

UNIVERSITE SAAD DAHLEB BLIDA 1
Institut d'Aéronautique et des Études Spatiales

THESE DE DOCTORAT

Spécialité : Aéronautique

**OPTIMISATION DES PROPRIETES NANOSTRUCTURES DES
ALLIAGES A BASE DE FER PAR LES ALGORITHMES
GENETIQUES ET RESEAUX DE NEURONES**

Par

Mamar BOUZIANE

Devant le jury composé de :

T. REZOUG	Professeur, U. Blida 1	Président
S. BERGHEUL	Professeur, U. Blida 1	Directeur de thèse
A. BENKHEDDA	Professeur, U. Blida 1	Examinatrice
F. SIAHMED	Maître de conférences A, USTHB	Examinatrice
A. YOUNES	Maitre de recherche A, CRTI, Chéraga	Examinateur
R. KERID	Maître de conférences A, ENSH	Examinateur
R. RENANE	Maître de conférences A, U. Blida 1	Invité

Blida, 2025

RESUME

Dans cette étude, un modèle basé sur la combinaison des algorithmes génétiques (GA) et des réseaux de neurones artificiels (ANN) est proposé comme outil d'analyse et d'optimisation des propriétés structurales et magnétiques des poudres de Fe-Ni nanostructurées préparées par broyage à billes à haute énergie. Initialement, le modèle ANN a été utilisé pour établir une corrélation entre les paramètres de broyage (temps de broyage, composition chimique et vitesse de rotation du disque et des jarres) et le produit final (taille des cristallites et coercivité). Différentes structures ANN ont été développées à partir des données expérimentales recueillies des travaux antérieurs. La meilleure structure, basée sur les critères de performance (le coefficient de corrélation, la racine carrée de l'erreur quadratique moyenne et l'erreur absolue moyenne) a ensuite été utilisée pour l'optimisation par GA. GA a été utilisé pour trouver les paramètres de broyage à billes optimaux. Des optimisations mono-objectif et multi-objectifs ont été appliquées dans le but de réduire le temps de broyage, la taille des cristallites et la coercivité. L'ensemble optimal de paramètres de broyage dans une optimisation multi-objectifs a été déterminé à partir d'une analyse du front de Pareto. Ensuite, le modèle programmation génétique multi-gène a été utilisé pour trouver une relation mathématique précise entre les données d'entrée et de sortie. Enfin, une méthode d'analyse de sensibilité a été appliquée pour déterminer les paramètres de broyage ayant le plus d'influence sur les paramètres de sortie.

Mots-clés : Alliages nanostructurés, mécanosynthèse, algorithmes génétiques, réseaux de neurones.

ABSTRACT

In this study a model based on the combined genetic algorithm (GA) and artificial neural network (ANN) method is proposed as a tool for the analysis and optimization of structural and magnetic properties of nanostructured Fe–Ni powders prepared by high energy ball milling. Initially, The ANN model was implemented to correlate the milling parameters (milling time, chemical composition and milling rotation speed of disc and vial) to end product (crystallite size and coercivity). Different ANN structures were developed using collected experimental data from previous work. The best structure based on the performance factors (the correlation coefficient, the root mean square error and the mean absolute error) was used for optimization by GA. Then, a GA was implemented to find the optimal ball milling parameters. A single- and multi-objective optimizations are applied to minimize milling time, crystallite size and coercivity. The optimal set of ball milling parameters in a multi-objective optimization was found based on a Pareto front analysis. After that, the multi-gene genetic programming model is used to find an accurate mathematical relationship between input and output data. Finally a sensitivity analysis method was applied to determine the most influential milling parameters on the outputs parameters.

Keywords: Materials nanostructured, mechanical alloying, genetic algorithms, neural networks.

ملخص

في هذه الدراسة، تم اقتراح نموذج يعتمد على دمج الخوارزمية الجينية (GA) بالشبكة العصبونية الاصطناعية (ANN) كأداة لتحليل وتحسين الخصائص الهيكلية والمغناطيسية في مساحيق الحديد والنيكل ذات البنية النانومترية المحضرة عن طريق السحق الميكانيكي عالي الطاقة. في البداية، تم تطبيق نموذج الشبكة العصبونية الاصطناعية لربط معلمات الطحن (وقت الطحن والتركيب الكيميائي وسرعة دوران الطحن للقرص والقارورة) بالمنتج النهائي (حجم البلورات والإكراه). تم تطوير هياكل مختلفة لشبكة ANN باستخدام المعطيات التجريبية المجمعة من الأعمال السابقة. تم استخدام أفضل بنية استنادًا إلى عوامل الأداء (معامل الارتباط، والجزر التربيعي لمتوسط الخطأ ومتوسط الخطأ المطلق) لتحقيق التحسين بواسطة GA. بعد ذلك، تم تنفيذ GA للعثور على معلمات الطحن المثلى. تم تطبيق تحسينات أحادية ومتعددة الأهداف لتقليل وقت الطحن وحجم البلورة والإكراه. تم العثور على المجموعة المثلى من معلمات الطحن بالكرات المعدنية في التحسين متعدد الأهداف بناءً على تحليل جبهة باريتو. بعد ذلك، تم استخدام نموذج البرمجة الجينية متعددة الجينات لإيجاد علاقة رياضية دقيقة بين بيانات المدخلات والمخرجات. وأخيرًا تم تطبيق طريقة تحليل الحساسية لتحديد معلمات الطحن الأكثر تأثيرًا على معلمات المخرجات.

الكلمات المفتاحية: المواد ذات البنية النانومترية، السحق الميكانيكي، الخوارزميات الجينية، الشبكات العصبونية.

REMERCIEMENTS

En premier lieu, j'adresse mes remerciements à DIEU tout puissant pour la santé, la volonté le courage et surtout la patience qui m'a donné pour mener ce travail à terme.

Je voudrais exprimer ma gratitude à monsieur Said BERGHEUL, Professeur à l'Université Saad Dahlab de Blida1 pour avoir suivi mon travail avec une attention soutenue, une grande disponibilité et beaucoup d'encouragements, et de m'avoir guidé par ses conseils pertinents et judicieux pendant toute la période de la thèse.

Je remercie très sincèrement, Monsieur Tahar REZOUG, Professeur à l'Université Saad Dahlab de Blida1 et Directeur de laboratoire des sciences aéronautiques, de m'avoir fait l'honneur d'accepter de présider le jury de cette thèse.

J'exprime ma profonde reconnaissance à Madame Amina Benkhedda, Professeur à l'Université Saad Dahlab de Blida1; à Madame Fatiha. SIAHMED, Maître de Conférences A à l'USTHB; à Monsieur Abderahmane YOUNES, Maître de Recherche A à CRTI de Chéraga, alger et à Monsieur Rachid KERID, Maître de Conférences A à l'ENSH, qui ont aimablement accepté d'être examinateurs de cette thèse.

Je tiens à remercier Monsieur Rachid RENANE Maître de Conférences A à l'Université Saad Dahlab de Blida1 pour avoir accepté d'être l'invité d'honneur de mon jury de thèse, qu'il trouve ici ma profonde reconnaissance et mes profonds respects.

Je tiens à remercier Monsieur Abderrahim ABADA Maître de Conférences A à l'Université Saad Dahlab de Blida1 pour leur aide indispensable.

Je ne peux pas oublier mes chers parents qui m'ont toujours soutenu et ont prié pour moi. Que dieu les protège.

Pour finir je voudrai remercier, ma femme et mes enfants ainsi que mon beau-père qui m'ont toujours soutenu et encouragé.

TABLE DES MATIERES

RESUME	1
REMERCIEMENTS	4
TABLE DES MATIERES	5
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	9
INTRODUCTION	13
1. ETUDE BIBLIOGRAPHIQUE	
1.1 Généralités sur les nanomatériaux	16
1.1.1 Les nanomatériaux	16
1.1.2 Classification des alliages nanocristallins	17
1.1.3 Méthodes d'élaboration des nanomatériaux	18
1.1.4 Propriétés des nanomatériaux	19
1.1.5 Domaines d'application des nanomatériaux	21
1.2 Elaboration des matériaux nanostructurés par broyage mécanique	22
1.3. Généralités sur les matériaux magnétiques	27
1.3.1 Matériaux magnétiques	27
1.3.2 Influence d'un champ magnétique extérieur	28
1.3.3 Mécanisme d'aimantation et cycle d'hystérésis	29
1.3.4 Les différentes formes du cycle d'hystérésis	30
1.4 Travaux antérieurs sur les propriétés structurales et magnétiques des alliages Fe-Ni nanocristallins obtenus par mécanosynthèse	32
1.4.1 Propriétés structurales	32
1.4.1.1 Influence de la durée de broyage	32

1.4.2 Influence de la vitesse de broyage	34
1.4.2 Propriétés magnétiques	35
1.4.2.1 Influence de la durée de broyage	35
1.4.2.2 Influence de la composition	37
1.5 Application des AG-ANN dans la mécanosynthèse	38
2. OPTIMISATION PAR ALGORITHMES GENETIQUES	
2.1 Introduction	41
2.2 Problème d'optimisation	42
2.3 Algorithmes génétiques mono-objectif	43
2.3.1 Codage et population initiale	44
2.3.1.1 Codage binaire	44
2.3.1.2 Codage réel	45
2.3.1.3 Codage en base n	45
2.3.2 Opérateur de croisement	46
2.3.3 Opérateur de mutation	47
2.3.4 Fonction d'évaluation	48
2.3.5 Opérateur de sélection	48
2.3.5.1 Sélection par tournoi	49
2.3.5.2 Sharing	50
2.3.5.3 Elitisme	51
2.4 Programmation Génétique Multi-Gène	51
2.5 Optimisation multi-objectifs	54
2.5.1 Méthodes agrégées	55
2.5.2 Méthodes basées sur l'équilibre de Pareto	56
2.5.3 Méthodes non agrégées et non Pareto	57
2.5.4 Etude de l'algorithme génétique multi-objectifs NSGA-II	57
2.6 Conclusion	62

3. RESEAUX DE NEURONES

3.1. Introduction	64
3.2. Notions sur les réseaux de neurones artificiels	65
3.2.1. Neurone biologique et neurone artificiel	65
3.2.2. Concept du réseau de neurones de type feedforward	67
3.2.3. Fonctions d'activation	69
3.2.4. Architecture des réseaux de neurones artificiels	70
3.2.5. Sélection d'un réseau de neurones artificiel	71
3.2.5.1. Sélection du type de réseau	71
3.2.5.2. Sélection de la structure du réseau	71
3.2.5.3. Sélection de la fonction d'activation	72
3.2.5.4. Sélection de type d'apprentissage : Apprentissage supervise	72
3.2.5.5. L'apprentissage	73
3.3. Conclusion	78

4. RESULTATS DE LA MODELISATION

4.1 Application du réseau de neurones	79
4.1.1 Introduction	79
4.1.2. Conception de la base de données	79
4.1.2.1. Collections de données	79
4.1.1.2. Normalisation des données	80
4.1.2. Développement du modèle ANN	81
4.1.2.1. Architecture de réseau de neurones	81
4.1.2.2. Algorithme d'apprentissage	82
4.1.2.3 Découpage de la base de données	83
4.1.2.4 Critère d'évaluation des performances du modèle	83
4.1.3 Détermination de l'architecture optimale de réseau	84
4.1.4 Résultats	84

4.1.4.1 Estimation de la taille des cristallites	84
4.1.4.2 Estimation de la coercivité	89
4.2 Optimisation par les Algorithmes génétiques	93
4.2.1 Introduction	93
4.2.2 Optimisation mono-objectif	93
4.2.2.1 Intégration d'un GA avec ANN	93
4.2.2.2 Paramétrage du GA	95
4.2.2.3 Résultats	96
4.2.3 Optimisation multiobjectif	99
4.2.3.1 Intégration de NSGA-II avec ANN	99
4.2.3.2 Paramétrage du NSGA-II	100
4.2.3.3 Résultats	100
4.3. Modélisation par Programmation génétique	107
4.3.1 Introduction	107
4.3.2 Implémentation de la méthode MGGP	107
4.3.2.1 Base de données	107
4.3.2.2 Paramétrage du MGGP	107
4.3.3 Résultats	108
4.4 Conclusion	111
CONCLUSION	113
APPENDICES	116
REFERENCES	121

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Représentation schématique des grains et joints de grains d'un nanomatériau	16
Figure 1.2	Evolution du pourcentage d'atomes situés au joint de grains d'un solide nanocristallin en fonction de la taille des grains	17
Figure 1.3	Classification des matériaux nanostructurés	18
Figure 1.4	Broyeurs attriteur et vibrant	24
Figure 1.5	Schématisation d'un broyeur à mouvement planétaire et de l'action des billes pendant le processus	24
Figure 1.6	a) Domaines de Weiss; b) Rotation des moments magnétiques dans une paroi de Bloch à 180°	28
Figure 1.7	Evolution des domaines de Weiss sous l'action d'un champ magnétique extérieur croissant dans un matériau ferromagnétique	28
Figure 1.8	Courbe de première aimantation (OA), et cycle d'hystérésis (ABCDEF) d'un matériau ferromagnétique	29
Figure 1.9	Cycle d'hystérésis pour un matériau magnétique dur (a) et un matériau magnétique doux (b)	31
Figure 1.10	Variation de D et ε de Fe-50 % Ni pour différents temps de broyage.	33
Figure 1.11	Evolution du paramètre de maille en fonction de la durée de broyage des alliages Fe-10%Ni et Fe-20%Ni.	33
Figure 1.12	Taille des particules (μm) et taille des cristallites (nm) de Fe-45%Ni en fonction de la vitesse de rotation des jarres	34
Figure 1.13	Alimentation à saturation pour les alliages 20%Ni, 35%Ni et 50%Ni en fonction du temps de broyage	35

Figure 1.14	Variation du champ coercitif en fonction de la durée de broyage pour les deux alliages $Fe_{50}Ni_{50}$ et $Fe_{80}Ni_{20}$	36
Figure 1.15	Evolution de l'aimantation à saturation en fonction de la teneur en nickel pour différentes intensités de broyage	37
Figure 1.16	Evolution du champ coercitif en fonction de la teneur en nickel pour différentes intensités de broyage	38
Figure 2.1	Croisement en un seul point	46
Figure 2.2	Principe de l'opérateur de mutation	47
Figure 2.3	Effet de sharing sur la répartition des solutions	50
Figure 2.4	Exemple typique d'un modèle MGGP représentée sous forme d'arbre.	52
Figure 2.5	Opération de croisement en MGGP	53
Figure 2.6	Opération de mutation en MGGP	53
Figure 2.7	Front et rangs de Pareto pour un problème de minimisation de deux objectifs.	56
Figure 2.8	Classification des individus selon le rang de Pareto.	58
Figure 2.9	Distance de crowding- les cercles noirs représentent les solutions appartenant au front optimal	59
Figure 2.10	Calcul de la distance de crowding pour chaque solution d'un front.	60
Figure 2.11	Principe de l'algorithme NSGA-II	61
Figure 3.1	Neurone biologique	65
Figure 3.2	Neurone artificiel	66
Figure 3.3	Réseau de neurones artificiels de type feedforward	68
Figure 3.4	Illustration d'apprentissage supervisé	73
Figure 3.5	Evolution schématique de l'erreur de l'erreur sur les sous-ensembles d'apprentissage et de validation en fonction du nombre d'itérations.	75

Figure 3.6	Comparaison entre l'apprentissage avec l'algorithme de LM seul et l'algorithme LM + regularisation bayesienne	76
Figure 4.1	L'architecture du réseau de neurones proposé	81
Figure 4.2	Organigramme ANN utilisé	82
Figure 4.3	Apprentissage d'un ANN avec 10 neurones dans une couche cachée.	86
Figure 4.4	Valeurs estimées de la taille des cristallites	87
Figure 4.5	Valeurs prédites du D en fonction des données expérimentales.	87
Figure 4.6	Apprentissage de l'ANN avec 10 neurones dans une couche cachée.	90
Figure 4.7	Valeurs estimées de la coercivité	91
Figure 4.8	Valeurs prédites du H_c en fonction des données expérimentales.	91
Figure 4.9	Chromosome typique	94
Figure 4.10	Organigramme de GA-ANN	95
Figure 4.11	Evolution de la fonction fitness (D) au cours des générations	97
Figure 4.12	Evolution de la fonction fitness (H_c) au cours des générations	98
Figure 4.13	Organigramme de NSGA-II	99
Figure 4.14	Front de Pareto multiojectif pour la taille des cristallites et le temps de broyage	101
Figure 4.15	Front de Pareto multiojectif de la coercivité et le temps de broyage	102
Figure 4.16	Front de Pareto multiojectif de la taille des cristallites et la coercivité	104
Figure 4.17	Comparaison de la taille des cristallites	105
Figure 4.18	Comparaison de la coercivité	106

Figure 4.19	Valeurs prédites de D, a) données d'entraînement; b) données de test	109
Figure 4.20	Valeurs prédites de Hc, a) données d'entraînement ; b) données de test	110
Figure 4.21	Pourcentage de contribution de chaque variable d'entrée.	111
Tableau 1.1	Domaines d'applications des nanomatériaux.	21
Tableau 3.1	Correspondance entre modèles neuronaux biologiques et artificiels	67
Tableau 3.2	Fonctions de transfert usuelles, $a = f(n)$,	70
Tableau 3.3	Résumé des caractéristiques des algorithmes d'apprentissage de rétro propagation implantés dans MatLab®	77
Tableau 4.1	Domaine de variations des paramètres d'entrés	80
Tableau 4.2	Performances du réseau de neurones pour D	85
Tableau 4.3	Poids de connexions pour D	89
Tableau 4.4	Performances du réseau de neurones pour Hc.	90
Tableau 4.5	Poids de connexions pour Hc	92
Tableau 4.6	Paramètres pour la mise en œuvre de l'algorithme AG	96
Tableau 4.7	Paramètres optimaux de la taille des cristallites	97
Tableau 4.8	Paramètres optimaux de la coecivité	98
Tableau 4.9	Paramètres pour la mise en œuvre de l'algorithme NSGA-II	100
Tableau 4.10	Solutions pour la taille des cristallites et le temps de broyage	102
Tableau 4.11	Solutions de la coercivité et le temps de broyage	103
Tableau 4.12	Solutions de la taille des cristallites et la coercivité	104
Tableau 4.13	Paramètres pour la mise en œuvre de l'algorithme MGGP	108
Tableau 4.14	Données statistiques pour les phases de formation et de test	109

INTRODUCTION

Les matériaux nanostructurés, aussi appelés matériaux nanocristallins ou nanomatériaux, ont suscité un intérêt croissant dans de nombreuses recherches en raison de leurs propriétés souvent plus intéressantes que celles des matériaux massifs de même composition. Cela est dû à la taille nanométrique des cristallites, qui confère aux matériaux des propriétés particulières et ouvre de nouveaux domaines d'application dans divers secteurs technologiques [1-2].

Les matériaux magnétiques nanocristallins ont été largement étudiés en raison de leurs propriétés remarquables qui diffèrent considérablement de celles des matériaux microcristallins et sont étroitement liées à la structure et à la microstructure [3-5].

Les alliages Fe-Ni sont connus depuis longtemps et largement utilisés dans les composants magnétiques de l'industrie de l'énergie électrique, tels que les transformateurs et les noyaux magnétiques, en raison de leurs excellentes propriétés magnétiques douces, à savoir une faible coercivité et une magnétisation à saturation élevée [6].

L'intérêt suscité par ces alliages magnétiques nanostructurés provient de leurs propriétés structurales et chimiques qui sont importantes dans la détermination de leurs propriétés magnétiques [7]. De plus, ces nanomatériaux revêtent un grand intérêt pour la recherche en magnétisme, car la réduction de la taille des grains à l'échelle du domaine magnétique ouvre de nouvelles perspectives pour les nanomatériaux magnétiquement doux [8].

Plusieurs méthodes ont été utilisées pour l'élaboration des alliages nanocristallins de Fe-Ni, parmi lesquelles figurent le broyage mécanique à haute énergie, également appelé mécanosynthèse. Cette méthode est une technique

d'élaboration hors équilibre puissante qui permet de synthétiser des alliages Fe-Ni sous forme de poudre nanocristalline [9-12].

Plusieurs travaux de recherche ont été menés sur les alliages Fe-Ni élaborés par mécanosynthèse pour différentes conditions d'élaboration à l'aide de plusieurs types de broyeurs. Ces études ont donné des résultats intéressants concernant la variation des paramètres magnétiques en fonction de la durée de broyage et donc de la taille des grains [13-17].

Le but de cette étude est d'optimiser les propriétés structurales (taille des cristallites) et les propriétés magnétiques (champ coercitif) en utilisant des techniques d'intelligence artificielle telles que les réseaux de neurones et les algorithmes génétiques.

Dans un premier temps, des modèles de prédiction ont été développés en utilisant la technique des réseaux de neurones. Ces modèles se basent sur les résultats expérimentaux disponibles dans la littérature et prennent en compte les paramètres les plus influents sur le phénomène de broyage, tels que la vitesse du plateau, la vitesse des jarres, le temps de broyage et la composition chimique de l'alliage (paramètres d'entrée). Les propriétés spécifiques recherchées, comme la taille des cristallites et la coercivité, sont considérées comme les paramètres de sortie.

Les réseaux de neurones ont été choisis pour leur capacité d'apprentissage des données, leur robustesse et leur adaptabilité aux données manquantes. En effet, ces dernières années, les réseaux de neurones sont devenus un sujet d'actualité et de recherche très réussi. Ils sont appliqués avec succès à la modélisation de nombreux systèmes industriels, dans lesquels les relations entre les variables d'entrée et de sortie sont complexes et les règles de fonctionnement sont inconnues.

Dans l'étape suivante, l'objectif principal est de définir une approche permettant d'aider le décideur à choisir les paramètres de broyage afin d'optimiser les propriétés des poudres nanocristallines Fe-Ni préparées par mécanosynthèse. Cette étape est divisée en trois parties principales. La première partie présente

une description détaillée des algorithmes génétiques mono-objectifs et rappelle les définitions relatives à leur fonctionnement. La deuxième partie est consacrée à la programmation génétique, qui s'inspire des algorithmes génétiques et vise à créer une relation mathématique entre les paramètres de broyage et les propriétés nanocristallines Fe-Ni. La troisième partie traite de l'optimisation multi-objectifs à l'aide de l'algorithme NSGA-II (Non-dominated Sorting Genetic Algorithm-II). Il s'agit de l'algorithme le plus connu et le plus utilisé actuellement pour résoudre des problèmes d'optimisation multi-objectifs.

Cette thèse est organisée en quatre chapitres :

Le premier chapitre présente une étude bibliographique composée de trois parties. La première partie aborde les généralités sur les nanomatériaux, en indiquant brièvement leurs propriétés, leurs applications et leurs procédés d'élaboration, notamment la mécanosynthèse qui est détaillée. La deuxième partie traite des notions sur le magnétisme et les matériaux magnétiques. La troisième partie est consacrée aux études antérieures sur les alliages nanocristallins de Fe-Ni ainsi que l'application des AG-ANN dans la mécanosynthèse.

Le deuxième chapitre est consacré à l'optimisation par les algorithmes génétiques. La première partie traite des algorithmes génétiques mono-objectifs. La deuxième partie présente la programmation génétique multi-gènes. La troisième partie explique les notions de base sur lesquelles reposent les algorithmes multi-objectifs et présente en détail le principe de fonctionnement de l'algorithme NSGA-II.

Le troisième chapitre présente la théorie des réseaux de neurones. Après avoir exposé les différents éléments de base de cet outil mathématique, nous nous limitons à des aspects utiles pour la suite de ce travail.

Le quatrième et dernier chapitre aborde dans sa première partie la mise en œuvre des réseaux de neurones pour modéliser la relation entre les paramètres sélectionnés et les propriétés spécifiques recherchées. Dans sa deuxième partie, un couplage de réseaux de neurones et d'algorithmes génétiques a été réalisé

dans le but d'optimiser les propriétés nanocristallines Fe-Ni obtenues par mécanosynthèse.

Enfin, cette étude se termine par une conclusion générale qui rappelle les principaux résultats de ce travail, et les perspectives qu'ils permettent d'envisager.

CHAPITRE 1

ETUDE BIBLIOGRAPHIQUE

1.1 Généralités sur les nanomatériaux

1.1.1 Les nanomatériaux

Les nanomatériaux sont définis comme des matériaux constitués de nano-objets dont la taille se situe entre 1 et 100 nm [18]. Comme illustré dans la figure 1.1, ces matériaux contiennent une proportion significative d'atomes situés aux limites des grains, ce qui leur confère des propriétés intéressantes qui ne se retrouvent pas dans l'équivalent massif [19].

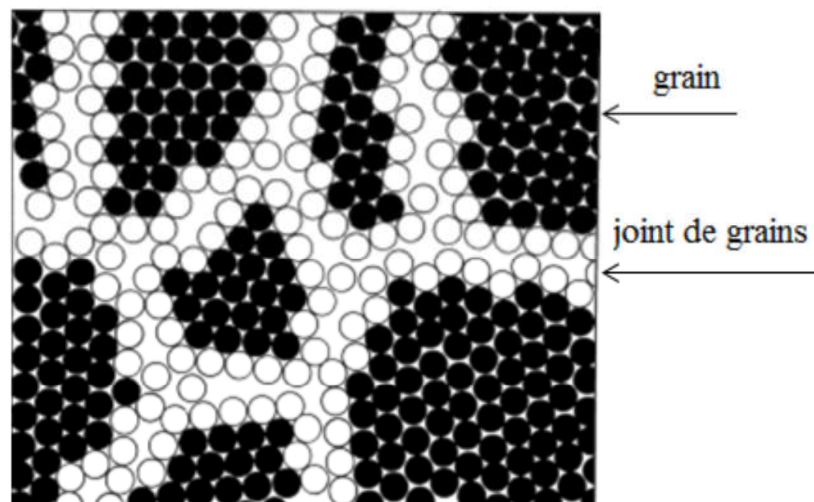


Figure 1.1 :Représentation schématique des grains et joints de grains d'un nanomatériau [19]

Selon le matériau et la technique d'élaboration, les joints de grains peuvent avoir différentes épaisseurs, allant d'une à plusieurs couches atomiques. Plus la taille des grains est petite, plus le rapport surface/volume est important. La fraction d'atomes localisés aux joints des grains est d'environ 3 % pour des grains de 100

nm et de plus de 50 % pour des grains de moins de 5 nm, avec une épaisseur de joint d'environ 1 nm [20]. La figure 1.2 illustre cette évolution du pourcentage d'atomes situés au joint de grains d'un solide nanocristallin en fonction de la taille des grains.

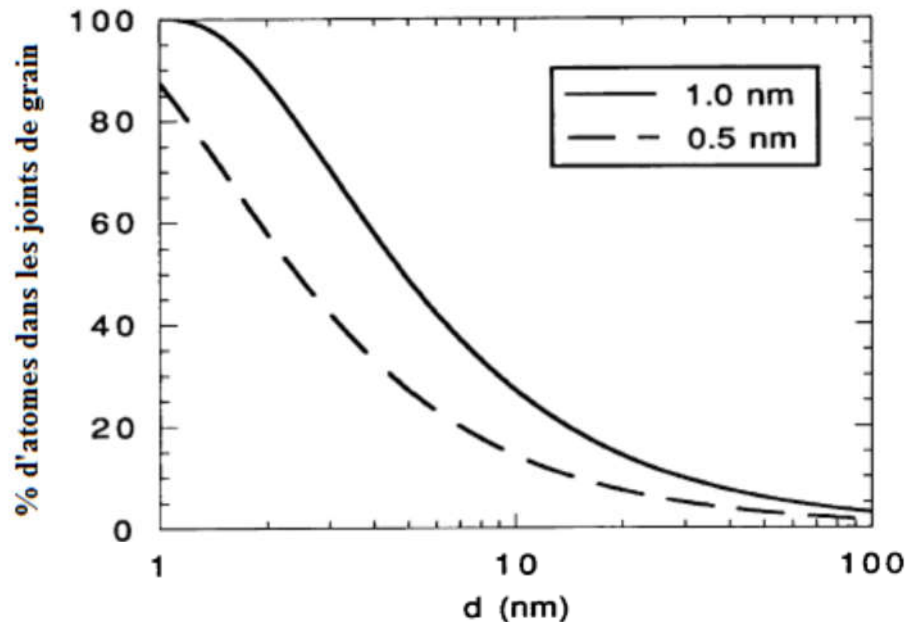


Figure 1.2 : Evolution du pourcentage d'atomes situés au joint de grains d'un solide nanocristallin en fonction de la taille des grains [20].

1.1.2 Classification des alliages nanocristallins

Les nano-objets Les nano-objets peuvent se présenter sous forme de particules, de fibres ou de tubes (appelés charges ou renforts), de couches minces ou de composants structuraux. Ils sont utilisés soit tels quels (comme catalyseurs, pour le transport de médicaments, pour le polissage de plaquettes de silicium et de disques durs en microélectronique, etc.), soit dans le but de développer de nouveaux matériaux. Ils peuvent être incorporés dans une matrice pour apporter de nouvelles fonctionnalités ou modifier les propriétés mécaniques, optiques, magnétiques ou thermiques.

Les nanomatériaux peuvent être classés en quatre catégories en fonction de la dimension de leur nanostructure [21]:

- **Matériaux de dimension zéro:** Une dimension zéro signifie que le matériau a une taille nanométrique dans les trois directions de l'espace et n'a donc aucune dimension micro ou macroscopique. Cela correspond à la définition des nanoparticules [21]. Les nanoparticules peuvent être isolées ou s'agréger pour former des amas nanostructurés. Elles peuvent être déposées sur une surface, en suspension dans un gaz ou dans un liquide, ou en état de poudre.
- **Matériaux de dimension 1:** Dans le domaine des nanomatériaux à une dimension, on trouve les nanofils et les nanotubes [21].
- **Matériaux de dimension 2:** Il s'agit de matériaux sous forme de couches minces, comme les dépôts d'agrégats ou les revêtements [21].
- **Matériaux de dimension 3:** Il s'agit de matériaux sous forme compacte (nanocomposites), comme les céramiques et les métaux nanostructurés [21].

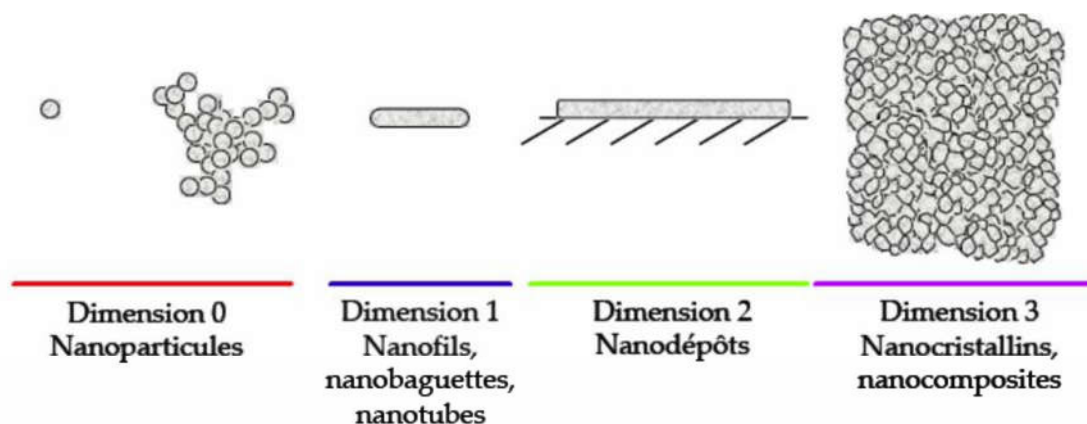


Figure 1.3 : Classification des matériaux nanostructurés [21]

1.1.3 Méthodes d'élaboration des nanomatériaux

Les méthodes d'élaboration des nanomatériaux sont classées en 3 grandes classes [22]:

a) Technique par voie physique

L'élaboration des nanoparticules peut être réalisée à partir d'une phase vapeur extraite d'un matériau source par chauffage (fusion en creuset ou sans) ou par bombardement (faisceau d'électrons). Dans la plupart des cas, la vapeur du

solide que l'on souhaite former est refroidie par collision avec un gaz neutre et le matériau est collecté sur une paroi froide.

b) Technique par voie chimique

Plusieurs techniques de fabrication sont couramment utilisées dans l'industrie. Dans les réactions en phase vapeur, les matériaux précurseurs vaporisés sont introduits dans un réacteur CVD (Chemical Vapor Deposition) dans lequel les molécules de précurseurs sont adsorbées à la surface d'un substrat maintenu à une température adaptée.

Les réactions en milieu liquide donnent lieu à des synthèses et sont le plus souvent effectuées à partir d'une solution aqueuse ou organique contenant les réactants. La précipitation des nanoparticules est obtenue par une modification des conditions de l'équilibre physico-chimique, soit par coprécipitation.

Enfin, les techniques sol-gel permettent de produire des nanomatériaux (optiques, magnétiques, électroniques, catalyseurs, énergétiques, capteurs...) par le biais de réactions de polymérisations inorganiques.

c) Technique par voie mécanique

Parmi les techniques d'élaboration par méthode mécanique, la technique dite mécanosynthèse consiste généralement à broyer des poudres micrométriques qui en général, varient de 1 à 30 μm d'un ou de plusieurs alliages.

C'est la technique sur laquelle nous nous sommes basés dans notre étude, en nous intéressant particulièrement à l'optimisation des matériaux nanostructurés élaborés par la mécanosynthèse.

1.1.4 Propriétés des nanomatériaux

Les nanomatériaux présentent des propriétés nouvelles et intéressantes : très souvent, leurs caractéristiques diffèrent notablement de celles des mêmes matériaux à l'état massif. Par exemple sur le plan mécanique, la microdureté du $\text{Fe}_{60}\text{Co}_{40}$ élaboré par broyage mécanique augmente à mesure que la taille des cristallites diminue, et cela est dû aux microdistorsions induites par les dislocations créées durant la phase de mécanosynthèse [23].

L'effet de la nanostructuration se traduit par un phénomène de super élasticité : un matériau en cuivre nanocristallin obtenu par laminage à froid peut avoir un allongement extrêmement important (de plus de 5000 %) [24]. La nanostructuration permet d'améliorer la dureté et la résistance des matériaux sans compromettre leur ductilité. En effet, dans un métal classique, il existe des dislocations, qui sont des défauts de la structure reliant les agrégats. Lorsqu'on exerce une contrainte sur le métal, cela provoque un déplacement des dislocations et le métal se déforme.

Cependant, dans les nanomatériaux, la fraction de dislocation est très faible par rapport à la fraction des joints des grains et des joints triples, ce qui rend la résistance à la déformation beaucoup plus importante: le matériau à l'état nanocristallin est plus dur que dans l'état massif [25].

Les propriétés électriques sont également influencées par la diminution de la taille des grains. En effet, lorsque la taille des grains des matériaux est réduite à l'échelle nanométrique, la résistivité électrique devient très élevée par rapport à celle des matériaux classiques à l'état massif. Par exemple, la résistivité mesurée dans le nickel nanométrique (avec une taille des grains de 11 nm) est trois fois plus grande que celle mesurée dans le nickel polycristallin à l'état massif (avec une taille des grains de 100 μm). Des augmentations similaires sont observées dans le cuivre, le fer et le plomb nanométriques [25].

L'introduction de nanoparticules et/ou de nanotubes peut modifier de manière drastique la conductivité électrique de matériaux considérés comme isolants [26]. Le fait que les nanoparticules aient des dimensions inférieures aux longueurs d'onde du visible (380-780 nm) permet d'améliorer les propriétés optiques du matériau [26]. L'utilisation de nanoparticules permet également d'améliorer certaines propriétés de transfert thermique, même pour de faibles fractions volumiques. Par exemple, l'ajout de 0,2 % en fraction volumique de nanoparticules de fer entraîne une augmentation de plus de 10 % de la conductivité thermique du fer massif [27]. Dans le domaine de l'enregistrement magnétique, l'utilisation des nanotechnologies a permis d'augmenter de près de huit ordres de grandeur la densité de stockage [27].

1.1.5 Domaines d'application des nanomatériaux

Durant ces dernières années, grâce à leurs nouvelles caractéristiques physico-chimiques, les structures à taille nanométrique ont été exploitées par les chercheurs dans différents domaines scientifiques (Nano-biologie, Nano-systèmes..). Les nanomatériaux sont toujours utilisés pour une fonction particulière. Toutes les grandes familles de matériaux sont concernées : les métaux, les céramiques, les oxydes magnétiques, les diélectriques, les carbones, les polymères, etc. Du fait de leurs propriétés variées et souvent inédites, les nanomatériaux recèlent de potentialités très diverses et leurs utilisations ouvrent plusieurs perspectives.

Les nanomatériaux permettent ainsi des innovations incrémentales et de rupture dans de nombreux domaines d'activité tels que : la santé, l'automobile, la construction, l'agroalimentaire ou encore l'électronique. L'application des nanomatériaux magnétiques a ainsi permis de développer de nouvelles gammes, d'utilisation à champ coercitif et à champ à saturation variable, non réalisables avec les matériaux classiques.

Tableau 1.1 : Domaines d'applications des nanomatériaux [28].

secteurs d'activité	exemples d'applications
Automobile, aéronautique et aérospatial	Matériaux renforcés et plus légers ; peintures extérieures avec effet de couleur, plus brillantes, anti-rayures, anti-corrosion et anti-salissures ; capteurs optimisant les performances des moteurs ; détecteurs de glace sur les ailes d'avion; pneumatiques plus durables et recyclables...
agroalimentaire	Emballages actifs ; additifs : colorants, anti-agglomérants,...
Electronique et communications	Mémoires à haute densité et processeurs miniaturisés; cellules solaires ; bibliothèques électroniques de poche ; ordinateurs et jeux électroniques ultra-rapides; écrans plats...

