الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي
جامعة وهران للعلوم و التكنولوجيا محمد بوضياف

THÈSE

En vue de l'obtention du Diplôme de Doctorat

Présentée par :

Si Larbi Samiya

Intitulé

Amélioration de l'Apprentissage du Réseau ANFIS par les Algorithmes Evolutionnaires

Faculté	*:* Mathématiques Et Informatique
Département	*:* Informatique
Domaine	*:* Informatique
Filière	*:* Informatique
Intitulé de la Forma-	*: Reconnaissance des Formes et Intelligence Artificielle RFIA*

Devant le Jury Composé de :

Membres de Jury	*Grade*	*Qualité*	*Domiciliation*
Chouraqui Samira	*Pr*	*Président*	*USTO-MB*
Tlemsani Redouane	*MCA*	*Encadreur*	*USTO-MB*
Benamrane Nacera	*Pr*	*Examinateur*	*USTO-MB*
Adjoudj Réda	*Pr*	*Examinateur*	*Université Djilali Liabes de Sidi Bel Abbès*
Boukli-Hacene Sofiane	*Pr*	*Examinateur*	*Université Djilali Liabes de Sidi Bel Abbès*

Année Universitaire : 2020-2021

REMERCIEMENT

Je voudrais exprimer ma profonde gratitude envers dieu tout puissant qui grâce à son aide, j'ai pu finir ce modeste travail.

Je voudrais remercier Dr. Tlemsani Redouane, mon directeur pour m'avoir encadré dans ce travail de thèse, pour la confiance qu'il m'a témoigné, pour ses précieux critiques et conseils qu'il veuille trouver dans ce mémoire ma profonde gratitude et mon grand respect.

Je remercie madame Chouraqui Samira, Professeur à l'université des Sciences et de la Technologie d'Oran USTO-MB, de m'avoir fait l'honneur de présider mon jury de soutenance.

Je remercie madame Benamrane Nacera, Professeur à l'université des Sciences et de la Technologie d'Oran USTO-MB, pour avoir accepté de juger mon travail. C'est pour moi un plaisir de vous siéger dans mon jury de soutenance.

Mes remerciements les plus intenses sont adressés à monsieur Boukli-Hacene Sofiane, Professeur à l'niversité Djilali Liabes de Sidi Bel Abbès, à monsieur Adjoudj Réda, Professeurà l'niversité Djilali Liabes de Sidi Bel Abbès, pour l'honneur qu'ils mon fait en acceptant de faire partie de ce jury et d'examiner ce travail.

Je remercie Pr. Benyettou Abdelkader pour la proposition de mon sujet, j'ai eu le privilège de bénéficier de ses compétences scientifiques.

Et bien sûr, ceux sans qui je ne serais rien mes parents, Merci Maman et Papa pour votre soutien et votre confiance.

Je tiens également à exprimer mes plus vifs remerciements à mon mari, pour son soutien moral indéfectible, ses encouragements depuis de longues années et sans lesquels ce travail n'aurait pas été possible, qu'ils trouvent dans cette thèse ma profonde gratitude.

Je tiens aussi à remercier Dr. Bendahmane Abderrahmane pour son aide si précieuse.

الملخص

تقدم هذه الرسالة نهجا تطوريا لتعلم نظام الاستدلال الغامض القائم على الشبكة التكييفية ا ذيص . الأعمال السابقة مبنية على الانتشار الخلفي للتدرج (جض)، هذه الخوارزمية تتقارب ببطء شديد وتلتصق بالحدود المحلية السيئة. في هذه الأطروحة، نقوم بتطبيق خوارزميات جينية (جا) وأسراب جسيمات (عصو) لتحسين العناصر والبارامترات الناتجة عن قواعد غامضة لنظام المعلومات المسبقة عن بعد ا ذيص . أولا، تم استخدام خوارزمية التجميع الطرحي لتحديد البنية المثالية لشبكة ا ذيص، أي أفضل تقسيم لمساحة الإدخال؛ ثم تعديل البارامترات السابقة واللاحقة للقواعد الغامضة بحيث يتم تقليل وظيفة موضوعية محددة إلى الحد الأدنى. تبدأ عملية التطور بالتوليد العشوائي لمجموعة أولية، كل مرشح يتم تمثيله بمتجه. ويستند طول هذه الأخيرة إلى عدد البارامترات السابقة واللاحقة في نموذج نظام المعلومات الإدارية المتكامل. ثم جعل كل السكان يتحسنون تدريجيا حتى الوصول إلى الحد الاقصى لعدد التكرار. تم تطبيق النهج المقترح للاعتراف بالصوتيات من قاعدة بيانات ةيبية والاعتراف بالمتكلم من قاعدة بيانات ثحاص. أظهرت النتائج التي تم الحصول عليها بواسطة النماذج الهجينة اجا ذيص و عصوا ذيص تحسنًا في الدقة مقارنةً بـ ا ذيص الكلاسيكي المماثل استنادًا إلى الانتشار الخلفي للتدرج.

الكلمات الرئيسية : نظام الاستدلال العصبي الضبابي التكيفي ، تحسين سرب الجسيمات ، الخوارزمية الجينية ـ نظام الاستدلال العصبي الضبابي المتكيف ، تحسين سرب الجسيمات ـ نظام استدلال عصبي غامض متكيف ، التجميع الطرحي ، المتحدث ، الصوت.

Résumé

Une approche évolutionnaire pour l'apprentissage du système d'inférence floue basé sur un réseau adaptatif (ANFIS) a fait l'objet de cette thèse. Les travaux précédents sont basés sur la descente du gradient (GD), cet algorithme converge très lentement et se bloque à de mauvais minima locaux. Dans cette thèse, on applique les algorithmes génétiques (AG) et les essaims de particules (PSO) pour optimiser les paramètres des antécédents et conséquents des règles floues de l'ANFIS. En premier, l'algorithme de regroupement soustractif a été utilisé pour la détermination de la structure optimale du réseau ANFIS, c'est-à-dire le meilleur partitionnement de l'espace d'entrée, ensuite, l'ajustement des paramètres antécédents et conséquents des règles floues afin qu'une fonction objectif spécifiée soit minimisée. Le processus évolutionnaire commence par la génération aléatoire d'une population initiale, chaque solution candidate est représentée par un vecteur. La longueur de ce dernier est basée sur le nombre de paramètres des antécédents et conséquents dans le modèle ANFIS. Ensuite, la population entière a été amenée à s'améliorer progressivement jusqu'à que le nombre maximal d'itération est atteint. L'approche proposée a été appliquée pour la reconnaissance des phonèmes de la base de données TIMIT et la reconnaissance du locuteur de la base de données CHAINS. Les résultats obtenus par les modèles hybrides AG-ANFIS et PSO-ANFIS ont montré une amélioration de la précision par rapport à un ANFIS classique similaire basé sur la rétro-propagation du gradient.

Mots clés : ANFIS, Regroupement Soustractif, AG, PSO, AG-ANFIS, PSO-ANFIS, locuteur, phonème.

Abstract

This thesis presents an evolutionary approach for the learning of the Adaptive Network Based Fuzzy Inference System (ANFIS). The previous works are based on the descent of the gradient (GD), this algorithm converges very slowly and stuck into bad local minima. In this thesis, we apply genetic algorithms (GA) and particle swarms (PSO) to optimize the antecedents and consequent parameters of ANFIS fuzzy rules. First, the subtractive clustering algorithm was used to determine the optimal structure of the ANFIS network, i.e., the best partitioning of the input space; then, adjusting the antecedent and consequent parameters of the fuzzy rules so that a specified objective function is minimized. The evolutionary process begins by randomly generating an initial population, each candidate solution is represented by a vector. The length of the latter is based on the number of antecedent and consequent parameters in the ANFIS model. Then, the entire population was made to improve gradually until the maximum number of iterations was reached. The proposed approach was applied for the recognition of phonemes from TIMIT database and speaker recognition from the CHAINS database. The results obtained by the hybrid models AG-ANFIS and PSO-ANFIS showed an improvement in precision compared to a similar classic ANFIS based on the back-propagation of the gradient.

Keywords: ANFIS, Subtractive Clustering, GA, PSO, AG-ANFIS, PSO-ANFIS, speaker, phoneme.

Table des matières

Table des matières **vi**

Liste des figures **ix**

Liste des tableaux **xi**

Préface **1**

1 Neuro flou Adaptive-Network-based Fuzzy Inference System ANFIS **4**

1.1 Introduction . . . 5

1.2 Les réseaux de neurones . . . 5

1.3 La logique Floue . . . 6

1.3.1 Les fondements essentiels de la logique floue . . . 6

1.3.2 Types des systèmes flous . . . 10

1.4 Les systèmes neuro-flou . . . 10

1.4.1 Définition . . . 12

1.4.2 Généralités . . . 12

1.4.3 Création d'un système neuro-flou . . . 14

1.4.4 Les combinaisons des réseaux de neurones avec la logique floue . . . 14

1.4.5 Systèmes neuro-flous hybrides . . . 17

1.4.6 Architecture neuro flou hybride . . . 18

1.5 Adaptive Network Fuzzy Inference System ANFIS . . . 24

1.5.1 Les caractéristique du réseau ANFIS . . . 24

1.5.2 Architecture du réseau ANFIS . . . 24

1.5.3 Les avantages du réseau ANFIS . . . 28

1.5.4 Apprentissage du réseau ANFIS . . . 28

1.6 Conclusion . . . 34

2 Reconnaissance Automatique de la Parole **35**

2.1 Introduction . . . 36

2.2 Principe de base du signal vocal . . . 37

2.2.1 Le système de production de la parole . . . 37
2.2.2 La perception de la parole . . . 40
2.3 Les caractéristiques du signal de parole . . . 42
2.4 La reconnaissance automatique de la parole . . . 43
2.4.1 Le prétraitement . . . 44
2.4.2 L'extraction des caractéristiques . . . 44
2.4.3 La correspondance des caractéristiques . . . 54
2.5 Reconnaissance du locuteur . . . 54
2.5.1 Domaines d'application de la reconnaissance du locuteur . . . 55
2.5.2 Pourquoi la reconnaissance du locuteur ? . . . 56
2.5.3 Informations spécifiques au locuteur dans le signal vocal . . . 56
2.5.4 Identification Vs Vérification . . . 58
2.5.5 Dépendance et indépendance du texte . . . 59
2.5.6 Développement d'un système de reconnaissance du locuteur . . . 61
2.5.7 Identification Vs Vérification . . . 61
2.6 Etat de l'art . . . 62
2.7 Conclusion . . . 67

3 Les Algorithmes évolutionnaires **68**
3.1 Introduction . . . 69
3.2 Les Algorithmes Evolutionnaires . . . 69
3.3 Terminologie . . . 71
3.4 Principe de base des algorithmes évolutionnaires . . . 71
3.4.1 Caractéristiques alternatives des algorithmes évolutifs . . . 73
3.4.2 Représentation de la population . . . 73
3.4.3 Fitness et coût . . . 74
3.4.4 La sélection . . . 75
3.4.5 Les opérateurs de reproduction . . . 75
3.5 Stratégies d'évolution . . . 77
3.6 Programmation génétique . . . 78
3.7 Programmation évolutionnaire . . . 78
3.8 L'évolution différentielle . . . 79
3.9 Les algorithmes génétiques . . . 81
3.9.1 Introduction . . . 81
3.9.2 Principe de base de l'algorithme génétique . . . 81
3.9.3 Le fonctionnement d'un Algorithme Génétique . . . 83
3.9.4 Forces et limites des Algorithmes Génétiques . . . 90

3.10 Optimisation par l'essaim de particules 92
3.10.1 Introduction 92
3.10.2 Fonctionnement du PSO 93
3.10.3 Topologies 96
3.10.4 Forces et limites des PSO 97
3.11 Conclusion 98

4 Experiences et Résultats **99**
4.1 Introduction 100
4.2 Problématique 100
4.3 L'approche proposée 101
4.3.1 Le modèle AG-ANFIS proposé : 103
4.3.2 Le modèle PSO-ANFIS proposé 104
4.4 Expérience et Résultats 106
4.4.1 Expérience 1 : La Reconnaissance des phonèmes 107
4.4.2 Expérience 2 : La Reconnaissance des locuteurs 113
4.4.3 Résultats et discussions 115
4.5 Conclusion 120

Conclusion générale **121**

Liste des figures

1.1 exemple de variable linguistique 'surface' avec cinq sous-ensembles flous : minuscule, petite, moyenne, grande et immense. 7
1.2 Formes usuelles des fonctions d'appartenance. 8
1.3 Structure interne d'un système flou. 9
1.4 Différents modèles d'inférences flous. 11
1.5 Principe du système neuro-flou. 12
1.6 Principe de fonctionnement d'un Réseau neuro-flou. 15
1.7 Les étapes de création d'un système neuro-flou. 16
1.8 Réseau neuro-flou réalisé en parallèle. 17
1.9 Réseau neuro-flou réalisé en série. 17
1.10 Réseau neuro-flou hybride. 18
1.11 Différentes architectures des systèmes neuro-flous. 20
1.12 Les trois différents types de SANFIS. 23
1.13 Architecture du réseau ANFIS. 25
1.14 (1) Type-1 raisonement flou, (b) equivalent ANFIS (type-1 ANFIS). 27
1.15 Résultats du C Moyens Flous FCM. 31
1.16 Méthode d'apprentissage Hybride. 33

2.1 Extraction automatiquement des informations transmises dans le signal vocal. 36
2.2 Système de production de la parole humaine. 39
2.3 Modèle de production de la parole. 39
2.4 Fréquences vocales. 40
2.5 Structure de l'oreille humaine. 42
2.6 Signal vocal. 43
2.7 Processus du système de reconnaissance de la parole. 44
2.8 Flux de travail de traitement d'extraction des caractéristiques MFCC. . . . 47
2.9 PSD du signal échantillonné à 44,1 KHz avant et après la préaccentuation Power Spectral Density PSD. 48
2.10 Spectrogramme d'un signal avant et après la préaccentuation. 48

2.11 La fenêtre Hamming. 49
2.12 Séquence de N = 10 échantillons DFT. 51
2.13 Ampleur du signal vocal fenêtré Banque Mel-Filter. 51
2.14 Ampleur du signal vocal fenêtré par banc de filtres de Mel. 52
2.15 Banc de filtres Mel triangulaires pour un système à 24 filtres. 52
2.16 Spectre de puissance de la trame. 52
2.17 Spectre Mel, spectre de puissance de la trame dans le domaine fréquentiel. 53
2.18 Diagramme schématique du système de reconnaissance du locuteur. . . . 55
2.19 Système de reconnaissance des locuteurs. 58
2.20 Identification du locuteur. 60
2.21 Vérification du locuteur. 60
2.22 Phases du système de reconnaissance du locuteur. 61

3.1 Classification des techniques d'optimisation. 70
3.2 Organigramme de l'algorithme évolutionnaire 72
3.3 Organigramme d'un algorithme évolutif. 76
3.4 Boucle de base d'un GA. 82
3.5 L'organigramme de l'algorithme génétique. 83
3.6 Sélection de la roue de roulette. 86
3.7 Croisement à un point. 87
3.8 Croisement à deux points. 87
3.9 Croisement Uniforme. 88
3.10 Mutation. 89
3.11 Organigramme de l'algorithme d'optimisation par essaims de particules. 94
3.12 Topologies d'essaim de particules couramment utilisées. 96

