, 1, 0, 1, 1], [1, 0, 0, 1, 0, 1, 1, 1, 0], [1, 0, 0, 0, 1, 1, 0, 1, 0],
[1, 0, 1, 0, 1, 1, 0, 0, 0], [1, 0, 0, 1, 1, 1, 1, 1, 0], [1, 0, 0, 0, 0, 1, 1, 0, 0], [0, 0, 1, 0, 1, 1, 0, 1, 0],
[1, 1, 0, 0, 0, 1, 0, 1, 1], [0, 1, 0, 0, 0, 1, 0, 1, 0], [1, 1, 0, 0, 1, 0, 1, 0, 1], [1, 1, 0, 1, 1, 1, 0, 1, 1]])
```

Figure IV.9 Population initiale

✓ Un exemple d'Individu

```
[0, 1, 0, 0, 0, 0, 0, 1, 0].
```

Figure IV.10 Représentation d'individu

L'individu (Chromosome) possède 9 gènes, tel que chaque gène représente un symptôme et ce dernier a deux valeurs {0,1} :

1 : Affecté par ce symptôme

0 : Non affecté

IV.3.4 Calcul des fonctions de fitness

Pour calculer les **fonctions de fitness** nous utilisons les équations suivantes :

$$F1(i) = \sum_{i=1}^{n} 1/i$$

Equation IV.1 Calcul de F1

$$F2(i) = IV(i)$$

$$IV(i) = \sum (C\% - NC\%) * WOE(i)$$

$$WOE = \ln [(ci / C) / (nci / NC)]$$

Equation IV.2 Calcul de F2

C% : le pourcentage de COVID-19

/ NC% : le pourcentage de NonCOVID-19

C:le nombre total COVID-19.

NC : le nombre total de NON COVID-19.

ci : le nombre de COVID 19 avec symptôme i.

nci : le nombre de NON COVID 19 avec symptôme i.

• Problème 1

- Lorsque nous utilisons ces lois, nous constatons que les deux fonctions prennent la mauvaise forme dans la formule.
- L'objectif est de créer un nuage.
- Mais les règles mathématiques précédentes ne peuvent pas atteindre cet objectif.

$$F1(i) = \sum 1 / i=1 \dots n$$

Equation IV.3 Cette formule de calcul de F1 ayant un problème

- ✓ Faisons d'abord connaissance de l'erreur dans la première fonction

La première fonction a pour objectif de **minimiser le nombre des symptômes**, et cela en comptant la somme des 1, c'est-à-dire en additionnant les symptômes qui ont affecté le patient. Cette idée ne peut pas avoir lieu car quelle que soit la valeur de F2, un nuage ne peut pas se former.

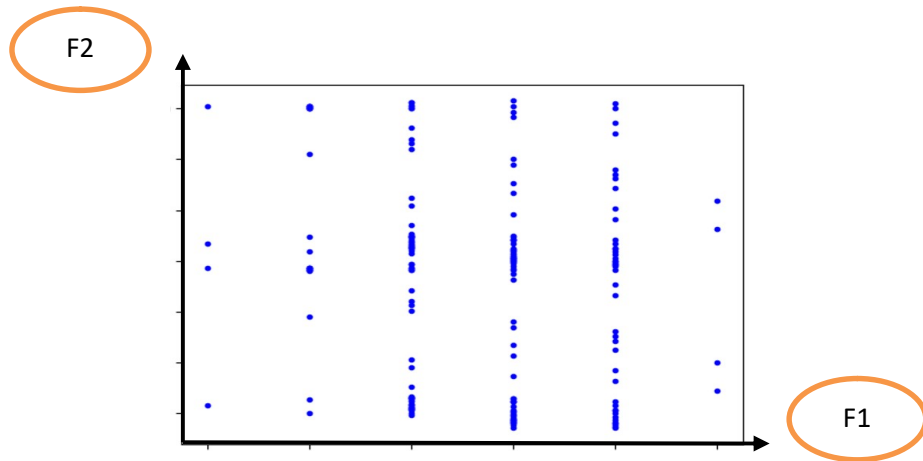


Figure IV.11 Représentation significative de l'erreur

- **Solution 1**

1. PROCEDURE CALCULERF1(R0) :

Si le patient est infecté par Covid-19 Alors

$$F1(i) = \sum (\text{pourcentage_symptôme_Covid}(i))$$

Sinon

$$F1(i) = \sum (\text{pourcentage_symptôme_NonCovid}(i))$$

Fin si

2. POURCENTAGE SYMPTOME COVID

Initialisation :

X_Covid = Nombre Totale avec symptôme(i) dans la population R0

Y_Covid = Nombre Totale avec symptôme(i) et infecté par le covid dans la population R0