Chimie et matériaux	Pigments ; charges ; poudres céramiques ; inhibiteurs de corrosion ; catalyseurs multi-fonctionnels ; textiles et revêtements anti-bactériens et ultra-résistants...
Pharmacie et santé	Médicaments et agents actifs ; surfaces adhésives médicales antiallergènes ; médicaments sur mesure délivrés uniquement à des organes précis ; vaccins oraux ; imagerie médicale...
Construction	Ciments autonettoyants et anti-pollutions, vitrages autonettoyants et anti-salissures ; peintures ; vernis ; colles ; mastics...
Défense	Détecteurs d'agents chimiques et biologiques ; systèmes de surveillance miniaturisés ; systèmes de guidage plus précis ; textiles légers et qui se réparent d'eux-mêmes...
Énergie	Cellules photovoltaïques nouvelle génération ; nouveaux types de batteries ; fenêtres intelligentes ; matériaux isolants plus efficaces ; entreposage d'hydrogène combustible...
Environnement et écologie	Diminution des émissions de dioxyde de carbone ; production d'eau ultra pure à partir d'eau de mer ; fertilisants plus efficaces et moins dommageables ; analyseurs chimiques spécifiques...

1.2 Elaboration des matériaux nanostructurés par broyage mécanique

L'élaboration de poudres nanostructurées et/ou de poudres dont la taille des particules est nanométrique repose sur des voies d'élaboration variées. Parmi celles-ci, on trouve le broyage à haute énergie (mécanosynthèse). Il s'agit d'une technique d'élaboration hors équilibre qui permet de produire des alliages nanostructurés à partir d'un mélange de poudre élémentaire.

a) Broyage mécanique à haute énergie

Contrairement au broyage classique utilisé depuis la préhistoire pour réduire en poudre les matériaux, le broyage à haute énergie permet de synthétiser, à température ambiante, des matériaux organisés à l'échelle nanométrique. Cette méthode est difficile, voire impossible à obtenir par des méthodes classiques telles que la fusion-solidification. J. Benjamin a développé cette méthode de synthèse dans les années 60 afin de réaliser des dispersions d'oxydes (Al_2O_3 , Y_2O_3) dans des alliages de nickel dans le but d'améliorer leur propriété mécanique [29].

À partir des années 80, cette technique a été utilisée pour réaliser des alliages amorphes [30, 31], des solutions solides sursaturées d'éléments immiscibles et des matériaux nanostructurés [32, 33]. Deux termes sont couramment utilisés dans la littérature anglo-saxonne pour désigner le processus de broyage à haute énergie : " mechanical alloying " qui est la mécanosynthèse proprement dite, puissant outil permettant la production de poudres nanostructurées métastables à partir d'un mélange de poudres élémentaires (les poudres initiales ayant une composition différente de celle du produit), et " mechanical milling ou grinding " qui est la mécanosynthèse, broyage mécanique non réactif qui consiste à confiner des poudres (de métal pur ou d'alliages) de composition stoechiométrique pour obtenir un matériau nanostructuré non nécessairement homogène.

b) Principe du broyage à haute énergie

Le principe du broyage mécanique à haute énergie consiste à agiter une jarre contenant des billes d'acier non oxydables et de la poudre. Cette poudre peut être un mélange de poudre d'éléments purs ou de poudre d'un pré-alliage.

Grâce à la force centrifuge et sous l'effet des collisions répétées des billes entre elles et des billes avec la paroi de la jarre, le composé va être fracturé jusqu'à l'obtention de particules de dimension nanométrique. Différents types de broyeurs peuvent être utilisés, tels que le broyeur attriteur, où les billes sont mises en mouvement par l'intermédiaire de palettes en rotation (figure 1.4 a), ou encore le broyeur vibrant où le mouvement des billes est provoqué par des vibrations haute fréquence (figure 1.4 b) et le broyeur planétaire, où le mouvement des billes

résulte de la rotation de la jarre (figure 1.5). Pour cette étude, nous avons utilisé le troisième type.

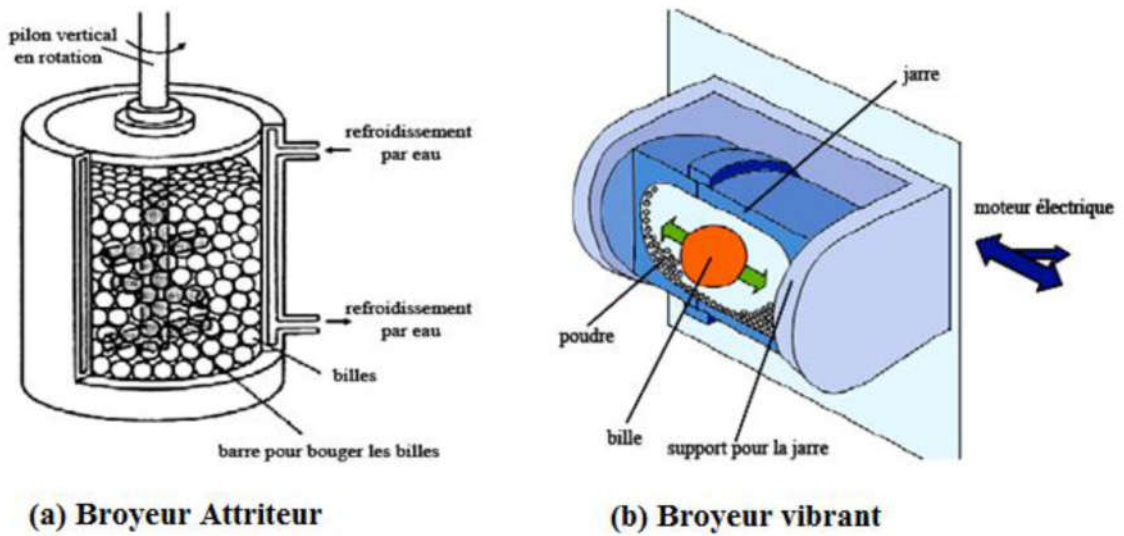


Figure 1.4 : Broyeurs attriteur et vibrant.

Le broyeur planétaire

Dans ces types de broyeurs, connus comme broyeurs centrifuges, les jarres dans lesquelles on place des billes de 10 ou 20 mm de diamètre avec la poudre, sont fixées sur des satellites tournants, qui sont montés sur un plateau tournant. Les jarres tournent avec une vitesse angulaire ω , tandis que le plateau tourne avec une vitesse Ω (figure 1.5).

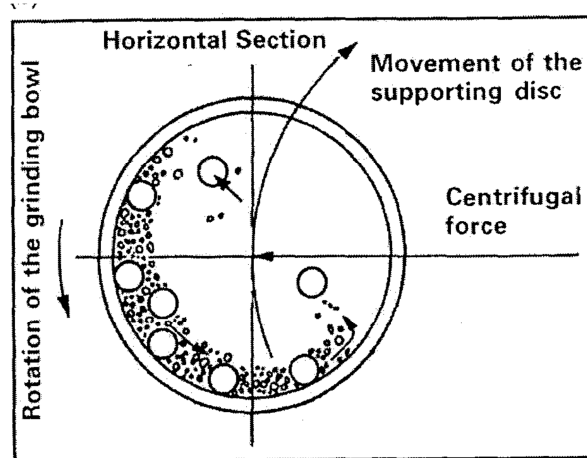


Figure 1.5 : Schématisation d'un broyeur à mouvement planétaire et de l'action des billes pendant le processus.

Le broyage est effectué soit par chocs, soit par frottements. En l'absence de chauffage externe, la température moyenne des récipients est comprise entre 50 et 120 °C, selon la vitesse des billes. L'élévation locale de température est comprise entre 60 et 300 °C. Afin d'éviter une élévation excessive de la température, le broyage est généralement réalisé en plusieurs cycles, avec des périodes de repos. L'état stationnaire est généralement atteint après 24 à 100 heures de broyage, selon les matériaux broyés [34].

c) Mécanisme physique du broyage

Le broyage mécanique pour la production d'un composé à partir d'éléments purs ou combinés repose sur les phénomènes répétés de soudage et de fracture résultant de l'action des chocs entre les billes, la poudre et les parois du récipient. Ces phénomènes de soudage et de fracture des poudres assurent l'échange de matière entre les particules pendant le broyage, ce qui conduit à un mélange intime à l'échelle atomique des éléments de départ. La fracture des particules de poudre entraîne la formation de nouvelles surfaces qui peuvent adhérer à d'autres particules ou sur lesquelles de nouveaux grains peuvent se former. Le mélange au niveau atomique est facilité par la diffusion à froid des atomes du soluté dans le réseau du solvant grâce aux défauts créés pendant l'action mécanique. L'énergie introduite dans la poudre peut conduire à la formation de nouvelles phases (amorphes, solutions solides, composés intermétalliques, etc.).

En effet, le produit final est obtenu lorsque l'équilibre entre les processus de fracture et de soudage est atteint, ce qui conduit à une dimension stable des particules. La différence entre les procédés classiques et le broyage mécanique réside dans le fait que dans ce dernier, les transformations de phase se produisent dans un processus dynamique sous l'effet continu des déformations mécaniques et de la chaleur générée par les chocs de déformation [35].

d) Les paramètres impliqués dans la mécanosynthèse

Le broyage mécanique est un processus qui implique un grand nombre de paramètres qui ont un effet considérable sur la nature du produit final. On peut distinguer les paramètres contrôlables et les paramètres non contrôlables (mal maîtrisés).

Les paramètres contrôlables comprennent :

- L'énergie du broyeur : elle dépend de chaque type de broyeur. En principe, une énergie élevée conduit à une obtention plus rapide du produit final [36]. L'énergie du broyeur est également liée à la vitesse de broyage. Une vitesse trop élevée fait que les billes restent collées sur les parois sans agir sur la poudre et augmente la température de la jarre, ce qui a des effets négatifs sur le produit final (par exemple, l'oxydation) [37-38].
- La nature des billes et des jarres : elle prend en compte le type d'acier des billes, leur dimension et leur nombre, qui ont une influence sur l'efficacité du broyage [37-39].
- Le rapport masse bille/masse poudre : il influence la formation des phases dans les poudres broyées. Il peut varier de 1/1 à 220/1, et à mesure que ce rapport augmente, le temps nécessaire pour obtenir le produit final peut être réduit [37-39].
- Le degré de remplissage (billes + poudre) : c'est un autre facteur important dans le broyage mécanique. Un remplissage trop faible ou trop important diminue l'efficacité du processus ; un taux de remplissage optimal est atteint à 50-60% du volume de la jarre. Le produit final peut dépendre du taux de remplissage de la jarre avec les billes [37-39].
- Le temps de broyage : c'est le paramètre qui décrit l'intervalle de temps nécessaire pour obtenir le produit final. Il dépend du type de broyeur utilisé, du mode d'action des billes sur la poudre (collisions élastiques ou frottements) et de la température de broyage [39].
- Broyage continu ou séquentiel : le broyage continu peut conduire à l'obtention d'un produit différent de celui obtenu lors d'un broyage avec des pauses [40].

Les paramètres non contrôlables comprennent :

- L'atmosphère dans laquelle se poursuit le broyage : elle peut influencer la nature et la taille du produit final. En général, des gaz inertes tels que l'argon (Ar) ou l'hélium (He) sont utilisés pour éviter la contamination, mais l'air, le diazote (N₂) ou l'hydrogène (H₂) peuvent également être utilisés pour produire des nitrures, des hydrures, des oxydes, etc. [37-38]. Il a été constaté que les gaz inertes

peuvent créer des défauts dans la poudre finale, mais aussi qu'ils peuvent être "piégés" dans le composé final [41]. Selon la nature de l'atmosphère de broyage, la mécanosynthèse peut être réalisée dans un environnement sec (air, Ar, He, N₂ ou H₂) ou humide (des composés organiques peuvent être introduits, par exemple, dans l'atmosphère de gaz inerte).

- La température de broyage : elle influence la formation de l'état final du produit. Une température élevée entraîne une augmentation de la taille des cristallites, mais réduit leurs contraintes et leur solubilité à l'état solide [37-38]. La température varie pendant le processus de broyage ; une température plus élevée est observée pendant la période de formation du produit final. On constate une différence de température entre les billes et la paroi ainsi qu'entre différents types de poudres broyées (ductiles, fragiles) [42].

- La contamination : c'est le principal défi de la mécanosynthèse. Parmi les facteurs contribuant à la contamination des poudres, citons :

- Les billes et les jarres : la contamination en fer dépend de l'énergie du broyeur, du temps de broyage et de la température atteinte dans la jarre [37-38] ;
- L'atmosphère de broyage : réaction avec la poudre broyée et formation de produits indésirables.

1.3. Généralités sur les matériaux magnétiques

1.3.1 Matériaux magnétiques

Les matériaux les plus importants du point de vue magnétique en génie électrique sont les matériaux ferromagnétiques principalement composés de fer, de nickel ou de cobalt, et dont la résultante globale des moments magnétiques microscopiques est différente de zéro. La direction et le sens de ces moments magnétiques microscopiques sont identiques.

Le phénomène d'orientation des moments magnétiques à l'intérieur des matériaux ferromagnétiques se produit de manière localisée [43]. L'expérience montre que tous les matériaux magnétiques d'un volume suffisant se divisent spontanément en régions plus petites appelées " domaines magnétiques " ou " domaines de Weiss " [44].

À l'intérieur d'un domaine, tous les moments magnétiques atomiques sont maintenus parallèles par les forces d'échange, de telle sorte que chaque domaine se présente comme un petit volume aimanté à saturation (J_s) avec une orientation différente de celle de ses voisins. Les interfaces entre les domaines sont appelées " parois de Bloch ", à travers lesquelles l'orientation des moments magnétiques passe progressivement d'un domaine à l'autre (figure 1.6 b).

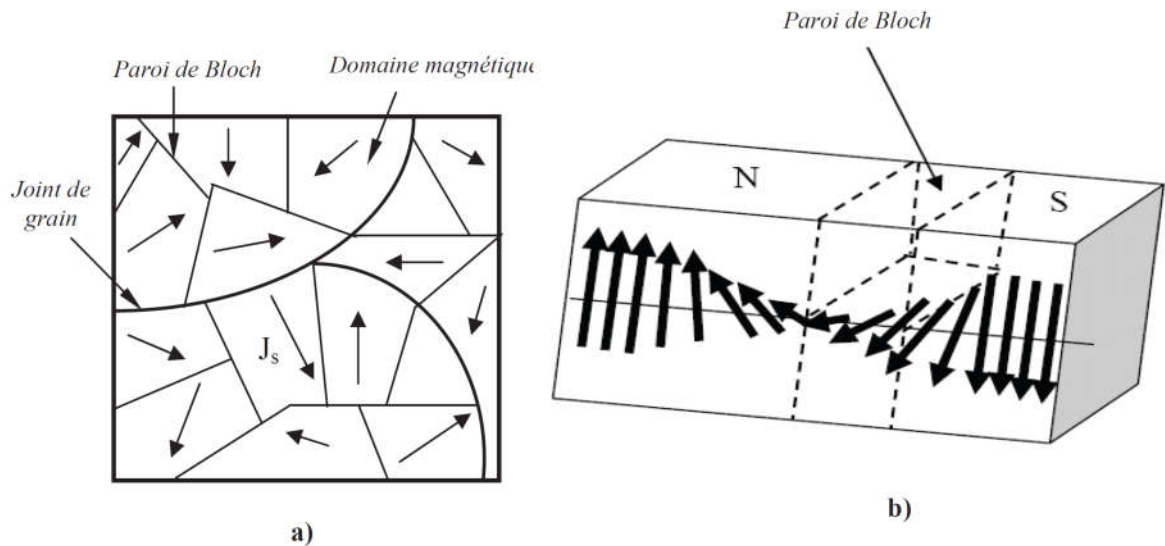


Figure 1.6. a) Domaines de Weiss; b) Rotation des moments magnétiques dans une paroi de Bloch à 180° [45].

1.3.2 Influence d'un champ magnétique extérieur

L'application d'un champ magnétique extérieur H se traduit par l'introduction d'un nouveau terme d'énergie qui modifie l'état primitif d'équilibre énergétique du matériau créant ainsi une nouvelle configuration [45 - 46].

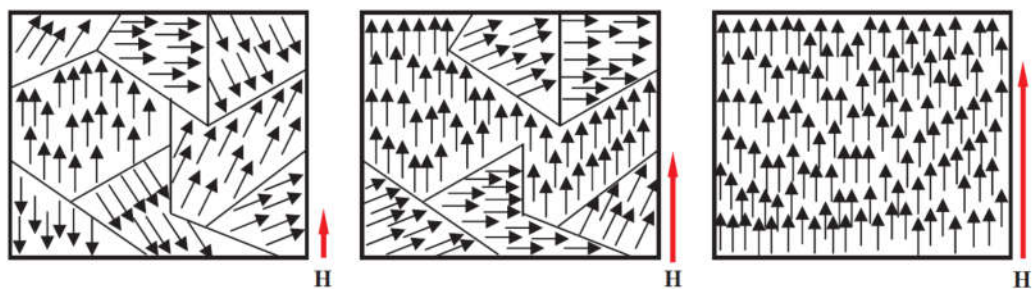


Figure 1.7 : Evolution des domaines de Weiss sous l'action d'un champ magnétique extérieur croissant dans un matériau ferromagnétique [45].

Tout d'abord, il y a une augmentation de certains domaines au détriment de leurs voisins par le déplacement des parois (figure 1.7). Pour de faibles champs, ce déplacement reste réversible. Lorsque les champs sont plus forts, les déplacements deviennent irréversibles : les domaines deviennent moins nombreux et leurs directions d'aimantation spontanée s'alignent progressivement dans la direction du champ H . Pour des champs importants, les dernières parois disparaissent et on obtient la saturation macroscopique du matériau (figure 1.7).

Le mouvement des parois dépend de plusieurs paramètres, notamment : la perméabilité du matériau, l'intensité de l'excitation, ainsi que le nombre d'obstacles.

1.3.3 Mécanisme d'aimantation et cycle d'hystérésis

On appelle hystérésis les phénomènes ou les mécanismes irréversibles qui se produisent lors de l'évolution de l'état du matériau. Le cycle d'hystérésis magnétique représente la variation de l'induction magnétique (ou de l'aimantation) d'un matériau en fonction du champ magnétique appliqué. Il fournit des informations sur les caractéristiques magnétiques intéressantes d'un échantillon.

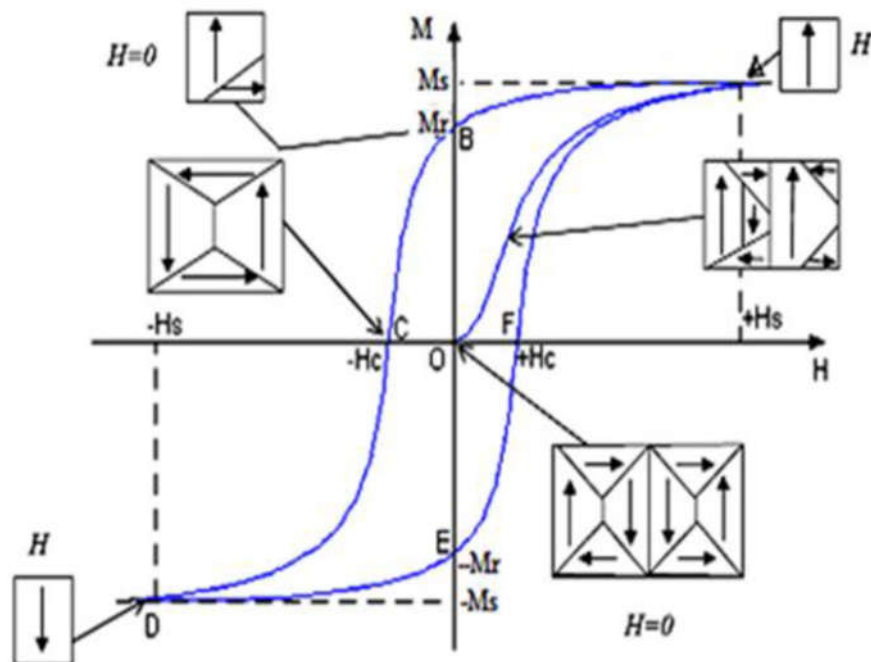


Figure 1.8 : Courbe de première aimantation (OA), et cycle d'hystérésis (ABCDEF) d'un matériau ferromagnétique [25].

En général, d'un point de vue pratique, on utilise le cycle d'hystérésis des matériaux magnétiques pour obtenir certaines grandeurs physiques telles que l'aimantation à saturation (M_s), l'aimantation rémanente (M_r) et le champ coercitif (H_c). Lorsqu'on place ce matériau ferromagnétique dans un champ H externe, les parois vont se déplacer de manière à renforcer ce champ H externe. Si H augmente considérablement, le domaine favorablement orienté occupera tout le volume du matériau qui sera alors aimanté à saturation. La courbe ci-dessous représente le cycle d'hystérésis du matériau ferromagnétique (figure 1.8).

Lorsque H augmente à partir de zéro, les parois de Bloch se déplacent, entraînant une aimantation de l'échantillon. Quand H est suffisamment intense (H_s), un seul domaine occupe tout l'échantillon. L'aimantation correspondante est l'aimantation à saturation (M_s). Si on diminue H , on oblige les parois à se déplacer à nouveau. Le mouvement de retour n'est pas le même que celui suivi lorsque H augmentait car une partie du mouvement des parois est irréversible (en raison des défauts cristallins). L'aimantation qui reste lorsque H est nul s'appelle aimantation rémanente à saturation (M_r). Le champ nécessaire pour annuler cette aimantation s'appelle le champ coercitif (H_c). Le champ permettant d'annuler l'aimantation rémanente s'appelle le champ coercitif rémanent (H_c). La courbe d'aimantation ainsi obtenue est une boucle d'hystérésis ou cycle d'hystérésis ($ABCDEF$); les paramètres caractéristiques de cette boucle sont les grandeurs M_s (H_s) et H_c . L'aire du cycle d'hystérésis représente l'énergie dépensée, par unité de volume du matériau, pour réorienter les moments magnétiques des domaines et déplacer les parois de Bloch, au cours d'un cycle complet de variation du champ magnétique externe [25].

1.3.4 Les différentes formes du cycle d'hystérésis

Les cycles d'hystérésis peuvent prendre diverses formes, qui peuvent varier fortement d'un matériau magnétique à l'autre.

Pour les matériaux ferromagnétiques et ferrimagnétiques, on distingue deux grandes classes de matériaux magnétiques en fonction de la forme du cycle d'hystérésis et de la valeur du champ coercitif : les matériaux magnétiques doux et les matériaux magnétiques durs. Dans la (figure 1.9), les cycles d'hystérésis

caractéristiques des matériaux magnétiques durs (a) et doux (b) sont présentés ; la courbe de première aimantation est indiquée avec une ligne en pointillé.

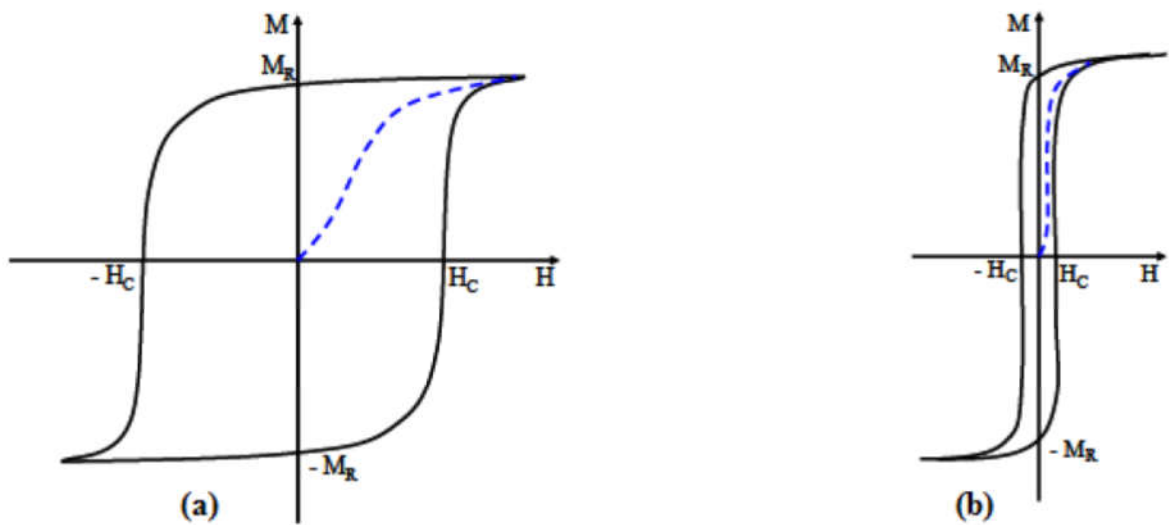


Figure 1.9 : Cycle d'hystérésis pour un matériau magnétique dur (a) et un matériau magnétique doux (b) [47]

a) Matériaux magnétique doux :

Les matériaux magnétiques doux se caractérisent par un cycle d'hystérésis étroit et possèdent une très grande perméabilité. Ils s'aimantent et se désaimantent facilement ; le champ coercitif ne dépasse pas 103A/m [48]. Pour obtenir un bon matériau magnétique doux, les parois de domaine doivent bouger facilement sous l'action du champ magnétique appliqué. Par conséquent, ces matériaux doivent avoir une faible quantité d'impuretés et de défauts. De nombreux matériaux magnétiques doux sont utilisés, tels que le fer pur, les alliages Fe-Ni (permalloy), les alliages Fe-Si, les alliages Fe-Al, les alliages Fe-Co, etc. [48].

b) Matériaux magnétiques durs :

Les matériaux durs présentent des cycles d'hystérésis extrêmement larges. Il s'agit d'aimants permanents qui ne se désaimantent pas facilement et qui ont une faible perméabilité ainsi qu'une forte valeur de champ coercitif (généralement supérieure à 10^4 A/m) [48].

Dans le cadre de notre étude, nous nous intéressons au comportement des matériaux ferromagnétiques doux. Ces matériaux doux possèdent un champ coercitif faible (en général inférieur à 1000A/m). Lorsque le champ extérieur varie, très peu d'obstacles s'opposent au déplacement des parois de Bloch et à la réorientation des domaines magnétiques. En d'autres termes, ces matériaux présentent peu de défauts dans leur structure cristalline ainsi qu'une anisotropie réduite. Ils sont caractérisés par une très haute perméabilité, une induction à saturation B_s élevée et de faibles pertes par hystérésis. Les matériaux magnétiques doux peuvent être aimantés facilement à l'aide de champs magnétiques faibles. Ils sont utilisés dans les noyaux (ou circuits magnétiques) des transformateurs, des moteurs et des générateurs, dans les inductances de précision des circuits électroniques, les écrans magnétiques, etc.

1.4 Travaux antérieurs sur les propriétés structurales et magnétiques des alliages Fe-Ni nanocristallins obtenus par mécanosynthèse.

Plusieurs chercheurs ont étudié les propriétés microstructurales et magnétiques des alliages Fe-Ni élaborés par mécano synthèse. D'après la littérature, nous avons constaté que ces propriétés dépendent de la composition et des conditions d'élaboration (composition de l'alliage, vitesse de broyage, temps de broyage, etc.). À cet égard, nous citerons quelques résultats des travaux de recherche qui ont été obtenus sur les alliages nanostructurés Fe-Ni obtenus par mécanosynthèse.

1.4.1 Propriétés structurales

1.4.1.1 Influence de la durée de broyage

Djekoun et al. [49] ont élaboré par mécanosynthèse les compositions $Fe_{(100-x)}Ni_x$ ($x = 22, 26, 30, 50$ at %) jusqu'à 96 h de broyage au moyen d'un broyeur planétaire de type Fritsch (Pulvérisette P7). La figure 1.10 présente l'évolution de la taille des cristallites D et de la microdistorsion e en fonction du temps de broyage pour le composé Fe-50 % Ni. Les auteurs ont observé que la taille des cristallites décroît de façon monotone tandis que la microdistorsion augmente avec l'augmentation de la durée de broyage.

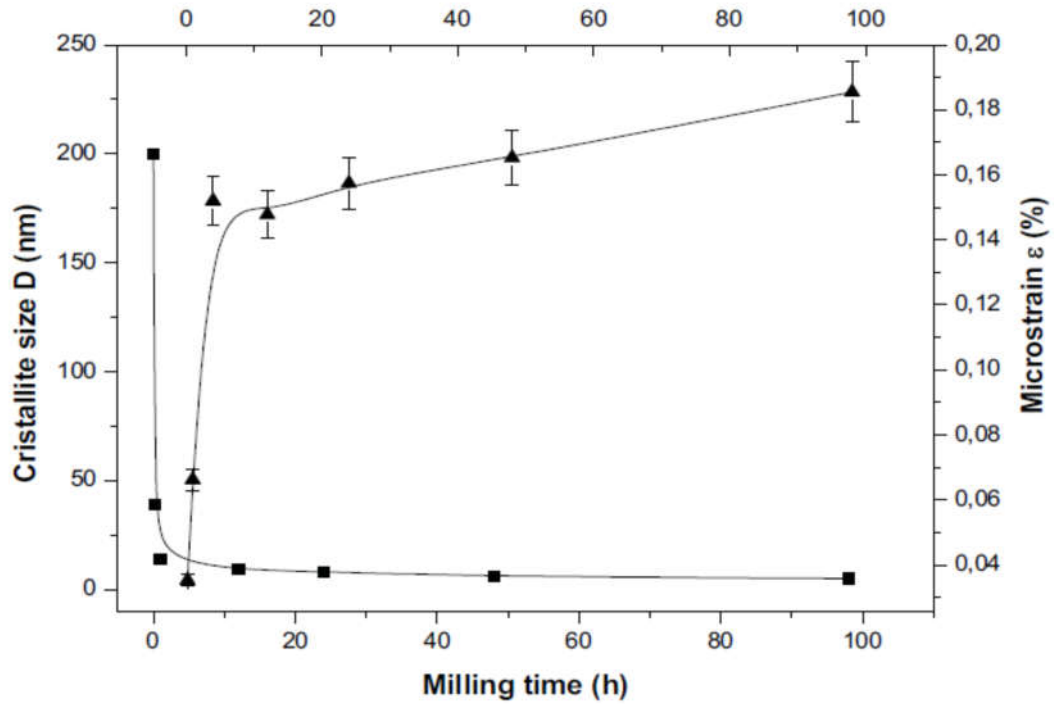


Figure 1.10 : Variation de D et ϵ de Fe-50 % Ni pour différents temps de broyage [49]

Hamzaoui et al. [50] ont élaboré dans un broyeur planétaire des alliages Fe-10%Ni et Fe-20%Ni. La figure 1.11 donne l'évolution du paramètre de maille en fonction de la durée de broyage.

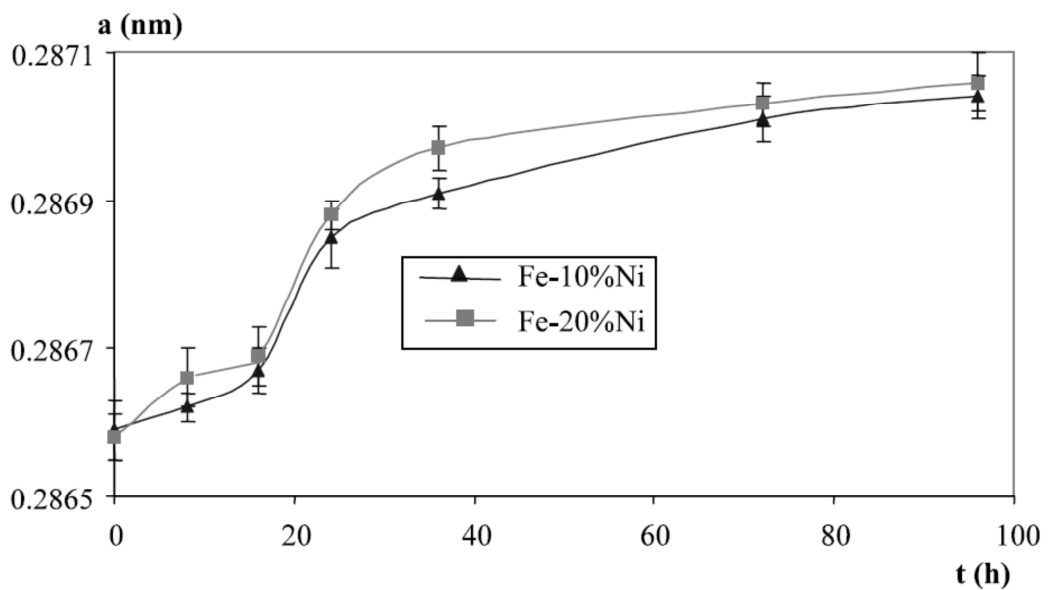


Figure 1.11 : Evolution du paramètre de maille en fonction de la durée de broyage des alliages Fe-10%Ni et Fe-20%Ni. [50]

Ils ont observé que la diminution de la taille des grains entraîne une augmentation des paramètres de maille, ce qui se traduit par une augmentation de la fraction volumique des joints de grains. Par conséquent, la structure moins dense des joints de grains peut exercer une pression négative sur les interfaces, ce qui peut entraîner une augmentation des paramètres de maille. On remarque également que les paramètres de maille augmentent lorsque la teneur en nickel augmente.

1.4.2 Influence de la vitesse de broyage

Kh. Gheisari et al. [51], ont étudié l'influence de la vitesse des jarres ω sur les propriétés structurales du composé Fe-45%Ni pendant 24 heures de broyage effectué à l'aide d'un broyeur planétaire à différentes vitesses du plateau Ω avec un rapport des vitesses $|\omega|/\Omega = 1.65$.

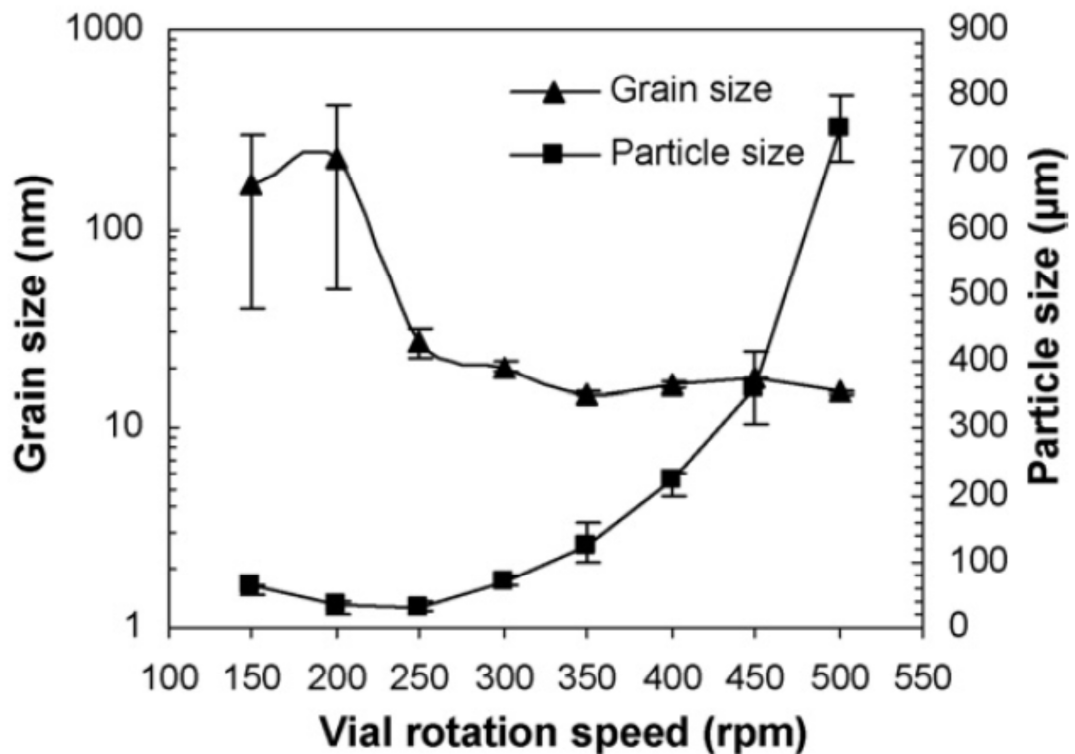


Figure 1.12 : Taille des particules (μm) et taille des cristallites (nm) de Fe-45%Ni en fonction de la vitesse de rotation des jarres [51]

Ces auteurs ont constaté que, lorsque l'énergie cinétique augmente et que la vitesse de broyage dépasse 250 rpm (ω), les phénomènes de soudage à froid

deviennent prédominants par rapport aux phénomènes de fragmentation, la taille des particules ont tendance à se souder et former des grosses particules, les fragments générés par ce mécanisme entraînent une réduction de la taille des grains, comme illustré dans la figure 1.12.

1.4.2 Propriétés magnétiques

1.4.2.1 Influence de la durée de broyage

Dans la figure 1.13, Jartych et al. [52] présentent l'évolution de l'aimantation à saturation en fonction de la durée de broyage des alliages Fe-20%Ni, Fe-35%Ni et Fe-50%Ni obtenus par mécanosynthèse avec un broyeur planétaire. Ils ont observé que l'aimantation à saturation augmente avec la durée de broyage. Cette augmentation est le résultat de la diminution de la taille des grains (lorsque la durée de broyage augmente, la taille des grains diminue), atteignant 10 nm après 50 heures de broyage.