4.1 L'organigramme du modèle AG-ANFIS proposé. 105
4.2 L'organigramme du modèle PSO-ANFIS proposé. 106
4.3 Taux de reconnaissance des phonèmes en fonction du rayon du model ANFIS. 110
4.4 L'erreur Quadratique Moyenne RMSE en fonction du rayon du modèle ANFIS utilisé pour la reconnaissance des phonèmes. 111
4.5 Taux de reconnaissance pour les 8 locuteurs en utilisant ANFIS. 115
4.6 Taux de reconnaissance pour les 16 locuteurs en utilisant ANFIS. 116

Liste des tableaux

1.1 Comparaison entre la logique floue et les réseaux de neurones 13
1.2 les différentes couches d'un système ANFIS 26
1.3 Les paramètres à ajuster d'un système ANFIS 33

2.1 Nombre total de caractéristiques résultantes dans un vecteur standard . . 53
2.2 Hiérarchie des informations spécifiques au locuteur et facteurs déterminants associés 57
2.3 État de l'art de l'ANFIS dans la parole. 63
2.4 Quelques travaux sur la reconnaissance des locuteurs. 66

4.1 Les paramètres du FIS utilisé 102
4.2 Les paramètres de l'AG utilisé 103
4.3 Les paramètres du PSO utilisé dans notre thèse 104
4.4 Sous corpus de la base de données TIMIT utilisé dans notre thèse 109
4.5 Taux de reconnaissance des voyelles en utilisant le modèle ANFIS 112
4.6 Taux de reconnaissance des fricatives en utilisant le modèle ANFIS 112
4.7 Taux de reconnaissance des plosives en utilisant le modèle ANFIS 113
4.8 Taux de reconnaissance pour les 8 locuteurs en utilisant AG-ANFIS . . . 116
4.9 Taux de reconnaissance pour les 16 locuteurs en utilisant AG-ANFIS . . . 116
4.10 Taux de reconnaissance pour les 8 locuteurs en utilisant PSO-ANFIS (1000 itérations) 117
4.11 Taux de reconnaissance pour les 16 locuteurs en utilisant PSO-ANFIS . . 117
4.12 Comparaison ANFIS, GA-ANFIS et PSO-ANFIS pour les 8 locuteurs . . . 117
4.13 Comparaison ANFIS, GA-ANFIS et PSO-ANFIS pour les 16 locuteurs . . 117
4.14 Comparaison des modèles proposés et des modèles existants 119

Introduction Générale

En cette ère d'avancées technologiques, les nouvelles techniques évoluent pour une meilleure interaction homme-machine. Initialement, cette envie d'interagir avec les machines était la principale raison de l'invention de périphériques d'entrée-sortie comme le clavier, la souris, le moniteur, l'imprimante, etc. Cependant, aucune de ces inventions ci-dessus n'est capable de fournir une interaction verbale de l'homme et des machines, qui est le moyen naturel de communication pendant de nombreux siècles. À la différence d'autres moyens électroniques de communication, les systèmes utilisant la parole offrent à l'utilisateur non entraîné un accès simple et naturel. Elle permet d'avoir un accès immédiat à une information sans avoir à parcourir toute une arborescence hiérarchique de menus.

La reconnaissance vocale fait référence à la capacité d'une machine ou d'un programme à reconnaître ou à identifier des mots prononcés, des personnes et aussi à interpréter la voix. Ces quatre dernières décennies ont été témoin d'une évolution impressionnante dans le développement du traitement automatique de la parole où un certain nombre d'avancées significatives ont été réalisées. Le développement d'un grand nombre d'applications commercialisés utilisant la reconnaissance automatique de la parole et mise à la disposition du public, notamment la commande et le contrôle, la dictée, les systèmes de dialogue pour les personnes handicapées, la traduction, etc.

Les outils intelligents sont de plus en plus utilisés dans la conception, la modélisation et la commande de systèmes complexes tels que les robots, les procédés biologiques, les véhicules routiers ... Ces outils s'adaptent en utilisant quelques exemples de situations (entrées d'un système) et ils corrigent automatiquement les décisions pour des situations futures. On entend par outils intelligents les techniques du soft computing à savoir : les réseaux de neurones, la logique floue et les systèmes neuro-flous les algorithmes d'optimisation.

Le terme neuro-flou est né de la fusion de la logique floue et des réseaux de neurones. Cette hybridation très approuvée par les chercheurs, qui tire profit des deux approches. Tel que la possibilité d'incorporer une base de connaissances, et de traiter les données imprécises et vagues par logique floue et la capacité d'apprentissage par les réseaux de neurones Plusieurs architectures neuro-floues ont été proposées dans la littérature sui-

vant le type de règles floues qu'elles intègrent (Mamdani ou Sugeno).
Le réseau adaptatif à système inférence flou (ANFIS), proposé par Jang, est l'un des systèmes neuro-fuzzy dans lesquels un système flou de type Sugeno de premier ordre est implémenté ; il applique la règle de la forme TSK Takagi Sugeno et Kang dans son architecture. Cette règle produit directement des sorties nettes, car elle utilise des polynômes comme conséquences de la règle.

La construction de modèles flous implique deux phases importantes, l'identification de la structure qui permet de déterminer la structure appropriée du réseau, c'est-à-dire le meilleur partitionnement de l'espace d'entrée (le nombre de fonctions d'appartenance pour chaque entrée et le nombre de règles). Ainsi, la détermination d'une structure appropriée devient un problème important, si bien que des techniques de regroupement sont appliquées pour surmonter ce problème ; dans notre thèse nous nous sommes intéressés au regroupement soustractif.

Ensuite, l'apprentissage paramétrique effectué pour ajuster les paramètres antécédents et conséquents afin qu'une fonction objectif spécifiée soit minimisée. L'optimisation de ces paramètres est l'un des principaux enjeux de l'apprentissage de l'ANFIS, l'apprentissage classique est basé sur la descente du gradient, cet algorithme à l'inconvénient de tomber dans des minimas locaux. Pour faire face à ce problème, les algorithmes d'optimisation peuvent être une solution idéale.

Le but principal de notre travail, est l'amélioration des performances du réseau ANFIS en termes de précision par les algorithmes évolutionnaires et plus précisément l'amélioration de l'apprentissage de paramètres internes du réseau par les algorithmes génétiques AG et l'essaim de particules PSO. L'approche proposée est appliqué dans le domaine de la reconnaissance de la parole. L'avantage de la méthode est clairement mis en évidence en offrant une nette amélioration de la précision par rapport à la méthode d'apprentissage classique.

Ce mémoire est organisé en quatre chapitres comme suit :

Le premier chapitre est divisé en quatre parties. La première traite les réseaux de neurones. La deuxième partie est consacrée aux systèmes flous. Nous rappelons d'abord les notions de bases sur lesquelles reposent ces systèmes puis nous décrivons leur principe de fonctionnement et leurs différentes composantes. Enfin, dans la troisième partie nous aborderons l'approche neuro-floue. L'architecture et l'apprentissage du réseau ANFIS est détaillé dans la dernière partie.

Le deuxième chapitre est consacré à la reconnaissance automatique de la parole dans le quel nous présentons les bases du signal de la parole et le principe des systèmes de RAP. Ensuite, la reconnaissance du locuteur est bien expliquée. En dernier, Un état

de l'art portant sur les travaux existants sur la reconnaissance de la parole par le réseau ANFIS, et aussi la reconnaissance du locuteur par différentes méthodes.

Le troisième chapitre est consacré aux algorithmes évolutionnaires. Il est constitué de trois parties principales. La première présente une description détaillée des algorithmes évolutionnaires simples et quelques algorithmes évolutionnaires dans laquelle nous rappelons les définitions relatives à leur fonctionnement. Dans la deuxième partie nous introduirons les algorithmes génétiques. La troisième partie décrit en détail les essaims de particules.

Le quatrième chapitre est dédié à la conception et la réalisation du système de reconnaissance de la parole (locuteur et phonème) par l'approche proposée. On clôture cette thèse par une conclusion générale et des perspectives.

1 Neuro flou Adaptive-Network-based Fuzzy Inference System ANFIS

Sommaire

1.1 Introduction . . . 5
1.2 Les réseaux de neurones . . . 5
1.3 La logique Floue . . . 6
1.3.1 Les fondements essentiels de la logique floue . . . 6
1.3.2 Types des systèmes flous . . . 10
1.4 Les systèmes neuro-flou . . . 10
1.4.1 Définition . . . 12
1.4.2 Généralités . . . 12
1.4.3 Création d'un système neuro-flou . . . 14
1.4.4 Les combinaisons des réseaux de neurones avec la logique floue . . . 14
1.4.5 Systèmes neuro-flous hybrides . . . 17
1.4.6 Architecture neuro flou hybride . . . 18
1.5 Adaptive Network Fuzzy Inference System ANFIS . . . 24
1.5.1 Les caractéristique du réseau ANFIS . . . 24
1.5.2 Architecture du réseau ANFIS . . . 24
1.5.3 Les avantages du réseau ANFIS . . . 28
1.5.4 Apprentissage du réseau ANFIS . . . 28
1.6 Conclusion . . . 34

1.1 Introduction

La combinaison de la logique floue et des réseaux de neurones a prouvé son efficacité dans divers problèmes. Ces systèmes hybrides très approuvé par les chercheurs. Chaque technique intelligente a des propriétés particulières (capacité d'apprentissage, explication de décisions). Chaque technique convient à résoudre certains problèmes particuliers. En effet, les réseaux neuronaux sont par exemple utilisés pour la reconnaissance des modèles. Cependant, ils sont incapables d'expliquer comment ils atteignent leurs décisions. Aussi pour les Systèmes de la logique floue qui peuvent raisonnez avec l'information imprécise et expliquer leurs décisions mais ne peuvent cependant pas acquérir automatiquement les règles qu'ils l'utilisent pour prendre ces décisions. Cela a mené au développement d'une autre approche qui est principalement connu comme approche neuro-floue. Avant de décrire cette dernière, il est important de présenter une brève introduction sur les réseaux de neurones et quelques notions sur la logique floue.

1.2 Les réseaux de neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base par des neurones biologiques que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles et à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

Un réseau de neurones (RN) est un système informatique qui a des caractéristiques semblables aux réseaux de neurones biologiques. Il est composé de plusieurs neurones arrangés sous forme de blocs différents appelés couches du réseau. Les neurones appartenant à la même couche utilisent le même type de fonction d'activation et possèdent les mêmes caractéristiques. La liaison entre deux couches voisines se fait par l'entremise de poids qui ont le rôle des synapses. La structure du réseau permet de traiter l'information portée par la valeur de ses poids et le diriger vers la sortie. En général, les structures des réseaux sont divisées en deux grandes familles : les réseaux de neurone non bouclés et les réseaux de neurones bouclés.

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés. L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. Dans

le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage [1].

1.3 La logique Floue

La logique floue a été développée par Lofti A. Zadeh en 1965 à partir de sa théorie des sous-ensembles flous. Les sous-ensembles flous sont une manière mathématique de représenter l'imprécision de la langue naturelle, ils peuvent être considérés comme une généralisation de la théorie des ensembles classiques. La logique floue est aussi appelée "logique linguistique" car ses valeurs de vérité sont des mots du langage courant : "plutôt vrai, presque_faux, loin, si loin, près de, grand, petit...". La logique floue a pour objectif l'étude de la représentation des connaissances imprécises, des raisonnements approchés et elle cherche à modéliser les notions vagues du langage naturel [2].

1.3.1 Les fondements essentiels de la logique floue

La notion de variable linguistique permet de modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise est inconnue. Une variable linguistique, ou variable floue, est donc une variable dont les valeurs floues appartiennent à des ensembles flous pouvant représenter des mots du langage naturel. Ainsi une variable floue peut prendre simultanément plusieurs valeurs linguistiques. Le domaine sur lequel ces termes et ces variables sont définis, constitue l'univers de discours. Le découpage de cet univers de discours par les termes flous est appelé une partition floue. Lorsque l'univers de discours est totalement recouvert par les termes flous, et que pour toutes valeurs, la somme des degrés d'appartenance est égale à 1, on parle alors de partition floue forte.

La notion d'ensemble flou a été proposée par Zadeh [2] en introduisant un caractère graduel de l'appartenance d'un élément à un ensemble donné. Cela permet une meilleure représentation des termes et des connaissances vagues que nous, les humains, manipulons au quotidien.

Mathématiquement, un ensemble flou A d'un univers de discours U, est caractérisé par une fonction d'appartenance, notée μA, à valeur dans l'intervalle [0,1] et qui associe à chaque élément x de U un degré d'appartenance $\mu A(x)$ indiquant le niveau d'appartenance de x à A. $\mu A(x) = 1$ et $\mu A(x) = 0$ correspondent respectivement à l'appartenance et la non-appartenance. La variable linguistique peut être représentée par un triplet $(x, T(x), U)$ où x est une variable définie sur l'univers de discours U et

$T_x = A1, A2,$.est un ensemble composé de sous-ensembles flous de U qui caractérise x. On associe souvent à chaque sous ensemble flou de TX une valeur ou un terme linguistique (étiquette) [3]. La figure 1.1 illustre un exemple de variable linguistique 'surface' avec cinq sous-ensembles flous : minuscule, petite, moyenne, grande et immense.