//Fonction F1 :

Pourcentage_symptôme_Covid(i) = X/Y

Retourner Pourcentage_symptôme_Covid

3. POURCENTAGE SYMPTÔME NON COVID

Initialisation

$X_{\text{Non_Covid}} = \text{Nombre Totale avec symptôme}(i)$ dans la population
R0

$Y_{\text{Non_Covid}} = \text{Nombre Totale avec symptôme}(i)$ et n'est pas infecté par le covid dans la population R0

//Fonction F1 :

Pourcentage_symptôme_Non_Covid(i) = X/Y

Retourner Pourcentage_symptôme_Non_Covid

- **Problème 2**

Quant à l'erreur dans la deuxième fonction, elle est la suivante : l'objectif de F2 et de maximiser le poids pour chaque symptôme ne se calcule pas de cette manière et c'est ce que j'utilisais avant.

$$F2(i) = IV(i)$$

$$IV(i) = \sum (C\% - NC\%) * \mathbf{WOE(i)}$$

$$WOE = \ln [(c_i / C) / (nc_i / NC)]$$

Equation IV.4 Cette formule de calcul de F2 ayant un problème

Cette loi a été élaborée avec la première base de données, qui est déséquilibrée. Mais une telle loi ne s'adapte pas à ce type de base de données. La chose qui le prouve est l'observation que nous avons mentionnée plus tôt, à savoir que les valeurs de F2 ou {Poids: poids pour chaque} doivent être comprises dans la plage [-1, +1] et c'est une erreur mentionnée dans l'article.

- **Solution 2**

Afin d'obtenir cette information, la loi correcte est la suivante:

$$\mathbf{WOE} = [\mathbf{Log}(n_{\text{nonCovid}}/n_{\text{Covid}}) - \mathbf{Log}(N_{\text{nonCovid}}/N_{\text{Covid}})]$$

Pour une base de données déséquilibrée, nous devons diviser l'ensemble des solutions (Population) en sous-groupes.

n_nonCovid : Nombre de patients qui ne sont pas infecté par Covid et avec un symptôme (i) dans chaque sous-groupe

n_Covid: Nombre de patients qui sont infecté par Covid et avec un symptôme (i) dans chaque sous-groupe

N_nonCovid :le nombre totale qui ne sont pas infecte par covid et avec un symptôme (i) dans tout la population

N_Covid :le nombre totale qui sont infecte par covid et avec un symptôme (i) dans tout la population

La loi précédente utilisée pour calculer F2 satisfait les informations importantes sur le poids à savoir que les valeurs se situent dans une plage spécifique. Nous signalons que la loi mentionnée dans l'article ne fonctionne pas avec une base de données déséquilibrée. Nous avons également donné une loi alternative pour ce problème.