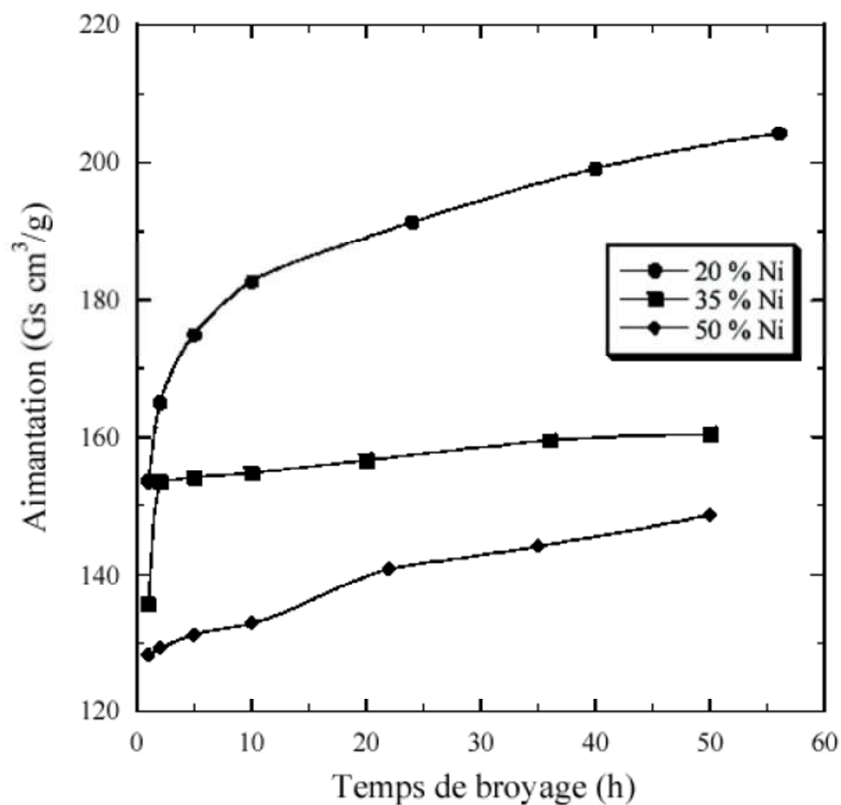


Figure 1.13 : Aimantation à saturation pour les alliages 20%Ni, 35%Ni et 50%Ni en fonction du temps de broyage [52].

Ces grains deviennent des domaines magnétiques éliminant ainsi l'influence des parois de Bloch. Ces auteurs ont également montré que les valeurs de l'aimantation à saturation décroissent lorsque la teneur en nickel augmente. Cette augmentation de l'aimantation est principalement liée à la formation de l'alliage [53].

Dans une autre étude, Guitoum et al [34] ont étudié le composé $\text{Fe}_{50}\text{Ni}_{50}$, élaboré dans un broyeur planétaire de type Retsch PM400. En ce qui concerne le comportement du champ coercitif H_c , ils ont constaté à partir de la figure 1.14, que pour un temps de broyage, t , inférieur à 8 h, H_c augmente lorsque t augmente. Cependant, pour $t > 8$ h, H_c augmente (diminue) de façon monotone pour les alliages $\text{Fe}_{50}\text{Ni}_{50}$ ($\text{Fe}_{80}\text{Ni}_{20}$). L'augmentation de H_c au début de la formation ($0 < t < 8$ h) du composé semble être liée à l'augmentation des contraintes dans les deux cas. De plus, au début de la formation du composé $\text{Fe}_{50}\text{Ni}_{50}$, la valeur minimale de H_c égale à 4.03 Oe a été enregistrée pour un temps de broyage de 8 h. Cependant pour l'alliage $\text{Fe}_{80}\text{Ni}_{20}$, la valeur minimale de H_c égale à 4.23 Oe a été atteinte après 36 h de broyage et pour une taille des grains de l'ordre de 11 nm.

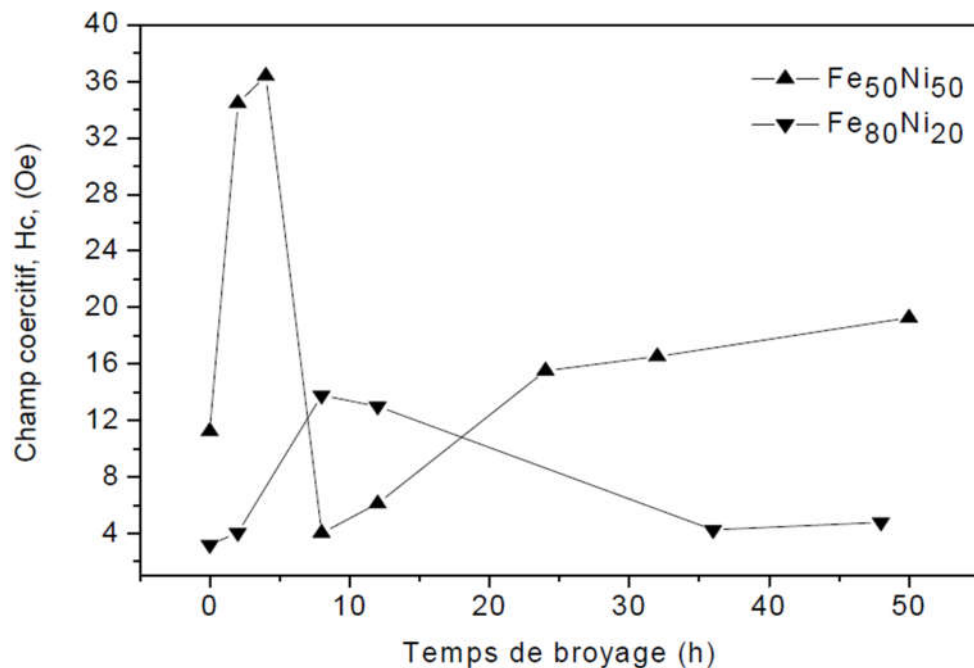


Figure 1.14 : Variation du champ coercitif en fonction de la durée de broyage pour les deux alliages $\text{Fe}_{50}\text{Ni}_{50}$ et $\text{Fe}_{80}\text{Ni}_{20}$. [34]

1.4.2.2 Influence de la composition

KUHRT et al [54] ont étudié l'évolution de l'aimantation à saturation M_s et du champ coercitif H_c en fonction de la composition de nickel dans un broyeur planétaire type P5, et cela pour différentes intensités de broyage. Ils ont remarqué d'après la figure 1.15 que l'aimantation à saturation décroît lorsqu'on passe de la région riche en fer à celle riche en nickel, avec une déviation aux alentours de 30 % Ni pour une intensité de 9, de 40% Ni pour une intensité de 7 et de 50% Ni pour une intensité de 5.

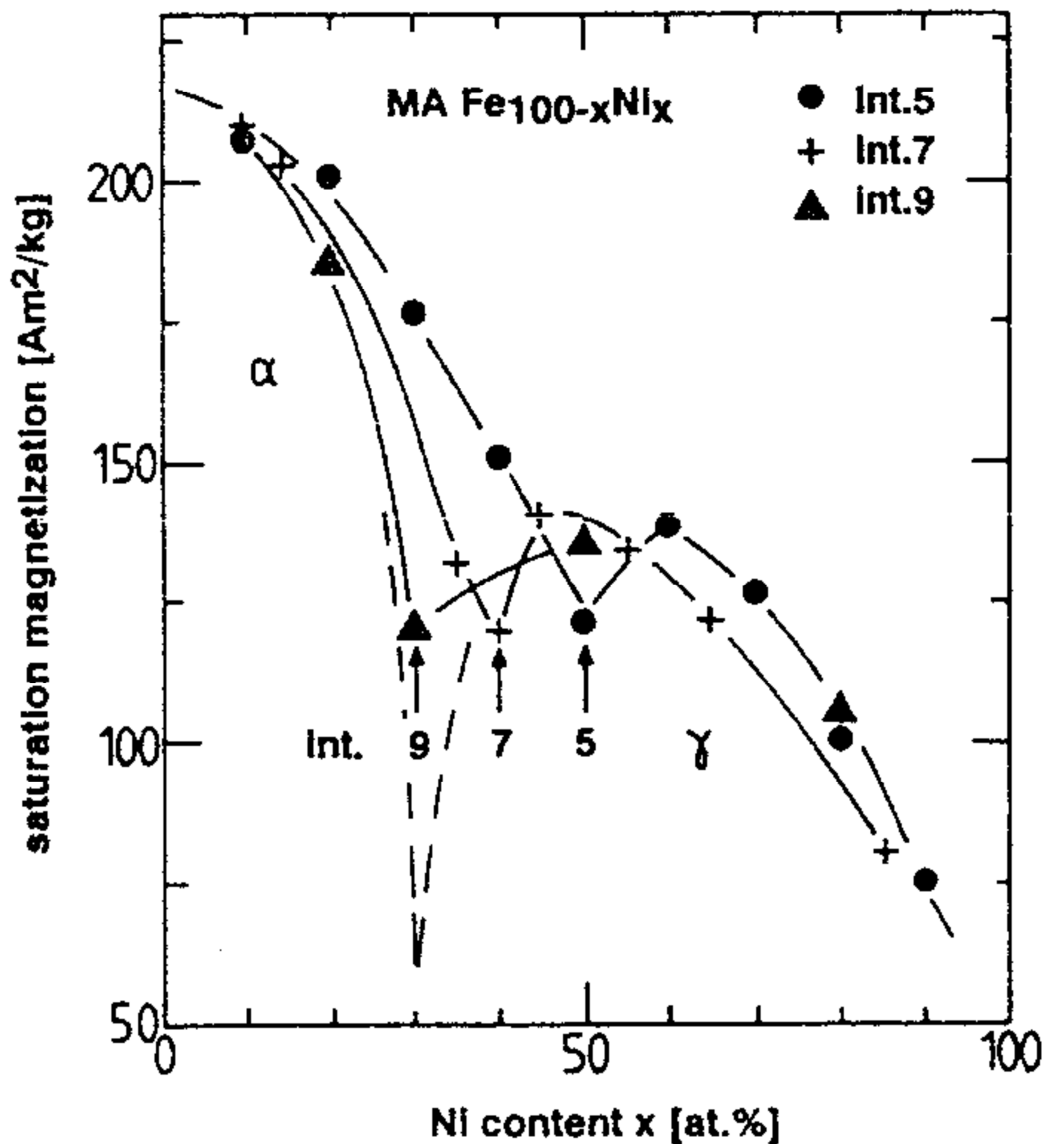


Figure 1.15 : Evolution de l'aimantation à saturation en fonction de la teneur en nickel pour différentes intensités de broyage [54]

Khurt et al. [54] ont étudié également l'évolution champ coercitif en fonction de la composition du nickel pour différentes intensités de broyage, ils ont constaté d'après la figure 1.16 que le changement de l'intensité de broyage n'a pas d'effet sur le comportement de la coercivité. Les valeurs élevées du champ coercitif sont dues aux fortes microdistorsions causées par le processus d'élaboration. Le maximum du champ coercitif est de 2000 A/m et le minimum est de ~ 400 A/m, situé à une composition correspondant à 80%Ni. Ils ont conclu que la mécanosynthèse n'est pas appropriée pour l'élaboration de matériaux magnétiquement doux à cause des microdistorsions engendrées.

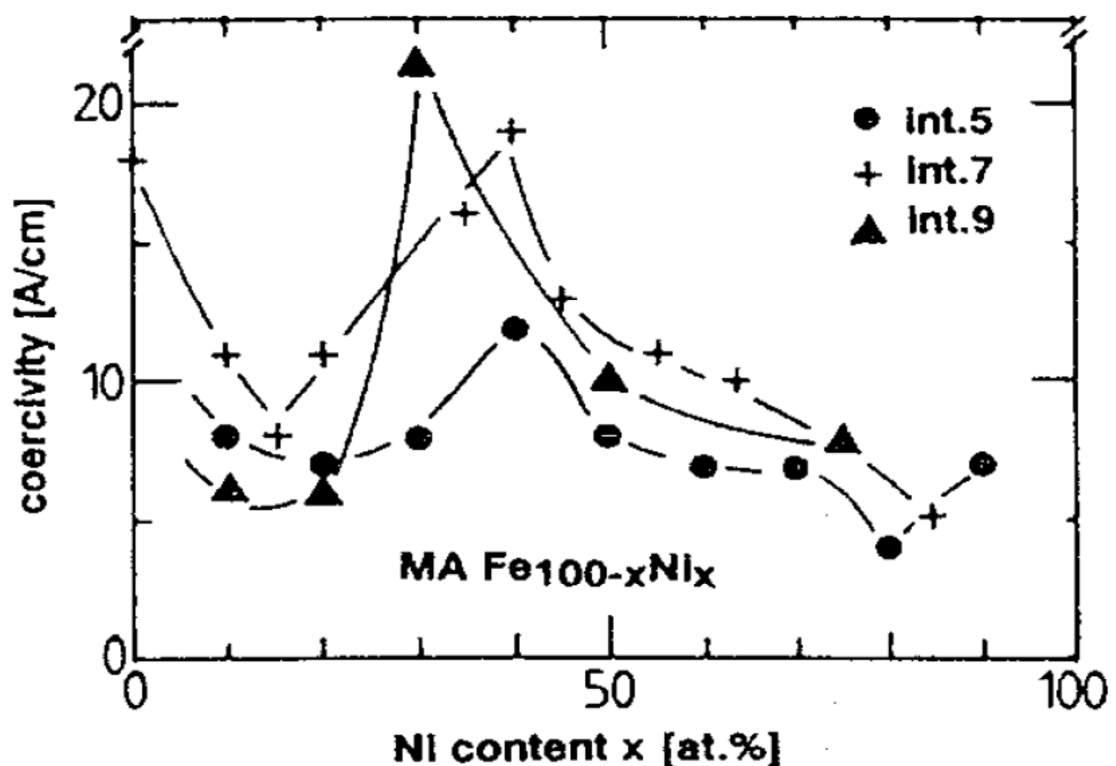


Figure 1.16 : Evolution du champ coercitif en fonction de la teneur en nickel pour différentes intensités de broyage [54]

1.5 Application des AG-ANN dans la mécanosynthèse

L'algorithme génétique est un outil d'optimisation très puissant qui permet de trouver de bonnes solutions pour les problèmes les plus complexes de plusieurs paramètres. De plus, il s'agit d'une technique couramment utilisée qui utilise les résultats des réseaux de neurones comme entrée de l'algorithme génétique pour déterminer les conditions optimales lors de la mécanosynthèse.

Torabi et al [55] ont utilisé différents pourcentages en poids de Cr et différents temps de broyage pour synthétiser un composé nanostructuré Cu-Cr préparé par broyage mécanique à haute énergie. Tout d'abord, un réseau neuronal artificiel à rétropropagation a été développé pour prédire la micro-dureté. Ensuite, un algorithme génétique a été développé pour spécifier les paramètres qui permettent d'obtenir la dureté maximale des alliages Cu-Cr.

Zeraati et al [56] ont utilisé des réseaux de neurones combinés avec des algorithmes génétiques multi-objectifs pour optimiser et améliorer les propriétés magnétiques et structurelles d'un alliage ternaire nanostructuré FeCoNi préparé par mécanosynthèse. Ils ont utilisé le pourcentage pondéral de Fe, Co et Ni en tant qu'éléments d'alliage, le temps et la vitesse de broyage, le temps et la température de recuit, ainsi que le rapport bille/poudre (RBP) comme paramètres d'entrée pour prédire les paramètres de sortie, à savoir la taille des grains, la saturation de l'aimantation et la coercivité.

Mahdavi et al [57] ont utilisé un réseau neuronal artificiel (ANN) et un algorithme génétique (GA) pour trouver les conditions optimales (température de frittage, temps de frittage, pression de compactage, temps de broyage et vitesse de rotation des jarres) permettant d'atteindre la dureté maximale de l'Al6061 renforcée par des nanotubes de carbone multiparois (MWCNT). Ils ont examiné différentes architectures neuronales pour déterminer la structure optimale du modèle, c'est-à-dire la structure neuronale qui donne une bonne corrélation. La structure trouvée a été utilisée comme fonction d'ajustement pour l'algorithme génétique. Après comparaison avec les données expérimentales, ils ont conclu que le modèle GA-ANN est une méthode puissante pour trouver les conditions optimales de préparation de l'Al6061-MWCNT.

Yazdanmehr et al [58] ont développé un modèle théorique basé sur un réseau neuronal artificiel (ANN) et un algorithme génétique (GA) pour optimiser les propriétés magnétiques des poudres nanocristallines Fe-Si préparées par mécanosynthèse. Le modèle ANN a été utilisé pour corrélérer le temps de broyage, la composition chimique, la vitesse de broyage et le rapport bille/poudre (BPR) à la coercivité et à la taille des cristallites des poudres. En combinant les deux méthodes (GA-ANN), ils ont pu trouver les conditions optimales pour obtenir une

coercivité minimale et une taille de cristallite minimale. Après comparaison avec les données expérimentales, ils ont conclu que l'algorithme GA-ANN est une méthode utile, efficace et solide pour trouver les conditions de broyage et la composition chimique optimale. Cette méthodologie permet de réduire le nombre d'essais expérimentaux nécessaires pour optimiser le comportement magnétique des poudres Fe-Si nanocristallines.

CHAPITRE 2

OPTIMISATION PAR ALGORITHMES GENETIQUES

2.1 Introduction

Les ingénieurs et concepteurs sont quotidiennement confrontés à des problèmes technologiques de plus en plus complexes, qui surviennent dans des secteurs très divers. Le problème à résoudre peut souvent être considéré comme un problème d'optimisation dans lequel une ou plusieurs fonctions objectives, ou fonctions de coût, sont définies et cherchées à être minimisées (ou maximisées) par rapport à l'ensemble des paramètres concernés [59]. La résolution de ce type de problème a conduit au développement de méthodes de résolution de plus en plus performantes, parmi lesquelles on compte les algorithmes génétiques.

Ce chapitre est organisé en trois parties. La première partie est consacrée aux algorithmes génétiques en tant que méthode d'optimisation mono-objectif. Après une présentation générale du principe de fonctionnement d'un algorithme génétique, nous décrivons en détail ses différentes composantes. La deuxième partie est consacrée à la programmation génétique, une méthode introduite par [60], dont le modèle s'inspire des algorithmes génétiques et vise à créer automatiquement un programme pour résoudre un problème. Elle suit le même schéma qu'un algorithme génétique [61].

Enfin, la dernière partie traite de l'optimisation multi-objectifs par le biais des algorithmes génétiques. Nous commencerons d'abord par un bref rappel des spécificités d'un problème à plusieurs objectifs, puis nous évoquerons le concept de dominance et la notion d'optimalité au sens de Pareto, qui caractérisent cette classe de problèmes. Enfin, nous détaillerons l'algorithme élitiste (NSGA-II) qui sera utilisé dans la suite de ce travail.

2.2 Problème d'optimisation

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum d'une fonction donnée. On peut également trouver des problèmes d'optimisation pour lesquels les variables de la fonction à optimiser sont contraintes à évoluer dans une certaine partie de l'espace de recherche. Dans ce cas, il s'agit d'un problème d'optimisation sous contraintes [62].

Un problème d'optimisation mono-objectif se définit par :

- Une fonction objective (de coûts) à optimiser (minimiser ou maximiser),
- Des variables de décision et leurs espaces de recherche,
- Un ensemble de contraintes à respecter.

D'une manière générale, un problème d'optimisation mono-objectif peut être écrit sous la forme générale suivante :

$$\begin{cases} \text{Min } f(x_k) \\ g_i(x_k) \leq 0 & i = 1, \dots, p \\ h_j(x_k) = 0 & j = 1, \dots, q \\ x_{kmin} \leq x_k \leq x_{kmax} \end{cases} \quad (1.1)$$

Où :

$f(x_k)$ est le critère à minimiser, également appelé fonction objectif

x_k est un vecteur à n variables. Ces variables sont les paramètres du problème à optimiser.

$g_i(x_k)$ et $h_j(x_k)$ représentent respectivement les contraintes d'inégalité et d'égalité.

x_{kmin} et x_{kmax} désignent les contraintes du domaine qui délimitent l'espace de recherche.

La solution d'un problème d'optimisation est alors donnée par un ensemble de paramètres pour lesquels la fonction objective présente une valeur minimale, en respectant les contraintes d'égalité et d'inégalité.

2.3 Algorithmes génétiques mono-objectif

Les algorithmes génétiques (AG) développés par John Holland [63] présentent des qualités intéressantes pour la résolution de problèmes d'optimisation complexes.

Leurs fondements théoriques ont été exposés par Goldberg [64]. Ils tentent de simuler le processus d'évolution des espèces dans leur milieu naturel : une transposition artificielle des concepts basiques de la génétique et des lois de survie énoncées par Darwin. L'algorithme génétique représente un individu par un code, c'est-à-dire un ensemble de données (appelées chromosomes) qui identifient complètement cet individu.

La reproduction est, dans ce domaine, un tirage aléatoire de chromosomes de deux individus, donnant naissance à des individus enfants ayant une empreinte génétique nouvelle, héritée des parents. La mutation génétique est caractérisée dans le code génétique de l'enfant par l'apparition d'un nouveau chromosome, inexistant chez les individus parents [65].

Ce phénomène génétique d'apparition de "mutants" est rare, mais il permet d'expliquer les changements dans la morphologie des espèces, toujours dans le sens d'une meilleure adaptation au milieu naturel. La disparition de certaines espèces est expliquée par "les lois de survie" selon lesquelles seuls les individus les mieux adaptés auront une longévité suffisante pour générer une descendance. Les individus peu adaptés tendent à disparaître. C'est une sélection naturelle qui conduit, de génération en génération, à une population composée d'individus de plus en plus adaptés.

Un algorithme génétique est construit de manière tout à fait analogue. Dans l'ensemble des solutions d'un problème d'optimisation, une population de taille N est constituée de N solutions (les individus de la population) qui sont convenablement marquées par un codage qui les identifie complètement. Une procédure d'évaluation est nécessaire pour déterminer la force de chaque individu de la population. Ensuite, une phase de sélection (qui sélectionne les individus en fonction de leur force) et une phase de recombinaison (utilisant des opérateurs

artificiels de croisement et de mutations) génèrent une nouvelle population d'individus, qui ont de bonnes chances d'être plus forts que ceux de la génération précédente. De génération en génération, la force des individus de la population augmente et après un certain nombre d'itérations, la population est entièrement composée d'individus forts, c'est-à-dire de solutions quasi-optimales du problème posé.

Le fonctionnement d'un GA est basé sur les phases suivantes [66]:

1. Initialisation: une population initiale de taille N chromosomes est tirée aléatoirement.
2. Evaluation: chaque chromosome est décodé puis évalué.
3. Reproduction: création d'une nouvelle population de N chromosomes en utilisant une méthode de sélection appropriée.
4. Opérateurs génétiques: croisement et mutations de certains chromosomes au sein de la nouvelle population.
5. Retour à la phase 2 tant que la condition d'arrêt du problème n'est pas satisfaite.

2.3.1 Codage et population initiale

Tout d'abord, il est nécessaire de représenter les différents états possibles de la variable dont on cherche la valeur optimale sous une forme utilisable pour un AG : c'est le codage. Cela établit une connexion entre la valeur de la variable et les individus de la population, imitant ainsi la transcription génotype-phénotype qui existe dans le monde vivant. Il existe principalement trois types de codage : le codage binaire, le codage réel et le codage en base n .

2.3.1.1 Codage binaire

Ce codage a été le premier à être utilisé dans le domaine des AG. Il présente plusieurs avantages : un alphabet minimum $\{0,1\}$, une facilité de mise en place des opérateurs génétiques et des fondements théoriques existants. Néanmoins, ce type de codage présente quelques inconvénients:

1. Les performances de l'algorithme sont réduites pour les problèmes d'optimisation de grande dimension nécessitant une haute précision numérique. Pour de tels problèmes, les GA basés sur les chaînes binaires ont des performances limitées, comme le montre Michalewicz [67].
2. La distance de Hamming entre deux nombres voisins (nombre de bits différents) peut être assez grande dans le codage binaire : l'entier 7 correspond à la chaîne 0111 et la chaîne 1000 correspond à l'entier 8. Or, la distance de Hamming entre ces deux chaînes est de 4, ce qui entraîne souvent une convergence plutôt que l'obtention de la valeur optimale.

2.3.1.2 Codage réel

Il a le mérite d'être simple. Chaque chromosome est en fait un vecteur dont les composantes sont les paramètres du processus d'optimisation. Par exemple, si on recherche l'optimum d'une fonction de n variables $f(x_1, x_2, \dots, x_{n-1}, x_n)$, on peut tout simplement utiliser un chromosome ch contenant les n variables :

$$ch: \begin{array}{|c|c|c|c|c|} \hline x_1 & x_2 & \dots & x_{n-1} & x_n \\ \hline \end{array}$$

Avec ce type de codage, la procédure d'évaluation des chromosomes est plus rapide grâce à l'absence de l'étape de transcodage (du binaire vers le réel). Les résultats donnés par Michalewicz [59] montrent que la représentation réelle conduit souvent à une meilleure précision et un gain important en termes de temps d'exécution.

2.3.1.3 Codage en base n

Dans ce type de codage, les gènes constituant un chromosome sont des chiffres exprimés dans une base de numération n , ce qui permet de représenter n valeurs discrètes.

L'algorithme génétique démarre avec une population composée de N individus dans le codage retenu. Le choix des individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace de recherche est totalement inconnue, il est intéressant que la population soit répartie sur tout l'espace de recherche. Si, par contre, des informations a priori sur le problème

sont disponibles, il paraît évident de générer les individus dans un espace particulier afin d'accélérer la convergence. Disposant d'une population initiale souvent non homogène, la diversité de la population doit être entretenue au cours des générations afin d'explorer le plus largement possible l'espace de recherche. C'est le rôle des opérateurs de croisement et de mutation.

2.3.2 Opérateur de croisement

Le croisement est le principal opérateur agissant sur la population des parents. Il permet de créer de nouveaux individus par l'échange d'informations entre les chromosomes par le biais de leur combinaison. La population courante est divisée en deux sous populations de même taille ($N/2$) et chaque couple formé par un membre provenant de chaque sous-population participe à un croisement avec une probabilité (pc) souvent supérieure à 0,5. Si le croisement a eu lieu entre deux chromosomes parents (ch_1 et ch_2), constitués de l gènes, on tire aléatoirement une position de chacun des parents. Ensuite on échange les deux sous chaînes terminales de chacun des chromosomes, ce qui produit deux enfants ch'_1 et ch'_2 comme indiqué sur la figure 2.1

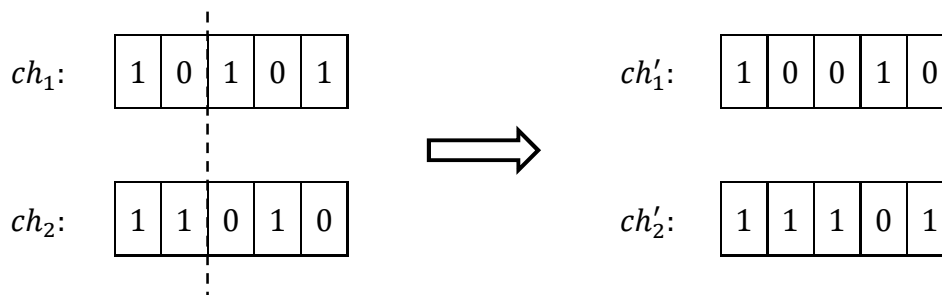


Figure 2.1 : Croisement en un seul point

Dans notre exemple, figure 2.1, un croisement localisé à la quatrième position a eu lieu entre les chromosomes ch_1 et ch_2 : il s'agit d'un croisement en un seul point. Ainsi, on peut étendre ce principe de combinaison en choisissant non pas un seul point, mais 2, 3, etc...(croisement en multipoints) [68].

Ce type de croisement est très efficace et peut s'étendre à n'importe quel type de chaînes (réelles, en base n , etc...). Néanmoins certains auteurs [69]

préfèrent utiliser dans le cas des chaînes réelles, un croisement de type barycentre (appelé croisement arithmétique): deux gènes $ch_1(i)$ et $ch_2(i)$ sont sélectionnés dans chacun des parents à la même position i . Ils définissent deux nouveaux $ch'_1(i)$ et $ch'_2(i)$ par combinaison linéaire :

$$\begin{cases} ch'_1(i) = \alpha \times ch_1(i) + (1 - \alpha) \times ch_2(i) \\ ch'_2(i) = (1 - \alpha) \times ch_1(i) + \alpha \times ch_2(i) \end{cases} \quad (2.2)$$

Où α est un paramètre de pondération aléatoire qui prend généralement ses valeurs dans l'intervalle $[0, 1]$.

Quoi qu'il en soit, il se peut que l'effet de l'opérateur de croisement soit insuffisant pour assurer une meilleure exploration de l'espace de recherche. Ainsi dans le cas du codage binaire, certaines chaînes peuvent totalement disparaître de la population. Par exemple, si aucun chromosome de la population initiale ne contient de 1 en première position et que ce 1 fait partie de la chaîne optimale à trouver, aucun croisement ne peut faire apparaître cet élément. Ce dernier ne peut s'introduire dans la population que si l'on introduit un autre opérateur et c'est, entre autres, pour remédier à ce problème que l'opérateur de mutation est utilisé.

2.3.3 Opérateur de mutation

Le rôle de cet opérateur est de modifier aléatoirement la valeur d'un gène dans un chromosome. Dans le cas du codage binaire, chaque bit $a_i \in \{0,1\}$ est remplacé par son complémentaire $\bar{a}_i = 1 - a_i$.

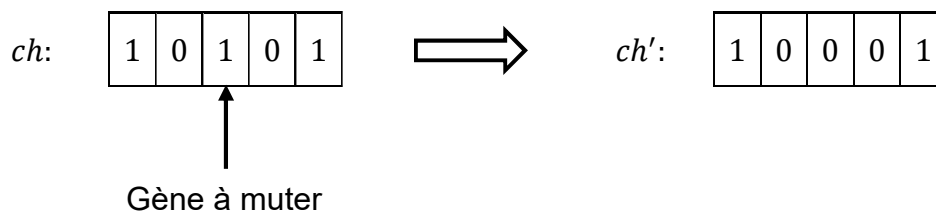


Figure 2.2 : Principe de l'opérateur de mutation

Dans l'exemple de la figure 2.2, une mutation a eu lieu sur le troisième gène du chromosome ch et elle l'a transformé de 1 en 0.

Pour les chaînes codées en base n , la mutation consiste à remplacer le gène initial par un chiffre en base n tiré au sort.

Dans le cas d'un codage réel, on utilise principalement deux opérateurs de mutation: la mutation uniforme et la mutation non uniforme [70]. En supposant une probabilité de mutation fixée p_m , un tirage au sort pour chaque gène x_k d'un chromosome ch permet de décider si ce gène doit être modifié ou non. Nous supposons que le gène prend ses valeurs dans un intervalle $[x_k^{min}, x_k^{max}]$. Pour la mutation uniforme, qui est une simple extension de la mutation binaire, on remplace le gène x_k sélectionné par une valeur quelconque x'_k tirée aléatoirement dans l'intervalle $[x_k^{min}, x_k^{max}]$.

2.3.4 Fonction d'évaluation

Un algorithme génétique nécessite généralement la définition d'une fonction rendant compte de la pertinence des solutions potentielles, à partir des grandeurs à optimiser. Cette fonction sera utilisée par les processus de sélection pour favoriser les individus les mieux adaptés, c'est à dire dit les meilleures solutions au problème. L'évaluation représente donc la performance de l'individu par rapport au problème à résoudre. De ce fait, cette fonction est en lien avec la fonction objectif de ce dernier. Aucune règle n'existe pour définir cette fonction d'évaluation. Tout comme le choix du mode de représentation des solutions, le choix du mode d'évaluation a un impact important sur la manière dont l'algorithme va parcourir l'espace de recherche, et donc sur les solutions qu'il renverra finalement.

2.3.5 Opérateur de sélection

La sélection crée une population intermédiaire constituée de copies des individus de la population courante. En règle générale, le nombre de copies d'un individu est directement lié à la fitness relative de l'individu au sein de la population. Il existe plusieurs méthodes heuristiques qui représentent la reproduction, la méthode la plus connue et la plus utilisée étant la sélection par roulette biaisée (roulette wheel selection) de Goldberg [71]. Selon cette méthode, chaque chromosome est copié dans la nouvelle population proportionnellement à sa fitness. On effectue en quelque sorte, autant de tirages avec remise qu'il y a

d'éléments existant dans la population. Ainsi, pour un chromosome particulier ch_i de fitness $f(ch_i)$, la probabilité de sa sélection dans la nouvelle population de taille N est :

$$p(ch_i) = \frac{f(ch_i)}{\sum_{j=1}^N f(ch_j)} \quad (2.3)$$

Plus un individu est performant par rapport aux autres, plus il a de chances d'être reproduit dans la population. Les individus ayant une grande fitness relative ont donc plus de chances d'être sélectionnés. On parle alors de sélection proportionnelle.

Le nombre de copies attendu pour chaque individu résultant de la sélection est alors égal à :

$$n_i = N \times p(ch_i) = \frac{f(ch_i)}{\frac{1}{N} \sum_{j=1}^N f(ch_j)} = \frac{f(ch_i)}{\bar{f}} \quad (2.4)$$

L'inconvénient majeur de ce type de reproduction réside dans le fait qu'il peut favoriser la domination d'un individu qui n'est pas nécessairement le meilleur. Cette méthode peut également entraîner une perte de diversité en favorisant la domination d'un super-individu. Pour remédier à cet inconvénient, on préfère souvent des méthodes qui n'autorisent en aucun cas l'apparition de super-individus. Par exemple, la sélection par tournoi (tournament selection) ou d'autres méthodes qui impliquent un changement d'échelle (Scaling) et/ou des notions de voisinage entre chromosomes (Sharing).

2.3.5.1 Sélection par tournoi

On tire aléatoirement deux individus dans la population et on reproduit le meilleur des deux dans la nouvelle population. On répète cette procédure jusqu'à ce que la nouvelle population soit complète. Cette méthode peut donner de bons résultats, mais elle a l'inconvénient d'éliminer de bons individus.

2.3.5.2 Sharing

Le principe du sharing consiste à ajuster la fitness des individus pour éviter qu'ils se concentrent dans une niche principale (optimum globale). La technique

de partage de la fitness (fitness sharing), introduite par Goldberg et Richardson [72], réduit la fitness de chaque individu d'un facteur correspondant environ au taux d'agrégation de la population autour de son voisinage :

$$f'(ch_i) = \frac{f(ch_i)}{m_i} \quad (2.5)$$

Où le compteur de niche m_i se calcule de la manière suivante:

$$m_i = \sum_{j=1}^N sh(d_{ij}) \quad (2.6)$$

Où N désigne la taille de la population et sh mesure la similarité entre deux individus i et j en fonction de la distance d_{ij} et le rayon de niche σ_{shar} :

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{shar}}\right)^\alpha & \text{si } d_{ij} < \sigma_{shar} \\ 0 & \text{sinon} \end{cases} \quad (2.7)$$

La figure 2.3 montre deux exemples de répartition de populations dans le cas d'une fonction multimodale: le premier sans sharing et le deuxième avec sharing.

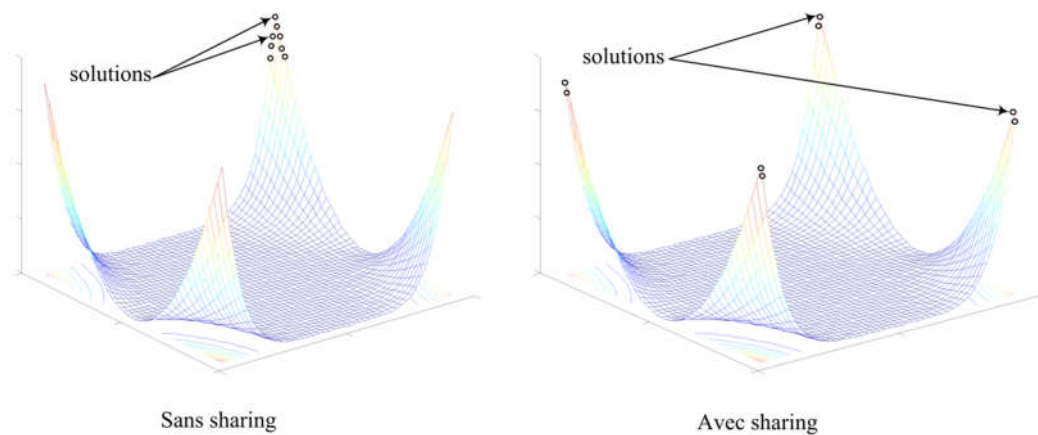


Figure 2.3 : Effet de sharing sur la répartition des solutions

Si le principe de sharing a été utilisé initialement dans l'espace des paramètres (le critère de distance étant fonction des paramètres), il est tout à fait possible de l'adapter dans l'espace des objectifs. C'est d'ailleurs cette idée qui est exploitée dans les algorithmes génétiques multi-objectifs.

2.3.5.3 Elitisme

La stratégie élitiste consiste à conserver le meilleur individu à chaque génération. Ainsi l'élitisme empêche l'individu le plus performant de disparaître au cours de la sélection ou que ses bonnes combinaisons soient affectées par les opérateurs de croisement et de mutation. Après chaque évaluation de la performance des individus à une génération t donnée, le meilleur individu de la génération précédente ($t - 1$) est réintroduit dans la population si aucun des individus de la génération t n'est meilleur que lui. Par cette approche, la performance du meilleur individu de la population courante est monotone de génération en génération.