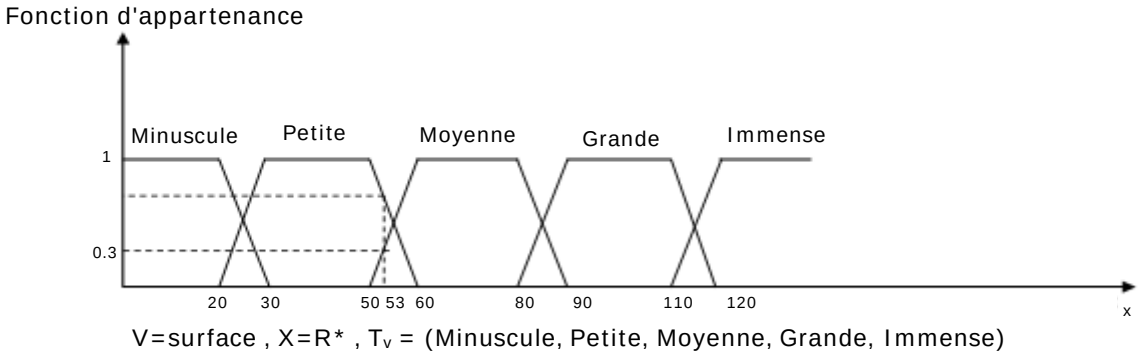


FIGURE 1.1 – *exemple de variable linguistique 'surface' avec cinq sous-ensembles flous : minuscule, petite, moyenne, grande et immense.*

1.3.1.1 Les fonctions d'appartenance

Soit un ensemble E et un sous-ensemble A de $E(A \subset E)$, et x un élément de E appartenant à $A(x \in A)$. Pour illustrer cette caractéristique, on utilise la fonction d'appartenance $\mu A(x)$ compris entre 0 et 1, qui représente le degré d'appartenance de x à l'ensemble flou A. Le plus souvent, la fonction d'appartenance est déterminée par l'une des fonctions suivantes [4][5](figure 1.2) :

Fonction triangulaire : elle est définie par trois paramètres a, b, c qui déterminent les coordonnées des trois sommets (figure 1.2(a)).

$$\mu_A(x) = max\left[min\left[\frac{(x-a)}{(b-a)}, \frac{(c-x)}{(c-b)}\right], 0\right] \tag{1.1}$$

Fonction trapézoïdale : elle est définie par quatre paramètres a, b, c, d (figure 1.2 (b)) :

$$\mu_A(x) = max\left[min\left[\frac{(x-a)}{(b-a)}, 1, \frac{(d-x)}{(d-c)}\right], 0\right] \tag{1.2}$$

Fonction gaussienne : elle est définie par deux paramètres σ, m (figure 1.3 (c))

$$\mu_A(x) = exp\left[-\frac{(x-m)^2}{(2\sigma^2)}\right] \tag{1.3}$$

Fonction sigmoïde : une fonction sigmoïde est définie par deux paramètres a, b (figure 1.3 (d))

$$\mu_A(x) = \frac{1}{1 + exp\,[-a(x-c)]} \tag{1.4}$$

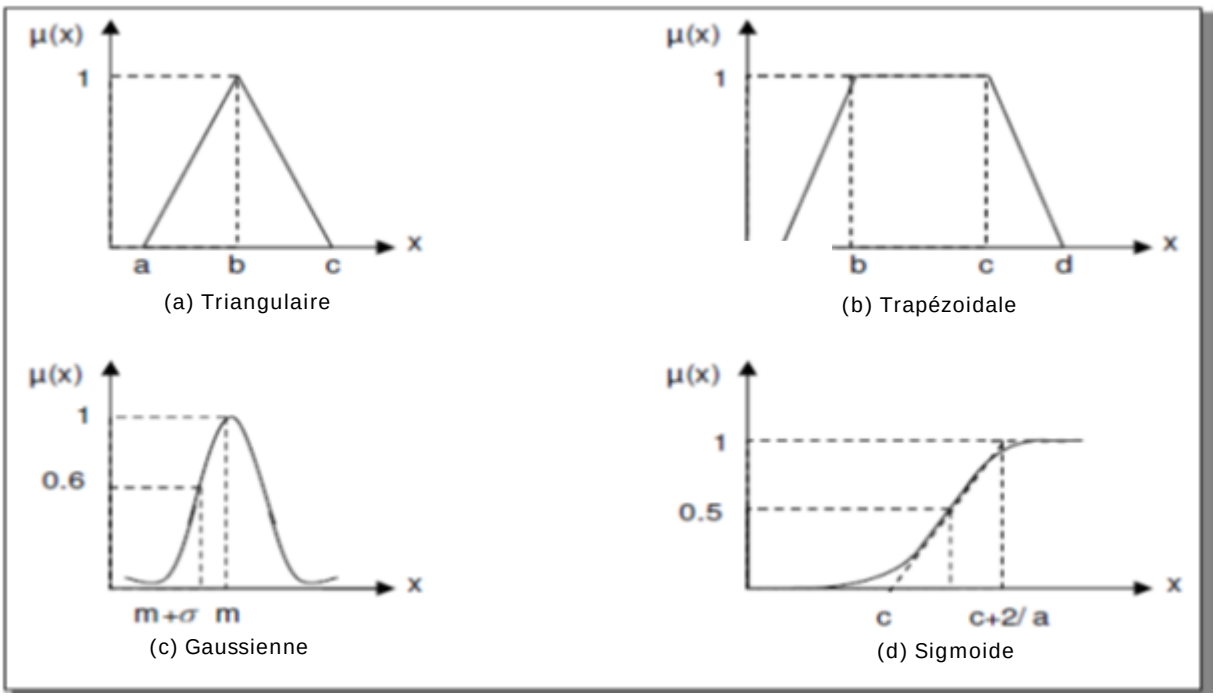


FIGURE 1.2 – *Formes usuelles des fonctions d'appartenance.*

1.3.1.2 les règles d'inférence floues

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sorties de ce système à l'aide des différents opérateurs. Ces règles se présentent sous la forme :
Si condition 1 ET/OU condition 2 (ET/OU...) **alors** action sur les sorties OU
Si condition 3 ET/OU condition 4 (ET/OU...) **alors** action sur les sorties OU
Si condition n ET/OU condition n+1 (ET/OU...) **alors** action sur les sorties OU

1.3.1.3 les systèmes d'inférence floues SIF

Un système d'inférence floue a comme but de transformer les données d'entrée en données de sortie à partir de l'évaluation d'un ensemble des règles. Un système d'inférence floue est formé de trois blocs (Figure 1.3) [3][2] :

La base de connaissances : elle contient les définitions des fonctions d'appartenance

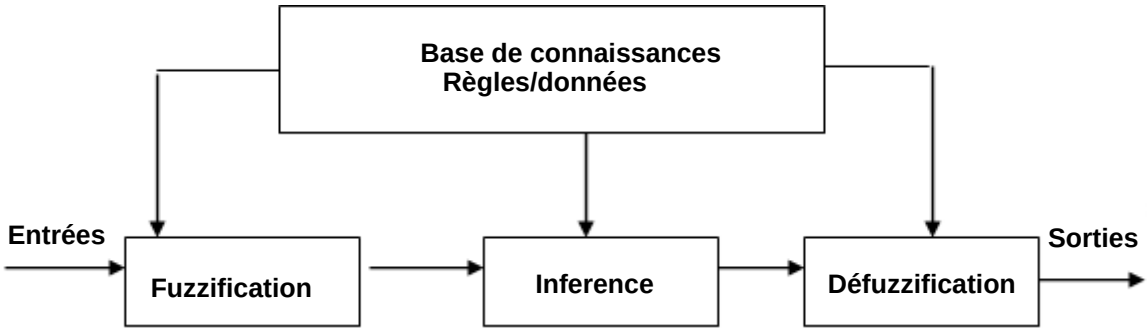


FIGURE 1.3 – *Structure interne d'un système flou.*

(formes et paramètres) associées aux variables d'entrée/sortie ainsi que l'ensemble des règles floues.

1.3.1.4 les règles floues

1. La fuzzification consiste à calculer, pour chaque valeur d'entrée numérique, les degrés d'appartenance aux ensembles flous associés et prédéfinis dans la base de données du système flou. Ce bloc réalise la transformation des entrées numériques en informations symboliques floues utilisables par le mécanisme d'inférence.

2. Le second bloc est le moteur d'inférence et il est constitué par l'ensemble des règles. Le mécanisme d'inférence consiste d'une part à calculer le degré de vérité des différentes règles du système et d'autre part à associer à chacune de ces règles une valeur de sortie. Cette valeur de sortie dépend de la partie conclusion des règles qui peut prendre plusieurs formes.

3. Enfin, une étape de défuzzification qui permet, si nécessaire, d'inférer une valeur nette. La défuzzification consiste à remplacer l'ensemble des valeurs des sorties des différentes règles résultant de l'inférence par une valeur numérique unique représentative de cet ensemble. Dans le cas des règles de type Sugeno, le calcul se fait simplement par une somme normalisée des valeurs associées aux règles floues. Dans le cas de règles de Mamdani, le calcul de la valeur numérique de sortie s'effectue en deux étapes : il s'agit de regrouper (par union) les sous ensemble flous issus de l'inférence pour en obtenir un seul ensemble représentatif des différentes conclusions des règles floues. Ensuite le passage du symbolique vers le numérique.

Il existe plusieurs méthodes de défuzzification [6], les plus communément employées sont : centre de gravité (COG), Moyenne des maxima (MOM), bissectrice de la zone (BIS), plus grand maximum (LOM), plus petit maximum (SOM).

1.3.2 Types des systèmes flous

Une règle floue est de la forme Si je rencontre telle situation Alors j'en tire telle conclusion. La situation, appelée prémisse ou antécédent de la règle, est définie par une combinaison de relations de la forme x est A pour chacune des composantes du vecteur d'entrée. La partie conclusion de la règle est appelée conséquence, ou encore simplement conclusion.

Dans le jeu de règles du système flou interviennent les opérateurs flous "ET(AND)" et "OU(OR)". L'opérateur "ET" s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur "OU" lie les différentes règles. Plusieurs types de raisonnement flou ont été proposés dans la littérature suivant la réalisation des opérateurs flous "ET" et "OU" et le type des règles floues utilisées. Les trois moteurs d'inférence floue les plus utilisés sont : le moteur de Mamdani, Sugeno et celui de Tsukumoto. Selon la structure particulière de la proposition conséquente, on peut distinguer trois types de modèles flous basés sur des règles [6] (Figure 1.4) :

1. Modèle flou linguistique (ou modèle Mamdani), dans lequel l'antécédent et le conséquent sont tous les deux des propositions floues qui utilisent des variables linguistiques.

2. Modèle flou Takagi-Sugeno, dans lequel le conséquent utilise des variables numériques plutôt que des variables linguistiques, sous la forme d'une constante, d'un polynôme ou de manière plus générale d'une fonction ou d'une équation différentielle dépendant des variables associées à la proposition antécédente.

3. Méthode de Tsukumoto, Dans ce cas, des fonctions monotoniques sont associées aux variables de sortie. La sortie totale est une moyenne pondérée des degrés de confiance des règles floues et des valeurs des fonctions des variables de sortie [7].

La figure 1.4 illustre les types du raisonnement flou pour un système flou à deux entrées et une base de connaissances de deux règles floues. On constate que les différences viennent de la spécification de la partie conclusion d'une part, et de la méthode de défuzzification d'autre

1.4 Les systèmes neuro-flou

L'utilisation conjointe des réseaux de neurones et de la logique floue permet de tirer avantage des deux méthodes. La capacité d'apprentissage de la première et la lisibilité et la souplesse de la seconde (figure 1.5). Plusieurs combinaisons de ces deux méthodes ont été développées dans la littérature [8][9][10] sont le plus souvent orientées vers la commandes de systèmes complexes et les problèmes de classification.

Les systèmes flous et les réseaux de neurones ont le même objectif, tous les deux

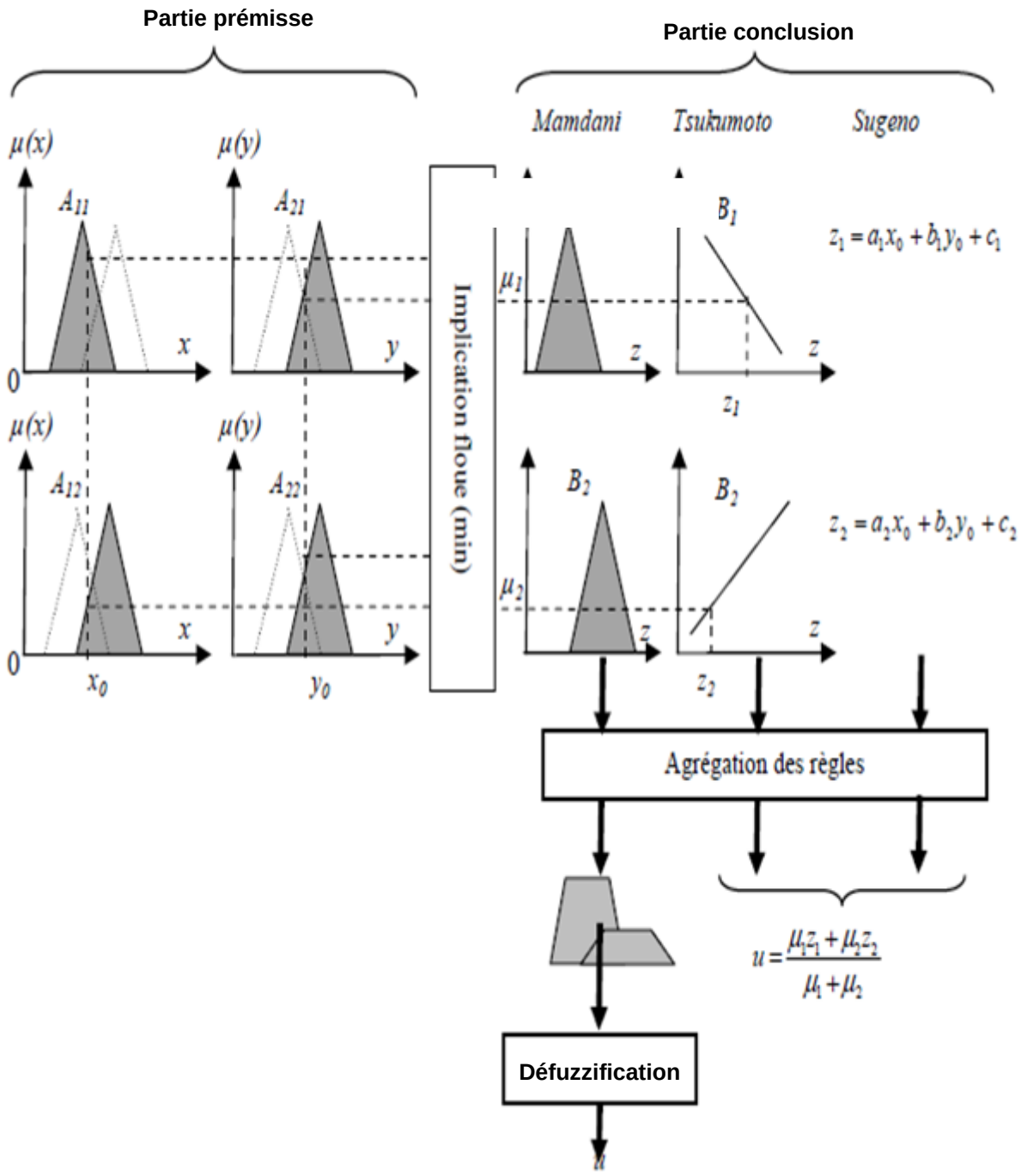


FIGURE 1.4 – *Différents modèles d'inférences flous.*

tentent de modéliser le fonctionnement du cerveau humain. Mais il y une nuance entre les deux. Les réseaux de neurones tentent de modéliser l'architecture du cerveau en créant une modélisation de l'entité du cerveau (le neurone). Les systèmes flous eux modélisent le cerveau par son mode de fonctionnement (apprentissage et déduction) et fournit des connaissances avec un certain degré d'incertitude (ou d'exactitude). D'où l'idée de combiner ces deux techniques pour créer un système artificiel intelligent qui

résout les problèmes de manière proche de celle de l'être humain. Afin de résumer

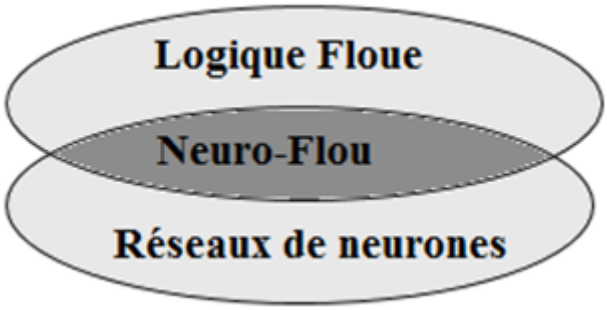


FIGURE 1.5 – *Principe du système neuro-flou.*

l'apport du neuro-flou, le tableau 1.1 regroupe les avantages et les inconvénients de la logique floue et des réseaux de neurones [11]. Les systèmes neuro-flous sont créés afin de synthétiser les avantages et de surmonter les inconvénients des réseaux neuronaux et des systèmes flous.