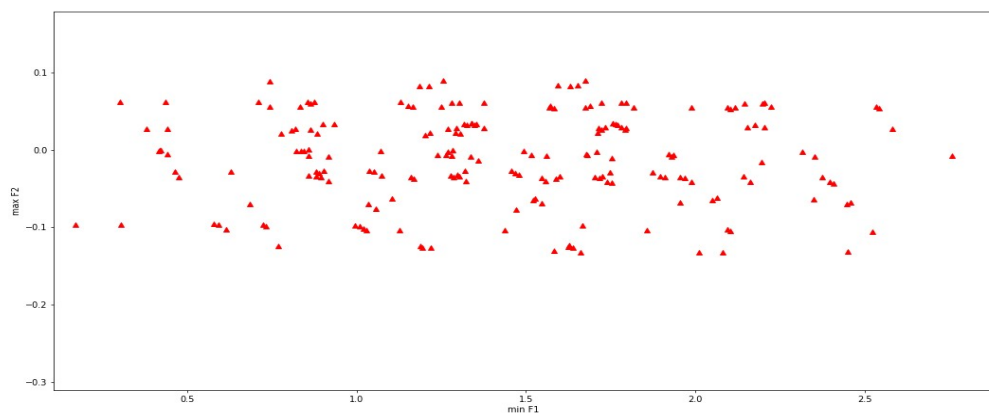
IV.3.5 Structure stockage

Dans cette section, nous allons décrire quelque structure utilisées pour le stockage de données.

IV.3.5.1 Stockage de la valeur fitness

Nous stockons les valeurs de notre fitness dans une matrice de 2 colonnes.

F1	F2
0,34	0,77
0,87	0,57
0,20	0,65
0,22	0,23



✓ **Figure IV.12 Nuage**

✓

IV.3.5.2 Stockage des fronts

Nous avons utilisé un dictionnaire pour stocker les fronts pour notre nuage.

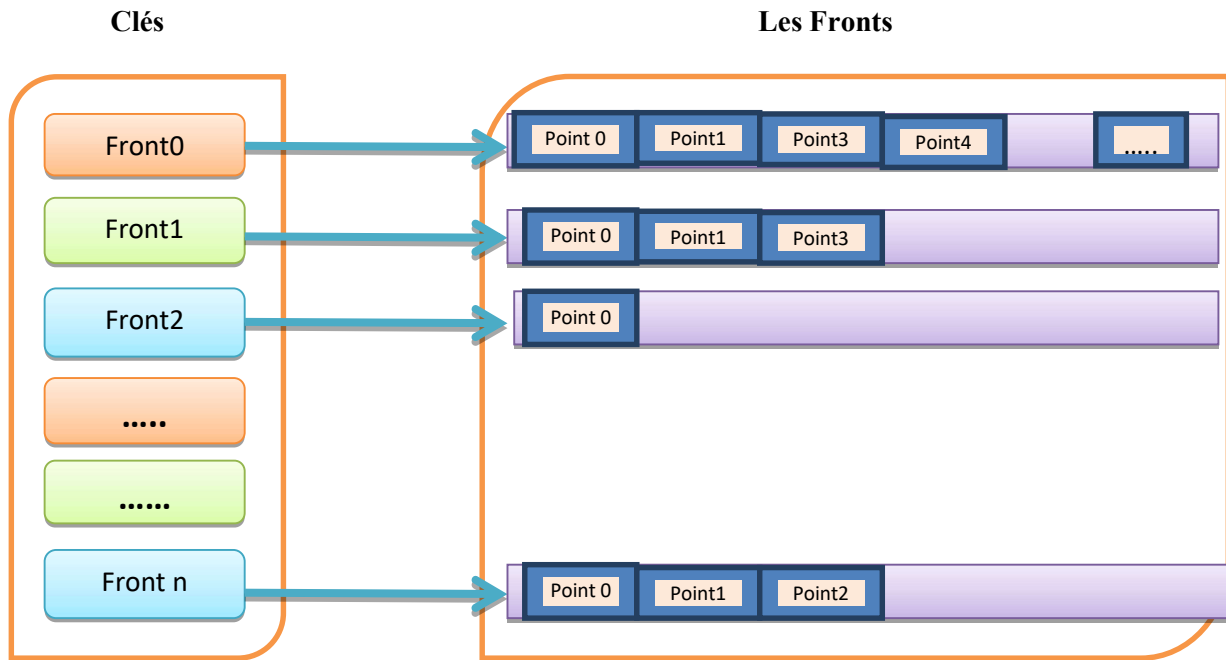


Figure IV.13 Structure du dictionnaire pour le stockage des fronts

✓ Les courbes a base de dominance

• Les figures suivantes sont les fronts dans la $i_{\text{ème}}$ exécution avec différentes itération tel que dans cette exécution :