2.4 Programmation Génétique Multi-Gène

La programmation génétique (PG) fait partie des algorithmes évolutionnaires, elle est basée sur le même principe d'évolution que les algorithmes génétiques. La PG possède plusieurs variantes, dont sa version robuste la programmation génétique multi-gène MGGP. Cette dernière est conçue pour développer un modèle mathématique empirique, qui est une combinaison linéaire pondérée d'un certain nombre d'arbres de GP et est appelée "gènes". "Multi-gène" fait référence à la combinaison linéaire de ces gènes [73].

La MGGP est capable de proposer des équations mathématiques concrètes que les réseaux de neurones ou d'autres techniques ne peuvent pas le faire [74].

Généralement, les structures évoluées par la programmation génétique sont représentées par l'une des trois formes suivantes : sous forme linéaire, sous forme d'arbre ou sous forme de graphe. Dans notre application, c'est la structure sous forme d'arbre, appelée "arbre syntaxique" et présentée sur la figure 2.4, qui sera utilisée.

La figure 2.4 montre un exemple de modèle MGGP dans lequel le résultat est représenté par une combinaison linéaire de deux gènes développés à l'aide de trois variables d'entrée (x_1, x_2, x_3). Chaque gène est un modèle non linéaire car il contient des termes non linéaires ($\cos(\cdot), \log(\cdot)$). Les coefficients linéaires (poids)

de Gene1 et Gene2 (d_1 et d_2) et le biais (d_0) du modèle sont obtenus à partir des données d'apprentissage à l'aide d'une analyse de régression (méthode des moindres carrés ordinaires).

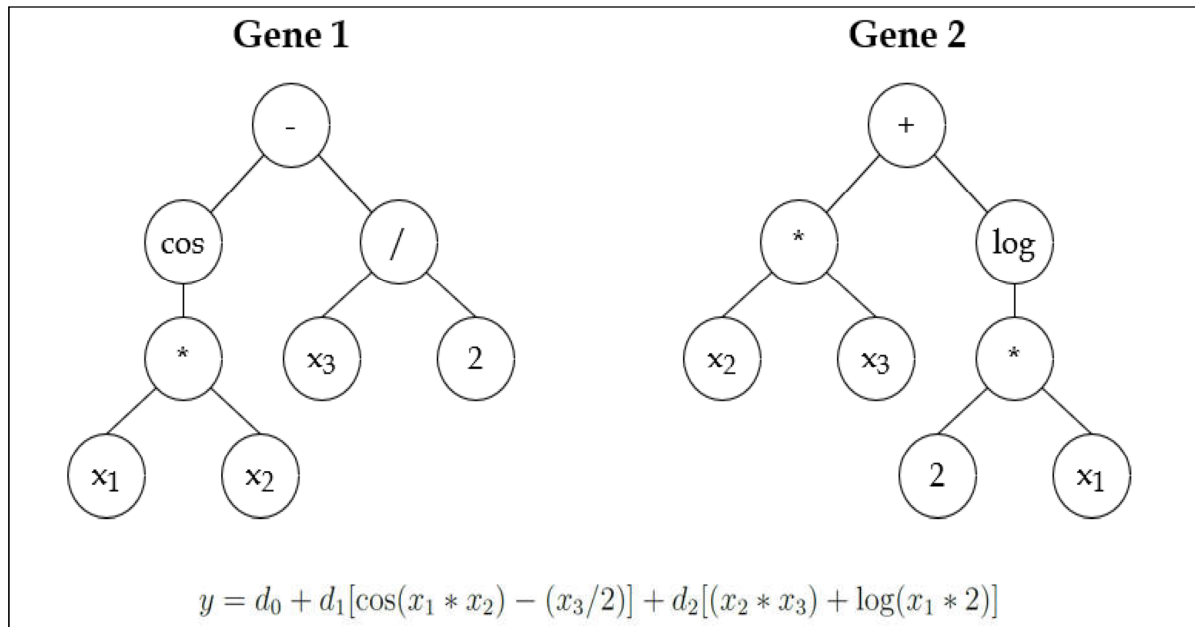


Figure 2.4 : Exemple typique d'un modèle MGGP représentée sous forme d'arbre

Pour les opérateurs génétiques, le croisement consiste à sélectionner deux arbres GP (Parent1 et Parent2) au hasard dans la population du pool d'accouplement. Un nœud de chaque arbre est sélectionné de manière aléatoire, les sous-arbres situés sous les nœuds sélectionnés sont échangés et deux descendants (Enfant1 et Enfant2) sont générés. Un exemple du fonctionnement du croisement est illustré à la figure 2.5.

Par contre, la mutation se concentre sur un parent unique et la modifie pour former le nouvel enfant. La mutation signifie l'évolution d'une structure complètement nouvelle déterminée au hasard au nœud sélectionné (c'est-à-dire que le sous-arbre du nœud choisi est supprimé et qu'un nouveau est créé au hasard, en "poussant" vers le point de mutation choisi).

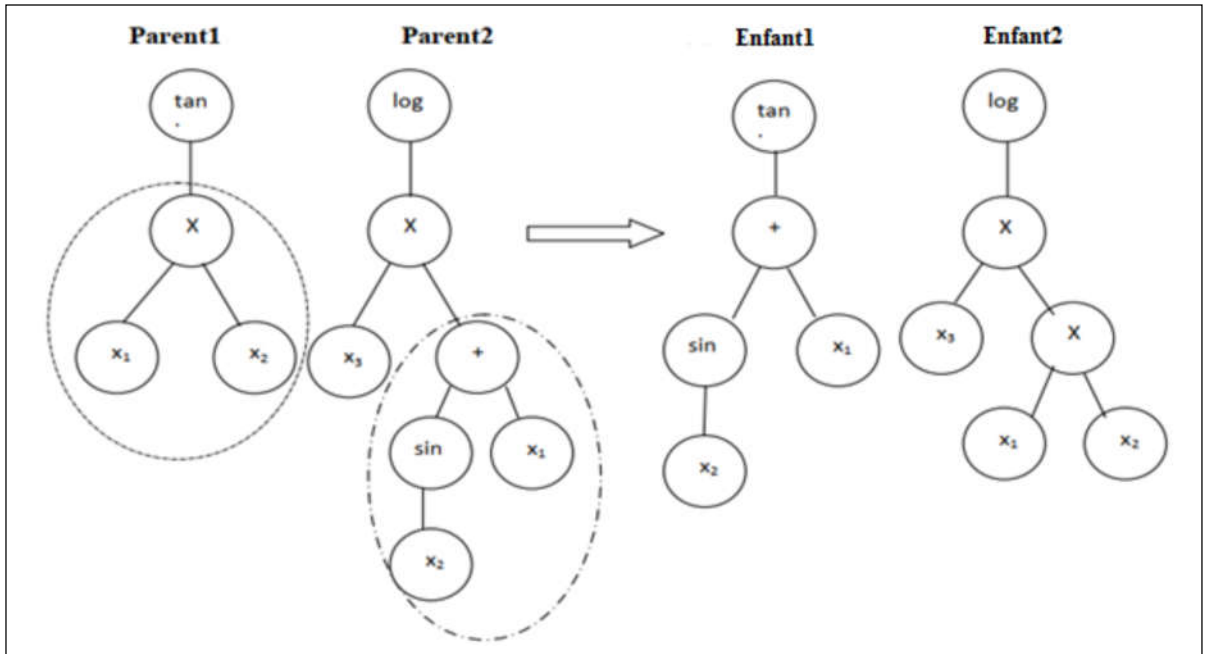


Figure 2.5 : Opération de croisement en MGGP

Ce changement aléatoire ou l'apparition de gènes, comme le montre la figure 2.6, permet à certains programmes de la nouvelle génération de s'éloigner du chemin évolutif de la population générale afin que différents programmes puissent être examinés.

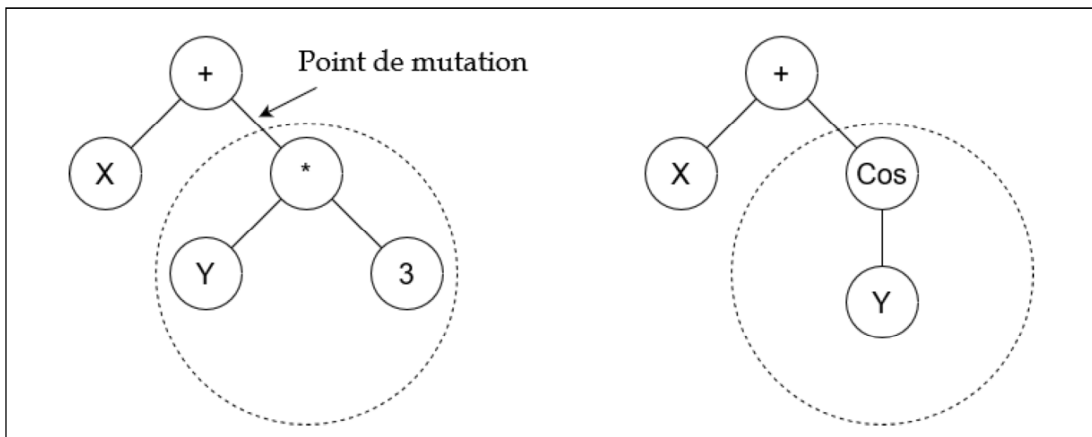


Figure 2.6 : Opération de mutation en MGGP

L'utilisateur peut décider d'un compromis entre la précision et la complexité du modèle de prédiction en contrôlant le nombre maximum de gènes autorisés dans chaque individu et la profondeur des arbres dans MGGP, ce qui permet de faire un compromis entre la simplicité et la précision de la méthode.

2.5 Optimisation multi-objectifs

Dans la section précédente, nous avons seulement considéré le cas de problèmes monoobjectif. Cependant, la plupart des problèmes réels sont en fait des problèmes multi-objectifs, c'est-à-dire que l'on cherche à optimiser simultanément plusieurs critères qui sont généralement conflictuels (par exemple, on peut avoir besoin de maximiser la qualité d'un produit tout en minimisant son prix de revient). L'optimisation multi-objectif (aussi appelée multicritère) consiste donc à optimiser simultanément plusieurs fonctions.

On peut formaliser un problème d'optimisation multi-objectif comme suit :

$$\begin{cases} \text{Min } (F(x) = (f_1(x), f_2(x), \dots, f_n(x))) \\ x \in C \end{cases} \quad (2.8)$$

Où n représente le nombre de fonctions objectifs et $x = (x_1, x_2, \dots, x_m)$ est le vecteur représentant les variables de décision. C représente l'ensemble des solutions réalisables associées à des contraintes d'égalité et d'inégalité et $F(x)$ est le vecteur d'objectifs à optimiser (ou critères de décision) avec $n \geq 2$ comme le nombre de fonctions objectifs. L'espace des objectifs (ou espace des critères) est l'image de l'espace de décision par la fonction objectif F (ou ensemble des points réalisables $Y = F(C)$).

On remarque que ce type de problème n'a pas une seule solution optimale, mais potentiellement plusieurs, en fonction des objectifs que l'on considère plus ou moins importants. Il est donc difficile d'établir une définition précise de ce que doit être l'optimum. C'est pourquoi on fait intervenir un décideur, qui choisira parmi les solutions possibles celle qui lui semble la meilleure.

Dans la littérature, on trouve deux classifications différentes des méthodes de résolution des problèmes d'optimisation multiobjectif.

Le premier classement adopte un point de vue centré sur le décideur et divise ces méthodes en trois familles selon la coopération entre la méthode d'optimisation et le décideur :

- Les méthodes a priori : Le décideur intervient avant le processus d'optimisation et définit ses préférences en pondérant les objectifs en fonction de leur importance.
- Les méthodes progressives ou interactives : Le décideur affine son choix au fur et à mesure du processus en écartant ou en favorisant des solutions.
- Les méthodes a posteriori : Le décideur intervient après l'optimisation et choisit la solution qui lui semble la plus intéressante parmi l'ensemble des solutions fournies par la méthode.

Le deuxième classement se base sur la façon dont les méthodes traitent les fonctions objectifs. On distingue les méthodes agrégées, les méthodes basées sur l'équilibre de Pareto, ainsi que les méthodes non agrégées et non Pareto, qui sont détaillées dans la suite.

2.5.1 Méthodes agrégées

Les méthodes agrégées transforment un problème multiobjectif en un problème monoobjectif, en regroupant les critères à optimiser dans une seule fonction objectif. Les méthodes d'agrégation les plus simples utilisent une fonction de mise à l'échelle pour chaque critère, afin de pouvoir les additionner (modèle additif) ou les multiplier (modèle multiplicatif)

La méthode d'agrégation la plus connue et la plus utilisée est la moyenne pondérée. Cette méthode consiste à additionner tous les objectifs en leur attribuant un coefficient de poids. Ce coefficient traduit l'importance relative que le décideur accorde à chaque objectif :

$$F(x) = \sum_{i=1}^m \omega_i f_i(x) \quad (2.9)$$

Où les poids $\omega_i \geq 0$ sont tels que $\sum_{i=1}^m \omega_i = 1$

Bien que largement utilisée, cette méthode présente toutefois quelques problèmes. D'un côté, le décideur doit déterminer à l'avance les poids de chaque critère, ce qui repose sur sa connaissance du problème. D'un autre côté, il doit être capable d'exprimer les interactions entre les différents critères.

2.5.2 Méthodes basées sur l'équilibre de Pareto

Ces méthodes reposent sur le postulat de V. Pareto [75] : "Il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins l'un des autres critères".

La notion de solution optimale unique dans l'optimisation mono-objectif disparaît pour les problèmes d'optimisation multi-objectifs, au profit de la notion d'ensemble de solutions Pareto optimales, selon le critère de dominance au sens de Pareto. On appelle front de Pareto (ou surface de compromis) du problème l'ensemble des points de l'espace de recherche tels qu'il n'existe aucun point qui soit strictement meilleur que les autres sur tous les critères simultanément. Il s'agit de l'ensemble des meilleurs compromis réalisables entre les objectifs contradictoires. L'objectif de l'optimisation est d'identifier cet ensemble de compromis optimaux entre les critères.

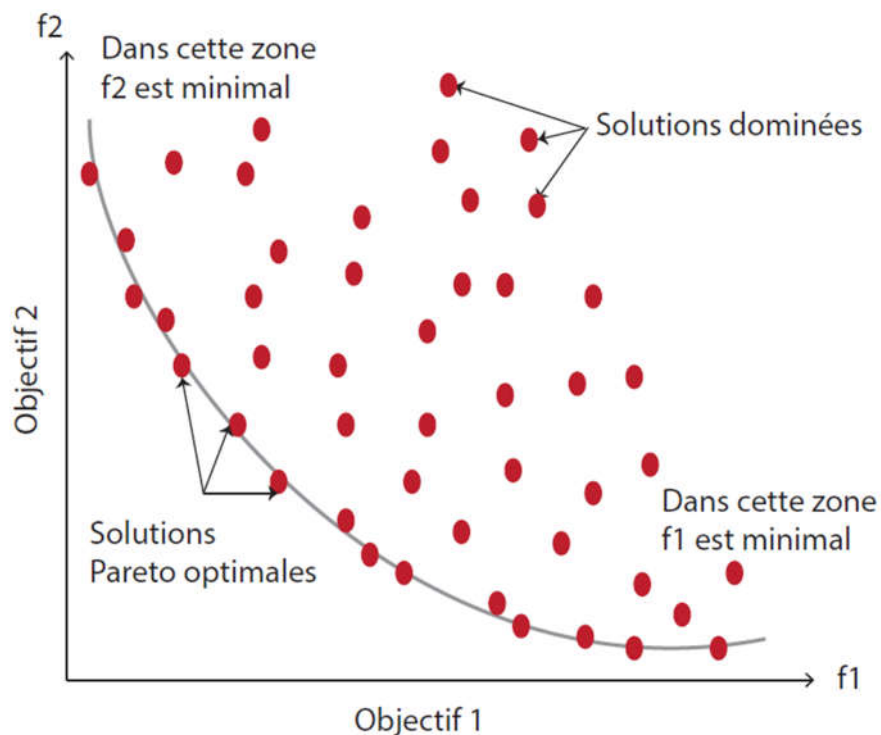


Figure 2.7: Front et rangs de Pareto pour un problème de minimisation de deux objectifs.

La figure 2.7 présente un exemple de front de Pareto. Cependant, il est important de noter que le front de Pareto peut prendre différentes formes, telles

que concave, discontinu, etc. Les solutions situées sur le front de Pareto ne peuvent pas être comparées, car aucune n'est systématiquement meilleure que les autres sur tous les objectifs. C'est donc au décideur de choisir la solution à retenir.

De nombreux algorithmes génétiques ont été développés pour résoudre ce type de problème. Parmi les plus significatifs figurent : l'algorithme Multi Objective Genetic Algorithm (MOGA) [76], l'algorithme Niched Pareto Genetic Algorithm (NPGA) [77], qui utilise une sélection par tournoi basée principalement sur la dominance de Pareto, l'algorithme Non Dominated Sorting Genetic Algorithm (NSGA) [78], et enfin l'algorithme NSGA-II [79], qui est sans doute l'algorithme le plus populaire.

2.5.3 Méthodes non agrégées et non Pareto

En général, les méthodes non agrégées et non Pareto traitent séparément les objectifs dans leur processus de recherche. Elles ne se basent donc sur aucun des deux principes énoncés précédemment. Un exemple de ce type de méthode est l'algorithme génétique à évaluation vectorielle (VEGA : Vector Evaluated Genetic Algorithm), présenté par Schaffer [80]. La seule différence avec un algorithme génétique classique réside dans le processus de sélection. Si nous avons m objectifs et une population initiale de n individus, alors la population est divisée en m sous-populations, chacune contenant les meilleurs individus (n/m) pour un objectif particulier. Ensuite, la population est reconstruite et les opérateurs génétiques de croisement et de mutation sont appliqués.

2.5.4 Etude de l'algorithme génétique multi-objectifs NSGA-II

Les méthodes d'optimisation utilisées dans le domaine de la conception en ingénierie ont été améliorées au fil du temps en intégrant de nouveaux opérateurs pour mieux résoudre les problèmes d'optimisation multi-objectifs. L'algorithme NSGA-II, proposé par [79], est mis en œuvre dans le chapitre 4 pour l'optimisation multi-objectifs des propriétés nanocristallines. Il s'agit d'une évolution de l'algorithme NSGA [79] qui intègre un mécanisme d'élitisme et utilise une technique de nichage simplifiée basée sur un mécanisme d'estimateur de densité (crowding). Les principales caractéristiques de cet algorithme sont détaillées dans

les sections suivantes, notamment la méthode de classification des individus, la définition de la distance de crowding, la description de la méthode de sélection et la structure de l'algorithme.

1. Classification des individus

Avant la sélection, chaque individu de la population se voit attribuer un rang correspondant au front auquel il appartient. Les individus non dominés de la population appartenant au front optimal (1^{er} front) reçoivent un rang égal à 1. Ces individus sont ensuite retirés de la population et l'opération est répétée pour les fronts successifs en incrémentant le rang.

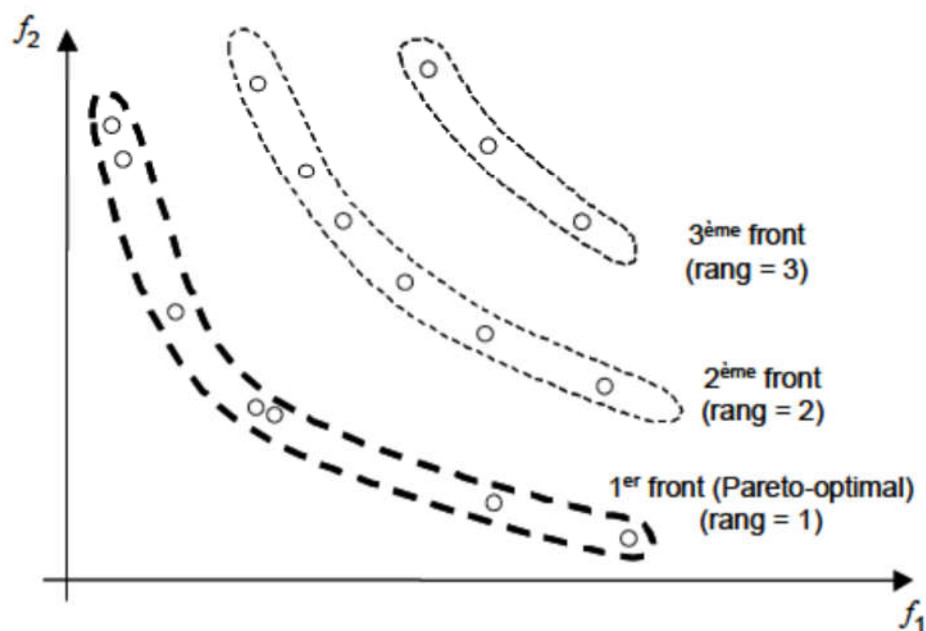


Figure 2.8 : Classification des individus selon le rang de Pareto.

2. Distance de crowding

La distance de crowding d'une solution i (ou d'un individu) est calculée en fonction du périmètre formé par les solutions du même front les plus proches de i sur chaque objectif. La figure 2.9 montre une présentation à deux dimensions associée à la solution i . Le calcul de la distance de crowding nécessite tout d'abord le tri des solutions selon chaque objectif, dans un ordre ascendant.

Ensuite, pour chaque objectif, les individus possédant les valeurs limites (la plus petite et la plus grande valeur de fonction objective) se voient attribuer à une distance infinie (∞). Pour les autres solutions intermédiaires, on calcule une distance de crowding égale à la différence normalisée des valeurs des fonctions objectives de deux solutions adjacentes. Ce calcul est réalisé pour chaque fonction objective. La distance de crowding d'une solution est calculée en sommant les distances correspondantes à chaque objectif.

L'algorithme de la figure 2.10 montre la procédure de calcul de la distance de toutes les solutions non-dominées de l'ensemble I . Dans cet algorithme, f_m^{i+1} et f_m^{i-1} représentent respectivement la valeur de la $m^{\text{ième}}$ fonction objectif de la solution $i+1$ et $i-1$, tandis que les paramètres f_m^{Max} et f_m^{Min} représentent les valeurs maximale et minimale de la $m^{\text{ième}}$ fonction objectif.

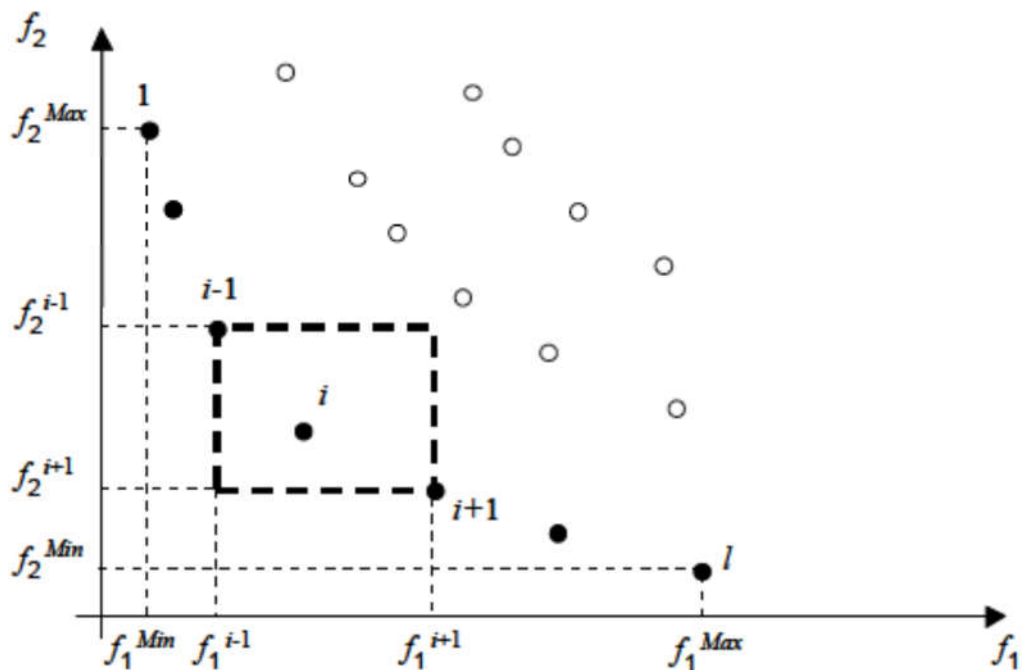


Figure 2.9 : Distance de crowding, les cercles noirs représentent les solutions appartenant au front optimal [79].

Après ce calcul, toutes les solutions seront caractérisées par un indice de nichage (la l -distance).

3. Opérateur de sélection

La sélection est basée sur un opérateur de comparaison, noté *crowded comparison* (\prec_n), qui utilise les concepts de rang de Pareto et de distance de crowding présentés précédemment. Cet opérateur guide le processus de sélection vers les solutions Pareto-optimales.

Algorithm 1 Calcul de la distance de crowding pour chaque solution d'un front

- 1: $l = |I|$, nombre de solutions dans le front I
- 2: Pour chaque solution i poser $I[i]_{distance} = 0$, Initialisation des distances.
- 3: Pour chaque objectif m :
 - $I = \text{trier}(I, m)$, trier I par ordre croissant selon le critère m
 - $I[1]_{distance} = I[l]_{distance} = +\infty$
 - Pour $i = 2$ jusqu'à $l - 1$ faire

$$I[i]_{distance} = I[i]_{distance} + (f_m^{i+1} - f_m^{i-1}) / (f_m^{Max} + f_m^{Min})$$

Figure 2.10 : Calcul de la distance de crowding pour chaque solution d'un front.

Chaque solution (i) de la population est identifiée par son rang (i_{rang}) et par sa distance de crowding ($i_{distance}$). L'opérateur défini ci-dessous permet d'identifier un ordre de préférence entre deux solutions i et j :

$$i \prec_n j \text{ si } (i_{rang} < j_{rang}) \text{ ou } ((i_{rang} = j_{rang}) \text{ et } i_{distance} > j_{distance}) \quad (2.10)$$

Lorsqu'il s'agit de deux solutions de rangs différents, on préfère la solution de rang inférieur (c'est-à-dire appartenant au meilleur front). Si les deux solutions appartiennent au même front, on privilégie celle qui est située dans la région où la densité de solutions est plus faible, c'est-à-dire celle qui possède la plus grande valeur de distance de crowding. Cet opérateur de sélection intensifie donc la recherche des solutions Pareto-optimales tout en préservant la diversité parmi les solutions équivalentes.

4. Etapes du NSGA-II

Dans cet algorithme, une population de parents (P_t) de taille (N) et une population (Q_t) de taille (N) sont assemblées pour former une population ($R_t = P_t \cup Q_t$), comme le montre la figure 2.11. Cet assemblage permet d'assurer l'élitisme. La population de taille ($2N$) est ensuite triée selon un critère de non-dominance pour identifier les différents fronts F_1, F_2 , etc. Les meilleurs individus vont se retrouver dans le ou les premiers fronts. Une nouvelle population parent (P_{t+1}) est formée en ajoutant les fronts (premier front F_1 , second front F_2 , etc.) tant que le nombre d'individus ajoutés ne dépasse pas la taille limite de l'archive N . Si le nombre d'individus présents dans (P_{t+1}) est inférieur à (N), un tri selon la distance de crowding est appliqué au premier front (F_i) non inclus dans (P_{t+1}).

Ce tri a pour objectif d'insérer les ($N - |P_{t+1}|$) meilleurs individus qui manquent dans la population (P_{t+1}). L'insertion des individus du front F_i se fait par troncation, relativement au tri effectué pour maintenir un maximum de diversité dans la population.

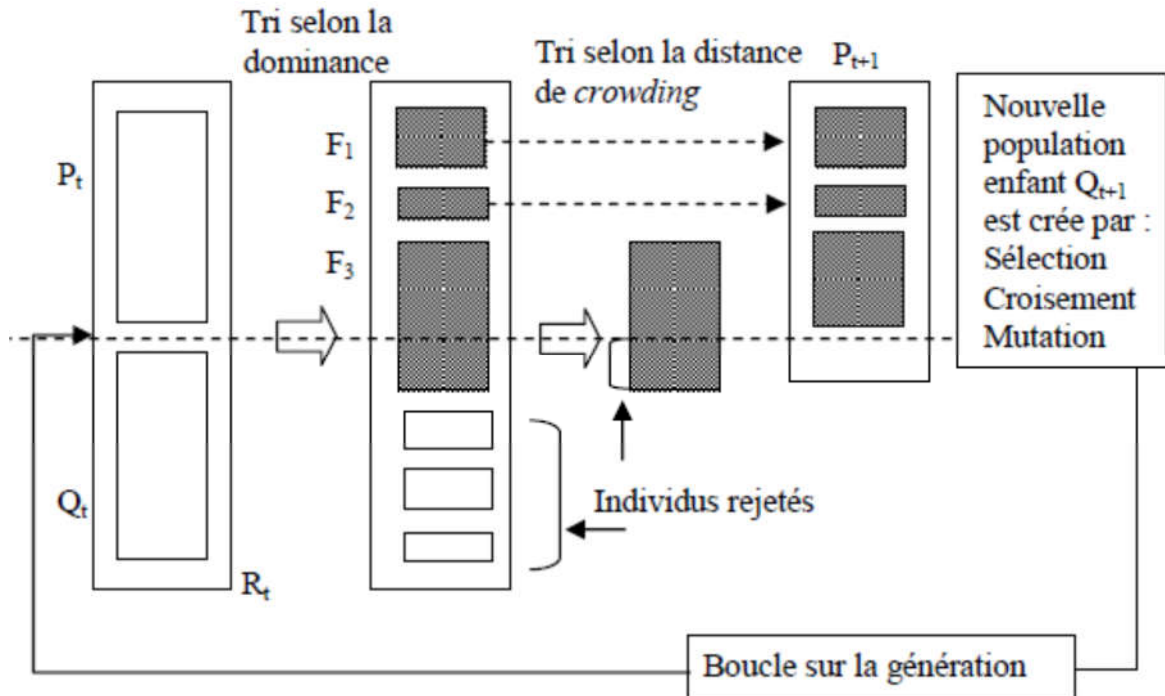


Figure 2.11 : Principe de l'algorithme NSGA-II [79].

Une fois que les individus appartenant à la population (P_{t+1}) sont identifiés, une nouvelle population enfant (Q_{t+1}) est créée par sélection, croisement et mutation. La sélection par tournoi est utilisée mais le critère de sélection est maintenant basé sur l'opération de comparaison \prec_n défini précédemment (1.6). Le processus continu d'une génération à la suivante, jusqu'à ce que un critère d'arrêt soit vérifié.

Après initialisation aléatoire de la population initiale P_0 , une itération de NSGA-II se déroule comme suit :

1. Créer Q_t à partir de P_t en utilisant le tournoi et en appliquant des opérateurs de variation génétique aux individus gagnants.
2. Réunir les populations des parents et des enfants $R_t = P_t \cup Q_t$. Trier l'ensemble résultant R_t en sous-ensemble F_i
3. Soit une nouvelle population $P_{t+1} = \phi$. Soit le compteur des sous-ensembles non-dominés.

$i = 1$.

4. Tant que $|P_{t+1}| + |F_i| \leq N, P_{t+1} = P_{t+1} \cup F_i$ et $i = i + 1$.

5. Ordonner l'ensemble F_i selon les "distance d'encombrement" (crowding distance) et Inclure $(N - |P_{t+1}|)$ solutions ayant les valeurs de distance les plus grandes dans la population P_{t+1} .

Il est important de noter que le tri de R_t en sous-ensembles non-dominés fait en stade 1 et le remplissage de la population P_{t+1} peuvent être effectués simultanément. Chaque fois qu'un nouveau front est trouvé on vérifie s'il peut rentrer dans P_{t+1} entièrement si ce n'est pas le cas, le processus du ranking s'arrête.

2.6 Conclusion

Dans ce chapitre, nous avons décrit le fonctionnement et les différents opérateurs d'un algorithme génétique mono-objectif. Nous avons également

abordé le problème de l'optimisation multi-objectifs et présenté quelques algorithmes génétiques de résolution mentionnés dans la littérature. Parmi les algorithmes examinés, on trouve NSGA-II, qui est actuellement l'une des techniques de référence garantissant la diversité des solutions Pareto optimales.

CHAPITRE 3

RESEAUX DE NEURONES

3.1. Introduction

Depuis quelques années, les réseaux de neurones connaissent un succès croissant dans divers domaines des sciences de l'ingénieur, y compris celui des matériaux nanocristallins.

Les réseaux de neurones artificiels (ANN) sont des systèmes parallèles et distribués de traitement de l'information inspirés par le fonctionnement du cerveau humain. Ce chapitre présente le principe de fonctionnement des ANN, leurs différentes algorithmes d'apprentissage, ainsi que les facteurs affectant l'application réussie des ANN de type feedforward, en particulier les perceptrons multicouches (MLP, Multi Layer Perceptron) comme approximateurs universels de fonctions.

Ils présentent deux similitudes avec le cerveau humain :

- La connaissance est acquise par le réseau à travers un processus d'apprentissage.
- Les forces de liaison entre neurones (poids synaptiques) sont utilisées pour stocker la connaissance acquise.

Dans ce chapitre, nous nous limitons à une présentation succincte des notions fondamentales des ANN et du vocabulaire nécessaire à la compréhension des parties suivantes de cette thèse.

3.2. Notions sur les réseaux de neurones artificiels

3.2.1. Neurone biologique et neurone artificiel

Les réseaux neuronaux artificiels sont inspirés du fonctionnement des neurones biologiques pour le traitement de l'information. Ce concept est utilisé pour simuler des processus massivement parallèles avec des éléments de traitement connectés dans une architecture de réseau. Dans le cerveau humain, l'apprentissage se produit à travers un réseau de neurones connectés par l'axone, les synapses et les dendrites (figure 3.1) [81].

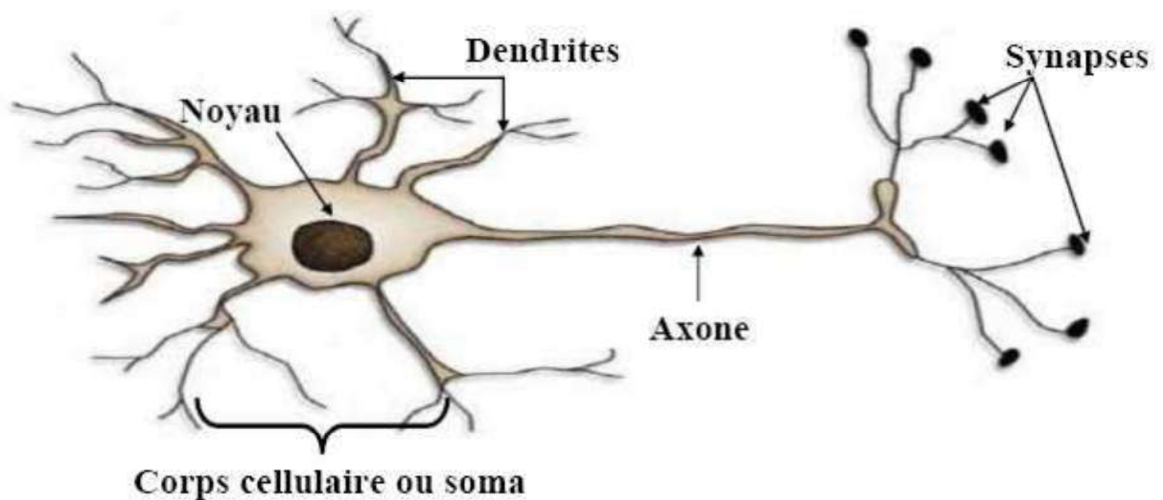


Figure 3.1 : Neurone biologique [81]

Le neurone biologique est composé d'un corps cellulaire et d'un noyau. Les dendrites se ramifient à partir du corps cellulaire. Les dendrites acheminent l'information de l'extérieur vers le soma, qui est le corps du neurone. L'information traitée par le neurone est ensuite transmise aux autres neurones le long de l'axone. La jonction entre deux neurones est appelée synapse. Les dendrites recueillent les signaux électrochimiques ou les potentiels d'action provenant d'autres cellules ou de l'extérieur via les synapses, et les acheminent vers le corps cellulaire ou le soma. En plus de son rôle dans le métabolisme de la cellule, le soma collecte et concentre les informations et en fait une sommation. Si le potentiel somatique dépasse un certain seuil, un potentiel d'action est émis, ce qui correspond à une oscillation électrique très brève. L'axone est un prolongement

cellulaire chargé de transmettre l'information nerveuse à d'autres cellules via les synapses grâce à son arborisation terminale.

Les synapses se trouvent au niveau du contact entre les neurones. Elles convertissent les impulsions nerveuses en signaux chimiques, c'est-à-dire en molécules de neurotransmetteurs. Lorsqu'un potentiel d'action atteint l'extrémité d'un axone, une quantité de neurotransmetteurs est libérée dans l'intervalle synaptique. Le nombre de synapses par neurone varie considérablement, allant de plusieurs centaines à une dizaine de milliers [81].

Le neurone artificiel (figure 3.2) reçoit des entrées qui sont analogues aux impulsions électrochimiques que les dendrites des neurones biologiques reçoivent d'autres neurones. Il est donc un opérateur mathématique très simple qui traduit la représentation mathématique d'un neurone biologique. Il possède des entrées qui peuvent être les sorties d'autres neurones ou des signaux extérieurs, et une sortie. La valeur de la sortie résulte du calcul de la somme des entrées, pondérées par des coefficients (poids synaptiques), et du calcul d'une fonction non linéaire bornée de cette somme pondérée. Cette fonction (fonction d'activation) est le plus souvent une fonction sigmoïde ou un échelon [82].