1.4.1 Définition

Un système Neuro-Flou est un réseau de neurones qui est typologiquement équivalent à la structure d'un système flou. Les entrées/sorties du réseau ainsi que les poids sont des nombres réels, mais les nœuds implémentent des opérations spécifiques aux systèmes flous : fuzzification, opérateurs flous et défuzzification [11]. En d'autres termes, un système Neuro-flou peut être vu comme un système flou pour lequel les opérations sont implémentées de façon parallèle par un réseau de neurone.

1.4.2 Généralités

Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine [12] . Généralement, les réseaux neuro-flous remplacent les différentes couches cachées des réseaux de neurones par des règles floues (i.e. règles linguistiques) qui peuvent être vues en tant que prototypes imprécis des données d'apprentissage (Figure 1.6). Ils utilisent ensuite des algorithmes d'apprentissage pour définir et optimiser ces paramètres. De plus, les règles d'un système neuro-flou sont transparentes, ce qui permet la validation et la manipulation par un expert.

L'introduction de la logique floue dans les réseaux neuronaux permet de résoudre les problèmes liés à la taille de la base de données, à l'initialisation des poids et à leur adaptation. Les développements actuels sur ce type de problématique montrent que

TABLE 1.1 – *Comparaison entre la logique floue et les réseaux de neurones*

Avantages	
RN	**LF**
• Capacité d'apprentissage • Ne nécessite pas un modèle mathématique • Capacité de généralisation • Robustesse • Capacité de calcul	• Le modèle mathématique non requis • Représentation des connaissances incertaines • Facilité d'interprétation des résultats • facilité d'extension de la base de connaissance
Inconvénients	
RN	**LF**
• Boite noire, manque d'interprétabilité • Difficulté de déterminer la topologie du réseau (nombre de neurone, nombre de couche cachée...) • Choix des valeurs initiales des poids et le pas d'apprentissage • Problème de convergence • Problème de sur apprentissage	• Les règles doivent être disponibles • Incapacité d'apprendre • Aucune méthode formelle pour l'ajustement • Adaptation difficile au changement de l'environnement

les performances des NF dépassent celles d'autres méthodes aussi bien en termes de précision des résultats que d'efficacité d'apprentissage [13].

1.4.3 Création d'un système neuro-flou

Les étapes de création d'un système neuro-flous (Figure 1.7) diffèrent selon l'application et les connaissances présentes, mais de façon générale, on peut les résumer en cinq étapes :

1. Définition des formes initiales des fonctions d'appartenance pour les entrées et les sorties .
2. Détermination de la base initiale des règles floues .
3. Réglage des règles et les fonctions d'appartenance de manière à avoir la meilleure approximation des règles d'après les données d'apprentissage, et aussi la meilleure généralisation .
4. Tester le système face à des données de test .
5. Elimination des règles faibles dans le but d'amélioré l'interprétation du système, en préservant ses performances en matière de tolérance.

Si les résultats ne sont pas satisfaisants, il est toujours possible de refaire l'apprentissage en modifiant les paramètres de l'algorithme.

1.4.4 Les combinaisons des réseaux de neurones avec la logique floue

Diverses associations de ces deux méthodes ont été développées depuis 1988 et sont le plus souvent orientées vers la commande de systèmes complexes et les problèmes de classification. Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue : réseau flou neuronal, système neuronal/flou concurrent, modèles neuro flous coopératifs et modèles neuro-flous hybrides [11][13][12] :

1.4.4.1 Réseau flou neuronal

Dans ce type de réseau, des techniques floues sont utilisées pour augmenter les possibilités d'apprentissage ou l'exécution d'un réseau neuronal.

1.4.4.2 Systèmes neuro-flous concurrents

Le réseau neuronal et le système flou fonctionnent ensemble sur la même tâche, sans qu'il y ait aucune influence de l'un sur l'autre, c'est-à-dire, aucun des systèmes ne détermine les paramètres de l'autre (figure 1.8).

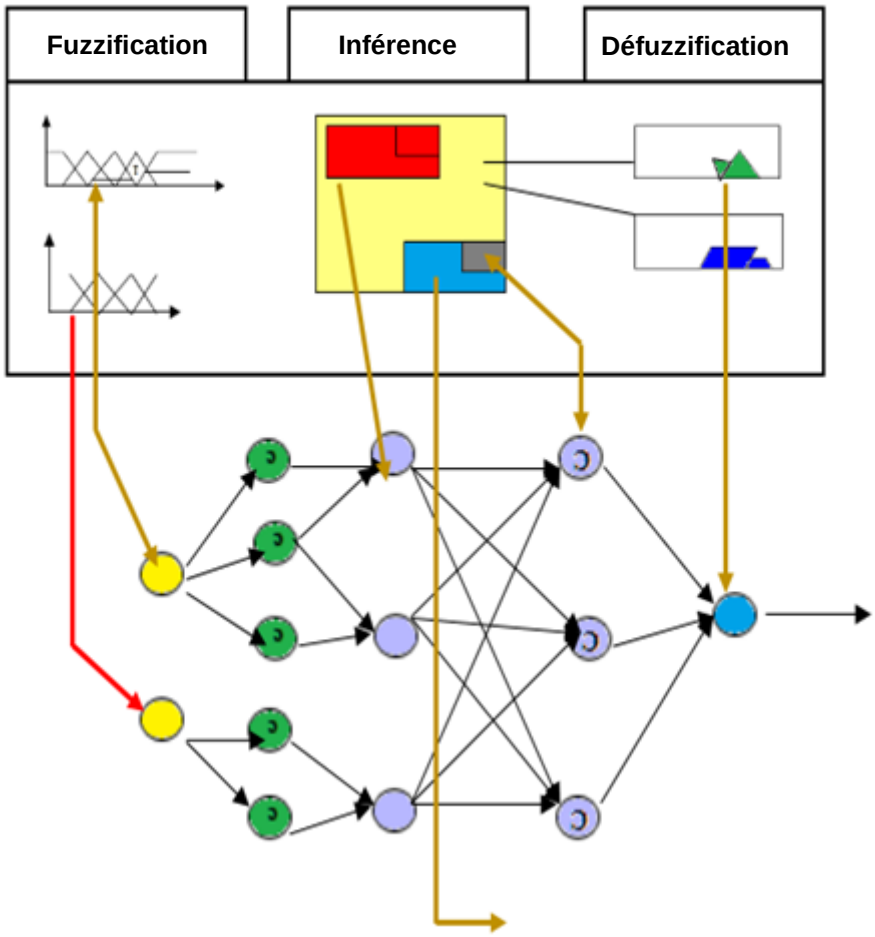


FIGURE 1.6 – *Principe de fonctionnement d'un Réseau neuro-flou.*

Le réseau neuronal est employé pour déterminer les paramètres (les règles et les ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau neuronal. C'est une forme simple des systèmes neuro-flous. En général, les réseaux de neurones se placent soit en amont pour le traitement des entrées, soit en aval pour la génération des sorties (figure 1.9). De tel réseaux présentent l'intérêt de résoudre deux problèmes importants de la logique floue : la détermination des fonctions d'appartenance et des règles ainsi leur adaptation à l'environnement du système.

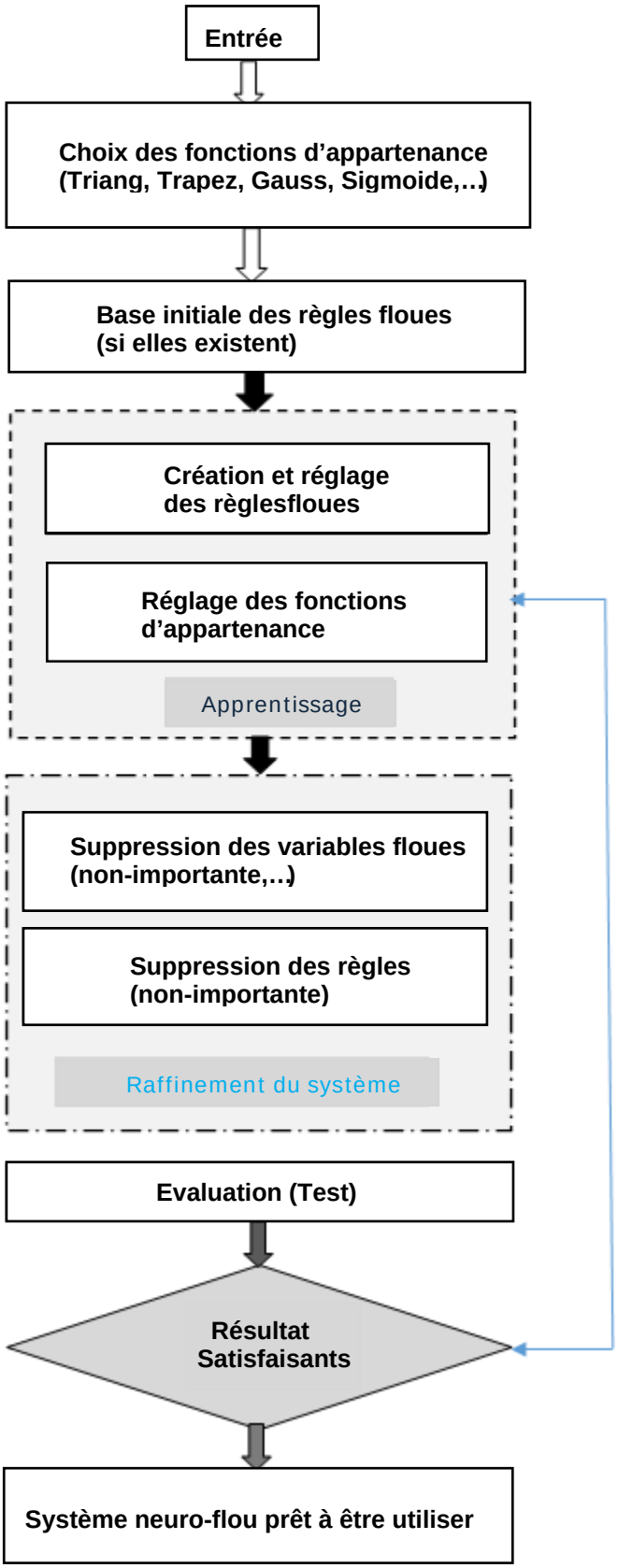


FIGURE 1.7 – *Les étapes de création d'un système neuro-flou.*

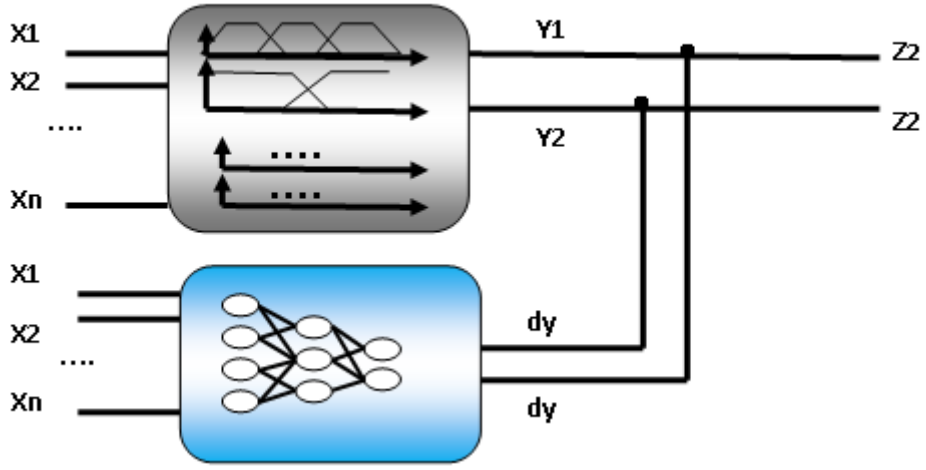


FIGURE 1.8 – *Réseau neuro-flou réalisé en parallèle.*

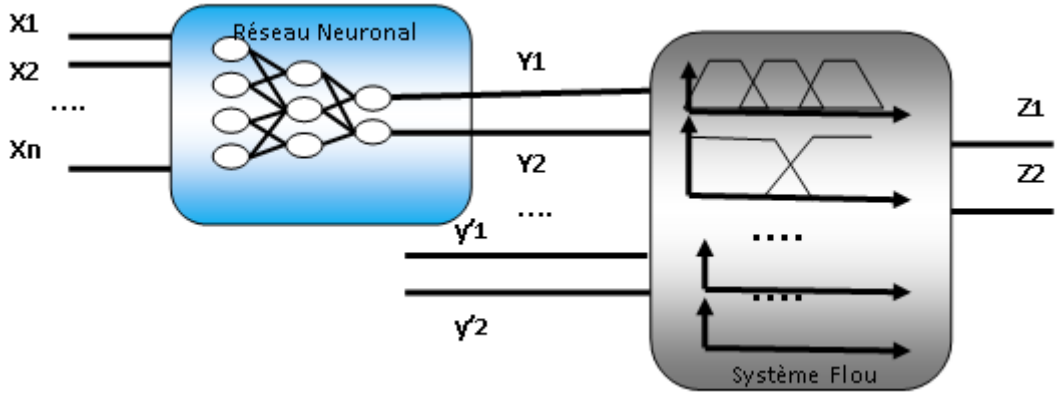


FIGURE 1.9 – *Réseau neuro-flou réalisé en série.*

1.4.5 Systèmes neuro-flous hybrides

Les deux paradigmes sont combinés dans une architecture homogène. Il peut être soit, un réseau neuronal mené des paramètres flous nommés réseau de neurones flou ou un système flou mis en application sous une forme distribuée parallèle dont les paramètres sont ajustes par apprentissage grâce au réseau de neurone nommé système flou neuronal. Le système d'inférence floue est mis sous la forme d'un réseau multicouche dans lequel les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règles et des méthodes d'inférence, d'agrégation et de défuzzification choisie. Par exemple, pour des règles de la forme : « si V1 est A11 et V2 est A21 alors W est W1 », on fait appel à un réseau de neurones qui admet pour entrées les valeurs X1 et X2 prises par les variables V1 et V2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance de A11 pour l'entrée X1 et de A21 pour X2 et à celui de la valeur prise par la conjonction

des conditions de chaque règle utilisant un opérateur de conjonction adéquat (figure 1.10).