* le rouge ■ (toujours le front0 -Front Pareto-)

* le vers ■ (toujours le front1)

* le jaune ■ (toujours le front2)

* le bleu ■ (toujours le front3)

-

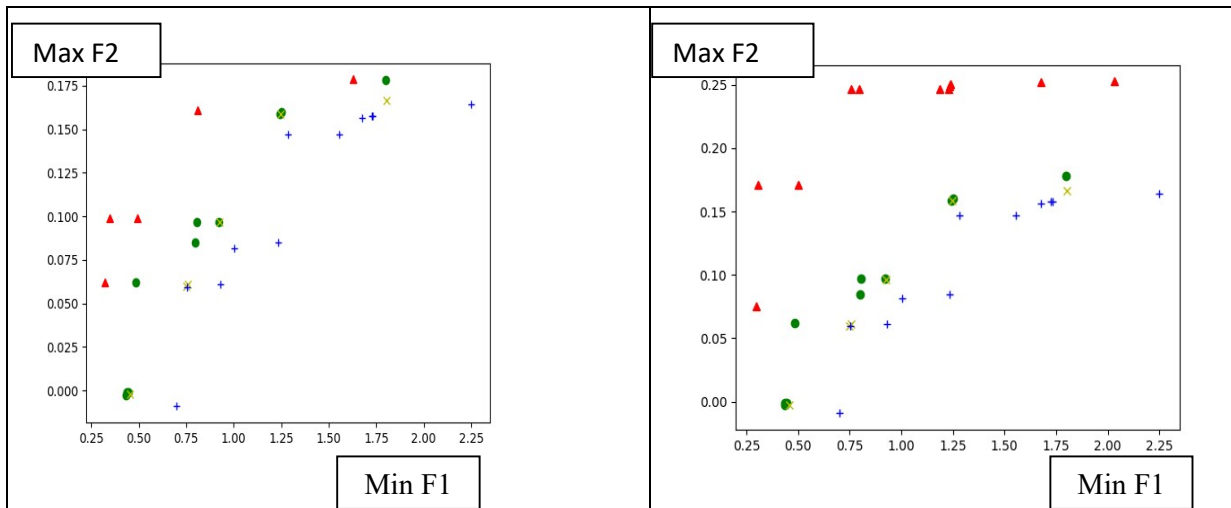


Figure IV.14 Affichage des fronts pour 4 exécutions (a)

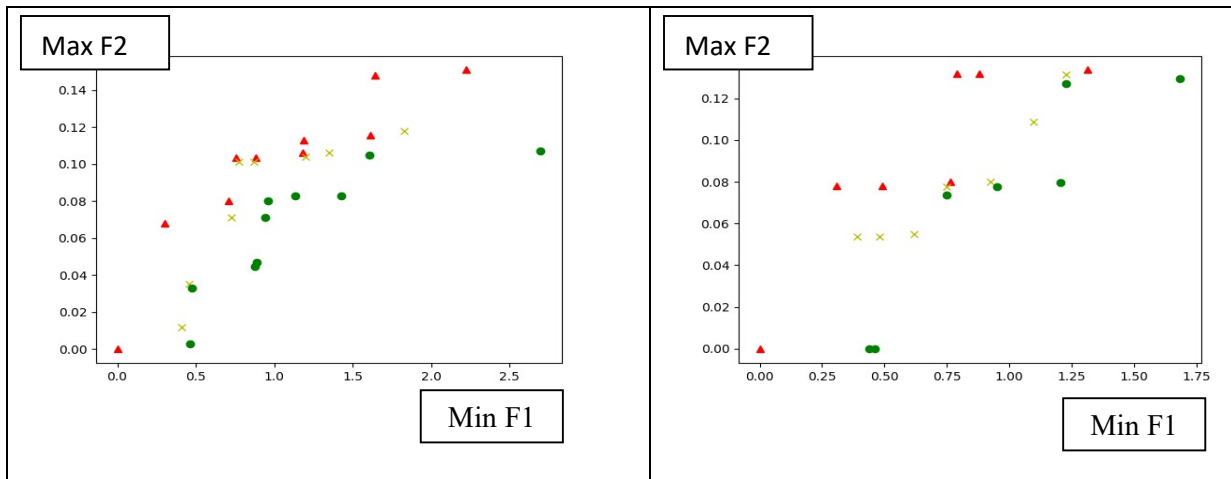


Figure IV.14 Affichage des fronts pour 4 exécutions (b)

IV.4 Conclusion

En conclusion de ce chapitre, nous signalons que les résultats obtenus sont satisfaisants et que les aspects conceptuels à base de NSGA II ont fait preuve de qualité dans le processus de classification de données. Le système de décision appliqué à COVID montre clairement que les idées inscrites dans le chapitre III possèdent des fondements théoriques satisfaisants.