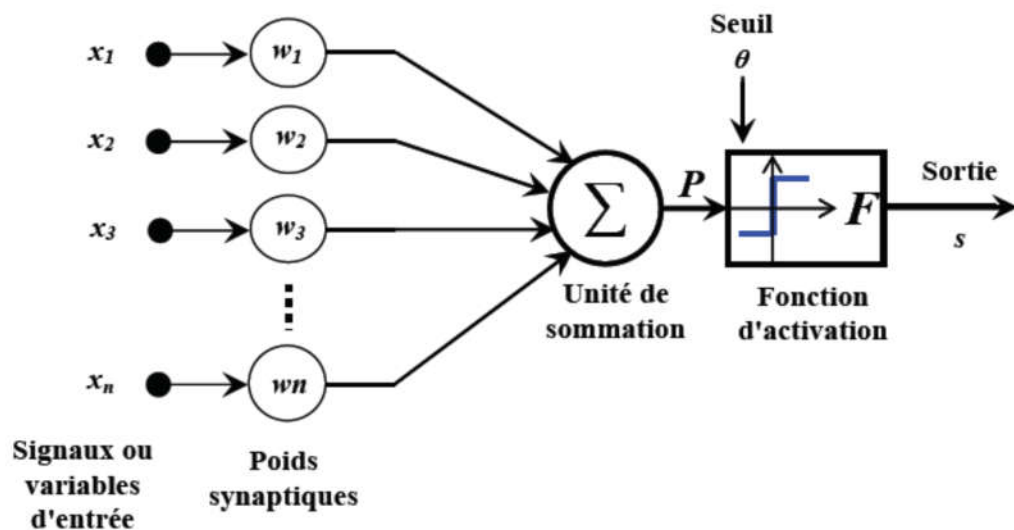


Figure 3.2 : Neurone artificiel, [81]

Le tableau 3.1 résume la correspondance entre le neurone biologique (figure 3.1) et le neurone artificiel (figure 3.2). On remarque que chaque composant ou constituant du neurone artificiel trouve son inspiration biologique.

Tableau 3.1 : Correspondance entre modèles neuronaux biologiques et artificiels
[83]

Cerveau	Réseau de neurones
Neurone	Élément processeur
Dendrites	Somme pondérée
Corps cellulaire	Fonction d'activation
Axone	Fonction de sortie
Synapse	Poids

3.2.2. Concept du réseau de neurones de type feedforward

Les neurones artificiels sont arrangés en couches (figure 3.3). La couche d'entrée reçoit des signaux d'entrée u_i du monde réel et chaque couche successive reçoit les sorties pondérées $w_{ij}u_i$ de la couche précédente comme entrée. Cela crée un ANN de type feedforward (cas de perceptrons multicouches), dans lequel chaque sortie d'une couche est alimentée en avant à la couche suivante où elle est traitée. Les sorties de la dernière couche constituent les sorties vers le monde réel [84].

Dans un tel ANN de type feedforward un neurone, dans une couche cachée ou une couche de sortie, a deux tâches [81].

- Il additionne le biais aux entrées pondérées de plusieurs connexions, puis applique une fonction d'activation (aussi appelée fonction de transfert ou fonction de seuillage) à cette somme comme donnée par l'équation 3.1 (pour le neurone j de la couche cachée de la figure 3.3) [85] :

$$Z_j = f_h(\sum_{i=1}^n w_{ji}^l u_i + b_{ij}); \quad j = 1, 2, \dots, m \quad (3.1)$$

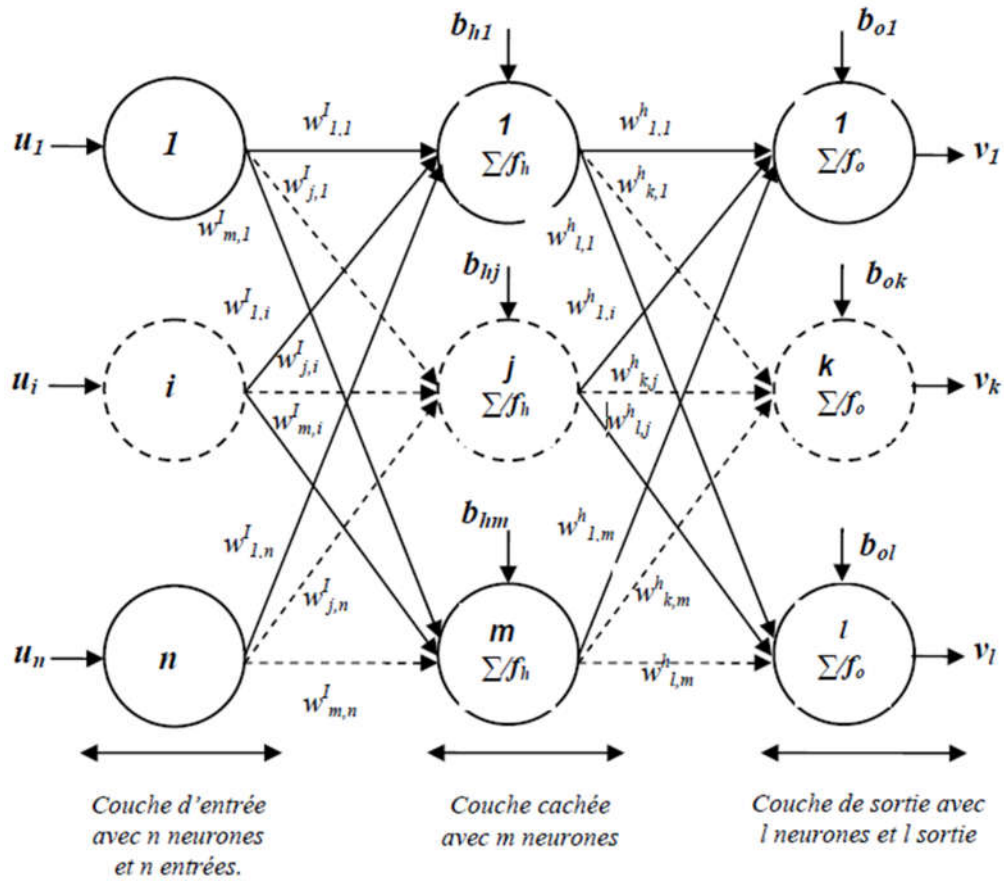


Figure 3.3: Réseau de neurones artificiels de type feedforward [81]

- Il propage la valeur résultante par les connexions partantes aux neurones de la couche suivante où elle subit le même processus tel que donné par l'équation 3.2 (par exemple les sorties de la couche cachée alimentée au neurone k de la couche de sortie donnent la sortie v_k) [85]

$$v_k = f_o\left(\sum_{j=1}^m w_{kj}^h z_j + b_{ok}\right); \quad k = 1, 2, \dots, l \quad (3.2)$$

En combinant des équations 3.1 et 3.2 on obtient la relation entre la sortie v_k et les entrées u_i d'un ANN:

$$v_k = f_o\left(\sum_{j=1}^m w_{kj}^h f_h\left(\sum_{i=1}^n w_{ji}^I u_i + b_{ij}\right) + b_{ok}\right) \quad (3.3)$$

Le nombre de neurones dans la couche d'entrée et la couche de sortie est déterminé par le nombre de variables indépendantes et dépendantes respectivement. Le modélisateur définit également le nombre de couches cachées et le nombre de neurones dans chaque couche cachée. Le modèle est développé

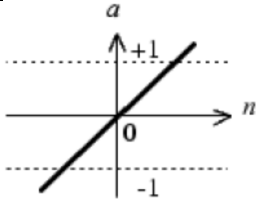
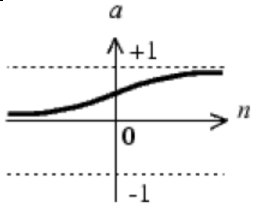
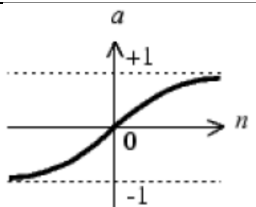
à travers un processus d'apprentissage où un ensemble de données expérimentales des variables indépendantes est présenté à la couche d'entrée du réseau. Les sorties de la couche de sortie comprennent une prédiction des variables dépendantes. Le réseau apprend les relations entre les variables indépendantes et dépendantes en comparant itérativement les sorties calculées avec les sorties expérimentales, puis en ajustant la matrice de poids et le vecteur des biais de chaque couche grâce à un algorithme d'apprentissage de rétropropagation du gradient d'erreur. Ainsi, le réseau développe un modèle neuronal capable de prédire avec une précision acceptable les variables de sortie dans le domaine des données d'apprentissage. Par conséquent, l'objectif de la modélisation neuronale est de minimiser les erreurs de prédiction de l'ensemble des données de validation présentées au réseau à la fin de l'étape d'apprentissage [86].

3.2.3. Fonctions d'activation

La fonction d'activation calcule la valeur de l'état du neurone, qui est transmise aux neurones aval. Elle sert également à introduire une non-linéarité dans le fonctionnement du neurone. Contrairement aux neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, +1]$ ou $[-1, +1]$. Il existe de nombreuses formes possibles pour la fonction de transfert. Le tableau 3.2 résume quelques fonctions de transfert utilisées dans le programme de calcul techniques MatLab® [87,84].

De plus, étant donné que la continuité et la différenciabilité des fonctions d'activation à tous les points sont les caractéristiques requises par les algorithmes d'optimisation actuels, les fonctions d'activation typiques qui répondent à ces exigences sont la fonction linéaire, la fonction tangente hyperbolique et la fonction sigmoïde logarithmique. Selon [88], la fonction linéaire est plus appropriée pour les neurones de sortie, tandis que les fonctions sigmoïdes sont plus appropriées pour les neurones des couches cachées.

Tableau 3.2: Fonctions de transfert usuelles, $a = f(n)$, [83]

Nom de la fonction	Relation entrée / sortie	Graphe de la fonction	Fonction MATLAB® correspondante
Linéaire	$a = n$		purelin
Logistique (Sigmoide logarithmique)	$a = \frac{1}{1 + e^{-n}}$		logsig
Tangente hyperbolique	$a = \frac{e^{+n} - e^{-n}}{e^{+n} + e^{-n}}$		tansig

3.2.4. Architecture des réseaux de neurones artificiels

Nous avons introduit précédemment le concept de ANN de type feedforward, il existe cependant d'autres types de ANN. En se basant sur le guide d'utilisateur de la boîte à outils des ANN de MatLab® [83] des principaux types d'ANN sont:

- Les réseaux de neurones non bouclés (perceptron multi couches, réseaux à fonctions de base Radiales).
- Les réseaux de neurones bouclés (cartes auto-organisatrices de Kohonen, réseaux de Hopfield).

3.2.5. Sélection d'un réseau de neurones artificiel

Il est clair que l'application réussie des ANN pour la modélisation des problèmes d'ingénierie est fortement affectée par quatre facteurs majeurs [81] :

- Le type de réseau ;
- La structure de réseau (nombre de couches cachées, nombre de neurones par couche cachée) ;
- Les fonctions d'activation ;
- L'algorithme d'apprentissage.

3.2.5.1. Sélection du type de réseau

Parmi les réseaux de neurones de type feedforward qui possèdent la propriété d'approximateur universel de fonction, on trouve les MLP. Ils présentent ces caractéristiques [81] :

- Un MLP peut avoir une ou plusieurs couches cachées.
- Pour un MLP, les neurones des couches cachées ou de la couche de sortie ont le même modèle.
- Les couches cachées et la couche de sortie d'un MLP, sont non linéaires. Cependant, en cas de régression non linéaire la couche de sortie est de préférence linéaire.

3.2.5.2. Sélection de la structure du réseau

Il est bien établi que la variation du nombre des neurones de(s) couche(s) cachée(s) a un effet significatif sur la capacité de prédiction du réseau. Le nombre adéquat de neurones cachés dépend d'une façon complexe de, [89] :

- Le nombre de neurones de la couche d'entrées et de sorties ;
- La taille ou la dimension de l'échantillon d'apprentissage ;
- La quantité de bruit dans les cibles (sorties du réseau) ;
- La complexité de la fonction ou de la classification à faire apprendre au réseau;
- L'architecture du réseau ;
- Le type de fonction d'activation des neurones cachés ;

- L'algorithme d'apprentissage ;
- La régularisation

Dans la plupart des situations, il est impossible de déterminer le nombre adéquat de neurones cachés sans entraîner plusieurs réseaux et évaluer ensuite l'erreur de généralisation de chaque réseau. Un nombre trop faible de neurones cachés conduit à des erreurs d'apprentissage et de généralisation élevées en raison du sous-apprentissage, tandis qu'un nombre trop élevé de neurones cachés entraîne une erreur d'apprentissage très faible mais une erreur de généralisation élevée en raison du sur-apprentissage [89].

Comme nous l'avons mentionné précédemment, il n'y a pas de règles générales pour déterminer le nombre de couches cachées d'un MLP pour un problème donné. Par conséquent, la stratégie d'essai et d'erreur est la plus raisonnable [90]. Selon [91], la méthode la plus courante pour optimiser les performances d'un ANN est de varier le nombre de neurones dans la couche cachée et de choisir l'architecture qui présente la meilleure capacité de généralisation.

3.2.5.3. Sélection de la fonction d'activation

Les fonctions d'activation pour les unités cachées sont nécessaires pour introduire la non-linéarité dans le réseau. Pour l'apprentissage par rétro propagation, la fonction d'activation doit être différentiable et il est préférable qu'elle soit bornée; les fonctions sigmoïdes comme "logsig" et "tansig" sont les choix les plus habituels. La fonction "tansig" qui produit des valeurs positives et négatives a tendance à permettre un apprentissage plus rapide que des fonctions qui produisent seulement des valeurs positives comme la fonction "logsig" [89].

3.2.5.4. Sélection du type d'apprentissage : Apprentissage supervisé

Cette technique d'apprentissage consiste à calculer les paramètres de connexion entre les différentes couches du réseau de neurones de manière à ce que les sorties du réseau soient aussi proches que possible des sorties désirées pour les exemples utilisés. Les combinaisons d'entrées et de sorties désirées étant préalablement connues, il s'agit d'adapter les paramètres du réseau afin

que, pour chaque exemple, la sortie du réseau correspond à une sortie désirée et connue. Ainsi, l'apprentissage supervisé force le réseau à converger vers un état final précis chaque fois qu'une donnée d'entrée lui est présentée [92]. Pour cela, des algorithmes d'optimisation sont utilisés, cherchant de manière itérative à minimiser une fonction de coût qui mesure l'écart entre la réponse réelle du réseau et la réponse désirée. Ce type d'apprentissage est utilisé dans de nombreuses architectures de réseaux de neurones artificiels non bouclés, tels que le MLP (Perceptron Multi-Couches).

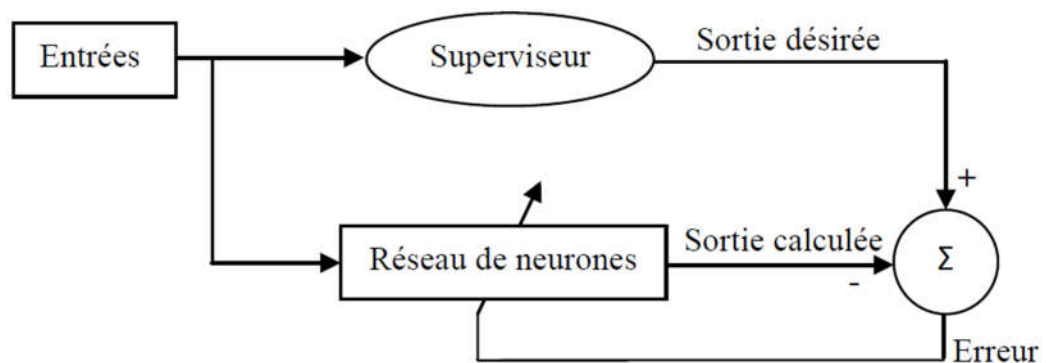


Figure 3.4: Illustration d'apprentissage supervisé

3.2.5.5. L'apprentissage

L'apprentissage est la procédure par laquelle les valeurs des poids synaptiques du réseau sont déterminées et la structure du réseau est établie. Plusieurs algorithmes d'apprentissage utilisent la procédure de rétropropagation et bien que chacun ait ses propres avantages, comme le taux de calcul et les exigences de calcul et de mémoire de stockage, il n'y a aucun algorithme qui convient à tous les problèmes. La performance de chaque algorithme dépend du processus modélisé, de l'échantillon d'apprentissage et du mode d'apprentissage. Cela explique que le choix des algorithmes, bien qu'il soit d'une importance primordiale, dépend étroitement de l'application concernée et demeure principalement expérimental [93]. L'algorithme de rétropropagation du gradient comporte donc deux phases [93].

La première phase "forward" concerne le signal qui se propage vers l'avant du réseau, d'un neurone à un autre, après initialisation des poids synaptiques et

des biais. Pendant cette phase, les poids synaptiques restent inchangés. Cette phase commence donc à la première couche cachée par le calcul du signal de tous les neurones qui la constituent et se termine à la couche de sortie par le calcul de la fonction d'erreur de tous les neurones qui la constituent également.

La deuxième phase, appelée "backward", concerne la propagation en sens inverse du signal d'erreur à travers les couches, en partant de la couche de sortie. Pour chaque neurone, on calcule le gradient local de l'erreur afin de corriger les poids synaptiques, une connexion à la fois, en utilisant une règle connue sous le nom de règle Delta. Pour un neurone de sortie, le gradient local est égal au signal d'erreur multiplié par la dérivée de la fonction d'activation appliquée à l'entrée nette de ce neurone.

L'algorithme de rétropropagation peut prendre deux formes ou modes. La première, appelée rétropropagation incrémentale ou séquentielle, consiste à ajuster les poids après chaque présentation. En revanche, la rétropropagation par lots (batch backpropagation) n'ajuste les poids qu'après un certain nombre de présentations (une époque), généralement égal au nombre d'exemples de l'échantillon d'apprentissage. Dans cette forme, on évite les interférences entre les modifications successives des poids : en calculant le gradient sur la somme des erreurs obtenues pour chaque exemple du lot, on effectue une sorte de moyenne. Cependant, si la base d'apprentissage est très importante, le besoin en capacité de mémoire de stockage et la redondance des informations rendent la procédure par lots moins efficace que la procédure séquentielle [89]. La rétropropagation se résume à un problème de recherche de minimum, ce qui soulève au moins trois questions liées à l'optimisation multidimensionnelle [94, 95]:

- Vers quel minimum la convergence s'effectue-t-elle ?
- À quelle vitesse l'algorithme converge-t-il ?
- Quelles sont les capacités de mémoire nécessaires ?

Les algorithmes de rétropropagation implantés actuellement dans les logiciels de réseaux de neurones artificiels (ANN) varient en fonction de la méthode d'optimisation utilisée et visent à apporter la meilleure solution à ces trois questions [81].

Un problème important qui peut survenir lors de l'apprentissage est celui du surapprentissage. Si le réseau de neurones apprend par cœur, il donnera de mauvais résultats lorsque des données exclues du jeu d'apprentissage lui seront présentées.

Il existe des méthodes pour optimiser la phase d'apprentissage afin d'éviter le surapprentissage ou le sous-apprentissage, notamment la technique de l'arrêt prématuré (early stopping) et la technique de régularisation.

- arrêt prématuré (Early stopping)

La technique de l'arrêt prématuré consiste à diviser les données disponibles en trois ensembles distincts. Le premier ensemble est utilisé pour l'apprentissage du réseau de neurones, le deuxième pour tester le réseau et le troisième pour vérifier la capacité de généralisation du réseau [96]. Pour le deuxième ensemble, l'erreur de test (entre les prédictions et les cibles) calculée pendant la phase d'apprentissage doit normalement diminuer au fil du processus d'apprentissage. Mais lorsque le réseau commence à apprendre par cœur (l'erreur de test recommence à croître), l'apprentissage est arrêté prématurément, d'où le nom d'"apprentissage avec arrêt prématuré" donné par [97] ou d'"apprentissage atténué" donné par [95]. L'apprentissage avec arrêt prématuré se comporte comme montré sur la figure 3.5

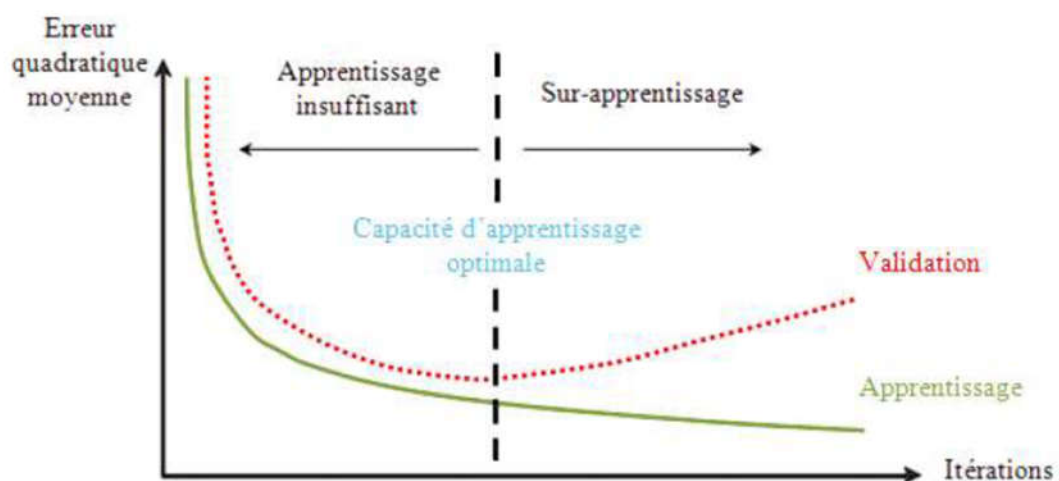


Figure 3.5 : Evolution schématique de l'erreur de l'erreur sur les sous-ensembles d'apprentissage et de validation en fonction du nombre d'itérations.

- Régularisation Bayésienne

La méthode de régularisation Bayésienne consiste à modifier la fonction de coût (MSE) en lui ajoutant un terme de régularisation (MSW) égal à la somme des carrés des poids W_{ij} et des biais b_k , afin de pénaliser les valeurs absolues élevées des poids et éviter ainsi le phénomène de sur-apprentissage [98]. On obtient donc une fonction du coût MSE " régulée " (MSE_{reg}) qui est donnée comme suit (Equation 3.4):

$$MSE_{reg} = \gamma MSE + (1 - \gamma)MSW \quad (3.4)$$

Avec $MSW = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2$ et $MSW = \frac{1}{N_p} \sum_{j=1}^{N_p} w_j^2$

Où γ est un scalaire positif qui pondère l'importance des valeurs des poids sur l'erreur quadratique moyenne, N l'ensemble d'apprentissage, d_i la valeur cible, y_i la valeur de sortie du réseau, N_p le nombre de paramètres du réseau et w_j un des poids W_{ij} ou des biais b_k du réseau de neurones.

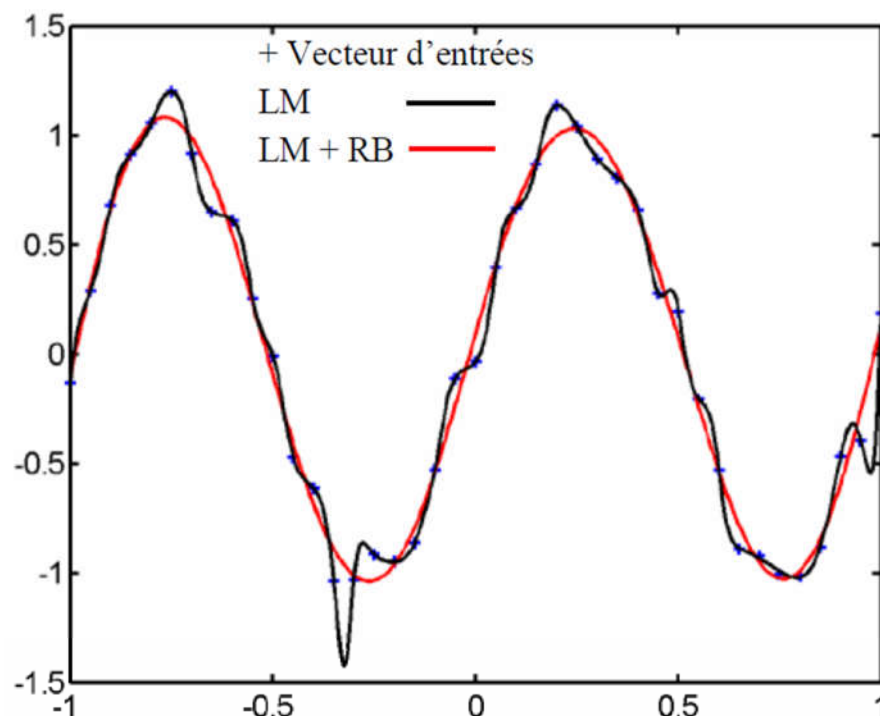


Figure 3.6 : Comparaison entre l'apprentissage avec l'algorithme de LM seul et l'algorithme LM+BR [100].

La régularisation bayésienne minimise la combinaison linéaire de MSE_{reg} afin d'obtenir une valeur optimale de γ . Cette modification entraîne une diminution des poids et des biais et force le réseau à donner une bonne réponse, évitant ainsi le sur-apprentissage.

La régularisation bayésienne est résolue à l'aide de l'algorithme de Levenberg-Marquardt [99]. Cet algorithme permet de mesurer le nombre de paramètres (poids et biais) réellement utilisés par le réseau. Cette méthode est souvent utilisée car elle est très simple à mettre en œuvre et ne nécessite pas l'utilisation d'un ensemble de validation pour contrôler l'apprentissage du réseau.

Tableau 3.3 : Résumé des caractéristiques des algorithmes d'apprentissage de rétro propagation implantés dans MatLab® [83].

Algorithme	Caractéristiques
Levenberg-Marquardt	
Levenberg-Marquardt (<i>trainlm</i>)	C'est l'algorithme d'apprentissage le plus rapide pour les réseaux de taille modérée. Il a la caractéristique de réduction de mémoire pour l'utilisation quand l'échantillon d'apprentissage est grand.
Régularisation	
Régularisation Bayésienne (<i>trainbr</i>)	Mise à jour des poids et des biais selon l'optimisation de Levenberg-Marquardt. Il réduit au minimum une combinaison d'erreurs quadratiques et des poids et détermine ensuite la combinaison correcte afin de produire un réseau avec une capacité de généralisation meilleure. Ce processus est appelé la régularisation Bayésienne. Il réduit la difficulté de déterminer l'architecture de réseau optimum.

La régularisation bayésienne, qui suppose que les poids et les biais suivent des distributions spécifiques (les paramètres sont estimés au fur et à mesure de l'apprentissage), donne en général des résultats très satisfaisants [100].

Le tableau 3.3 résume les algorithmes d'apprentissage par rétropropagation implantés dans la boîte à outils des ANN du programme MatLab® [83].

3.3. Conclusion

Tout au long de ce chapitre, nous avons fourni des informations sur les réseaux de neurones de manière générale, en mettant l'accent sur les réseaux de neurones de type feedforward qui sont les plus couramment utilisés dans les modélisations statiques en raison de leur capacité d'approximation universelle. Nous avons présenté un algorithme d'apprentissage très utilisé dans la communauté des réseaux neuronaux : l'algorithme de rétropropagation du gradient de l'erreur. De nombreux travaux lui ont été consacrés et plusieurs modifications ont été proposées pour accélérer le temps d'apprentissage, réduire les besoins en mémoire et améliorer la capacité de généralisation ainsi que les performances du réseau obtenu.

CHAPITRE 4

RESULTATS DE LA MODELISATION

4.1 Application du réseau de neurones

4.1.1 Introduction

Nous avons utilisé le logiciel Matlab R2014a lors de la phase de prédiction pour établir un modèle de réseaux de neurones. Ce logiciel contient une boîte à outils dédiée au développement de réseaux de neurones.

4.1.2 Conception de la base de données

4.1.2.1 Collections de données

Dans ce travail, nous avons collecté une base de données composée de 38 ensembles de données. Ces ensembles de données sont issus des résultats expérimentaux disponibles dans la littérature et comprennent les paramètres d'entrée suivants : la vitesse de rotation du disque (Ω), la vitesse de rotation des jarres (ω), la composition chimique (pourcentage en poids de nickel $\%Ni$) et le temps de broyage (t). Les paramètres de sortie sont la taille des cristallites (D), et la coercivité (Hc). Toutes ces données ont été obtenues à partir de l'alliage Fe-Ni broyé dans un broyeur planétaire à haute énergie (Retsch PM 400) et dans un broyeur à billes P4 vario (Fritsch). Les deux broyeurs utilisent des billes d'acier de 20 millimètres de diamètre et des jarres d'acier d'un volume de 50 ml. Le poids total des poudres est de 10 g et le rapport entre le poids des billes et celui des poudres est de 10:1 [50,53]. Les données complètes sont présentées dans l'Annexe B.

Nous avons utilisé cette base de données pour entraîner les réseaux de neurones et obtenir des prédictions sur les matériaux nanocristallins. Le tableau 4.1 présente les plages de valeurs retenues pour chaque paramètre d'entrée.

Tableau 4.1 : Domaine de variations des paramètres d'entrés

Paramètre	Minimum	Maximum	Unité
Vitesse de rotation du disque (Ω)	212	424	<i>rpm</i>
Vitesse de rotation des jarres (ω)	50	800	<i>rpm</i>
Composition chimique (pourcentage en poids de nickel %Ni)	10	20	%
Temps de broyage (t)	0	96	<i>h</i>

4.1.1.2. Normalisation des données

Avant d'injecter la base de données dans le réseau de neurones, un prétraitement est nécessaire pour rendre les données plus adaptées à l'entrée et à la sortie du réseau. En effet, lorsque les variables d'entrée présentent des valeurs d'ordres de grandeur et de nature très différentes, les valeurs les plus petites n'ont aucun effet sur l'apprentissage. Par conséquent, dans ce cas, il est préférable de normaliser les données afin qu'elles aient le même impact statistique sur la conception du modèle. Il existe plusieurs approches pour normaliser les données, nous avons choisi une méthode qui permet d'avoir des données normalisées comprises entre 0.1 et 0.9, en utilisant l'équation de normalisation suivante :

$$N_i = 0.1 + 0.8 \left(\frac{X_i - X_{min}}{X_{max} - X_{min}} \right) \quad (4.1)$$

Où N_i est la valeur normalisée, X_i est la valeur originale et X_{min} et X_{max} sont respectivement la valeur minimale et maximale de X_i [101].

La fonction "mapminmax" de Matlab (boîte à outils de réseaux de neurones de Matlab) réalise la normalisation selon l'équation 4.1, et la fonction inverse de normalisation est réalisée par la fonction "reverse".

4.1.2 Développement du modèle ANN

4.1.2.1. L'architecture de réseau de neurones

Le réseau de neurones que nous avons mis en place est composé de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. Ce réseau contient quatre paramètres d'entrée et deux paramètres de sortie qui correspondent respectivement aux caractéristiques d'entrée et de sortie du matériau nanocristallin. Le nombre de neurones dans la couche cachée n'est pas prédéterminé, il a été déterminé pendant la phase d'apprentissage. La fonction d'activation utilisée pour les neurones de la couche cachée est la fonction tangente hyperbolique, tandis que celle utilisée pour les neurones de sortie est la fonction linéaire.

La figure 4.1 présente une représentation symbolique de cette architecture.

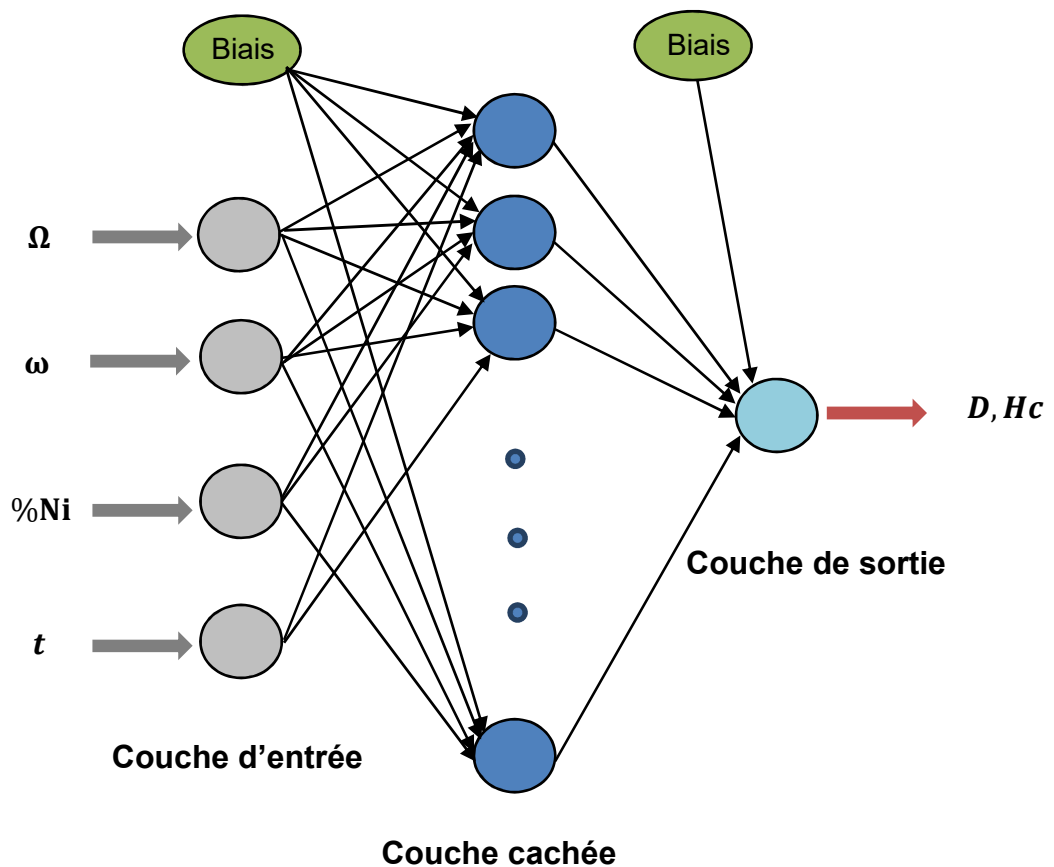


Figure 4.1: Architecture du réseau de neurones proposé

La méthodologie d'exécution ANN est décrite dans l'organigramme présenté sur la figure 4.2.

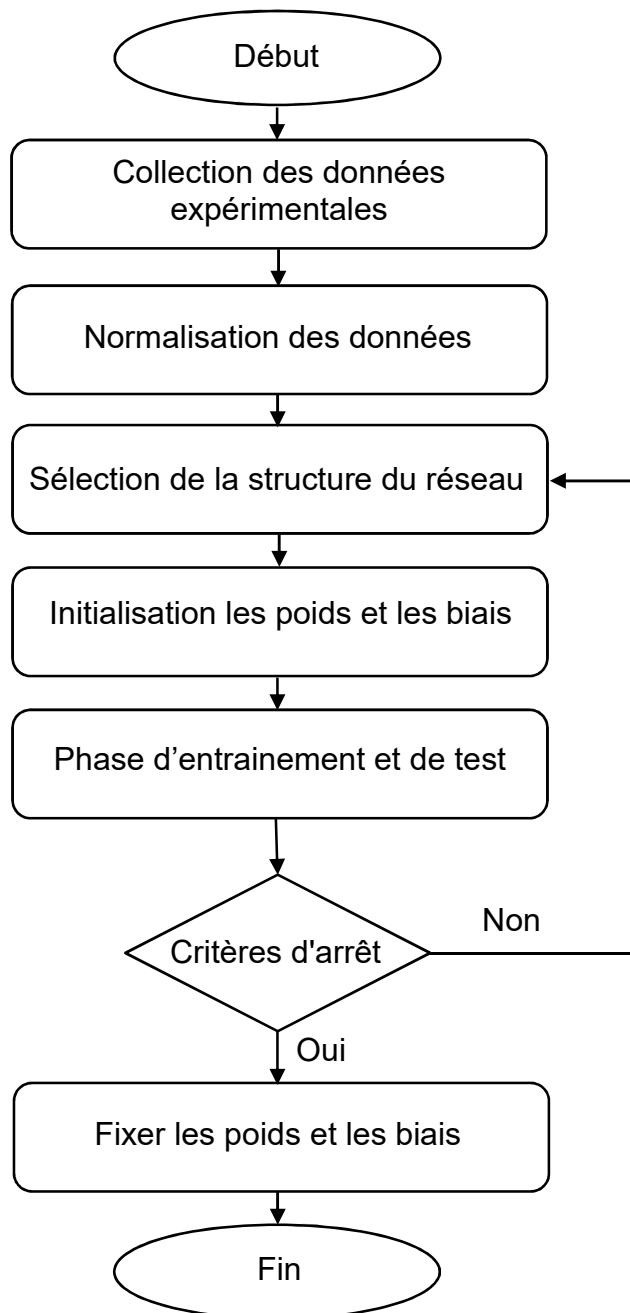


Figure 4.2 : Organigramme ANN utilisé.

4.1.2.2 Algorithme d'apprentissage

Nous avons utilisé la technique de régularisation bayésienne, comme mentionné dans le chapitre 3. Cette technique permet d'entraîner le réseau à l'aide de l'algorithme "trainbr" de Levenberg-Marquardt, qui modère les poids afin

de réduire le sur-apprentissage. Cet algorithme est disponible dans l'outil réseau de neurones de Matlab. L'avantage de cette technique est sa simplicité de mise en œuvre, car elle ne nécessite pas l'utilisation d'un ensemble de validation pendant l'apprentissage.