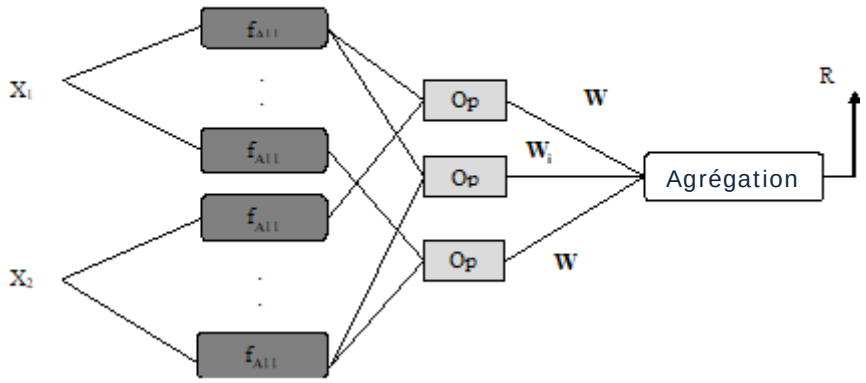


FIGURE 1.10 – *Réseau neuro-flou hybride.*

Les architectures Neuro-flous, sont divisées en deux classes :
a- Les modèles de réseaux neuro-flous à apprentissage supervisé
ANFIS : Adaptive Network Fuzzy Inference System.
NNDFR : Neuronal Network Driven Fuzzy Reasoning.
NEFCLASS : Neuro Fuzzy CLAASSification.
NEFPROX : Neuro-Fuzzy Function Proximation.
FuNe : Neuro Fuzzy model.
FRN : Fuzzy Rule Net.

b- Les modèles de réseaux neuro-flous à apprentissage par renforcement
ARIC : Approximation Reasoning-based Intelligent Control.
NEFCON : Neuro-Fuzzy CONtroller

1.4.6 Architecture neuro flou hybride

Plusieurs architectures, mettant en œuvre cette approche hybride, sont décrites dans la littérature [14][15]. Ces architectures peuvent être classées en trois groupes [13] selon le type de règles floues qu'elles intègrent :

Si (X_1 est A_1) ET (X_2 est A_2) ET ... ET (X_N est A_N) **Alors** (Y est C) où :

$$C = \begin{cases} B & (Type\ I) \\ f(x_1, x_2, \ldots, x_n) & (Type\ II) \\ \theta & (Type\ III) \end{cases} \tag{1.5}$$

Où X_i (i = 1,2, $\cdots$,n), Y représentent respectivement les variables d'entrée et de sortie, $A_j i$ les ensembles flous d'entrée , B, f(x_1,x_2, $\cdots$,xn) et β représentant respectivement, l'ensemble flou de sortie, une fonction linéaire des variables d'entrée et un composant singleton.

Notons que le type III est souvent considéré dans la littérature comme un cas particulier du type I [16].

La figure 1.11 montre quelques architectures : FALCON (a) et GARIC (b) interprètent la règle floue avec une structure à 5 couches, NEFCLASS (c) interprète la règle floue avec une structure à 3 couches et ANFIS (d) interprète la règle floue avec une structure à 6 couches.

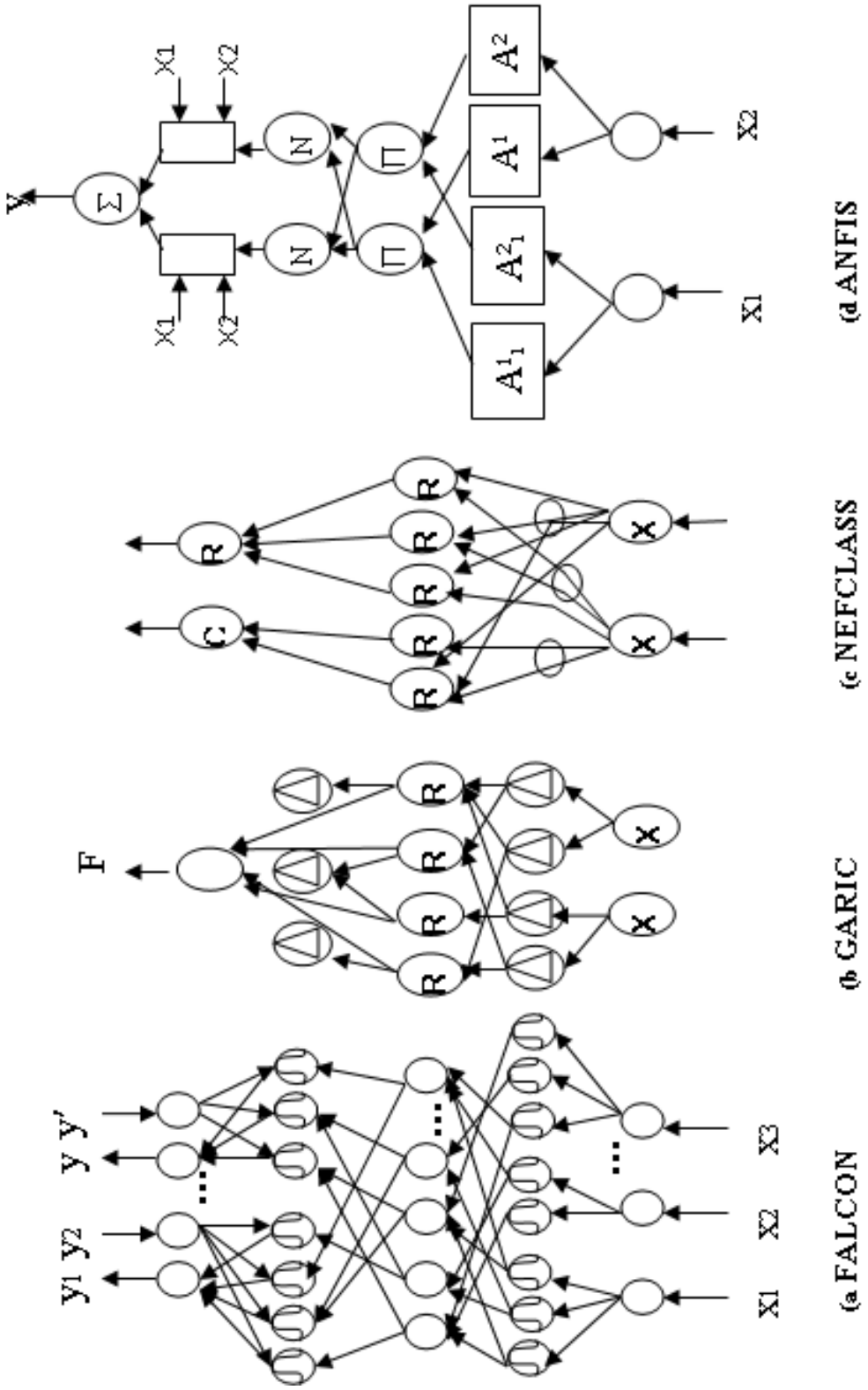


FIGURE 1.11 – *Différentes architectures des systèmes neuro-flous.*

ANFIS (Adaptive-Network-based Fuzzy Inference System)

ANFIS représente un système à inférence floue mis en application dans le cadre des réseaux adaptatifs. Il a été proposé par Jang [8]. II utilise la procédure d'apprentissage

hybride. Cette architecture affine les règles floues obtenues par des experts humains pour décrire le comportement d'entrée-sortie d'un système complexe. Un système ANFIS réalise une approximation linéaire de la variable de sortie en décomposant l'espace des entrées en différents espaces flous.

FALCON (Fuzzy Adaptive Learning Control Network) et GARIC (Generalized Approximate Reasoning-base Intelligent Controller)

Il s'agit de modèle à 5 couches [17][18], qui s'accorde bien à la technique de Mamdani, utilisant la fuzzification en entrée et la défuzzification en sortie. L'augmentation de la précision des résultats effectue une très lente exécution du système. Ce modèle est rarement utilisé en pratique mais il reste le meilleur pour la commande (Figure 1.11 a, b).

NEFCLASS (Neuro Fuzzy CLAASSification) suprimer la numérotation

Ce modèle utilisé généralement en classification [19], est constitué de 3 couches : une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes (Figure 1.11 c). Ce modèle est facile à mettre en application, il évite l'étape de défuzzification, et se caractérise donc par une rapidité bien supérieure aux autres architectures.

NEFCON (Neuro-Fuzzy CONtroller)

Ce modèle [20] formé de 3 couches. Une couche cachée formée par des règles, une couche d'entrée incluant les nœuds d'entrée avec les sous-ensembles flous et une couche de sortie avec un nœud de sortie et les sous-ensembles des conséquences. Le processus d'apprentissage du NEFCON peut être divisé en deux phases. La première consiste à trouver les règles de base initiales. Si les connaissances antérieures ne sont pas disponibles les règles de base seront difficiles à déterminer. Et si cette règle est définie par un expert l'algorithme les complète. Dans la seconde phase, les règles de base sont optimisées par modification des sous-ensembles flous des règles. Les deux phases utilisent l'erreur floue, cette erreur peut être trouvée avec la différence entre la sortie désirée et celle obtenue. Cette architecture est généralement utilisée en approximation de fonctions et en contrôle flou.

NEFPROX (Neuro Fuzzy function Approximator)

Ce modèle obtenu par l'association des deux architectures : NEFCLASS et NEFCON [21], est utilisé dans différentes applications comme la classification et l'approximation de fonctions. Le NEFCLASS utilise un algorithme supervisé pour définir les

règles floues, le NEFCON utilise un algorithme d'apprentissage non supervisé avec le calcul de l'erreur de sortie. Les deux modèles emploient la rétropropagation afin de définir les sous-ensembles flous. Le NEFPROX est le premier système interprétable et lisible, il est plus rapide que le modèle ANFIS, mais ce dernier donne de meilleurs résultats en approximation.

Architecture SANFIS (Self-adaptive neuro-fuzzy inference systems)

Une autre structure connexionniste basée sur la fonction floue de base (FBF), a été développée récemment [22]. D'après la Figure 1.12, les entrées de ces 3 modèles sont les mêmes, alors que les sorties sont différentes, ces modèles sont utilisés selon le type d'applications :

• **Le type I** (modèle de Mamdani) est linguistiquement compréhensible, car il utilise les ensembles flous dans les précédences et les conséquences des règles floues. Avec sa meilleure capacité de généralisation, ce modèle est plus convenable pour la commande et les systèmes experts.

• **Le type II** est une version simplifiée du premier. Il réduit le nombre de paramètres et évite le processus de défuzzification, ce type est généralement utilisé en classification.

• **Le type III** réduit encore le nombre des règles floues pour modéliser les systèmes évolués ; il est utilisé dans les fonctions d'approximation non linéaires, la commande dynamique et le traitement du signal.

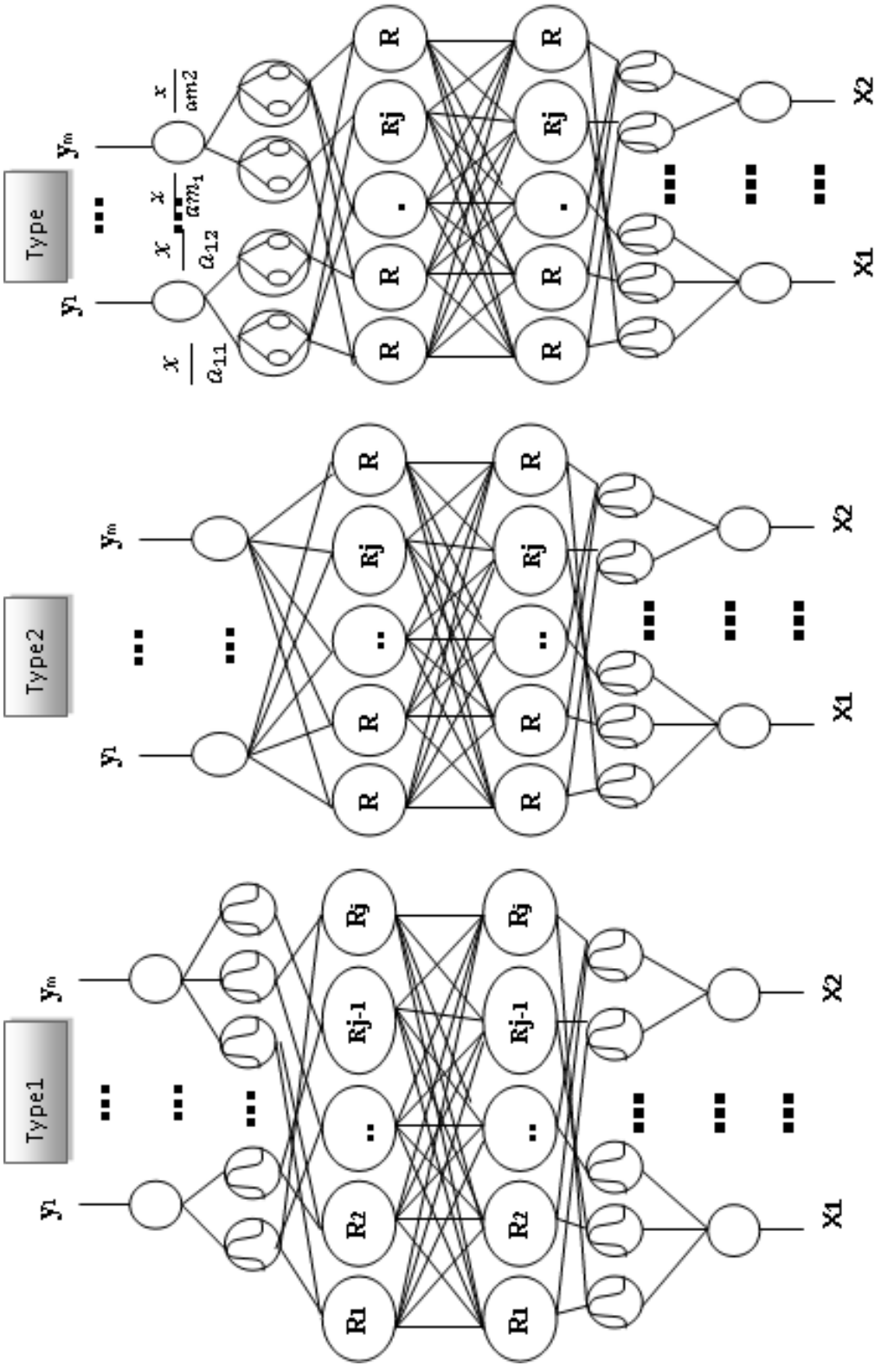


FIGURE 1.12 – *Les trois différents types de SANFIS.*

1.5 Adaptive Network Fuzzy Inference System ANFIS

ANFIS proposé par Jang en 1993, est un réseau neuronal multicouche dont les connexions ne sont pas pondérées ou tous les poids sont égaux à 1 [8], il combine les avantages de deux approches. Le réseau obtenu possède une excellente capacité d'apprentissage grâce aux réseaux de neurones et une interprétation linguistique des variables via la logique floue. ANFIS implémente un système flou de type Sugeno de premier ordre ; il applique la règle de la forme TSK Takagi Sugeno et Kang dans son architecture [23]. Cette règle produit directement des sorties nettes, car elle utilise des polynômes comme conséquences de la règle [15].

1.5.1 Les caractéristique du réseau ANFIS

ANFIS ne prend en charge que les systèmes de type Sugeno et ceux-ci doivent avoir les propriétés suivantes [24] :

- Être des systèmes de type Sugeno de premier ou de zéro ordre.
- Avoir une sortie unique, obtenue en utilisant la défuzzification moyenne pondérée. Tous les MF de sortie doivent être du même type linéaire ou constant.
- Pas de partage de règles. Différentes règles ne peuvent pas partager le même MF de sortie, à savoir que le nombre de MF de sortie doit être égal au nombre de règles.
- Avoir un poids unitaire pour chaque règle.