CONCLUSION GENERALE

En conclusion de ce mémoire, nous synthétisons les principales activités et synthèses développées dans le cadre de notre projet. Ainsi, nous signalons que nous avons établi :

- Une synthèse bibliographique portant sur le domaine de l'optimisation multi-objective et les techniques d'optimisation
- Une seconde synthèse bibliographique concernant les maladies respiratoire, le COVID-19 suivi des travaux développés pour l'analyse de données des paramètres affectant le malade. Nous avons aussi développés une synthèse des travaux pour les applications d'aide à la décision prédictive appliqués à COVID-19.
- Modélisation d'un système d'aide a la décision pour le diagnostic de **Covid 19** basée sur NSGA II.
- Une étude expérimentale qui nous a permit de valider nos choix conceptuels et d'avoir une vision très claire et pratique du comportement de COVID-19. Nos résultats sont très réconfortants.

Nous signalons qu'à travers ce projet nous avons largement investi deux grand domaines de recherche à savoir l'optimisation multi-objective et les systèmes de classification et d'aide à la décision.

Comme perspective à notre projet, il serait très bénéfique de :

- intégrer ce travail dans le domaine du Bigdata pour le traitement d'un grand volume de données,
- combiner NSGA-II avec d'autres classifieurs comme le CNN,
- exploiter NSGA-II pour une approche de sécurité avec le blockchain.
- Combiner NSGA-II avec la logique floue
- Appliquer NSGA-III pour COVID-19.

Chapitre I

Etat de l'art sur COVID-19

CHAPITRE II

L'optimisation multi-objectif, l'algorithme NSGA-II

CHAPITRE III

Modélisation d'un système d'aide a la décision pour la prédiction de Covid 19 basé sur NSGA II

Chapitre IV

Résultats expérimentaux

INTRODUCTION GENERALE

Le COVID-19 est une pandémie qui a touché le monde au début du 3^{ième} millénaire dont les statistiques ont montré un taux de mortalité très grand suite à un modèle de propagation basé sur le contact phasique et l'inspiration de d'oxygène portant le germe de contamination.

A cet effet, un intérêt primordial et rapide a eu lieu de la part surtout des chercheurs premièrement pour comprendre et ensuite maîtrisé le propagation de la pandémie COVID-19.

La conséquence de ces recherches est orientée surtout vers la spécification des principaux symptômes du COVID-19 qui au départ n'étaient ni connu ni contrôlés.

Un autre problème concerne l'analyse comportementale et surtout le processus de contamination des personnes par le COVID-19.

Tous ces facteurs ou des verrous scientifiques nécessaires à la prise de décision qualitative restent des enjeux de grande importance dans le domaine de la recherche scientifique affectant directement le secteur socio-économique médical.

Les résultats obtenus par les chercheurs sont confortables et satisfaisants mais les exigences des experts médicaux dans le domaine des maladies contagieuses restent toujours à leur gourmandise en matière de grande précision dans la décision préventive.

A cet effet, dans notre projet nous essayons de développer une approche d'aide à la décision et d'analyse de données appliquée au COVID-19 en utilisant une méta-heuristique qui a fait preuve dans d'autre domaine en matière de classification de données pour laide à la décision, il s'agit de NSGA II.

Nous signalons que « Machine learning » dépend des algorithmes de classification et des algorithmes de sélection (sélectionner les meilleures des caractéristiques), NSGA II est l'un de ces algorithmes.

Le mémoire est organisé comme suit :

- Le premier chapitre est dédié un état de l'art portant sur un état de lart sur COVID-19.