4.1.2.3 Découpage de la base de données

La base de données est divisée en deux ensembles : un ensemble pour l'apprentissage du réseau et un autre pour tester les prédictions du réseau. 75 % des données (28 exemples) de l'ensemble de la base de données sont utilisées pour l'entraînement du réseau de neurones. Les 25 % restants (10 exemples) sont réservés pour tester le réseau de neurones une fois l'apprentissage terminé. L'objectif est de vérifier si le réseau de neurones est capable de généraliser et de prédire des valeurs pour des données sur lesquelles il n'a pas été entraîné.

4.1.2.4 Critère d'évaluation des performances du modèle

L'évaluation de la capacité de prédiction d'un modèle mathématique qui représente au mieux toutes les données numériques de la base de données d'apprentissage doit être effectuée. Il existe plusieurs paramètres statistiques, dont trois sont couramment utilisés : la racine carrée de l'erreur quadratique moyenne (RMSE), le coefficient de corrélation (R) et l'erreur absolue moyenne (MAE). Voici leurs définitions :

Le coefficient de corrélation (R) : Ce coefficient exprime une relation linéaire entre deux variables. Si R est égal à zéro, les deux variables sont linéairement indépendantes, et si R est égal à 1, elles sont linéairement dépendantes.

$$R = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 (Y_i - \bar{Y})^2}} \quad (4.2)$$

La racine carrée de l'erreur quadratique moyenne (RMSE) : Il s'agit de la racine carrée de la moyenne des carrés des différences entre les prévisions et les observations. Plus la valeur de cette grandeur se rapproche de zéro, moins la dispersion est importante et plus le modèle prédit correctement les résultats expérimentaux.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_i - Y_i)^2}{n}} \quad (4.3)$$

L'erreur absolue moyenne (MAE) : Il s'agit de la moyenne arithmétique des valeurs absolues des différences. Plus cette grandeur tend vers zéro, moins il y a d'écart, en termes d'erreur absolue, entre les prédictions du modèle et les résultats expérimentaux.

$$MAE = \frac{\sum_{i=1}^n |X_i - Y_i|}{n} \quad (4.4)$$

Où:

X_i : Sorties mesurées

Y_i : Sorties prédites

\bar{X} : Moyenne de sorties mesurées

\bar{Y} : Moyenne de sorties prédites

n : Nombre d'échantillons

En combinant ces critères mentionnés ci-dessus, nous sommes en mesure d'évaluer de manière satisfaisante la capacité de prédiction du modèle ANN.

4.1.3 Détermination de l'architecture optimale du réseau

Le choix de l'architecture du réseau (les entrées externes, le nombre de couches cachées, le nombre de neurones cachés, les sorties, les connexions entre les neurones et la fonction d'activation des neurones dans chaque couche) influence considérablement la qualité des résultats. Selon [102], une seule couche cachée utilisant des fonctions de transfert sigmoïde est suffisante pour approximer n'importe quelle fonction non linéaire avec la précision souhaitée.

4.1.4 Résultats

4.1.4.1 Estimation de la taille des cristallites

Pour obtenir un réseau optimal, nous avons fait varier le nombre de neurones dans la couche cachée de 5 à 20, avec un pas de 5 neurones. Le seuil

d'apprentissage a été fixé à $e-06$. Le nombre maximum d'itérations a été fixé à 1000.

L'architecture neuronale est considérée comme optimale lorsque les racines carrées de l'erreur quadratique moyenne et de l'erreur absolue moyenne sur les ensembles d'apprentissage et de test sont du même ordre de grandeur et les plus petites possibles, et que les coefficients de corrélation sont proches de 1.

Des illustrations statistiques et graphiques ont été utilisées pour évaluer les modèles neuronaux. Nous avons fait varier le nombre de neurones pour sélectionner les meilleures valeurs pour le nombre de neurones et de couches cachées.

Le tableau 4.2 résume les performances de l'ANN. On peut constater que le nombre optimal de neurones dans la couche cachée est de 10.

Tableau 4.2 : Performances du réseau de neurones pour D

Performance	Nombre de neurones de la couche cachée pour D							
	Apprentissage				Test			
	5	10	15	20	5	10	15	20
R	0.9990	0.9990	0.9989	0.9990	0.9939	0.9983	0.9979	0.9983
$RMSE$	0.4476	0.4462	0.4525	0.4490	1.8930	1.5517	1.6311	1.5739
MAE	0.2950	0.2866	0.2957	0.2945	1.3322	0.9758	1.0408	0.9731

Il est possible de conclure que l'ajout de neurones cachés ne fait qu'augmenter la taille des réseaux de neurones et n'a aucun effet positif sur l'amélioration de la précision [103]. En se basant sur les résultats de performance des réseaux de neurones présentés dans le tableau 4.2, il a été déterminé qu'une seule couche cachée avec 10 neurones est le meilleur modèle pour prédire la taille des cristallites.

La figure 4.3 illustre la variation de l'erreur quadratique moyenne (correspondant à l'erreur entre les valeurs normalisées des prédictions et les mesures expérimentales de la taille des cristallites D) en fonction du nombre d'itérations (epochs) pour la meilleure architecture neuronale trouvée.

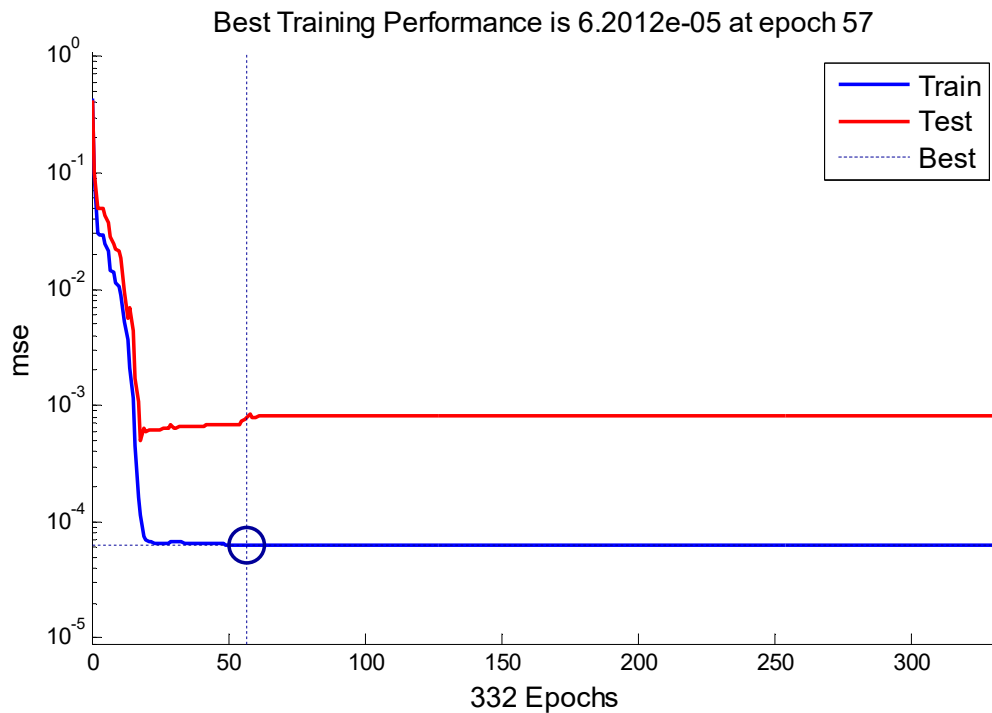


Figure 4.3 : Apprentissage d'un ANN avec 10 neurones dans une couche cachée.

Comme mentionné précédemment, la procédure d'ajustement des poids et des biais du modèle neuronal est effectuée par une technique d'optimisation, qui est un algorithme d'apprentissage. Cette technique vise à minimiser l'erreur quadratique moyenne (MSE) par itération.

On observe dans la figure 4.3 que lorsque le nombre d'itérations (ou époques) augmente, les poids et les biais du modèle neuronal convergent vers leurs valeurs optimales, ce qui entraîne une diminution continue de la MSE observée. La meilleure performance obtenue lors de la phase d'entraînement est de $6.2012e-5$ après 57 itérations.

La figure 4.4 présente le diagramme de parité, qui représente les valeurs mesurées expérimentalement et les valeurs prédites par le ANN, pour l'apprentissage et le test.

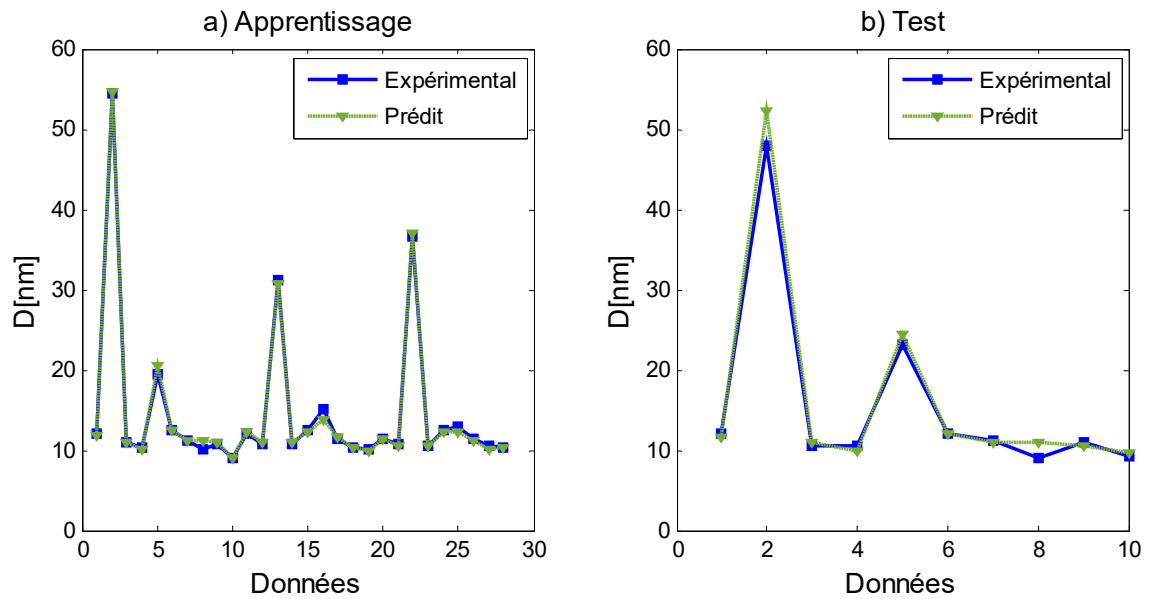


Figure 4.4: Valeurs estimées de la taille des cristallites

On remarque une légère différence entre les valeurs expérimentales et les valeurs prédites par l'ANN.

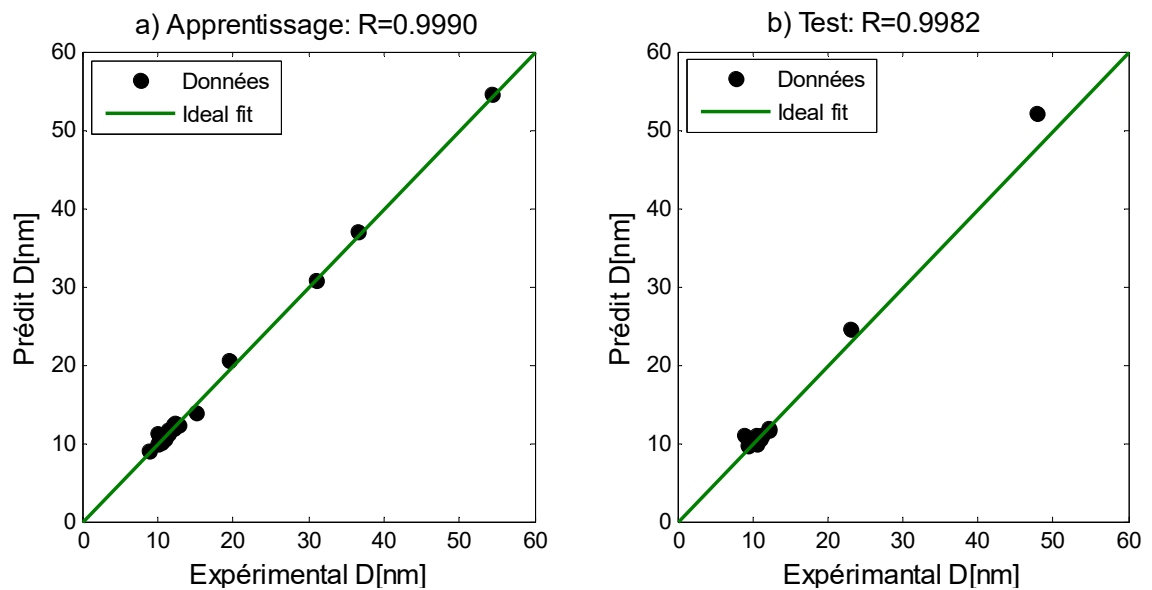


Figure 4.5: Valeurs prédites du D en fonction des données expérimentales.

En conséquence, le réseau neuronal artificiel (ANN) est capable de généraliser et de prédire des valeurs pour des données sur lesquelles il n'a pas été entraîné.

La comparaison des données prédites (données de test) par ce réseau avec les données expérimentales permet de conclure que les prédictions sont relativement proches de la réalité. Ces résultats prouvent que la capacité de généralisation de ce réseau est satisfaisante, ce qui confirme son efficacité.

La figure 4.5 a) montre la comparaison des valeurs prédites obtenues pour la taille des cristallites avec les résultats expérimentaux en utilisant le modèle neuronal optimal. La valeur R de 0,9990 pendant la phase d'apprentissage indique un coefficient de corrélation élevé. On peut également constater sur la figure 4.5 que les points de données sont proches de la ligne diagonale du graphique, ce qui démontre un bon modèle de prédiction.

Pour la phase de test, comme le montre la figure 4.5 b), la valeur R est égale à 0,9982. Cela signifie que la capacité de prédiction de ce réseau est satisfaisante, ce qui confirme son efficacité. Les neurones sont connectés les uns aux autres par des connexions pondérées. Ce sont les poids de ces connexions qui régissent le fonctionnement du réseau et permettent une application de l'espace des entrées vers l'espace des sorties, grâce à une transformation non linéaire. Le modèle neuronal optimal pour la taille des cristallites est constitué de 4 neurones d'entrée, 10 neurones cachés et deux neurones de sortie, dont la sortie est estimée comme suit :

$$D = \text{purelin}(LW \times \text{tansig}(IW \times [x_1; x_2; x_3; x_4] + b_1) + b_2) \quad (4.5)$$

Où $[x_1; x_2; x_3; x_4]$ est le vecteur d'entrée

$x_1 = \Omega$ est la vitesse de rotation du disque, $x_2 = \omega$ est la vitesse de rotation des jarres, $x_3 = \%Ni$ est la composition chimique (pourcentage en poids de nickel), $x_4 = t$ est le temps de broyage

IW et b_1 sont les poids et les biais de la couche cachée et LW et b_2 sont les poids et les biais de la couche de sortie. Ces quantités sont présentées dans le tableau 4.3.

Tableau.4.3 : Poids de connexions pour D

IW				LW	b1	b2
-0.0039	0.0444	-0.0316	-0.0652	0.1485	0.1150	1.6383
-0.1602	0.1686	0.2235	-0.8481	-1.0019	-0.4344	
-0.0305	0.0306	0.0938	-0.4441	0.5995	0.4572	
-0.1524	0.1531	-0.1250	-0.2804	0.4516	0.3415	
0.2931	-0.2096	0.0665	2.7624	-0.9506	-0.0302	
-0.2297	0.1898	0.1607	6.6689	-3.1800	0.4063	
0.0157	-0.0526	0.0219	0.0699	-0.1777	-0.1418	
0.0140	0.0336	-0.0134	-0.0313	0.0737	-0.0502	
-0.4110	0.4148	0.5286	3.0219	0.8767	0.4158	
0.2385	-0.2485	0.4209	0.6394	0.9204	0.6676	

4.1.4.2 Estimation de la coercivité

De manière similaire à la modélisation précédente, la meilleure structure pour le modèle neuronal est obtenue en variant le nombre de neurones de la couche cachée de 5 à 20, avec un pas de 5 neurones. L'évaluation des modèles a été réalisée en utilisant les critères d'évaluation des performances suivants : R, RMSE et MAE.

La variation de R, RMSE et MAE pour l'ensemble des données d'apprentissage et de test en fonction du nombre de neurones dans la couche cachée est illustrée dans le tableau 4.4.

En comparant les résultats de ce tableau, on peut conclure que le nombre optimal de neurones dans les couches cachées est de 10.

Tableau 4.4: Performances du réseau de neurones pour H_c .

Performance	Nombre de neurones de la couche cachée pour H_c							
	Apprentissage				Test			
	5	10	15	20	5	10	15	20
R	0.9752	0.978	0.9720	0.972	0.964	0.9769	0.9324	0.933
$RMSE$	84.214	82.495	89.399	89.148	116.310	107.449	145.548	144.647
MAE	55.525	57.000	59.274	59.155	88.046	67.647	123.573	122.456

La figure 4.6 illustre la variation de l'erreur quadratique moyenne, qui correspond à l'erreur entre les valeurs normalisées des prédictions et les mesures expérimentales de la coercivité H_c , en fonction du nombre d'itérations (epochs) pour la meilleure structure neuronale trouvée.

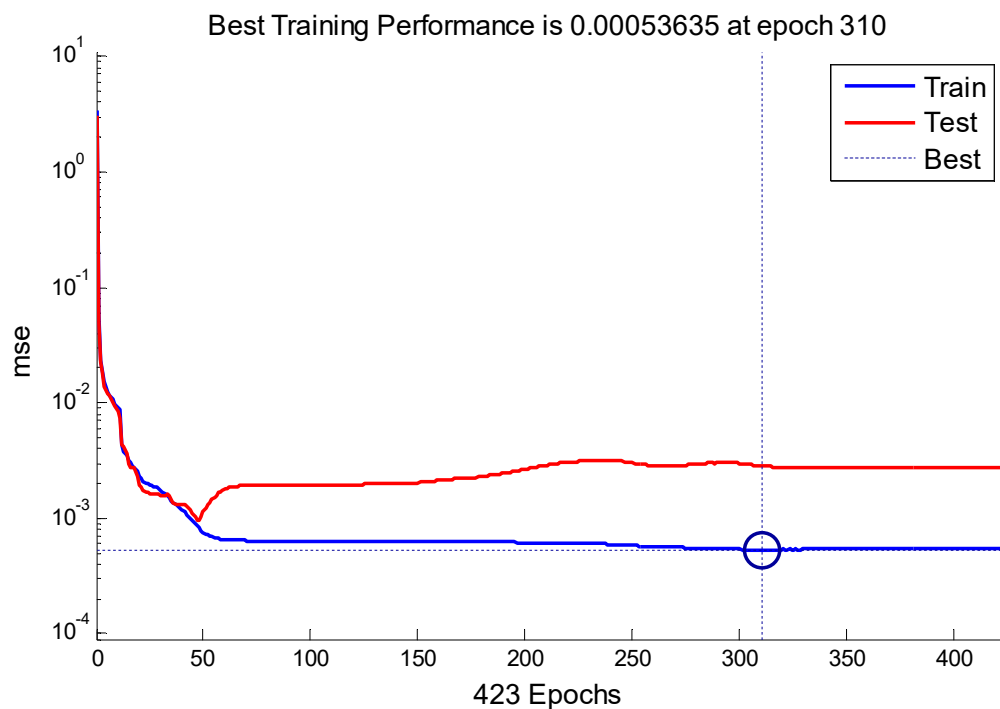


Figure 4.6: Apprentissage de l'ANN avec 10 neurones dans une couche cachée.

Les performances du modèle ANN pendant la phase d'apprentissage pour la coercivité sont présentées dans la figure 4.6. On remarque une diminution continue et la meilleure performance de la phase d'entraînement est de 0,00053635, observée après 310 itérations.

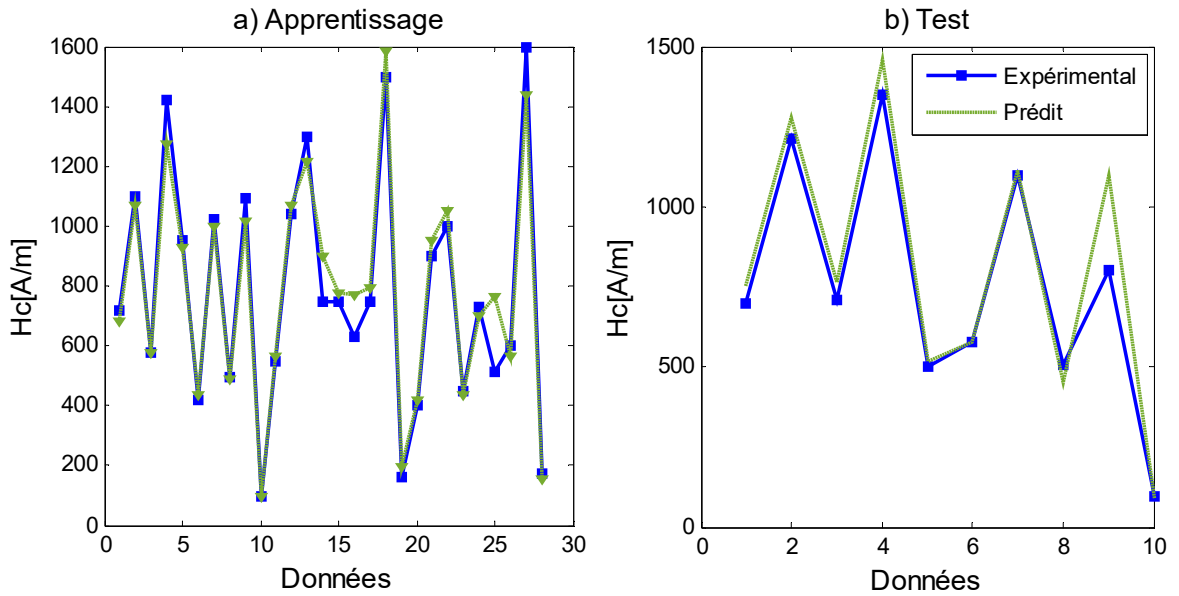


Figure 4.7: Valeurs estimées de la coercivité.

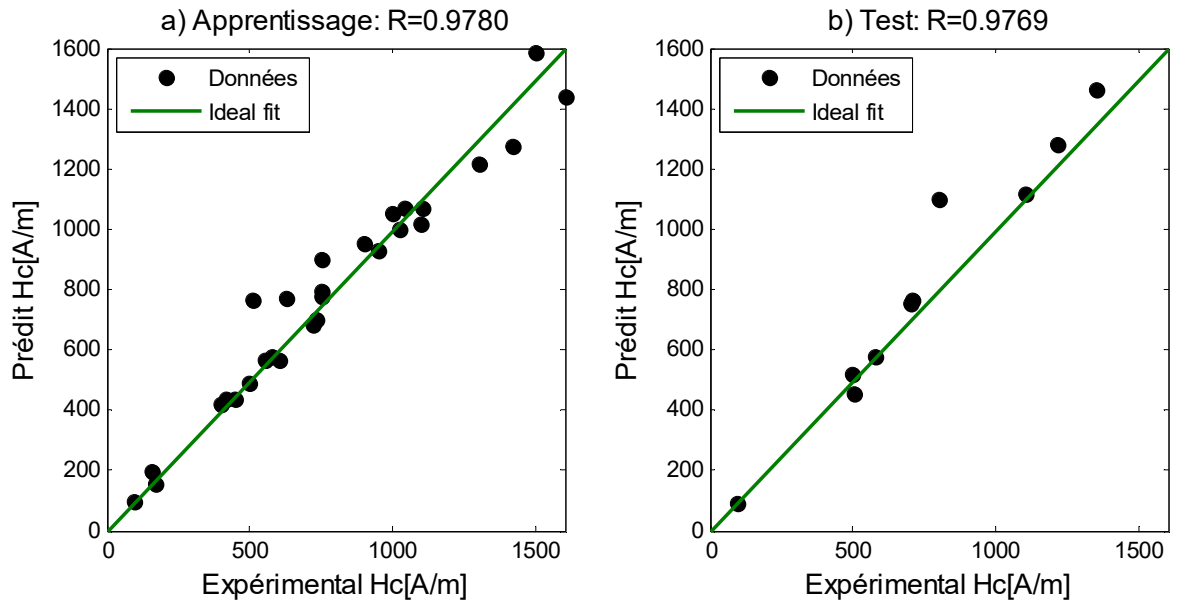


Figure 4.8: Valeurs prédites du Hc en fonction des données expérimentales.

La figure 4.7 présente une comparaison entre les prédictions du réseau neuronal pour la coercivité et les données expérimentales de l'ensemble de

données d'apprentissage et de test. On observe que les valeurs prédites de coercivité sont assez proches des valeurs mesurées expérimentalement.

La figure 4.8 illustre la comparaison entre les résultats prédits et les résultats expérimentaux pour la coercivité en utilisant le modèle neuronal optimal. Les valeurs de R sont de 0,9780 et 0,9769 pour la phase d'apprentissage et de test respectivement, ce qui indique une concordance acceptable entre les valeurs mesurées expérimentalement et les valeurs prédites. Cela signifie que la capacité de prédiction de ce réseau est satisfaisante.

Tableau 4.5 : Poids de connexions pour H_c .

IW		LW		b1	b2
-1.1147	0.0188	0.1350	0.6910	1.0015	0.4357
0.1513	-1.7290	0.3513	0.4417	0.9643	-0.2693
0.5368	0.4419	-0.01042	0.9363	0.9640	-0.0077
-0.4833	1.5781	-0.0942	1.5683	-1.1781	-0.7241
-0.4028	0.6426	-0.05917	0.4018	1.0241	0.5035
0.0175	-0.0117	0.0152	0.0061	-0.0312	-0.0161
-0.3455	0.9668	0.1336	-1.5521	1.2203	-0.2924
-0.0196	0.0131	-0.0171	-0.0069	0.0350	0.0181
-1.0102	-1.0609	0.0840	-0.0225	-0.8534	0.7069
0.0193	-0.0129	0.0168	0.0068	-0.0344	-0.0178

Le modèle de réseau de neurones optimal pour la coercivité comprend 4 neurones d'entrée, 10 neurones cachés et 2 neurones de sortie, dont la sortie est estimée comme suit :

$$H_c = \text{purelin}(LW \times \text{tansig}(IW \times [x_1; x_2; x_3; x_4] + b_1) + b_2) \quad (4.6)$$

Où $[x_1; x_2; x_3; x_4]$ est le vecteur d'entrée

$x_1 = \Omega$ représente la vitesse de rotation du disque, $x_2 = \omega$ représente la vitesse de rotation des jarres, $x_3 = \%Ni$ représente la composition chimique (pourcentage en poids de nickel), et $x_4 = t$ représente le temps de broyage

IW et b_1 sont les poids et les biais de la couche cachée, et LW et b_2 sont les poids et les biais de la couche de sortie. Ces quantités sont illustrées dans le Tableau 4.5

4.2 Optimisation par les Algorithmes génétiques

4.2.1 Introduction

Nous avons utilisé le logiciel Matlab R2014a lors de la phase d'optimisation. Ce logiciel contient une boîte à outils (toolbox) dédiée au développement des algorithmes génétiques.

4.2.2 Optimisation mono-objectif

4.2.2.1 Intégration d'un GA avec ANN

Grâce à la modélisation ANN, nous avons obtenu une fonction non linéaire (équation 4.7) qui représente la relation entre les variables d'entrée et les sorties, et qui est considérée comme une cartographie entrée-sortie. Nous avons utilisé un modèle ANN pour représenter le système complexe de broyage à billes, et la prédiction de la sortie correspondante du modèle a été rapidement estimée en fonction des modifications des paramètres d'entrée. Le modèle ANN s'est donc avéré très utile en termes de temps de calcul pour l'optimisation, car un algorithme génétique nécessite de nombreuses itérations pour calculer la sortie en fonction de la modification des variables d'entrée. Par conséquent, l'utilisation du système intégré ANN et GA nous permet de rechercher rapidement les paramètres optimaux de broyage. L'expression est la suivante :

$$D, Hc = f(\Omega, \omega, \%Ni, t) \quad (4.7)$$

Sous les contraintes :

$$212 \leq \Omega \leq 424$$

$$50 \leq \omega \leq 800$$

$$10 \leq \%Ni \leq 20$$

$$0 \leq t \leq 96$$

Où Ω est la vitesse de rotation du disque, ω est la vitesse de rotation des jarres, $\%Ni$ est la composition chimique (pourcentage en poids de nickel), et t est le temps de broyage

L'organigramme présenté à la figure 4.9 illustre le processus d'optimisation, qui est basé sur l'intégration d'un algorithme génétique (GA) dans un réseau neuronal artificiel (ANN). Le GA fonctionne en quatre étapes : génération (création d'une population aléatoire), évaluation (comparaison des individus), sélection (conservation uniquement des meilleurs individus) et croisement/mutation (permettant l'évolution des meilleurs individus en les reproduisant).

La population initiale est définie comme un ensemble d'individus. Chaque individu est représenté par un chromosome. Chaque chromosome comprend un nombre de gènes égal aux variables à optimiser. La figure 4.9 illustre un chromosome composé de 4 gènes, représentés par x_1 , x_2 , x_3 et x_4

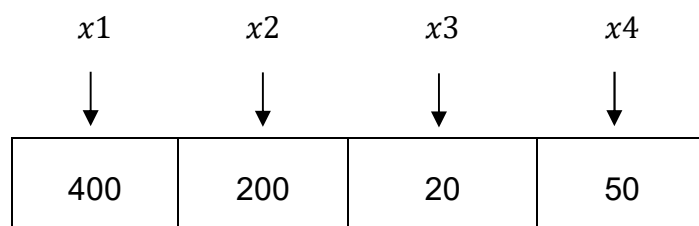


Figure 4.9 : Un chromosome typique

Un algorithme génétique (GA) utilise la fonction objectif (fournie par le modèle neuronal le plus prédictif) pour classer les individus de la population en fonction de leur qualité. Cette fonction objectif peut être définie par les équations (4.5) et (4.6). Les valeurs de D et Hc sont extraites des meilleurs modèles d'ANN entraînés.

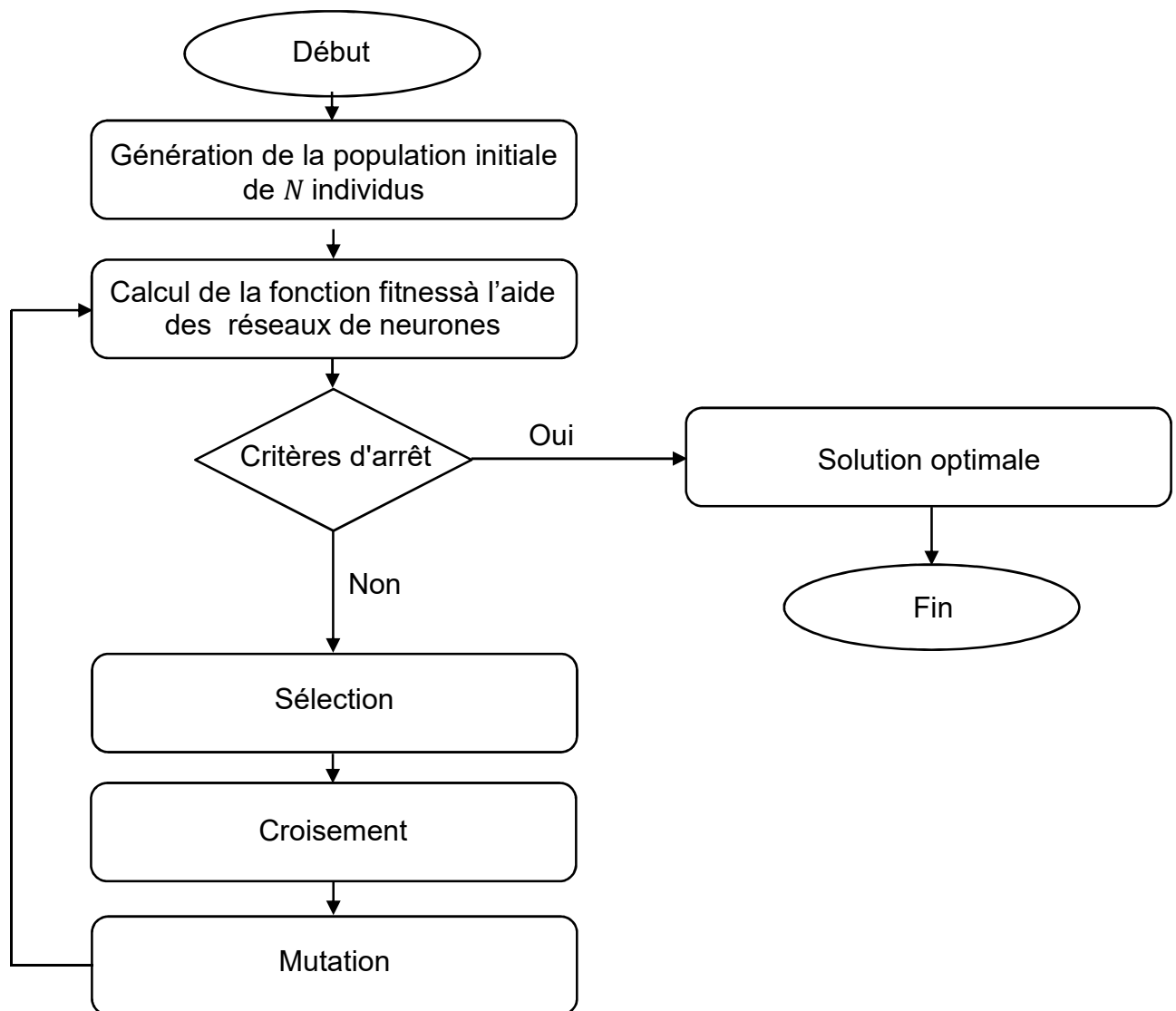


Figure 4.10 : organigramme de GA-ANN

4.2.2.2 Paramétrage du GA

Chaque problème à résoudre est unique, et une méthode d'optimisation ne peut pas être la plus efficace pour toutes les applications. La littérature offre des informations sur la mise en place des algorithmes génétiques, y compris de nombreux paramètres liés à leur évolution, tels que la taille de la population, le nombre de générations, la fonction de sélection, etc.

Tableau 4.6 : Paramètres pour la mise en œuvre de l'algorithme AG

Paramètres	Valeur
Nombre de variables	4
Fonction fitness	ANN : 4 :10 :1
Taille de la population	30
Codage de population	Réelle
Fonction de scalling	Rank
Fonction de selection	Tournoi
Elitisme	0.04
Fonction de croisement / fraction	Croisement arithmétique/0.85
Fonction de mutation/fraction	Mutation uniforme/0.1
Nombre de générations	100

L'influence de ces paramètres sur la convergence des algorithmes a été évaluée en utilisant une approche de l'essai et de l'erreur afin de déterminer les meilleurs choix pour optimiser notre méthode. Les principaux paramètres utilisés sont présentés dans le tableau 4.6.

4.2.2.3 Résultats

- Optimisation de D

L'optimisation de D mono-objectif vise à minimiser une fonction objectif, qui est exprimée de la manière suivante :

$$\text{Min}(D) \quad \text{Avec } D = f(\Omega, \omega, \%Ni, t) \quad (4.7)$$

La figure 4.11 illustre comment la fonction fitness (D) varie au fil des générations, en diminuant jusqu'à atteindre une valeur optimale minimale de 8,58 nm. Nous avons utilisé une méthode de sélection par tournoi combinée avec une approche élitiste, ce qui permet de conserver la meilleure solution obtenue au cours des générations et se traduit par une diminution de la fonction objectif dans le cas d'un problème de minimisation.

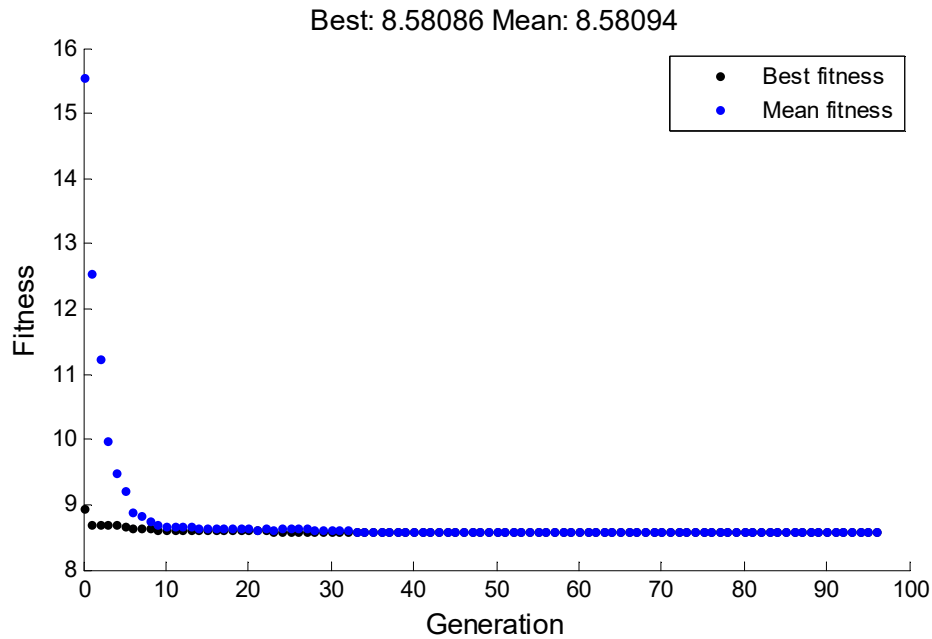


Figure 4.11 Evolution de la fonction fitness (D) au cours des générations

Les résultats de l'optimisation et les paramètres de broyage correspondants sont présentés dans le tableau 4.7.