1.5.2 Architecture du réseau ANFIS

Une architecture classique peut être décrite de la manière suivante [8] (figure 1.13) ; les nœuds circulaires sont fixes tandis que les nœuds carrés ont des paramètres à apprendre.

1. La première couche comporte autant de neurones qu'il y'a de sous-ensembles flous dans le système d'inférence représenté. Chaque neurone calcule le degré de vérité d'un sous ensemble flou particulier par sa fonction de transfert. La seule restriction sur le choix de cette fonction concerne sa dérivabilité. En retrouve dans la littérature, l'utilisation, de fonctions gaussiennes et les paramètres modifiables sont le centre et la pente de la gaussienne (variance). La fonction d'activation des neurones i de la première couche :

$$f_i^1 = \mu A_i(X) \tag{1.6}$$

Tel que x est l'entrée au neurone i, et Ai est un sous ensemble floue correspondant au variable x. En d'autres termes, f1 i est la fonction d'appartenance du Ai et il indique le degré auquel donné x satisfait le quantifier Ai. Nous choisissons $\mu A_i(x)$ pour être en

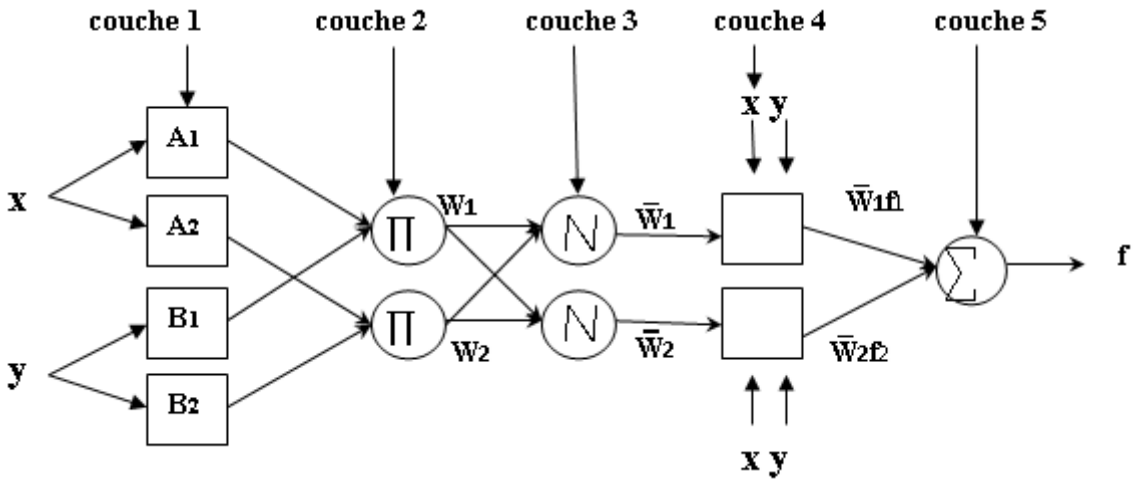


FIGURE 1.13 – *Architecture du réseau ANFIS.*

forme de (Gaussien, triangle, trapézoïdal) avec le maximum égal à 1 et le minimum égal à 0, tel que les fonctions généralisées de ces formes est :

Fonction triangulaire :

$$\mu_A(x) = max\left[min\left[\frac{(x-a)}{(b-a)}, \frac{(c-x)}{(c-b)}\right], 0\right] \tag{1.7}$$

Fonction trapézoïdale :

$$\mu_A(x) = max\left[min\left[\frac{(x-a)}{(b-a)}, 1, \frac{(d-x)}{(d-c)}\right], 0\right] \tag{1.8}$$

Fonction gaussienne :

$$\mu_A(x) = exp\left[-\frac{(x-m)^2}{(2\sigma^2)}\right] \tag{1.9}$$

Où a , b , c , s est l'ensemble des paramètres. Pendant que les valeurs de ces paramètres changent, les fonctions en forme précédente changent en conséquence, de ce fait présenter de diverses formes de fonction d'appartenance sur la variable linguistique Ai. Les paramètres dans cette couche désignés sous le nom des paramètres de fonction d'appartenance.

2. La deuxième couche sert à calculer le degré d'activation des prémisses. Les neurones de cette couche représentent chacun la prémisse d'une règle. Ils reçoivent en entrée le degré de vérité des différents sous-ensembles flous composant cette prémisse et ont en charge le calcul de son propre degré de vérité. Les fonctions d'activation utilisées pour ces neurones dépendant des opérateurs présents dans les règles (ET ou OU). La fonction d'activation des neurones i de la première couche :

$$W_k = \mu_A i(x) * \mu_B j(y) \tag{1.10}$$

Ou k : représente le nombre de règle, i : représente le nombre de partition de x, et j : le nombre de partition de y.

3. La troisième couche Chaque neurone dans cette couche est un neurone de cercle noté N. le i^{mme} neurone calcule le rapport entre i^{mme} poids de règles et la somme de tous les poids des règles. Cette opération est appelée la normalisation des poids.

$$\overline{W}_K = W_K / (\sum W_i) \tag{1.11}$$

L'ensemble des sorties de cette couche seront appelées les poids normalisés.

4. La quatrième couche sert à déterminer les paramètres la partie conséquence des règles (p, q, r). La fonction de chaque neurone dans cette couche est la suivante

$$f_k^4 = \overline{W}_k * f_k = \overline{W}_k * (p_K x + q_k y + r_k) \tag{1.12}$$

Où W_K est la sortie de la troisième couche, et p ,q , r est l'ensemble des paramètres.

5. La couche de sortie contient un seul neurone dans cette couche, est un neurone de cercle noté S qui calcule la sortie globale comme addition de tous les signaux entrants, c'est-à-dire :

$$f^5 = \sum_k \overline{W}_k * f_k^4 \tag{1.13}$$

TABLE 1.2 – *les différentes couches d'un système ANFIS*

Les différentes couches	**Type de couches**	**Le nombre de neurone dans la couche**
Couche 0	Les entrées	N
Couche 1	Les valeurs	(p,n)
Couche 2	Les règles	pn
Couche 3	La normalisation	pn
Couche 4	La normalisation des fonctions	pn
Couche 5	Somme	1

Tel que :

n : le nombre des entrées.

p : le nombre des sous-ensembles flous d'entrée (partition flou).

Noter que les neurones dans ANFIS ont différentes structures :

— Les valeurs [fonction d'appartenance définie par différentes formes].

— Les règles [habituellement produit].

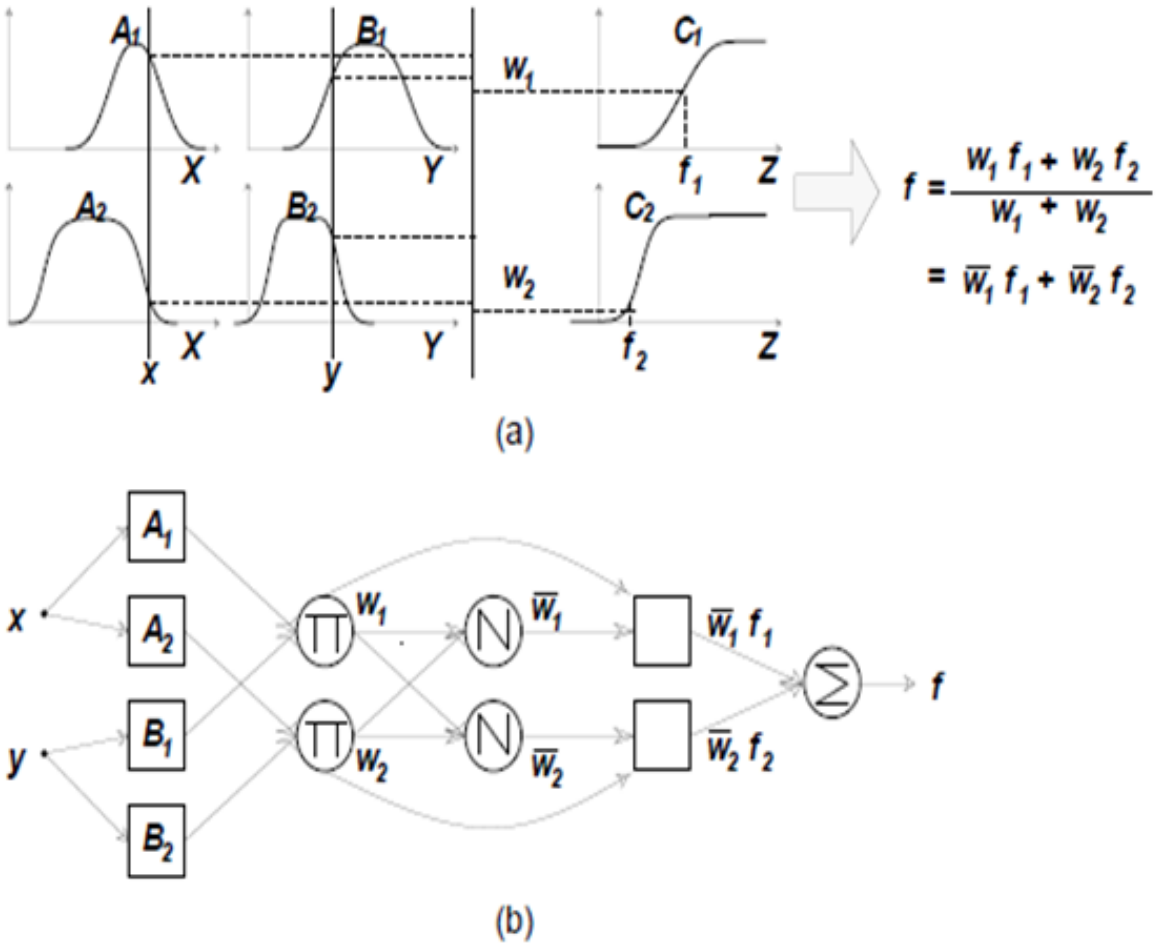


FIGURE 1.14 – *(1) Type-1 raisonement flou, (b) equivalent ANFIS (type-1 ANFIS).*

— La normalisation [division de somme et d'arithmétique].
— Les fonctions [régressions linéaires et multiplication avec w, tel que w est la normalisation du poids w].
— La sortie [Somme Algébrique].

L'algorithme d'inférence floue en tant que fondement d'ANFIS est une méthode dans laquelle des règles floues sont utilisées pour déduire une nouvelle conclusion approximative d'ensemble flou tout en prenant l'ensemble flou comme prémisse. Le système d'inférence floue (SIF) est principalement appliqué aux cas où les systèmes sont difficiles à modéliser avec précision ou la description des problèmes étudiés est vague et équivoque [25]. Un ANFIS est utilisé pour mapper les caractéristiques d'entrée aux fonctions d'appartenance d'entrée (MF), l'entrée MF à un ensemble de règles if-then, les règles à un ensemble de caractéristiques de sortie, les caractéristiques de sortie aux MF de sortie et les MF de sortie à une valeur unique sortie ou une décision associée à la sortie (figure 1.4).

1.5.3 Les avantages du réseau ANFIS

Le réseau ANFIS a plusieurs avantages, on peut citer :

— Exploitation de la connaissance disponible, grâce à la base de règles.

— Réduction de la taille de la base de règles : il suffit d'avoir des règles générales, les détails seront fournis par le RN.

— Réduction de la complexité de l'apprentissage : le RN doit simplement apprendre les cas particuliers ou les exceptions, pas le problème complet.

— Efficacité immédiate dès le début de l'apprentissage et possibilité d'éviter des comportements initiaux erratiques.

1.5.4 Apprentissage du réseau ANFIS

Deux étapes sont nécessaires, à savoir : L'apprentissage de structure qui permet de déterminer la structure de réseau appropriée, et un apprentissage paramétrique effectué pour ajuster les fonctions d'appartenance et les paramètres qui en découlent.

1.5.4.1 Apprentissage de la structure

L'apprentissage de la structure Permet de déterminer la structure appropriée du réseau, c'est-à-dire le meilleur partitionnement de l'espace d'entrée (nombre de fonctions d'appartenance pour chaque entrée et nombre de règles). Si la dimension en entrée est trop grande ; alors les nombres de règles auront une croissance exponentielle. Ainsi, la détermination d'une structure appropriée devient un problème important, de sorte que des techniques de clustering sont appliquées pour surmonter ce problème ; ANFIS propose trois approches pour identifier la structure :

— le partitionnement en grille (grid partioning)

— Le regroupement soustractif (substractive clustering)

— Les C moyens flous (le fuzzy c-means).

a) Le partitionnement en grille

L'approche de partitionnement en grille est utile si le nombre d'entrées n'est pas supérieur à 6 ou 7. Si le nombre d'entrées est trop élevé, cette méthode ne pourra pas être utilisée car la mémoire requise sera insuffisante lors de l'utilisation de MATLAB.

b) Le regroupement soustractif

Le regroupement soustractif a été introduit par [26]. Pour cette méthode, les points de données doivent être redimensionnés à [0,1] dans chaque dimension. Chaque point

de données $z_j = (x_j, y_j)$ se voit attribuer un potentiel P_j, en fonction de son emplacement, à tous les autres points de données :

$$P_i^* = \sum_{j=1}^{n} e^{-\alpha * ||x^i - x^j||} \ , \ \ \alpha = \frac{y}{r_a} \tag{1.14}$$

P_i^* est la valeur potentielle de la donnée i en tant que centre de cluster.

α est le poids entre la donnée i et la donnée j

x est le point de données.

Y est une variable (généralement définie sur 4)

r_a est une constante positive appelée rayon du cluster.

Le potentiel d'un point de données à être un centre de cluster est plus élevé lorsque plusieurs points de données sont plus proches. Le point de données avec le potentiel le plus élevé, noté P_i^* est considéré comme le premier centre de cluster $c_1 = (d_1, e_1)$. Le potentiel est ensuite recalculé pour tous les autres points à l'exclusion de l'influence du premier centre de cluster selon :

$$P_i^* = P_i^* - P_k^* \ e^{-\beta \|x^i - c^k\|^2} \ , \ \beta = \frac{4}{r_b^2} \ , \ r_b = r_a * \eta \tag{1.15}$$

P_i^* est la nouvelle valeur potentielle de la i eme donnée.

P_k^* est la valeur potentielle des données i en tant que centre de cluster.

c est le centre de cluster des données.

β est le poids de la i eme donnée au centre du cluster.

r_i est la distance entre le centre du cluster.

η est le facteur d'annulation (quash factor).

Encore une fois, le point de données avec le potentiel le plus élevé P^*k est considéré comme le prochain centre de cluster c_k, si :

$$\frac{d_{min}}{r_a} + \frac{(P_k^*)}{(P_1^*)} \geq 1 \tag{1.16}$$

Avec d_{min} est la distance minimale entre c_1 et tous les centres de cluster précédemment trouvés, le point de données est toujours accepté comme le prochain centre de cluster c_1. D'autres itérations peuvent ensuite être effectuées pour obtenir de nouveaux centres de cluster c_2. Si un centre de cluster possible ne remplit pas les conditions décrites ci-dessus, il est rejeté en tant que centre de cluster et son potentiel est défini sur o. Le point de données avec le potentiel le plus élevé suivant P_k^* est sélectionné comme nouveau centre de cluster possible et testé. Le clustering se termine si la condition suivante est remplie :

$$P_k^* < EP_i^* \tag{1.17}$$

Où : ϵ est le taux de rejet.