- Le second chapitre contient une synthèse exhaustive sur l'optimisation combinatoire.
- La conception de notre approche à base de NSGA-II pour l'aide à la décision concernant COVID-19 est développée sur les deux volets à savoir la structure et le fonctionnement.

- Le chapitre quatre décrit les résultats expérimentaux obtenus.

Le mémoire est clôturé par une conclusion générale résumant l'essentiel de notre projet, elle est suivie par des perspectives ouvrant d'autres chantiers scientifiques comme continuité de ce travail.

REFERENCES BIBLIOGRAPHIQUE

- [1] <https://www.ameli.fr/assure/sante/themes/pneumonie/definition-facteurs-risque>
- [2] https://www.passeportsante.net/fr/Maux/Problemes/Fiche.aspx?doc=pneumonie_pm
- [3] <https://www.noovomoi.ca/vivre/sante/article.pneumonie.1.1028536.html>
- [4] <https://sante.journaldesfemmes.fr/fiches-maladies/2503576-pneumonie-virale-symptomes-contagieuse-fievre-traitement/>
- [5] <https://www.msmanuals.com/fr/professional/maladies-infectieuses/virus-respiratoires/coronavirus-et-syndrome-respiratoire-aigu-s%C3%A9v%C3%A8re-covid-19-mers-et-sras>
- [6] <https://urgences-serveur.fr/covid-19-mise-au-point.html>
- [7] <https://apps.who.int/iris/bitstream/handle/10665/339920/WHO-2019-nCoV-clinical-2021.1-fre.pdf>
- [8] <https://www.afro.who.int/sites/default/files/Covid19/Technical%20documents/Cas%20asymptomatiques%2C%20pr%C3%A9-symptomatiques%20et%20signes%20et%20sympt%C3%B4mes.pdf>
- [9] Liotta E.M., Batra A., Clark J.R. Frequent neurologic manifestations and encephalopathy-associated morbidity in Covid-19 patients. *Ann ClinTransl Neurol.* 2020;7(11):2221–2230
- [10] OMS . 2021. COVID-19 prise en charge clinique : orientations évolutives. [<https://apps.who.int/iris/bitstream/handle/10665/339920/WHO-2019-nCoV-clinical-2021.1-fre.pdf?sequence=1&isAllowed=y>]
- [11] Remuzzi A., Remuzzi G. COVID-19 and Italy: what next? *Lancet.* 2020;395:1225–1228.
- [12] Albadr MAA, Tiun S, Ayob M, AL-Dhief FT, Omar K, Hamzah FA (2020) Optimised genetic algorithm-extreme learning machine approach for automatic COVID-19 detection. *PLoS ONE* 15(12): e0242899. <https://doi.org/10.1371/journal>.
- [13] MakramSoui · NesrineMansouri, RaedAlhamad, MarouaneKessentini, KhaledGhedira « NSGA-II as feature selection technique and AdaBoost classifier for COVID-19 prediction using patient’s symptoms », *Nonlinear Dyn* (2021) 106:1453–1475 <https://doi.org/10.1007/s11071-021-06504-1>
- [14] Sambhe, V., Rajesh, S., Naredo, E., Dias, D., Kshirsagar, M. and Ryan, C., Multi-objective Classification and Feature Selection of Covid-19 Proteins Sequences using NSGA-II and MAP-Elites. DOI: 10.5220/0010388512411248, In Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2, pages 1241-1248, ISBN: 978-989-758-484-8,

- [15] Mohammad Saber Iraj, Mohammad-Reza Feizi-Derakhshi, JafarTanha, "COVID-19 Detection Using Deep Convolutional Neural Networks and Binary Differential Algorithm-Based Feature Selection from X-Ray Images", *Complexity*, vol. 2021, Article ID 9973277, 10 pages, 2021. <https://doi.org/10.1155/2021/9973277>
- [16] Almalki, Y.E.; Qayyum, A.; Irfan, M.; Haider, N.; Glowacz, A.; Alshehri, F.M.; Alduraibi, S.K.; Alshamrani, K.; AlkhalikBasha, M.A.; Alduraibi, A.; et al. A Novel Method for COVID-19 Diagnosis UsingArtificial Intelligence in Chest X-ray Images.*Healthcare* 2021, 9, 522. <https://doi.org/10.3390/healthcare9050522>
- [17] Jonathan Ozik, Justin M Wozniak, Nicholson Collier, Charles M Macal, MickaëlBinois, A population data-driven workflow for COVID-19 modeling and learning, Volume: 35 issue: 5, page(s): 483-499, September 10, 2021, <https://doi.org/10.1177/10943420211035164>
- [18] R. Karthik, R. Menaka, M. Hariharan, G.S. Kathiresan, AI for COVID-19 Detection from Radiographs: Incisive Analysis of State of the Art Techniques, Key Challenges and Future Directions, *IRBM*, 2021, ISSN 1959-0318, <https://doi.org/10.1016/j.irbm.2021.07.002>.