Tableau 4.7 : Paramètres optimaux de la taille des cristallites

Ω [rpm]	ω [rpm]	%Ni[%]	t [h]	D [nm]
424	50	20	96	8.58

✓ Optimisation de H_c

La fonction fitness de l'optimisation mono-objectif pour le paramètre H_c est exprimée de la manière suivante :

$$\text{Min}(H_c) \quad \text{Avec} \quad H_c = f(\Omega, \omega, \%Ni, t) \quad (4.8)$$

La courbe de la figure 4.12 illustre la variation de la fonction fitness (H_c) au fil des générations. Elle diminue jusqu'à atteindre une valeur minimale de 46,63 A/m. Les résultats de l'optimisation et les paramètres correspondants sont répertoriés dans le tableau 4.8.

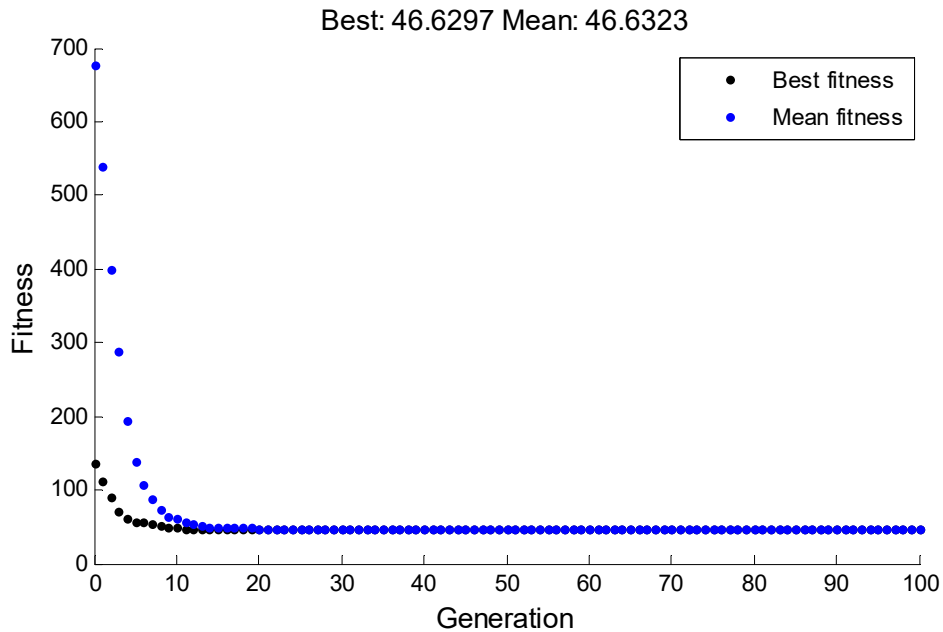


Figure 4.12 : Evolution de la fonction fitness (H_c) au cours des générations

Tableau 4.8 : Paramètres optimale de la coercivité

$\Omega[rpm]$	$\omega[rpm]$	%Ni[%]	$t[h]$	$H_c[A/m]$
212	390	20	84	46.63

Les résultats d'optimisation (Tableau 4.7 et Tableau 4.8) concordent bien avec les résultats expérimentaux. Vives et al [104] ont montré que lorsque la puissance du choc augmente, la taille des cristallites diminue. La puissance du choc correspond aux vitesses élevées du plateau et aux faibles vitesses des jarres. Hamzaoui et al [105] ont observé que de faibles valeurs du champ coercitif sont enregistrées pour de faibles vitesses du plateau Ω et des vitesses élevées des jarres ω . Ceci peut s'expliquer par une dépendance (en D^6) entre le champ coercitif et la taille des cristallites. Ils ont conclu que pour les propriétés magnétiques, la coercivité dépend du mode d'élaboration utilisé (choc : plateau \gg jarres ou friction : plateau \ll jarres). De plus, Herzer [106] a montré que les matériaux nanocristallins dont la taille des grains est inférieure à 20 nm présentent de faibles valeurs de champ coercitif.

4.2.3 Optimisation multiobjectif

4.2.3.1 Intégration de NSGA-II avec ANN

L'optimisation multi-objectif repose sur la prédiction de réseaux de neurones. À partir de ces prédictions, un programme Matlab génère l'algorithme d'optimisation NSGA-II.

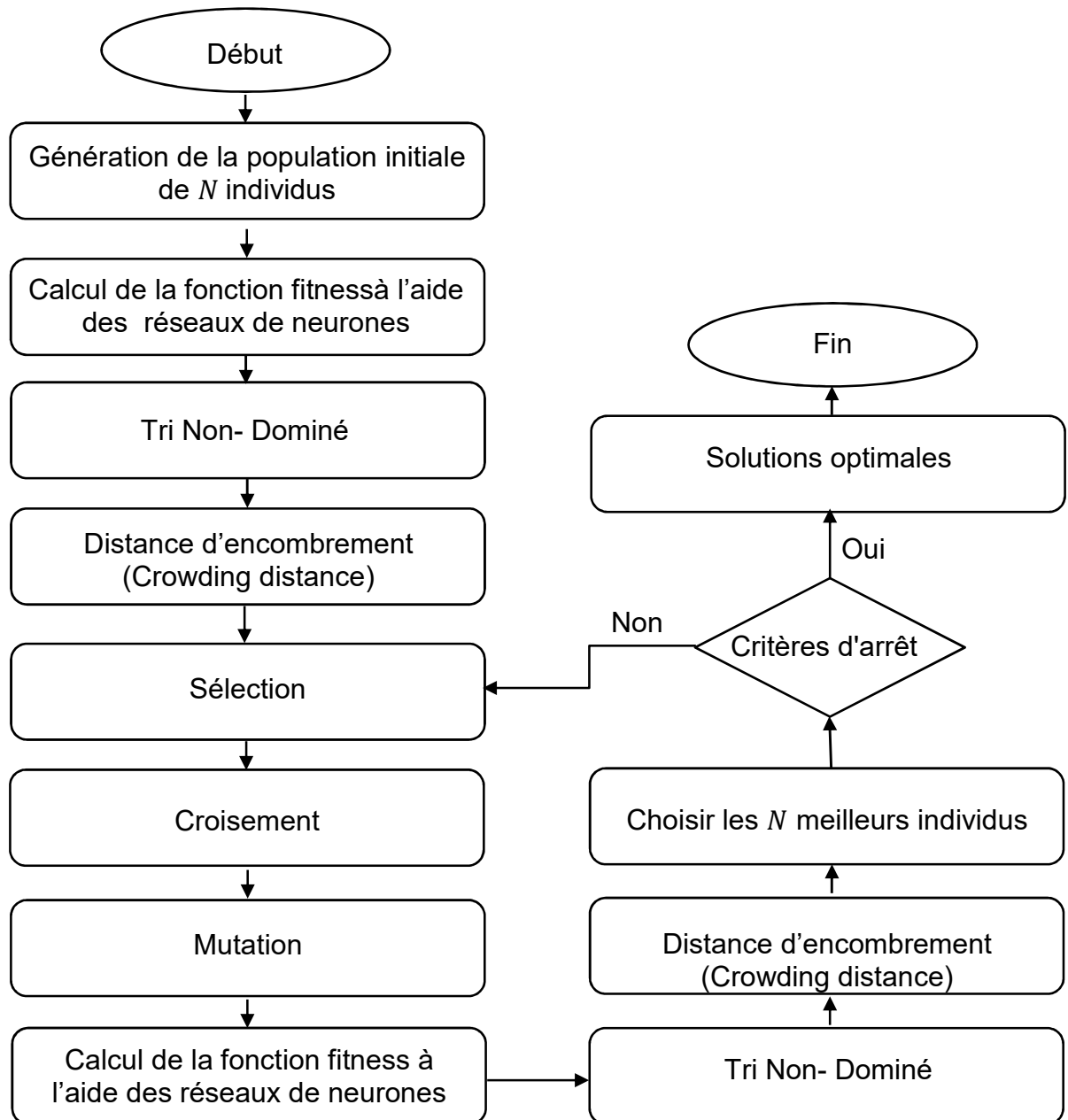


Figure 4.13 Organigramme de NSGA-II.

À son tour, l'algorithme d'optimisation NSGA-II génère aléatoirement un ensemble de solutions initiales dans leur espace de recherche tout en fixant les

paramètres de la méthode d'optimisation. Ensuite, l'algorithme calcule la valeur de la fonction objective de chaque solution en communiquant avec les résultats du premier programme (réseaux de neurones). Les opérations d'optimisation sont ensuite appliquées pour garantir que le meilleur individu soit présent à chaque génération. Cette boucle se répète jusqu'à ce que la condition d'arrêt soit atteinte (voir Figure 4.13).

4.2.3.2 Paramétrage du NSGA-II

Les opérateurs de l'algorithme NSGA-II sont guidés par un certain nombre de paramètres fixés au début de l'optimisation. De nombreuses simulations ont été réalisées sur la base de l'essai et de l'erreur afin de trouver des valeurs adéquates pour ces paramètres. Le tableau 4.9 présente les principaux paramètres utilisés dans la toolbox MATLAB.

Tableau 4.9 : Paramètres pour la mise en œuvre de l'algorithme NSGA-II.

Paramètres	Valeur
Nombre de variables	4
Fonction fitness	ANN1 : 4 :10 :1 ; ANN2 : 4 :10 :1
Taille de la population	30
Codage de population	Réelle
Fonction de selection	Tournoi
Elitisme	0.04
Fonction de croisement / fraction	Croisement arithmétique/0.85
Fonction de mutation/fraction	Mutation uniforme/0.1
Nombre de générations	200

4.2.3.3 Résultats

Dans ce cas, trois types de problèmes d'optimisation multi-objectifs sont définis, à savoir : optimisation de D par rapport à t , optimisation de Hc par rapport à t et optimisation de D par rapport à Hc .

- ✓ Optimisation de D par rapport à t

Les résultats sont présentés dans la figure 4.14 sous forme de front de Pareto. L'exemple traité concerne la taille des cristallites et le temps de broyage. Dans ce cas, nous avons un problème de minimisation à deux objectifs. Nous observons que les deux objectifs sont contradictoires, car en diminuant $Ob2:t$, l'objectif $Ob1:D$ tend à se détériorer, et vice versa. Au fur et à mesure des générations, le NSGA-II tend à regrouper les populations autour du front de Pareto, tout en minimisant les deux objectifs $Ob1$ et $Ob2$.

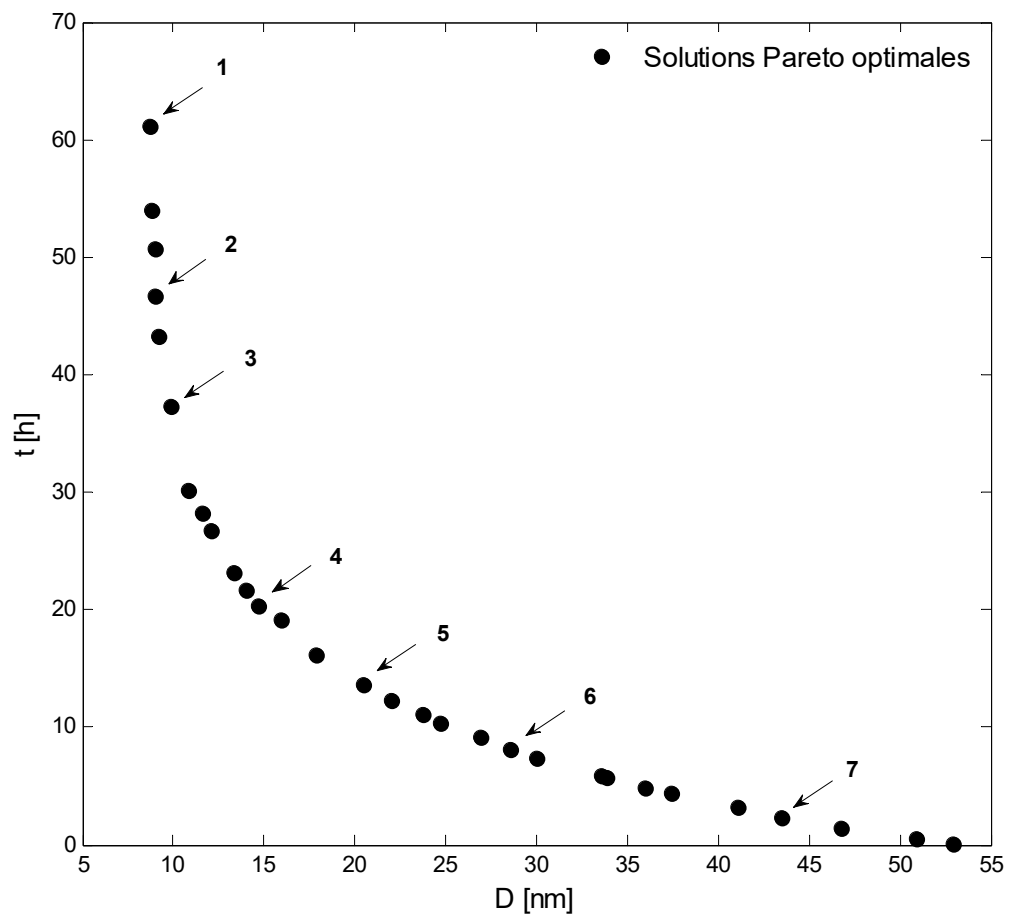


Figure 4.14 Front de Pareto-multiojectif pour la taille des cristallites et le temps de broyage

Les solutions non dominées correspondent à un compromis entre les deux objectifs. Étant donné qu'aucune des solutions dans l'ensemble Pareto-optimal n'est absolument meilleure qu'une autre, toutes les solutions sont acceptables. Le choix d'une solution par rapport à une autre dépend du décideur. Les résultats sont détaillés dans le tableau 4.10.

Tableau 4.10 : Solutions pour la taille des cristallites et le temps de broyage

	Ω [rpm]	ω [rpm]	%Ni[%]	t [h]	D [nm]	t [h]
1	414	122	20	61,18	8,71	61,18
2	418	112	18	46.72	9.02	46.72
3	420	102	20	37,29	9,90	37,29
4	366	135	19	20,33	14,70	20,33
5	417	128	17	12.3	21.99	12.3
6	420	88	16	8.12	28.55	8.12
7	424	90	18	2.32	43.45	2.32

✓ Minimisation de H_c par rapport à t

La figure 4.15 montre les résultats obtenus sous forme de front de Pareto.

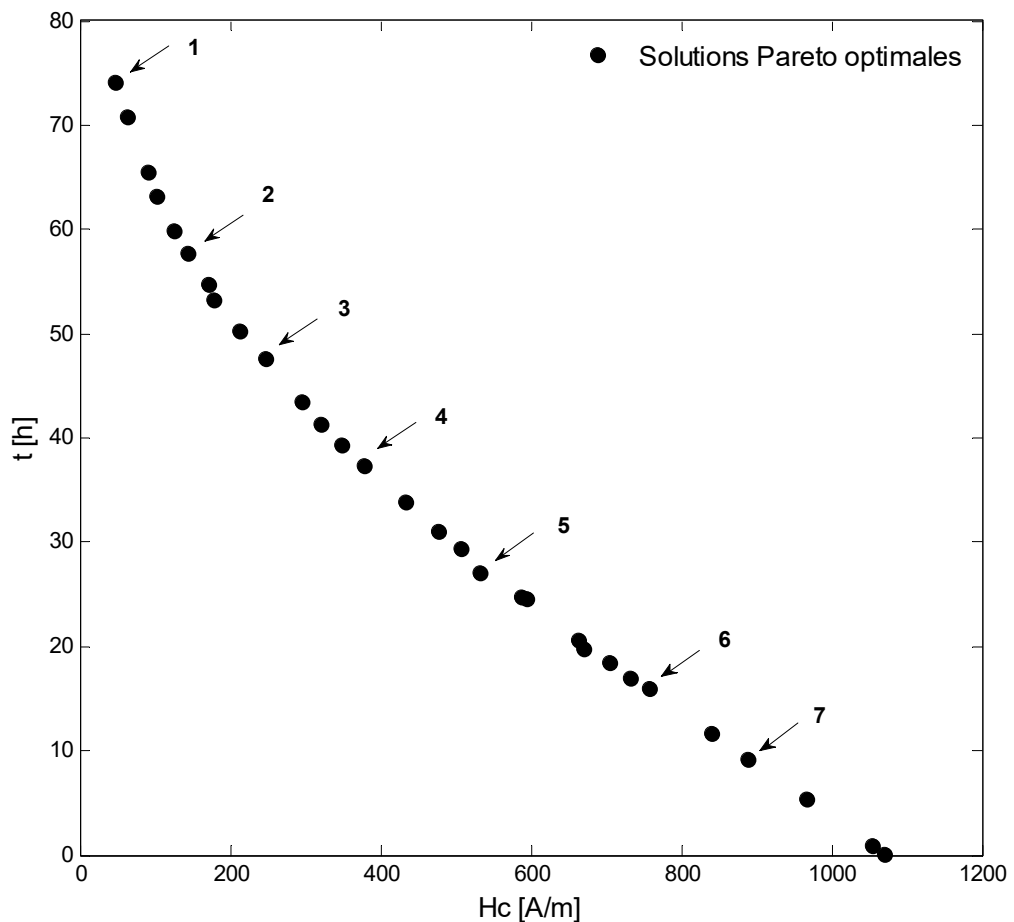


Figure 4.15 Front de Pareto du champ coercitif et le temps de broyage

Dans l'exemple traité, nous abordons le champ coercitif et le temps de broyage. Comme précédemment, nous rencontrons un problème de minimisation à deux objectifs. Les résultats de l'optimisation sont détaillés dans le tableau 4.11.

Tableau 4.11 : Solutions de la coercivité et le temps de broyage

	$\Omega[rpm]$	$\omega[rpm]$	%Ni[%]	$t[h]$	$Hc[A/m]$	$t[h]$
1	214	387	20	74,14	46,74	74,14
2	212	339	19	57,69	143,43	57,69
3	217	352	18	47.63	246.60	47.63
4	213	348	20	37.36	378.5	37.36
5	218	342	19	24.56	594.6	24.56
6	221	344	18	15.94	757.4	15.94
7	215	346	17	9.14	888.7	9.14

✓ Minimisation de D par rapport à Hc

La figure 4.16 illustre la répartition des solutions de la dernière population dans l'espace objectif ($Ob1: D$ et $Ob2: Hc$). On remarque clairement que la distribution des solutions n'est pas uniforme sur le front de Pareto représenté dans cette figure, où certaines régions sont occupées par plusieurs solutions et d'autres sont complètement vides. Cette répartition non uniforme est due à la nature des deux objectifs qui ne sont pas contradictoires [107], contrairement à ce qui est généralement observé en optimisation multi-objectif. En cas de objectifs contradictoires, l'algorithme est en mesure de garantir une distribution uniforme des solutions, comme le montrent les figures D par rapport à t et Hc par rapport à t . Il est intéressant de noter que la meilleure solution obtenue précédemment se trouve à une extrémité du front de Pareto, ce qui démontre la robustesse et la fiabilité de l'approche NSGA-II.

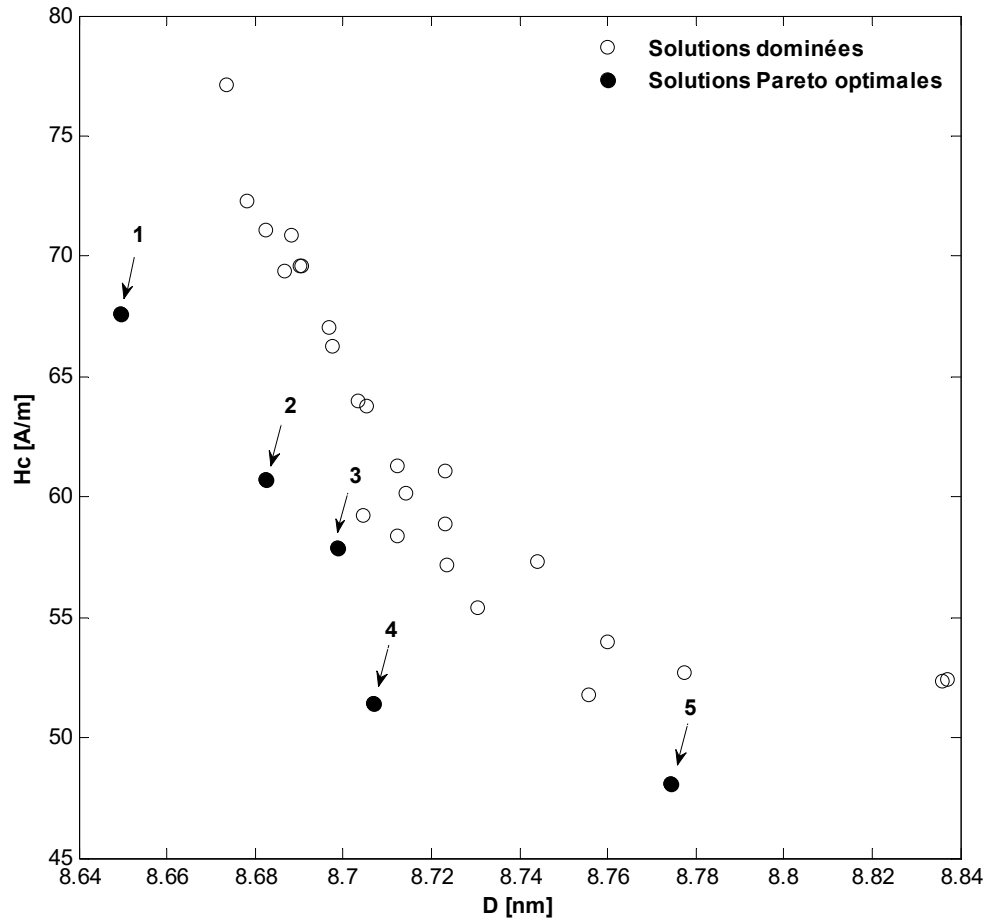


Figure 4.16 Front de Pareto de la taille des cristallites et la coercivité

Le tableau 4.12 donne les solutions de Pareto du 1^{er} rang (non dominés).

Tableau 4.12 : Solutions de la taille des cristallites et la coercivité

	Ω [rpm]	ω [rpm]	%Ni[%]	t [h]	D [nm]	H_c [A/m]
1	220	377	20	76	8,65	67,60
2	216	375	20	77	8,68	60,73
3	217	377	20	79	8,70	57,92
4	214	378	20	78	8,71	51,45
5	212	379	20	78	8,77	48,10

À titre de comparaison, les figures 4.17 et 4.18 présentent les résultats d'optimisation de cette étude ainsi que des données expérimentales provenant de la littérature, respectivement sur la taille des cristallites et la coercivité.

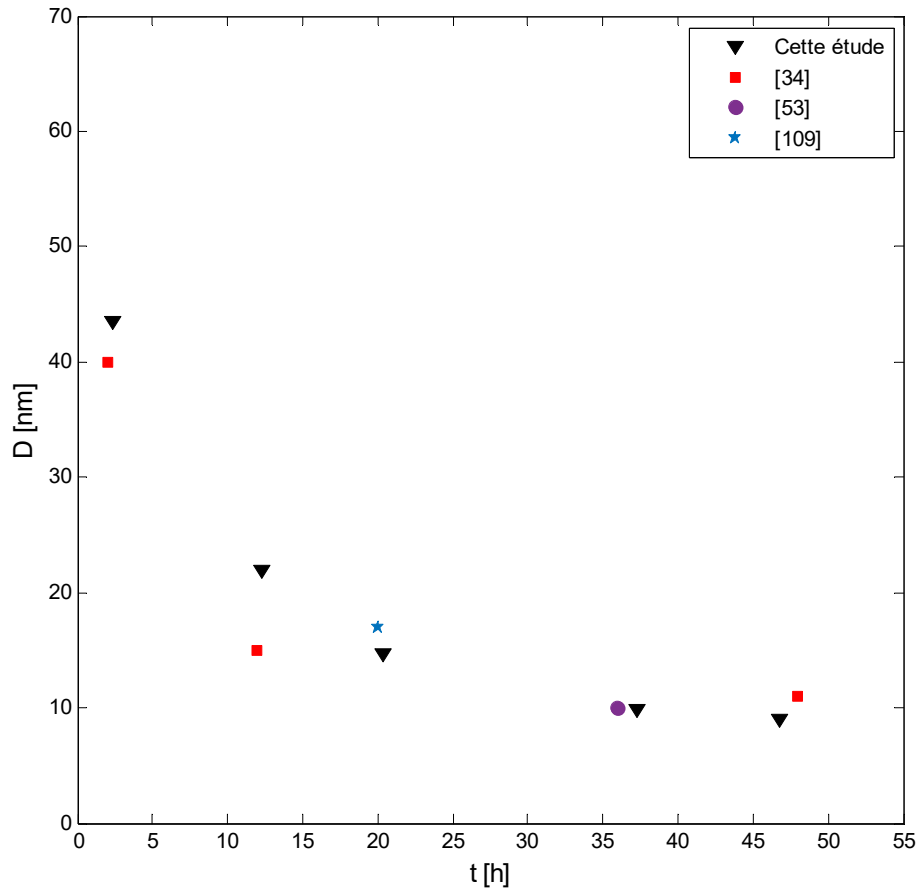


Figure 4.17 Comparaison de la taille des cristallites

Guittoum et al [34] ont étudié le composé $\text{Fe}_{80}\text{Ni}_{20}$, élaboré dans un broyeur planétaire de type Retsch PM400. La vitesse de rotation du plateau était de 240 rpm et le rapport de masse bille/poudre était de 45:1. Après 2 heures de broyage, ils ont obtenu une taille de cristallites de 40 nm, puis de 11 nm après 48 heures de broyage (étant donné que la taille des cristallites diminue avec l'augmentation du rapport de masse bille/poudre [108]). Ils ont trouvé aussi une valeur de H_c égale, dans la limite de l'incertitude expérimentale, à environ 4 Oe ($\approx 318 \text{ A/m}$) pour 48 h de broyage.

Hamzaoui et al [53] ont obtenu, pour le $\text{Fe}_{80}\text{Ni}_{20}$ en utilisant un broyeur de type FRITSCH P4, avec un rapport de masse bille/poudre de 10:1. La vitesse de rotation du disque était de 412 rpm et celle des jarres était de 100 rpm, pour un temps de broyage de 36h. Ils ont également mesuré une coercivité H_c de 400 A/m

pour une vitesse de rotation du disque de 212 rpm et des jarres de 400 rpm. Ils ont observé que de faibles valeurs de champ coercitif étaient enregistrées pour de faibles vitesses du plateau et des vitesses élevées des jarres. En revanche, de faibles valeurs de taille de cristallites étaient enregistrées pour de faibles vitesses des jarres et des vitesses élevées du plateau.

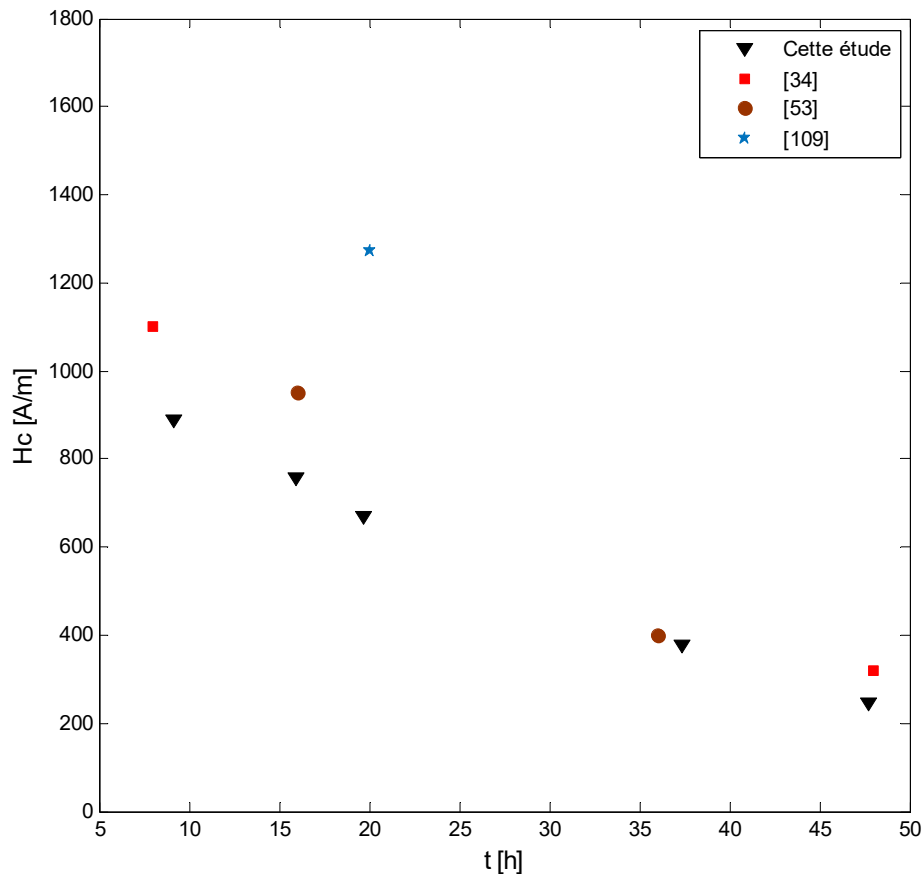


Figure 4.18 Comparaison de la coercivité

Dans une autre étude, Younes et al [109] ont examiné le composé $\text{Fe}_{80}\text{Ni}_{20}$, élaboré dans un broyeur planétaire de type Retsch PM400. La vitesse de rotation du plateau était de 300 rpm le rapport de masse bille/poudre était 10:1. Ils ont obtenu une taille des cristallites de 17 nm et une valeur de H_c de 16,01 Oe (= 1274 A/m) après 20 heures de broyage.

Comme le montrent ces figures, la méthode d'optimisation utilisée dans cette étude nous a permis de déterminer les meilleures conditions de broyage afin d'améliorer les propriétés des matériaux nanocristallins.

4.3. Modélisation par Programmation génétique

4.3.1 Introduction

Une fois que les paramètres d'entrée du modèle ont été déterminés, nous avons préparé une base de données d'apprentissage et de test. En utilisant ces données, le modèle prédictif établit une relation mathématique entre les paramètres d'entrée et de sortie.

4.3.2 Implémentation de la méthode MGGP

4.3.2.1 Base de données

Tout comme les réseaux de neurones, la méthode MGGP nécessite une base de données pour générer un modèle prédictif. La qualité de cette base de données influence l'apprentissage de l'algorithme. Nous avons utilisé les mêmes données collectées et utilisées dans la partie consacrée à la modélisation par les réseaux de neurones (paragraphe 4.1.2.1).

Sur l'ensemble de la base de données, 75% des données (soit 28 exemples) sont utilisées pour entraîner le modèle MGGP. Les 25% restants (soit 10 exemples) sont réservés aux tests. Cette base de données comprend les variables d'entrée (temps de broyage, composition chimique et vitesse de rotation du disque et des jarres) ainsi que les sorties qui correspondent aux deux propriétés à modéliser : la taille des cristallites et la coercivité.

4.3.2.2 Paramétrage du MGGP

Un ensemble de paramètres doit être ajusté afin de lancer le processus de recherche du modèle optimal. Ces algorithmes nécessitent également la connaissance des paramètres pertinents (appelés terminaux) et des fonctions, en plus de la base de données qui sert de terrain d'apprentissage pour affiner et proposer un modèle mathématique. Généralement, il n'existe pas de méthode précise pour trouver une configuration optimale de l'algorithme, car cela dépend de la nature du problème étudié. Dans notre cas, nous avons utilisé l'approche de l'essai et de l'erreur, qui nous permet d'ajuster les paramètres en fonction des

résultats obtenus à chaque fois. Le tableau 4.13 présente les principaux paramètres utilisés :

Tableau 4.13 : Paramètres pour la mise en œuvre de l'algorithme MGGP

Paramètres	Valeurs
Fonction set	+, -, ×, /, ^2, cos, tanh, exp
Fonction fitness	RMSE
Taille de population	200
Nombre de génération	200
Elitisme	0,02
Nombre maximal de gènes	4
Profondeur maximale de l'arbre	4
Taux de croisement	0.85
Taux de reproduction	0.10
Taux de mutation	0.05
Méthode de sélection	Tournoi

4.3.3 Résultats

Dans cette partie de notre travail, nous avons effectué plusieurs simulations. Pour chaque simulation, nous avons utilisé les critères d'évaluation des modèles définis dans le paragraphe 4.1.2.4 afin de sélectionner les meilleurs modèles.

Les modèles les plus performants sont présentés sous forme d'équations mathématiques (4.9) et (4.10). Ces équations décrivent respectivement la taille des cristallites D et la coercivité H_c en fonction des paramètres d'entrée du broyage.

$$D = 7.55 + 24.04 \tanh(x_2/x_3(x_4)^2) - 2051 \tanh(\cos(x_4)/(x_4)^2) + 935.5 \exp(\cos(x_4)) \tanh(x_4)/x_1 + 0.05184 x_1 \tanh(x_3)/x_4 \quad (4.9)$$

$$H_c = 1435 + 0.0005486((x_1)^2 + x_1 x_2)/\cos(2 x_2) - 211 \cos(x_4 - x_2 x_4) + 2.148 ((x_1)^2 + (x_2)^2)/(x_3 (x_2 + x_4)) - 6362 x_1 x_4/(x_1 + x_4)^2 \quad (4.10)$$

Où $x_1 = \Omega$ est la vitesse de rotation du disque, $x_2 = \omega$ est la vitesse de rotation des jarres, $x_3 = \%Ni$ est la composition chimique (pourcentage en poids de nickel), $x_4 = \text{test le temps de broyage}$

Tableau 4.14 : Données statistiques pour les phases de formation et de test

statistique	Modèle MGGP			
	<i>D</i>		<i>Hc</i>	
	Apprentissage	Test	Apprentissage	Test
<i>R</i>	0.999	0.998	0.989	0.984
<i>RMSE</i>	0.34	0.75	55.91	65.79
<i>MAE</i>	0.28	0.53	47.80	56.77

Les performances des modèles prédictifs de la taille des cristallites, ainsi que la coercivité développée par la MGGP, sont présentées dans le tableau 4.14, la figure 4.19 et la figure 4.20.

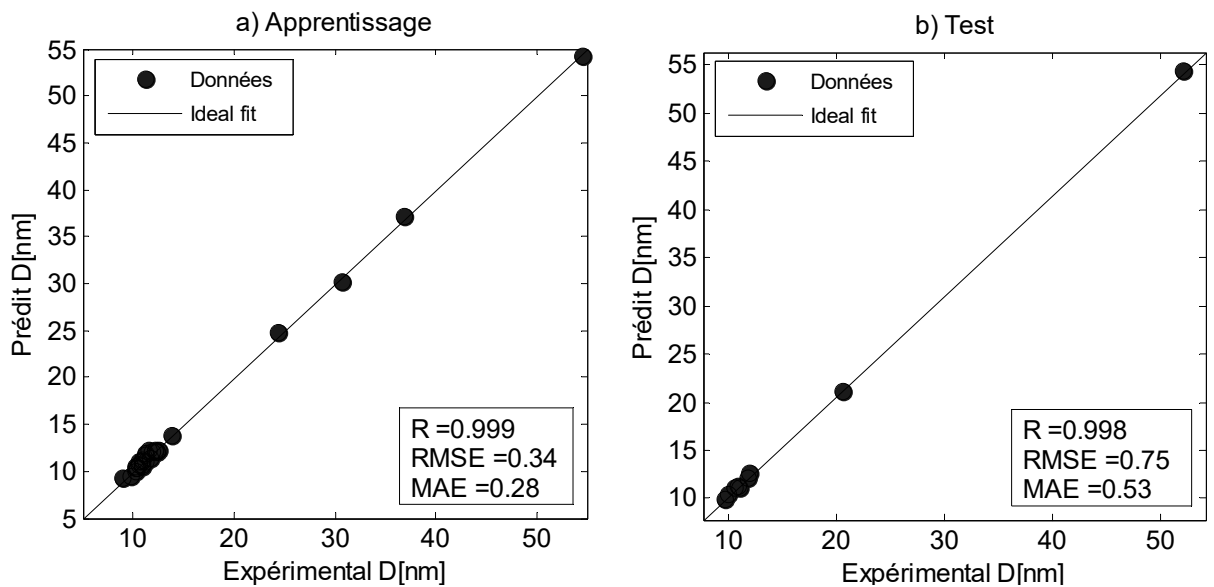


Figure 4.19 : Valeurs prédites de D.