Des valeurs indicatives de paramètres pour $r_a, \eta, \epsilon \; et \; \epsilon^*$ ont été suggérées par [27]. Chaque centre de cluster est considéré comme une règle floue qui décrit le comportement du système de la distance aux centres de cluster définis :

$$\mu_j^{ik} = e^{-\alpha \| x_j^i - c_j^k \|^2} \tag{1.18}$$

c) Les C Moyens Flous(Fuzzy C Means)

Les C moyens flous (FCM) est une méthode de regroupement dans laquelle un ensemble de données est regroupé en N clusters où chaque point de données appartient à deux clusters ou plus à un certain degré (voir figure 1.15). Par exemple, un point de données situé près du centre d'un cluster aura un haut degré d'appartenance à ce cluster, et un autre point de données qui se trouvera loin du centre d'un cluster aura un faible degré d'appartenance à ce cluster. Cette méthode développée par Dunn en 1973 [28] et améliorée par Bezdek en 1981 [29] est fréquemment utilisée dans la reconnaissance des formes. Elle est basé sur la minimisation de la fonction objectif suivante [30] :

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} U_{ij}^m \; \| \; x_i - c_j \; \|^2 \quad 1 \leq m < \infty \tag{1.19}$$

Où m est un nombre réel supérieur à 1, u_{ij} est le degré d'appartenance de x_i dans le cluster j, x_i est le ième des données mesurées en dimension d, c_j est le centre de dimension d du cluster, et $\| * \|$ est la distance euclidienne entre les données mesurées et le centre. Le partitionnement flou est réalisé grâce à une optimisation itérative de la fonction objective présentée ci-dessus, avec la mise à jour de l'appartenance u_{ij} et des centres de cluster c_j par :

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \; , \; c_j = \frac{\sum_{i=1}^{N} U_{ij}^m \; X_i}{\sum_{i=1}^{N} U_{ij}^m} \tag{1.20}$$

N est le nombre de point de donnes et C représente le nombre de clusters

Algorithme du C Moyens flous (Fuzzy C means)

Soit $X = x_1, x_2, x_3 ..., x_n$ l'ensemble des points de données et

$V = v_1, v_2, v_3 ..., v_c$ l'ensemble des centres

1) sélectionnez les centres de cluster 'c' au hasard aléatoirement.

2) Calculer l'appartenance floue μ_{ij} en utilisant :

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} * \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{(m-1)}}} \tag{1.21}$$

3) Calculer les centres flous $'v'_j$ en utilisant :

$$v_j = \frac{\sum_{i=1}^{n}(U_{ij})^m x_i}{\sum_{i=1}^{n}(U_{ij})^m}, \ \forall_j = 1, 2, \ldots..c) \tag{1.22}$$

4) Répétez les étapes 2) et 3) jusqu'à ce que la valeur 'J' minimale soit atteinte ou $||U(k+1) - U(k)|| < \beta$.

Où : 'k' est l'étape d'itération.

'β' est le critère de terminaison entre [0, 1].

'$U = (u_{ij})n * c$' est la matrice d'appartenance floue.

'J' est la fonction objectif.

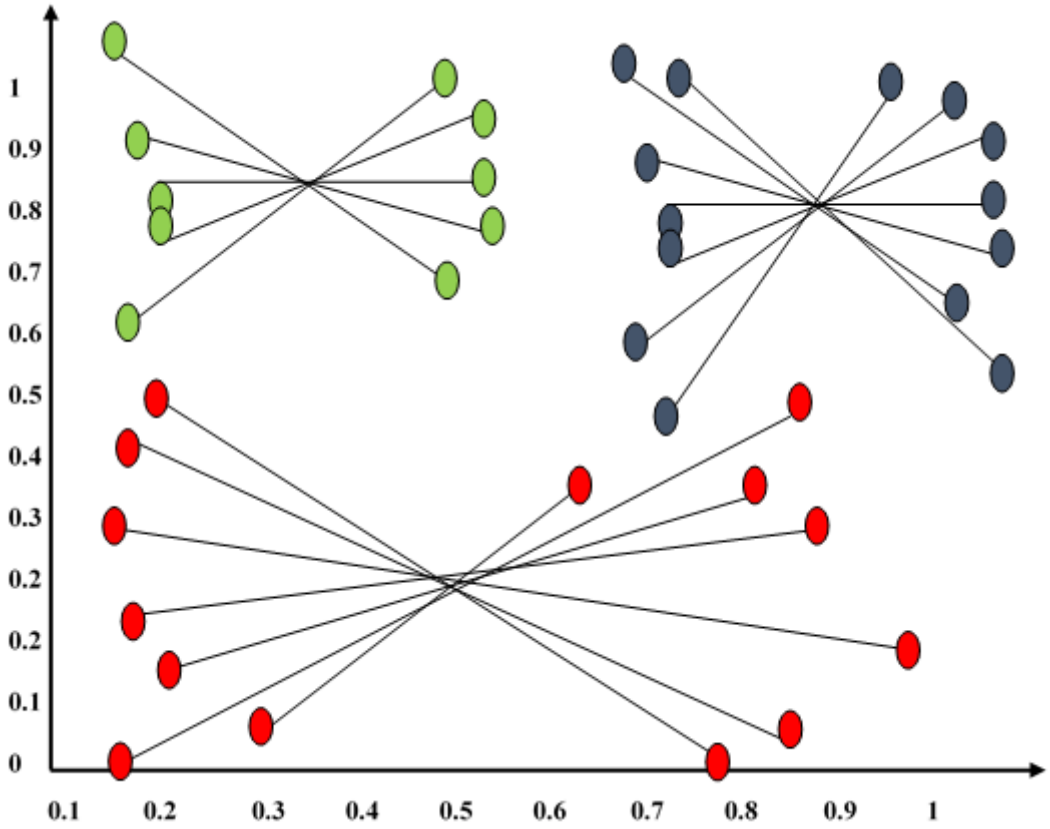


FIGURE 1.15 – *Résultats du C Moyens Flous FCM.*

1.5.4.2 Apprentissage paramétrique

L'apprentissage paramétrique réalisé pour ajuster les paramètres antécédents et conséquents de manière à minimiser une fonction objectif spécifiée. La complexité de calcul du LSE est généralement supérieure à celle de la méthode GD. Cependant, pour atteindre un niveau de performance prescrit, l'estimation par les moindres carrés (LSE) est généralement beaucoup plus rapide ; par conséquent, en fonction des ressources disponibles et du niveau de performance requis, cinq types de règles d'apprentissage

hybrides combinant l'algorithme de rétro-propagation du gradient et la méthode des moindres carrés à différents degrés ont été proposé par [31]

1. Tous les paramètres ne sont mis à jour que par l'algorithme de rétro-propagation du gradient.
2. Une seule application du LSE : les paramètres non linéaires sont fixés tandis que les paramètres linéaires sont identifiés par une seule application du LSE.
3. Une seule application du LSE suivie par GD : En premier lieu, les paramètres résultants sont obtenus en appliquant l'estimation des moindres carrés LSE une seule fois, puis l'algorithme de rétro-propagation du gradient met à jour tous les paramètres.
4. LSE séquentiel utilisant un filtre de Kalman étendu pour mettre à jour tous les paramètres.
5. Apprentissage hybride : combine l'algorithme de rétro-propagation du gradient et LSE pour trouver un ensemble réalisable d'antécédents et de paramètres conséquents.

Le choix de l'une des méthodes ci-dessus doit être fondé sur un compromis entre la complexité de calcul et la performance.

1.5.4.3 L'apprentissage hybride

L'apprentissage hybride est la combinaison de l'algorithme de rétro-propagation du gradient et la méthode des moindres carrés LSE

• **S1** représente les paramètres des ensembles flous utilisés pour la fuzzification dans la première couche du réseau ANFIS

• **S2** représente les coefficients des fonctions linéaires (les paramètres conséquentes) $S_2 = p_1, p_2, p_3, \ldots, q_1, q_2, q_3, \ldots r_1, r_2, r_3 \ldots$ (figure1.16)

L'apprentissage hybride passe par deux phases (tableau 1.3) :

— Le passage en avant : S_1 est fixe et S_2 est calculé en utilisant l'algorithme de Moindre carré de l'erreur (LSE).

— Passage en arrière : S_2 est fixe et S_1 est calculé en utilisant l'algorithme de Rétro – Propagation.

Cette procédure hybride est ensuite itérée jusqu'à ce que l'erreur de sortie soit réduite à un objectif souhaité ou jusqu'à ce qu'un nombre maximum de cycles ait été atteint.

La fonction à minimiser est l'erreur quadratique moyenne (RMSE) :

$$RMSE = \sqrt{\frac{1}{N \sum_{i=1}^{n} (d_i - 0_i)^2}} \tag{1.23}$$

TABLE 1.3 – *Les paramètres à ajuster d'un système ANFIS*

Les paramètres	**Passage en avant**	**Passage en arrière**
Paramètres des fonctions d'appartenance (non linéaire...)	Fixe	l'algorithme de rétro-propagation du gradient
Paramètres des coefficients (linéaire p,q,r...)	Estimation des moindres carrés LSE	Fixe

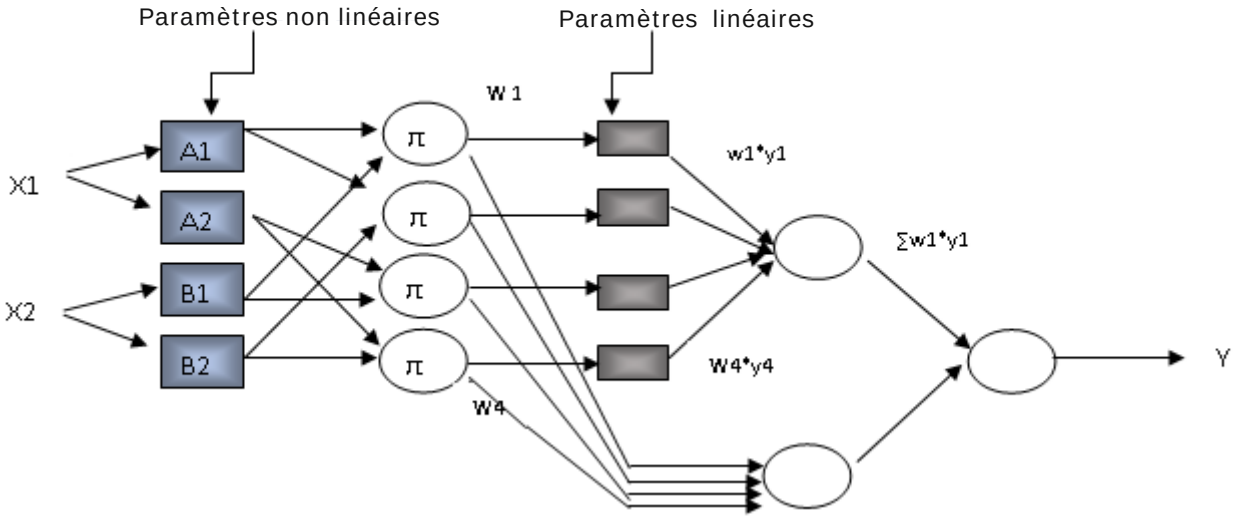


FIGURE 1.16 – *Méthode d'apprentissage Hybride.*

Où d_i est la sortie souhaitée et 0_i est la sortie de l'ANFIS pour le ième échantillon des données d'apprentissage et N est le nombre d'échantillons d'apprentissage.

1.6 Conclusion

La modélisation neuro-floue est une combinaison de logique floue et des réseaux de neurones qui tire parti des deux approches, le traitement des données imprécises ou vagues par logique floue et en même temps en introduisant l'apprentissage par le réseau neuronal. Dans ce chapitre nous avons exposé l'un des modèles flous les plus influents, Adaptive Network Based Fuzzy Inférence System ANFIS, la base de règle de ce dernier contient la règle floue si-alors du type de Takagi et Sugeno dans laquelle les parties conséquentes sont des fonctions linéaires d'entrées au lieu d'ensembles flous, réduisant le nombre de règles floues requises. L'identification du modèle flou comprend deux phases principales : l'identification de la structure et l'optimisation des paramètres. Ainsi, la détermination d'une structure appropriée à travers les techniques de regroupement tel que le regroupement soustractif et les C moyens flou.

Reconnaissance Automatique de la Parole

2

Sommaire

2.1 Introduction 36
2.2 Principe de base du signal vocal 37
2.2.1 Le système de production de la parole 37
2.2.2 La perception de la parole 40
2.3 Les caractéristiques du signal de parole 42
2.4 La reconnaissance automatique de la parole 43
2.4.1 Le prétraitement 44
2.4.2 L'extraction des caractéristiques 44
2.4.3 La correspondance des caractéristiques 54
2.5 Reconnaissance du locuteur 54
2.5.1 Domaines d'application de la reconnaissance du locuteur 55
2.5.2 Pourquoi la reconnaissance du locuteur ? 56
2.5.3 Informations spécifiques au locuteur dans le signal vocal 56
2.5.4 Identification Vs Vérification 58
2.5.5 Dépendance et indépendance du texte 59
2.5.6 Développement d'un système de reconnaissance du locuteur 61
2.5.7 Identification Vs Vérification 61
2.6 Etat de l'art 62
2.7 Conclusion 67

2.1 Introduction

Le son est un élément majeur permettant à l'être humain d'appréhender son environnement. Il est également, par le biais de la parole, le vecteur naturel de la communication humaine. Présent dans de nombreux documents multimédias, il est, de ce fait, porteur d'une information précieuse pour leur compréhension. Il existe des différences fondamentales entre la voix et la parole. La voix est l'ensemble des bruits vocaux qu'un humain peut produire. Il se matérialise par des ondes mécaniques se propageant dans l'air et oscillant à des fréquences allant de 40 Hz à 8 kHz. Pour rappel, plus la fréquence est élevée, plus l'onde vibre rapidement, produisant des sons aigus. La parole est désormais un plus petit ensemble de sons vocaux dédiés à la communication orale dans une langue donnée. La plage de fréquence pour la production d'un message intelligible est plus étroite, avec des valeurs comprises entre 300 Hz et 3 kHz 4. Dans le passé, les téléphones analogiques transmettaient des informations sur ces fréquences, provoquant par conséquent un effet de voix nasale.

En cette ère de progrès technologique, les nouvelles techniques évoluent pour une meilleure interaction homme-machine. Initialement, cette envie d'interagir avec les machines était la principale raison de l'invention de périphériques d'entrée-sortie tels que clavier, souris, moniteur, imprimante, joystick, scanner, écrans tactiles et trackball, etc. Cependant, aucune de ces inventions susmentionnées n'est capable de fournir une interaction verbale entre l'homme et les machines, qui est le moyen naturel de communication pendant de nombreux siècles. Ce manque de communication avec les machines utilisant la parole conduit les chercheurs à inventer les systèmes de traitement de la parole pour une meilleure interaction homme-machine utilisant des signaux vocaux.