- [19] Khanday, A.M.U.D., Rabani, S.T., Khan, Q.R. et al. Machine learning based approaches for detecting COVID-19 using clinical text data. *Int. j. inf. tecnol.*12, 731–739 (2020). <https://doi.org/10.1007/s41870-020-00495-9>
- [20]https://elearningfacsci.univannaba.dz/pluginfile.php/9177/mod_resource/content/1/Chapitre%203%20-%20M%C3%A9thodes%20de%20r%C3%A9solution.pdf
- [21] Ehrgott, M., Gandibleux, X., Przybylski, A. (2016). Exact Methods for Multi-Objective Combinatorial Optimisation. In: Greco, S., Ehrgott, M., Figueira, J. (eds) Multiple Criteria Decision Analysis. International Series in Operations Research & Management Science, vol 233. Springer, New York, NY. https://doi.org/10.1007/978-1-4939-3094-4_19
- [22]<https://www.enseignement.polytechnique.fr/profs/informatique/Eric.Goubault/poly/cours009.html>
- [23] Bhargava, S. (2013). A Note on Evolutionary Algorithms and Its Applications. *Adults Learning Mathematics: An International Journal*, 8(1), 31-45
- [24] <https://www.geeksforgeeks.org/genetic-algorithms/>
- [25] http://igm.univ-mlv.fr/~dr/XPOSE2013/tleroux_genetic_algorithm/fonctionnement.html
- [26] Xin-She Yang, Chapter 6 - Genetic Algorithms, Editor(s): Xin-She Yang, *Nature-Inspired Optimization Algorithms (Second Edition)*, Academic Press, 2021, Pages 91-100, ISBN 9780128219867, <https://doi.org/10.1016/B978-0-12-821986-7.00013-5>

[27] M. Akbari, P. Asadi, M.K. BesharatiGivi, G. Khodabandehlouie,13 - Artificial neural network and optimization, Editor(s): Mohammad KazemBesharatiGivi, ParvizAsadi, In Woodhead Publishing Series in Welding and Other Joining Technologies, Advances in Friction-Stir Welding and Processing, Woodhead Publishing, 2014, Pages 543-599, ISBN 9780857094544, <https://doi.org/10.1533/9780857094551.543>.

[28] <http://eprints.univ-batna2.dz/1957/1/Thesis-Aouadj-Wafa-Optimisation-multi-objectif-etudes-de-cas.pdf>

[29] R. Marler et J. Arora, «Survey of multi-objective optimization methods for engineering,» Structural and multidisciplinary optimization, vol. 26, n°6, pp. 369-395, April 2004.

[30] I. P. Stanimirovic, M. L. Zlatanovic et M. D. Petkovic, «On the linear weighted sum method for multi-objective optimization,»FactaActaUniversitatis, vol. 26, n°4, pp. 49-63, 2011.

[31] Cours MAE, SihemSlatnia, universite de Biskra, 2022

[32] Clarisse Dhaenens. Optimisation Combinatoire Multi-Objectif : Apport des méthodes coopératives et contribution à l'extraction de connaissances. Modélisation et simulation. Université des Sciences et Technologie de Lille - Lille I, 2005. fftel-00178895

[33] G. Chiandussi, M. Codegone, S. Ferrero et F. E. Varesio, «Comparison of multi-objective optimization methodologies for engineering applications,» Computers& Mathematics with Applications, vol. 63, n°5, pp. 912-942, 2012.