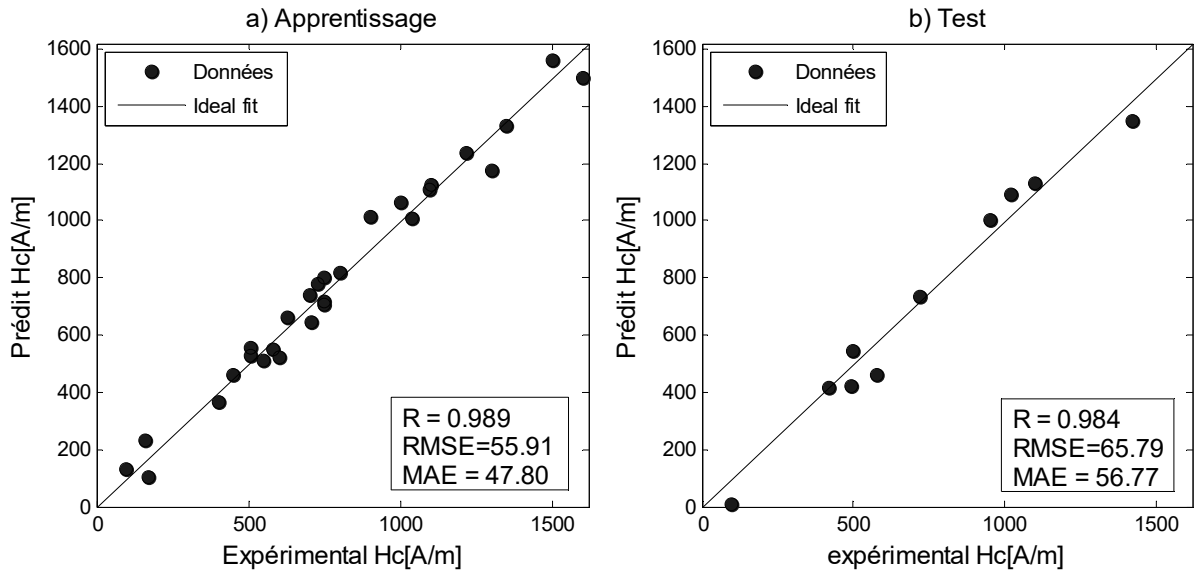


Figure 4.20 : Valeurs prédites de H_c .

Nous constatons que les deux modèles sont performants pour la prédiction de la taille des cristallites et la coercivité, en effet sur la figure 4.19 et figure 4.20, on constate clairement moins de dispersion autour de l'axe ($x=y$) qui représente théoriquement le modèle idéal. Ce qui exprime aussi une liaison fortement linéaire entre les résultats théorique de ces modèles et les résultats expérimentaux. En plus, la MGGP est capable de proposer des équations mathématiques concrètes contrairement aux réseaux neuronaux [74].

Les performances de ces deux modèles sont satisfaisantes, selon les coefficients de corrélation R , la racine carrée de l'erreur quadratique moyenne RMSE, ainsi que l'erreur absolue moyenne MAE.

L'analyse de sensibilité (SA) permet de déterminer la contribution d'une variable d'entrée en mesurant son impact et son influence sur la variable de sortie. Nous utilisons une procédure simple proposée par Garg et al [110]. Le pourcentage d'analyse de sensibilité de chaque paramètre d'entrée est déterminé à l'aide des formules suivantes :

$$SA_i = \frac{L_i}{\sum_{j=1}^n L_j} \quad (4.11)$$

$$\text{Avec } L_i = f_{max}(x_i) - f_{min}(x_i)$$

Où $f_{max}(x_i)$ et $f_{min}(x_i)$ sont, respectivement, le maximum et le minimum de la sortie prédite sur le $i^{\text{ème}}$ paramètre d'entrée, les autres variables étant égales à leurs valeurs moyennes.

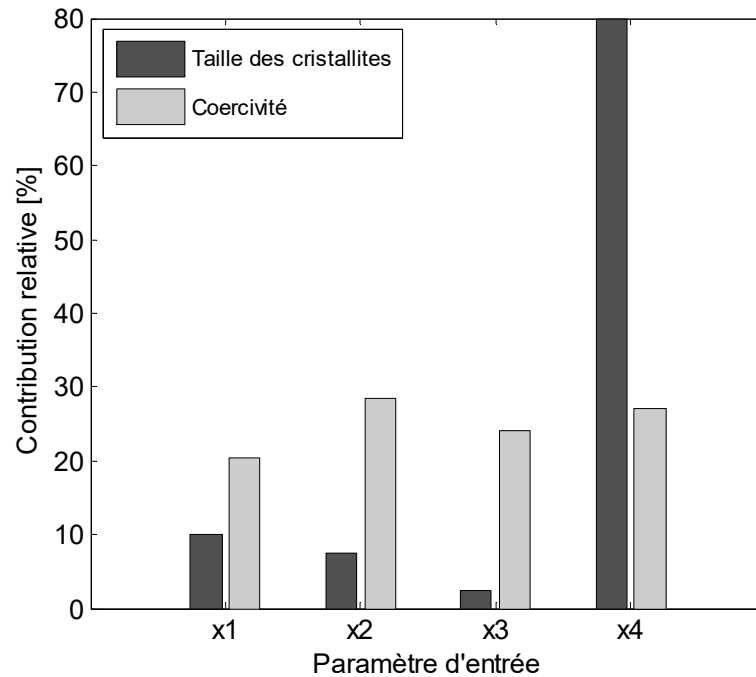


Figure 4.21 : Pourcentage de contribution de chaque variable d'entrée.

Le graphique à barres de la figure 4.21 illustre l'impact des variables d'entrée sur les variables de sortie, en utilisant les équations mathématiques fournies par le modèle MGGP (équations (4.9) et (4.10)). Ce graphique indique que parmi les quatre variables d'entrée, le temps de broyage a le plus grand impact sur D , suivi de la vitesse de rotation du disque et de la vitesse de rotation des jarres, tandis que le pourcentage de nickel a un impact faible. En ce qui concerne la variable de sortie H_c , les quatre variables d'entrée ont un impact relativement similaire, bien que la vitesse de rotation du disque et le pourcentage en poids de nickel aient légèrement plus d'effet sur H_c que les autres variables.

4.4 Conclusion

Dans cette partie du travail, nous avons développé un modèle en utilisant des algorithmes génétiques et un réseau neuronal artificiel (ANN) pour optimiser la taille des cristallites et la coercivité de la poudre de Fe-Ni nanostructurée obtenue par mécanosynthèse.

Tout d'abord, nous avons utilisé un modèle prédictif basé sur un réseau de neurones à trois couches, avec 10 neurones dans la couche cachée (4:10:1), des fonctions de transfert tansig-purelin et l'algorithme d'apprentissage de régularisation bayésienne. Ce modèle présente une capacité prédictive élevée.

Ensuite, nous avons utilisé le réseau neuronal le mieux entraîné comme fonction objectif pour l'algorithme génétique. Nous avons utilisé deux approches mono-objectifs (GA-ANN) et multi-objectifs (ANN-NSGA-II) pour déterminer les propriétés de broyage optimales.

Enfin, nous avons utilisé une procédure basée sur la méthode MGGP pour établir des relations fonctionnelles entre les propriétés recherchées (taille des cristallites et coercivité) et les conditions de broyage. Nous avons également utilisé une méthode d'analyse de sensibilité pour déterminer la contribution en pourcentage de chaque paramètre de broyage sur les paramètres de sortie, en utilisant les relations mathématiques fournies par le modèle MGGP.

CONCLUSION

Les matériaux nanostructurés à base de fer offrent un avantage technologique en raison de leurs propriétés structurales et magnétiques. La mécanosynthèse est l'une des techniques utilisées pour obtenir ces matériaux.

L'étude bibliographique réalisée sur ces matériaux a révélé que leurs propriétés dépendent principalement de nombreux paramètres qui influencent le processus de broyage dans un broyeur planétaire (vitesse du plateau, vitesse des jarres, temps de broyage, pourcentage d'alliages, etc.).

Notre première contribution dans cette thèse consiste à utiliser des réseaux de neurones artificiels pour modéliser les propriétés nanostructurales d'un alliage Fe-Ni produit par mécanosynthèse dans un broyeur à haute énergie. Dans ce but, nous avons construit et optimisé un modèle neuronal de type feed-forward afin de prédire l'évolution des paramètres de sortie, à savoir la taille des cristallites et le champ coercitif.

Le modèle neuronal que nous avons développé est composé de quatre neurones dans la couche d'entrée (vitesse du plateau, vitesse des jarres, pourcentage en poids de nickel et temps de broyage), d'une couche cachée avec un nombre de neurones optimisé lors du processus d'apprentissage, et d'un neurone dans la couche de sortie (taille des cristallites ou champ coercitif).

La fonction de transfert utilisée pour la couche cachée est la fonction tangente hyperbolique, et celle utilisée pour la couche de sortie est linéaire. L'algorithme d'optimisation choisi est celui de Levenberg-Marquardt, qui est réputé pour sa fiabilité et sa rapidité. Pour éviter le phénomène de sur-apprentissage, nous avons utilisé une approche de régularisation bayésienne, qui consiste à pénaliser la fonction de coût afin d'empêcher les poids de prendre des valeurs élevées. De plus, nous avons développé un algorithme itératif pour déterminer le

nombre optimal de neurones dans la couche cachée, ainsi que les fonctions de transfert appropriées. Nous avons obtenu une bonne concordance entre les valeurs expérimentales et les valeurs prédites par le modèle neuronal. En outre, le modèle neuronal utilisé a démontré une excellente capacité prédictive, avec des coefficients de corrélation approchant l'unité.

Notre deuxième contribution s'est basée sur le développement d'une méthodologie hybride d'optimisation, qui combine un réseau de neurones artificiels (ANN) avec un algorithme d'optimisation par les algorithmes génétiques. Ce dernier est un outil très puissant pour résoudre des problèmes d'optimisation. Cependant, l'utilisation efficace de tels algorithmes dépend du bon choix de différents paramètres de l'algorithme, tels que la fonction coût qui permet de classer les solutions appropriées, ainsi que le type de codage et la méthode de sélection. Le premier choix dépend du type de variable optimisée (continue ou discrète) et de la précision désirée. Lorsqu'il s'agit de choisir une méthode de sélection, il est important de prendre en compte le nombre d'objectifs définis. Pour un problème d'optimisation mono-objectif, la méthode de sélection utilisée est celle du tournoi. En revanche, pour un problème multi-objectif avec des objectifs contradictoires, la sélection par tournoi est utilisée, mais il est nécessaire d'opter pour un critère de sélection basé sur le principe de dominance selon Pareto. Un algorithme basé sur les algorithmes génétiques est chargé de proposer aux décideurs un ensemble de solutions potentielles pour résoudre le problème.

Le couplage des réseaux de neurones artificiels avec les algorithmes génétiques a été utilisé pour optimiser les conditions de broyage. Dans cette approche, une méthode d'optimisation mono-objectif est utilisée pour déterminer les valeurs optimales de la taille des cristallites et du champ coercitif. En parallèle, une méthode d'optimisation multi-objectif est utilisée pour obtenir l'ensemble des solutions optimales de Pareto. Le programme NSGA-II est utilisé pour les trois cas suivants : taille des cristallites vs temps de broyage, champ coercitif vs temps de broyage et taille des cristallites vs champ coercitif.

Les résultats de test ont montré clairement que le modèle a donné des résultats très satisfaisants, avec des coefficients de corrélation R très élevés, de 0.9983 et 0.9769 pour la taille des cristallites et la coercivité respectivement. Les

résultats obtenus sont optimisés par rapport à les constatations disponibles dans la littérature.

Notre troisième contribution consiste en l'application de la méthode de programmation génétique pour trouver des formules mathématiques reliant les paramètres de sortie aux variables d'entrée du broyage. Ensuite, une analyse de sensibilité a été réalisée pour identifier les paramètres d'entrée les plus influents sur le processus de broyage.

Cette thèse a montré plusieurs pistes de recherche qui feront l'objet de travaux futurs :

- Conception d'un modèle plus général reliant l'effet de tous les paramètres de broyage sur des propriétés nanostructures.
- Valider expérimentalement ce qu'on a obtenu par notre étude théorique.
- Modélisation et optimisation des propriétés nanostructures de différents alliages (binaire, ternaire,...).
- Combinaison des techniques utilisées avec d'autres méthodes d'intelligence artificielle.

ANNEXE A

LISTE DES SYMBOLES ET DES ABREVIATIONS

ANN :	Réseaux de neurones artificiels
BR :	Régularisation bayésienne
GA :	Algorithme génétique
GP :	Programmation génétique
LM :	Algorithme de Levenberg-Marquardt.
MAE :	Erreur absolue moyenne
MGGP :	Programmation génétique multi-gène
MLP :	Perceptron multi couches
MSE :	Erreur quadratique moyenne ou fonction de coût
MSW :	Terme de régularisation
MSE_{reg} :	Fonction du coût régulée
NSGA-II	Algorithme génétique de tri non dominé
RMSE :	Racine carrée de l'erreur quadratique moyenne
D :	Taille des cristallites
E :	Fonction d'erreur
F_i :	Fronts de Pareto
F :	Critères de décision
H :	Champ magnétique
H_c :	Champ coercitif
IW :	Poids de la couche cachée
LW :	Poids de la couche de sortie
Mr :	Aimantation rémanente

M_s :	Aimantation à saturation
N :	Taille de la population
O :	Sortie réelle.
P_t :	Population de parents
P_0 :	Population initiale
Q_t :	Population d'enfants
R :	Coefficient de corrélation
X_i :	Sorties mesurées
Y_i :	Sorties prédites
\bar{X} :	Moyenne de sorties mesurées
\bar{Y} :	Moyenne de sorties prédites
W :	Poids synaptique
Z_j :	Sortie de la couche cachée
b_1 :	Biais de la couche cachée
b_2 :	Biais de la couche de sortie
d_{ij}	Distance de niche
ch :	Chromosome
f :	Fonction fitness ou objectif
$logsig$:	Sigmoïde logarithmique
m_i :	Compteur de niche
n :	Nombre d'échantillons
pc :	Probabilité de croisement
pm :	Probabilité de mutation
$purlin$:	Fonction d'activation linéaire
sh :	Fonction de sharing
t :	Temps de broyage
$tansig$:	Tangente hyperbolique
u :	Signal d'entrée

v :	Signal de sortie
x :	Vecteur d'entrée
Ω :	Vitesse de rotation du plateau
α :	Paramètre de pondération
σ_{shar} :	Rayon de niche:
ω :	Vitesse de rotation de la jarre.
ω_i :	Pondérations des objectifs
$\%Ni$:	Pourcentage en poids de nickel
\prec_n :	Opérateur de comparaison de densité
$i_{distance}$:	Distance de crowding
i_{rang} :	Rang d'un individu

ANNEXE B

PARAMETRES D'ENTRES ET DE SORTIES DE BROYAGE MECANIQUE

Ω (rpm)	ω (rpm)	%Ni	t (h)	D (nm)	Hc (A /m)	References
212	200	20	36	12.31	550	[53]
212	50	20	36	11.90	700	
212	100	20	36	11.70	720	
212	400	20	36	11.50	400	
300	200	20	36	11.00	710	
300	400	20	36	10.90	580	
300	50	20	36	11.00	1040	
300	100	20	36	10.60	900	
424	200	20	36	11.00	750	
424	400	20	36	10.60	450	
424	50	20	36	10.20	1350	
424	100	20	36	10.00	1420	
212	50	10	36	12.39	750	
212	100	10	36	12.20	730	
212	200	10	36	12.60	580	
212	400	10	36	12.00	420	
300	50	10	36	11.25	1100	
300	100	10	36	11.00	1020	
300	200	10	36	11.70	750	
300	400	10	36	11.20	600	
424	50	10	36	10.40	1500	
424	100	10	36	10.20	1600	
424	200	10	36	11.00	800	
424	400	10	36	10.60	500	

400	800	10	0	54.60	1215	[50]
400	800	10	8	36.96	1300	
400	800	10	16	24.48	1095	
400	800	10	24	13.80	630	
400	800	10	36	11.28	505	
400	800	10	72	9.84	160	
400	800	10	96	10.32	170	
400	800	20	0	52.20	1100	
400	800	20	8	30.72	1000	
400	800	20	16	20.64	950	
400	800	20	24	12.24	510	
400	800	20	36	11.00	495	
400	800	20	72	9.00	95	
400	800	20	96	9.60	95	

REFERENCES

1. Charra, F., Gota-Goldmann, S. et Warlimont, H., "Nanostructured materials", Springer handbook of materials data, (2018), 1041-1080.
2. Kushwaha, A.K., John, M., Misra, M. et al., "Nanocrystalline Materials: Synthesis, Characterization, Properties, and Applications", Crystals, V. 11, n° 11, (2021), 1317.
3. Herzer, G., "Nanocrystalline soft magnetic alloys", Handbook of magnetic materials, V. 10, (1997), 415-462.
4. Herzer, G., "Soft magnetic nanocrystalline materials", Scripta Metallurgica et Materialia, V. 33, n° 10-11, (1995), 1741-1756.
5. Shull, R.D., "Nanocrystalline and nanocomposite magnetic materials and their applications", Journal of Iron and Steel Research International, V. 14, n° 4, (2007), 69-74.
6. Gehrmann, B., "Nickel-iron alloys with special soft magnetic properties for specific applications", Journal of magnetism and magnetic materials, V. 290, (2005), 1419-1422.
7. Mchenry, M.E., Willard, M.A., et Laughlin, D.E., "Amorphous and nanocrystalline materials for applications as soft magnets", Progress in materials Science, V. 44, n° 4, (1999), 291-433.
8. Pękała, M., Oleszak, D., Jartych, E. et al., "Structural and magnetic study of crystalline Fe₈₀Ni₂₀ alloys with nanometer-sized grains", Journal of non-crystalline solids, V. 250, (1999), 757-761.
9. Koohkan, R., Sharafi, S., Shokrollahi, H. et al., "Preparation of nanocrystalline Fe-Ni powders by mechanical alloying used in soft magnetic

- composites”, *Journal of Magnetism and Magnetic Materials*, V. 320, n° 6, (2008), 1089-1094.
10. Du, S.W., Ramanujan, R.V., “Mechanical alloying of Fe–Ni based nanostructured magnetic materials”, *Journal of magnetism and magnetic materials*, V. 292, (2005), 286-298.
 11. Shokrollahi, H., “The magnetic and structural properties of the most important alloys of iron produced by mechanical alloying”, *Materials & Design*, V. 30, n° 9, (2009), 3374-3387.
 12. Suryanarayana, C., “Mechanical alloying and milling”, *Progress in materials science*, V. 46, n° 1-2, (2001), 1-184.
 13. Taghvaei, A.H., Ebrahimi, A., Ghaffari, M. et al., “Investigating the magnetic properties of soft magnetic composites based on mechanically alloyed nanocrystalline Fe–5 wt% Ni powders”, *Journal of Magnetism and Magnetic Materials*, V. 323, n° 1, (2011), 149-155.
 14. Valderruten, J.F., Alcazar, GA P., et Greneche, J.M., “Study of Fe–Ni alloys produced by mechanical alloying”, *Physica B: Condensed Matter*, V. 384, n° 1-2, (2006), 316-318.
 15. Zhu, L.H., Huang, Q.W., et Zhao, H.F., “Effect of nickel content and milling parameters on martensitic transformation of Fe–Ni during mechanical alloying”, *Scripta materialia*, V. 51, n° 6, (2004), p. 527-531.
 16. Zhu, L.H., Huang, Q.W., “Study on martensitic transformation of mechanically alloyed nanocrystalline Fe–Ni”, *Materials Letters*, V. 57, n° 24-25, (2003), 4070-4073.
 17. Vitta, S., Khuntia, A., Ravikumar, G. et al., “Electrical and magnetic properties of nanocrystalline Fe_{100-x}Ni_x alloys”, *Journal of magnetism and magnetic materials*, V. 320, n° 3-4, (2008), 182-189.
 18. Suryanarayana, C.J.I.M.R., “Nanocrystalline materials”, *International materials reviews*, V. 40, n° 2, (1995), 41-64.

19. Lu, K., "Nanocrystalline metals crystallized from amorphous solids: nanocrystallization, structure, and properties", *Materials Science and Engineering: R: Reports*, V. 16, n° 4, (1996), 161-221.
20. Tjong, S.C., Chen, H., "Nanocrystalline materials and coatings", *Materials Science and Engineering: R: Reports*, V. 45, n° 1-2, (2004), 1-88.
21. Ibaseta, N., "Etude expérimentale et modélisation de l'émission d'aérosols ultrafins lors du déversement de poudres nanostructurées", thèse doctorat, Institut National Polytechnique de Toulouse, (2007).
22. Siegel, R.W. et Fougere, G.E., "Mechanical properties of nanophase materials. In: Nanophase materials: synthesis—properties—applications", Dordrecht : Springer Netherlands, (1994), 233-261.
23. Bergheul, S., Tafat, H., et Azzaz, M., "Formation and magnetic properties of nanocrystalline Fe₆₀ Co₄₀ alloys produced by mechanical alloying", *Journal of materials engineering and performance*, V. 15, (2006), 349-351.
24. Siegel, R.W., "Nanostructured materials-mind over matter", *Nanostructured Materials*, V. 3, n° 1-6, (1993), 1-18.
25. Boukherroub, N., "Elaboration et étude des propriétés structurales et hyperfines des nanomatériaux Fe_{100-x}Al_x", Thèse de doctorat, Université M'hamed Bougara de Boumerdès, (2010).
26. Fongang, B., "Etude par simulation numérique des propriétés structurales et magnétiques des systèmes ioniques nanostructurés: description des joints de grains", Thèse de doctorat, Le Mans, (2009).
27. Gleiter, H., "Nanostructured materials: basic concepts and microstructure" *Acta materialia*, V. 48, n° 1, (2000), 1-29.
28. Ourihane, H., "Etude des propriétés structurales microstructurale et magnétiques des nanomatériaux ternaires (Fe₈₀Ni₂₀)_{100-x}Y_x (Y= Co, Si)", Thèse de doctorat, Université M'hamed Bougara de Boumerdès, (2023).

29. Benjamin, J.S., "Dispersion strengthened superalloys by mechanical alloying Metallurgical transactions", V. 1, (1970), 2943-2951.
30. Koch, C.C., Cavin, O.B., Mckamey, C.G. et al., "Preparation of amorphous Ni₆₀Nb₄₀ by mechanical alloying", Applied Physics Letters, V. 43, n° 11, (1983), 1017-1019.
31. Hellstern, E. et Schultz, L., "Glass formation in mechanically alloyed transition metal-titanium alloys", Materials Science and Engineering, V. 93, (1987), 213-216.
32. Schwarz, R.B., Petrich, R.R., et Saw, C.K., "The synthesis of amorphous Ni-Ti alloy powders by mechanical alloying", Journal of non-crystalline solids, V. 76, n° 2-3, (1985), 281-302.
33. Fecht, H.J., Hellstern, E., Fu, Z. et al., "Nanocrystalline metals prepared by high-energy ball mill", Metallurgical Transactions A, V. 21, (1990), 2333-2337.
34. Guittoum, A., "Etude des propriétés structurales, microstructurales et magnétiques des nanoparticules FeNi", Thèse de doctorat, Université Ferhat Abbas de Setif, (2008).
35. Johnson, W.C., Lee, J.K., et Shiflet, G.J., "Thermodynamic treatment of cyclic amorphization during ball mill", Actamaterialia, V. 54, n° 19, (2006), 5123-5133.
36. Koch, C.C. et Whittenberger, J.D., "Mechanical milling/alloying of intermetallics", Intermetallics, V. 4, n° 5, (1996), 339-355.
37. Suryanarayana, C., "Mechanical alloying and mill", Progress in materials science", V. 46, n° 1-2, (2001), 1-184.
38. Suryanarayana, C., Al-Joubori, A.A., et Wang, Z., "Nanostructured materials and nanocomposites by mechanical alloying: an overview", Metals and Materials International, V. 28, n° 1, (2022), 41-53.

39. Gaffet, E. et Le Caër, G., "Mechanical processing for nanomaterials", In : Encyclopedia of nanoscience and nanotechnology, Stevenson Ranch, Calif, USA: American scientific publishers, V. 5, N°. 129, (2004), 91-129.
40. Spârchez, Z. et Chicinas, I., "Proceeding of European Congress on Powder Metallurgy", EuroPM2005, Prague, (2005), 99-104.
41. Sparchez, Z., Chicinas, I., Isnard, O. et al., "Mechanical alloying of Ni₃Fe in the presence of Ni₃Fe nanocrystalline germs", Journal of alloys and compounds, V. 434, (2007), 485-488.
42. Kwon, Y.S., Gerasimov, K.B., et Yoon, S.K., "Ball temperatures during mechanical alloying in planetary mills", Journal of alloys and compounds, V. 346, n° 1-2, (2002), 276-281.
43. Jamil, M.T. "Caractérisation et investigation des effets de la température d'utilisation sur les propriétés électromagnétiques des aciers magnétiques doux: application à la roue polaire de la machine à griffes", Thèse de doctorat, Université de Lille, (2021).
44. Brissonneau, P., "Magnétisme et matériaux magnétiques pour l'électrotechnique", Hermes, (1997).
45. Miouat, A., "Identification-optimisation paramétrique et quantification des pertes dans les matériaux ferromagnétiques doux", Mémoire Magistère, Université El Hadj Lakhder de Batna, (2005).
46. Benabou, A., "Contribution à la caractérisation et à la modélisation de matériaux magnétiques en vue d'une implantation dans un code de calcul de champ", Thèse de doctorat, Lille 1, (2002).
47. Kolhatkar, A.G., Jamison, A.C., Litvinov, D. et al., "Tuning the magnetic properties of nanoparticles", International journal of molecular sciences, V. 14, n° 8, (2013), 15977-16009.

48. Askeland, D.R., Fulay, P.P., et Wright, W.J., "The science and engineering of materials", 6th ed, Cengage learning, (2010).
49. Djekoun, A., Boudinar, N., Chebli, A. et al., "Structure and magnetic properties of Fe-rich nanostructured Fe_{100-x}Ni_x powders obtained by mechanical alloy", Physics Procedia, V. 2, n° 3, (2009), 693-700.
50. Hamzaoui, R, Elkedim, O, Fenineche, N. et al., "Structure and magnetic properties of nanocrystalline mechanically alloyed Fe–10% Ni and Fe–20%", Materials Science and Engineering: A, V. 360, n° 1-2, (2003), 299-305.
51. Gheisari, K., Javadpour, S., OH, J.T. et al., "The effect of milling speed on the structural properties of mechanically alloyed Fe–45% Ni powders", Journal of Alloys and Compounds, V. 472, n° 1-2, (2009), 416-420.
52. Jartych, E., Żurawicz, J.K., Oleszak, D. et al., "X-ray diffraction, magnetization and Mössbauer studies of nanocrystalline Fe–Ni alloys prepared by low-and high-energy ball mill", Journal of magnetism and magnetic materials, V. 208, n° 3, (2000), 221-230.
53. Hamzaoui, R., Elkedim, O. et Gaffet, E., "Milling conditions effect on structure and magnetic properties of mechanically alloyed Fe–10% Ni and Fe–20% Ni alloys", Materials Science and Engineering: A, V. 381, n° 1-2, (2004), 363-371.
54. Kuhrt, C. et Schultz, L., "Phase formation and martensitic transformation in mechanically alloyed nanocrystalline Fe–Ni", Journal of applied physics, V. 73, n° 4, (1993), 1975-1980.
55. Torabi, A., Babaheydari, R.M., Akbari, G.H. et al., "Optimizing of micro-hardness of nanostructured Cu–Cr solid solution produced by mechanical alloying using ANN and genetic algorithm", SN Applied Sciences, V. 2, (2020), 1-9.
56. Zeraati, M., Arshadizadeh, R., Chauhan, N.P.S. et al., "Genetic algorithm optimization of magnetic properties of Fe-Co-Ni nanostructure alloys prepared by the mechanical alloying by using multi-objective artificial neural

- networks for the core of transformer”, *Materials Today Communications*, V. 28, (2021), 102653.
57. Mahdavi Jafari, M., Soroushian, S. et Khayati, G.R., “Hardness optimization for Al6061-MWCNT nanocomposite prepared by mechanical alloying using artificial neural networks and genetic algorithm”, *Journal of Ultrafine Grained and Nanostructured Materials*, V. 50, n° 1, (2017), 23-32.
 58. Yazdanmehr, M., Anijdan, SH M. et Bahrami, A., “Using GA–ANN algorithm to optimize soft magnetic properties of nanocrystalline mechanically alloyed Fe–Si powders”, *Computational Materials Science*, V. 44, n° 4, (2009), 1218-1221.
 59. Hamzaçebi, C. et Kutay, F., “A heuristic approach for finding the global minimum: Adaptive random search technique”, *Applied Mathematics and Computation*, V. 173, n° 2, (2006), 1323-1333.
 60. Koza, J., “Genetic programming: on the programming of computers by means of natural selection”, MIT Press, (1992).
 61. Madeline, B., “Algorithmes évolutionnaires et résolution de problèmes de satisfaction de contraintes en domaines finis”, Thèse de doctorat, Université Nice Sophia Antipolis, (2002).
 62. Collette, Y et Siarry, P., “Optimisation multiobjectif”, Editions Eyrolles, (2002).
 63. Holland, J.H., “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence”, MIT press, (1992).
 64. Golberg, D.E., “Genetic algorithms in search, optimization, and machine learn”, Addison Wesley, Reading, (1989).
 65. Haupt, R.L. et Haupt, S.E., “Practical genetic algorithms”, John Wiley & Sons, (2004).
 66. Sivanandam, S.N. et Deepa, S.N., “Introduction to genetic algorithms”, Springer Science & Business Media, (2007).

67. Michalewicz, Z., and Zbigniew M., "Evolutionary programming and genetic programming", Genetic Algorithms+ Data Structures= Evolution Programs, (1996), 283-287.
68. Man, K.F., Tang, K.S., et Kwong, S., "Genetic algorithms: concepts and applications [in engineering design]", IEEE transactions on Industrial Electronics, V. 43, n° 5, (1996), 519-534.
69. Wu, C.J. et Lin, G.Y., "Design of fuzzy logic controllers using genetic algorithms", IEEE SMC'99 Conference Proceedings, V.6, (1999), 104-109.
70. Zbigniew, M., "Genetic algorithms + data structures= evolution programs", Computational statistics, V. 24, (1996), 372-373.
71. Goldberg, D.E. et Corruble, V., "Algorithmes génétiques: exploration, optimisation et apprentissage automatique", Ed. Addison-Wesley France, (1994).
72. Goldberg, D.E., Richardson, J. et al., "Genetic algorithms with sharing for multimodal function optimization" Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms. V. 4149, (1987), 414-425.
73. Gandomi, A.H. et Alavi, A.H., "A new multi-gene genetic programming approach to nonlinear system model", Part I: materials and structural engineering problems. Neural Computing and Applications, V. 21, (2012), 171-187.
74. Bayazidi, A. M., Wang, G.G., Bolandi, H. et al., "Multigene genetic programming for estimation of elastic modulus of concrete", Mathematical Problems in Engineering, V. 2014. (2014), 1-10.
75. Pareto, V., "Cours d'Economie Politique", Droz, Genève, V.1, (1964).
76. Fonseca, C.M., Fleming, P.J., "Genetic algorithms for multiobjective optimization: formulation discussion and generalization", In: ICGA, V.93, (1993), 416-423.

77. Horn, J., Nafpliotis, N., et Goldberg, D.E., "A niched Pareto genetic algorithm for multiobjective optimization", In: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, IEEE, (1994), 82-87.
78. Srinivas, N. et Deb, K., "Multiobjective optimization using nondominated sorting in genetic algorithms", Evolutionary computation, V. 2, n° 3, (1994), 221-248.
79. Deb, K., Agrawal, S., Pratap, A. et al., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II", In: Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, (2000) Proceedings 6, Springer Berlin Heidelberg, (2000), 849-858.
80. Schaffer, J.D., "Multiple objective optimization with vector evaluated genetic algorithms", In : Proceedings of the first international conference on genetic algorithms and their applications, Psychology Press, (2014), 93-100.
81. Si Moussa, C., "Modélisation des équilibres de phases à pressions élevées par les réseaux de neurones artificiels", Thèse de doctorat, Ecole Nationale Polytechnique d'Alger, (2010).
82. Laidi, M., "Etude et modélisation d'un réfrigérateur solaire", Thèse de doctorat, Université Yahia Fares de Médéa, (2013).
83. Beale, M.H., Hagan, M.T. et Demuth, H.B., "Neural network toolbox", User's Guide, MathWorks, V. 2, (2010), 77-81.
84. Rafinia, A. et Moshtagh, J., "A new approach to fault location in three-phase underground distribution system using combination of wavelet analysis with ANN and FLS", International Journal of Electrical Power & Energy Systems, V. 55, (2014), 261-274.
85. Melzi, N., Khaouane, L., Hanini, S. et al., "Optimization methodology of artificial neural network models for predicting molecular diffusion coefficients

- for polar and non-polar binary gases”, *Journal of Applied Mechanics and Technical Physics*, V. 61, (2020), 207-216.
86. Hornik, K., Stinchcombe, M. et White, H., “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks”, *Neural networks*, V. 3, n° 5, (1990), 551-560.
 87. Hagan, M.T., Demuth, H.B. et Beale, M., “*Neural Network Design*”, Boston, MA, USA: PWS Publishing Company, (1996).
 88. Samarasinghe, S., “*Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*”, Crc Press, (2016).
 89. Behrang, M.A., Assareh, E., Ghanbarzadeh, A. et al., “The potential of different artificial neural network (ANN) techniques in daily global solar radiation modeling based on meteorological data”, *Solar Energy*, V. 84, n° 8, (2010), 1468-1480.
 90. Curry, B. et Morgan, P.H., “Model selection in neural networks: some difficulties”, *European Journal of Operational Research*, V. 170, n° 2, (2006), 567-577.
 91. Plumb, A.P., Rowe, R.C., York, et al., “Optimisation of the predictive ability of artificial neural network (ANN) models: a comparison of three ANN programs and four classes of training algorithm”, *European Journal of Pharmaceutical Sciences*, V. 25, n° 4-5, (2005), 395-405.
 92. Chaturvedi, D.K., “Artificial neural network and supervised learn”, *Soft Computing: Techniques and its Applications in Electrical Engineering*, (2008), 23-50.
 93. Hagan, M.T. et Menhaj, M.B., “Training feedforward networks with the Marquardt algorithm”, *IEEE transactions on Neural Networks*, V. 5, n° 6, (1994), 989-993.

94. Henrique, H.M., Lima, E.L. et Seborg, D.E., "Model structure determination in neural network models", *Chemical Engineering Science*, V. 55, n° 22, (2000), 5457-5469.
95. Plumb, A.P., Rowe, R.C., York, P. et al., "The effect of experimental design on the modeling of a tablet coating formulation using artificial neural networks", *European journal of pharmaceutical sciences*, V. 16, n° 4-5, (2002), 281-288.
96. Mellit, A., Kalogirou, S. A., Hontoria, L. et al., "Artificial intelligence techniques for sizing photovoltaic systems: A review", *Renewable and Sustainable Energy Reviews*, V. 13, n° 2, (2009), 406-419.
97. Bourquin, J., Schmidli, H., Van Hoogevest, P. et al., "Application of artificial neural networks (ANN) in the development of solid dosage forms", *Pharmaceutical development and technology*, V. 2, n° 2, (1997), 111-121.
98. Dreyfus, G. et al., "Réseaux de Neurones Méthodologie et Applications", Dreyfus, Editions Eyrolles, Paris, (2002).
99. Mackay, D.J.C., "Bayesian interpolation", *Neural Computation*, V. 4, n° 3, (1992), 415-447.
100. Bouhila Ep Khodja, Z., "Contribution à l'évaluation des ETM dans les retombées atmosphériques par l'utilisation d'une technique d'analyse nucléaire", Thèse de doctorat, Ecole Nationale Polytechnique d'Alger, (2016).
101. Pradhan, B., "Landslide susceptibility mapping of a catchment area using frequency ratio, fuzzy logic and multivariate logistic regression approaches", *Journal of the Indian Society of Remote Sensing*, V. 38, (2010), 301-320.
102. Cybenko, G., "Approximation by superpositions of a sigmoidal function. Mathematics of control", *signals and systems*, V. 2, n° 4, (1989), 303-314.
103. Vaferi, B., "Application of artificial neural networks for accurate prediction of thermal and rheological properties of nanofluids", *Deterministic Artificial Intelligence*, IntechOpen, (2020), 69-98.

104. Vives, S., Gaffet, E. et Meunier, C., "X-ray diffraction line profile analysis of iron ball milled powders", *Materials Science and Engineering: A*, V. 366, n° 2, (2004), 229-238.
105. Hamzaoui, R. et Elkedim, O., "Magnetic properties of nanocrystalline Fe-10% Ni alloy obtained by planetary ball mills", *Journal of Alloys and Compounds*, V. 573, (2013), 157-162.
106. Herzer, G., "Nanocrystalline soft magnetic alloys", *Handbook of magnetic materials*, V. 10, (1997), 415-462.
107. Badreddine, B., "Conception d'un réseau WLAN par algorithmes génétiques hiérarchisés", Thèse de doctorat, Université Abderrahmane Mira de Béjaia, (2007).
108. Ismail, I. et al., "Milling time and BPR dependence on permeability and losses of Ni_{0.5}Zn_{0.5}Fe₂O₄ synthesized via mechanical alloying process", *Journal of magnetism and magnetic materials*, V. 323, n° 11, (2011), 1470-1476.
109. Younes, A. et al., "Magnetic properties and Structural characterization of nanocrystalline Fe-20% A (Ni, Co and Si) alloys powders synthesized by mechanical alloying process", *ENP Engineering Science Journal*, V. 3, n° 1, (2023), 3-6.
110. Garg, A., Tai, K., Vijayaraghavan, V. et al., "Mathematical modelling of burr height of the drilling process using a statistical-based multi-gene genetic programming approach", *The International Journal of Advanced Manufacturing Technology*, V. 73, (2014), 113-126.