[34]https://pepite-depot.univ-lille.fr/LIBRE/Th_Num/2004/50376-2004-209-210.pdf

[35]<https://asana.com/resources/pareto-principle-80-20-rule>

[36]Gardeux, V., Chelouah, R., Siarry, P. and Glover, F. (2011) EM323: A Line Search Algorithm for Solving High Dimensional Continuous Non-Linear Optimization Problems. Soft Computing, 15,2275-2285. <https://doi.org/10.1007/s00500-010-0651-6>

[37]E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," in *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, April 2003, doi: 10.1109/TEVC.2003.810758.

[38] Fonseca, C.M. and Fleming, P.J. (1993), 'Genetic algorithms for multi-objective optimization: formulation, discussion and generalization', Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco, CA, Morgan Kaufmann Publishers Inc.

[39] Ali Nikjoofar, Mahdi Zarghami, Water Distribution Networks Designing by the Multiobjective Genetic Algorithm and Game Theory, Editor(s): Xin-She Yang, Amir Hossein Gandomi, Siamak Talatahari, Amir Hossein Alavi, Metaheuristics in Water, Geotechnical and Transport Engineering, Elsevier, 2013, Pages 99-119, ISBN 9780123982964, <https://doi.org/10.1016/B978-0-12-398296-4.00005-2>.

[40] Zolpakar N.A., Lodhi S.S., Pathak S., Sharma M.A. (2020) Application of Multi-objective Genetic Algorithm (MOGA) Optimization in Machining Processes. In: Gupta K., Gupta M. (eds) Optimization of Manufacturing Processes. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-030-19638-7_8

[41] <http://bib.univ-oeb.dz:8080/jspui/bitstream/123456789/12798/1/these%20de%20doctorat%20en%20science%20de%20mahdi%202021.pdf>

[42] K. Tan, E. Khor, and T. Lee, Multiobjective Evolutionary Algorithms and Applications, ser. Advanced Information and Knowledge Processing. Berlin, Germany: Springer-Verlag, 2005

[43] Voloch, José Felipe. Chebyshev's method for number fields. Journal de Théorie des Nombres de Bordeaux, Tome 12 (2000) no. 1, pp. 81-85. http://www.numdam.org/item/JTNB_2000__12_1_81_0/

[44] IEEE transactions on evolutionary computation, vol. 11, no. 6, december 2007 MOEA/D: A Multiobjective Evolutionary Algorithm, Based on Decomposition Qingfu Zhang, Senior Member, IEEE, and Hui Li, pp 712-131

[45] Yufang Qin, Junzhong Ji, Chunnian Liu, "An Entropy-Based Multiobjective Evolutionary Algorithm with an Enhanced Elite Mechanism", Applied Computational Intelligence and Soft Computing, vol. 2012, Art, ID 682372, 11 pages, 2012. <https://doi.org/10.1155/2012/682372>

[46] Berro, Alain (2006) *Algorithmes évolutionnaires pour l'optimisation multi-objectif*. Technique et Science Informatiques, 25. pp. 991-1021.

[47] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," in *Proceedings of the International Conference on Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems*, pp. 95–100, 2002.

[48] <https://abaqus-docs.mit.edu/2017/English/IhrComponentMap/ihr-c-Reference-NSGAIL.html>

[49] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[50] https://www.etsmtl.ca/ETS/media/ImagesETS/Labo/LIVIA/memoires/memoire_pascal.pdf

[51] YuslizaYusoff, MohdSalihinNgadiman, AzlanMohdZain, Overview of NSGA-II for Optimizing Machining Process Parameters, *Procedia Engineering*, Volume 15, 2011, Pages 3978-3983, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2011.08.745>.

[52] Deb, K. ,& Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* , 18 , 577–601. doi:10.1109/TEVC.2013.2281535 [Crossref], [Web of Science ®], [Google Scholar]

[53] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," in *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, Aug. 2014, doi: 10.1109/TEVC.2013.2281535.

[54] Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization* 8(3), 631–657 (1998)

[55] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8129611/>

[56] <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>