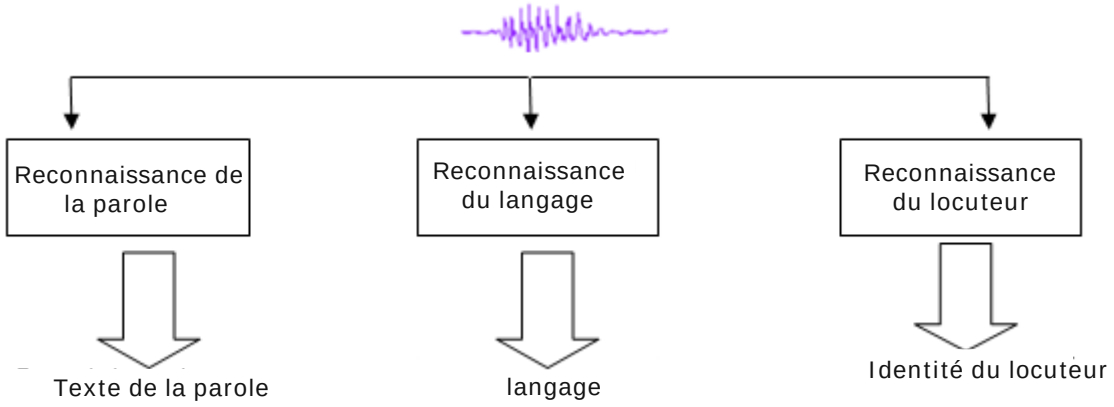


Figure 2.1 – *Extraction automatiquement des informations transmises dans le signal vocal.*

Trois principaux champs de reconnaissance à partir du signal vocal, qui sont les plus intéressants et ont été étudiés depuis plusieurs décennies, sont la reconnaissance vocale, la reconnaissance de la langue et la reconnaissance du locuteur (figure 2.1).

Dans cette thèse, nous concentrons notre attention sur le domaine de la reconnaissance du locuteur. La reconnaissance vocale par machine peut être définie comme la conversion automatique du signal vocal humain en forme textuelle par la machine sans aucune intervention humaine, fournissant une transcription / interprétation de ce qu'un humain parle pendant que la machine écoute. La plupart des applications en reconnaissance de la parole peuvent être regroupées en quatre catégories : commande et contrôle, accès à des bases de données ou recherche d'information, dictée vocale et transcription automatique de la parole.

Le but de ce chapitre est de fournir des informations générales sur la reconnaissance automatique de la parole et ses composants ainsi que la production et la perception de la parole humaine. Nous décrivons aussi la reconnaissance automatique du locuteur (automatic speaker recognition) ASR et ses méthodes d'évaluation des performances et d'extraction de caractéristiques, et nous explorons les travaux réalisés dans ce domaine.

2.2 Principe de base du signal vocal

Nous utilisons la parole tous les jours presque inconsciemment, mais une compréhension des mécanismes sur lesquels elle repose aidera à clarifier comment le cerveau traite l'information et conduira également au développement de dispositifs vocaux plus humains grâce à l'imitation de ces fonctions par des ordinateurs. La parole est fortement affectée par les accents, l'articulation, la prononciation, la rugosité, l'état émotionnel, le sexe, la hauteur, la vitesse, le volume, le bruit de fond et les échos [32] Pour toute technique de traitement de la parole, l'entrée est un signal de parole. Par conséquent, comprendre les bases des signaux vocaux a beaucoup d'importance. Les techniques de paramétrisation de la parole utilisées en RAL se basent généralement sur les paramètres extraits visant à caractériser l'appareil de production de la parole (Un modèle de production) ; ou bien la conception de paramètres s'inspirant de la perception humaine de la parole (Un modèle de perception) [33]

Dans cette section, les bases des signaux vocaux sont présentées. Tout d'abord, Pour mieux comprendre les principes de bases des techniques de paramétrisation utilisées en RAL, on commence par présenter le modèle de production de la parole ainsi que le modèle de perception. Par la suite, on fait le lien avec les paramètres acoustiques utilisés.

2.2.1 Le système de production de la parole

La production de la parole est basée sur des phénomènes complexes, largement étudiés pour leur rôle dans la cognition et la communication humaines. Ici, nous nous concentrons sur les aspects physiologiques. La production des sons de la parole

commence dans le cerveau. Après la création du message et de la structure lexico-grammaticale dans notre esprit, nous avons besoin d'une représentation de la séquence sonore et d'un certain nombre de commandes qui seront exécutées par nos organes de parole pour produire l'énoncé. Un plan phonétique et un plan moteur sont nécessaires. Ensuite, la production physique des sons commence.

Ainsi, Pour générer du son, le diaphragme est déplacé soit pour renifler dans les poumons, soit pour émettre de l'air. Pendant que l'air est émis, il se déplace de la trachée (trachée) et du larynx, où les vibrations des cordes vocales peuvent influencer cet air. Ensuite, cet air circule dans la gorge (cavité pharyngée), et à travers la cavité buccale ou nasale dans la dernière. Maintenant, la disposition des articulateurs décide de la sortie sonore finale : velum (palais mou), langue, lèvres et mandibule (mâchoire).La cartographie de ces vibrations acoustiques à des positions séparées dans la cochlée de l'oreille humaine dépend de la fréquence du signal vocal. Par conséquent, la cochlée se comporte comme un banc de filtres qui mappe diverses composantes de fréquence du signal vocal à différentes positions[34].
La modulation est effectuée dans le tractus vocal, à travers les mouvements de nombreux articulateurs, tels que le velum, les dents, les lèvres et la langue. Les mouvements des articulateurs modifient la forme du tractus vocal, ce qui crée de nombreuses fréquences de résonance et, par conséquent, différents sons de parole (Figure 2.2).

Enfin, le processus d'articulation qui a lieu dans la bouche et c'est le processus par lequel nous pouvons différencier la plupart des sons de la parole. En bouche, on peut distinguer :

- La cavité buccale, qui agit comme un résonateur, et les articulateurs, qui peuvent être actifs ou passifs.
- Lèvres supérieures et inférieures.
- Dents supérieures et inférieures.
- Langue (pointe, lame, avant, arrière) .
- Toit de la bouche (crête alvéolaire, palais et velum).

L'ensemble des organes agit comme un filtre, considéré comme linéaire, dont la réponse impulsionnelle comporte des fréquences de résonance caractérisées par des pics, appelés formants, dans le spectre du signal de sortie. Le signal résultant est globalement non stationnaire mais peut être considéré comme stationnaire sur de très courtes périodes, de l'ordre de 20ms (signal pseudo-stationnaire). Les sons de la parole se distinguent les uns des autres en termes de lieu et de manière dont ils sont articulés. En plaçant simplement votre main sur votre gorge, vous pouvez distinguer deux types de sons.

A. Les sons voisés sont produits par la vibration des cordes vocales et correspondent

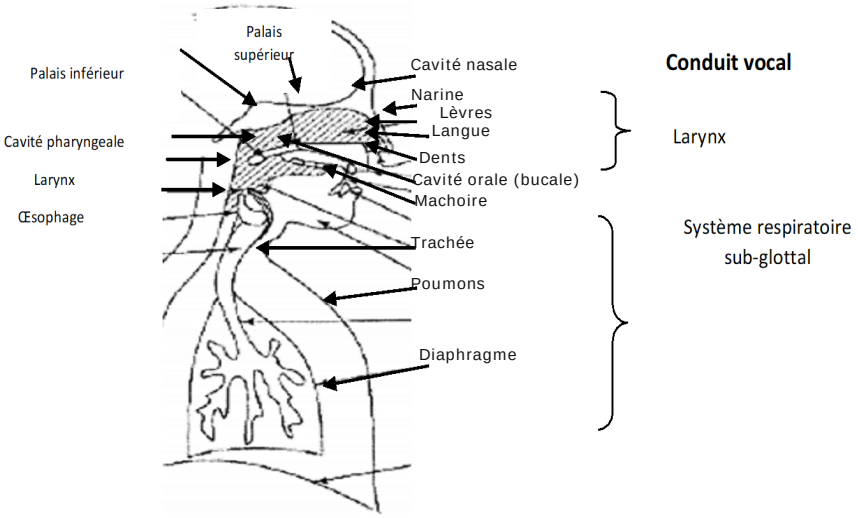


FIGURE 2.2 – *Système de production de la parole humaine.*

aux voyelles / a / et / o /. À gauche des courbes rouges de la figure 2.4, ces sons exprimés montrent des pics de résonance dans les basses et moyennes fréquences. La source d'excitation est modélisée par un train d'impulsions périodique, de fréquence dite de voisement Fo, qui correspond à la fréquence de vibration des cordes vocales, la fréquence fondamentale ou pitch ;où P représente la période de la fréquence fondamentale (en anglais, « pitch »).

$$u(n)^x x = \sum_{k}^{x} x\partial(n - k.P) \tag{2.1}$$

B. Les sons non voisés tels que la respiration sifflante / s / et explosif / p / ne nécessitent pas de plis à vibrer vocale. Dans ce cas, les positions de la langue et des lèvres conduiront à des distributions d'énergie totalement différentes. La source est modélisée par un bruit blanc. Cette représentation binaire de la production de la parole a été introduite par [35]. Elle est reprise sur la figure 2.3.

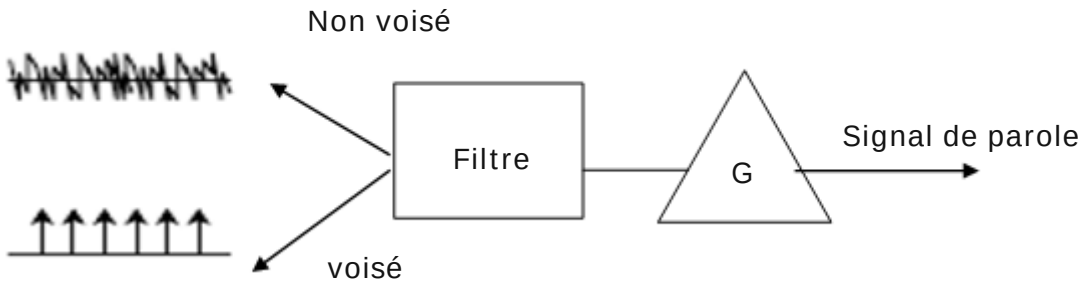


FIGURE 2.3 – *Modèle de production de la parole.*

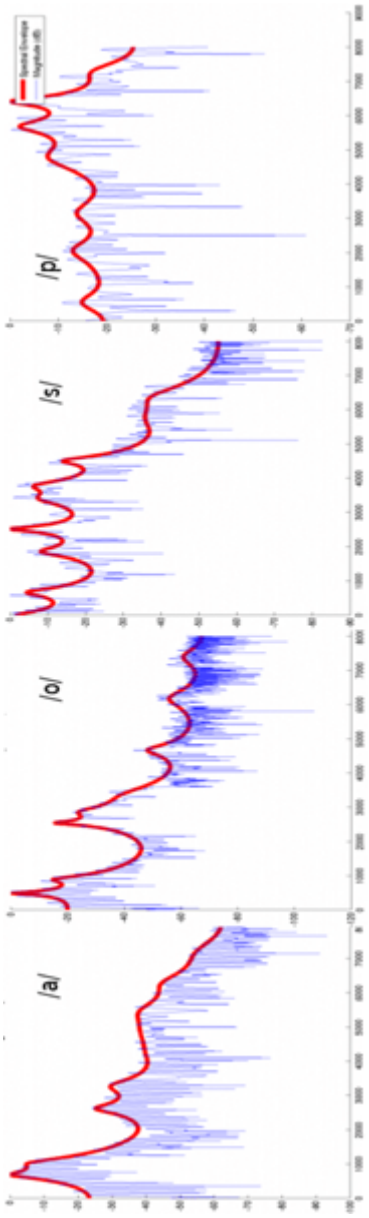


FIGURE 2.4 – *Fréquences vocales.*

2.2.2 La perception de la parole

On pourrait s'attendre à ce que les performances d'un système de reconnaissance automatique de la parole s'améliorent en utilisant les connaissances sur le système auditif humain. Le paragraphe ci-dessous présente brièvement le système auditif humain. Au niveau linguistique de la communication, l'idée se forme d'abord dans l'esprit du locuteur. L'idée est ensuite transformée en mots, phrases et phrases selon les règles grammaticales de la langue. Au niveau physiologique de la communication, le cerveau crée des signaux électriques qui se déplacent le long des nerfs moteurs.

Le système auditif tel que décrit par Munkong et Juang[36] est anatomiquement et fonctionnellement divisé en trois zones : l'oreille externe, l'oreille moyenne et l'oreille in-

terne. L'oreille externe est composée du pavillon (oreille externe, la partie que l'on peut voir) et du canal externe (ou méat). Le pavillon modifie le son entrant (en particulier les hautes fréquences) et le dirige vers le canal externe. L'effet de filtrage du pavillon humain sélectionne préférentiellement les sons dans la gamme de fréquences de la parole humaine. Il ajoute également des informations directionnelles au son. L'oreille moyenne est une cavité remplie d'air (cavité tympanique) qui couple le son de l'air aux fluides via une fenêtre ovale dans la cochlée. L'oreille interne est constituée d'un labyrinthe osseux rempli de liquide qui comporte deux parties fonctionnelles principales : le système vestibulaire et la cochlée.

La cochlée effectue une sorte d'analyse de fréquence qui n'est pas sans rappeler la transformée en ondelettes : elle contient la membrane basilaire et y est attachée des cellules ciliées reliées à des fibres nerveuses (neurones) se terminant dans le nerf auditif. Chaque composante de fréquence d'un stimulus (considérée comme composée de tons purs et sinusoïdaux) passant à travers le liquide cochléaire excite les cellules ciliées à d'autres endroits le long de la membrane basilaire. De ce fait, une autre transduction a lieu d'une excitation continue des cellules ciliées à des rafales de pics électriques à l'intérieur des neurones, où l'information est codée non pas dans l'amplitude des pics, mais dans leur schéma temporel. Chaque stimulus de ton pur excite non seulement une cellule ciliée, mais une gamme d'entre eux appelée « filtre auditif », comparable à une banque de filtres dans le traitement numérique du signal. Pourtant, cet ensemble de filtres n'est pas discret et semble s'assembler juste à temps qu'il existe un filtre auditif centré sur chaque fréquence d'apparition Les organes concernés sont représentés sur la figure 2.5.

Les humains perçoivent les sons de la parole différemment par rapport aux ordinateurs. Le spectre sonore n'est pas analysé par des méthodes mathématiques complexes[37]. Le cerveau est capable de distinguer avec succès les sons informatifs des sons bruyants. Ces sons bruyants peuvent être de différents types comme le bruit d'un énorme moteur. Presque tous les types de bruit sont ignorés par le cerveau humain, de sorte que les humains peuvent extraire les informations pertinentes. Un autre grand avantage du système de perception de la parole humaine est la capacité des humains à concentrer leur concentration sur une voix particulière à partir de plusieurs voix.

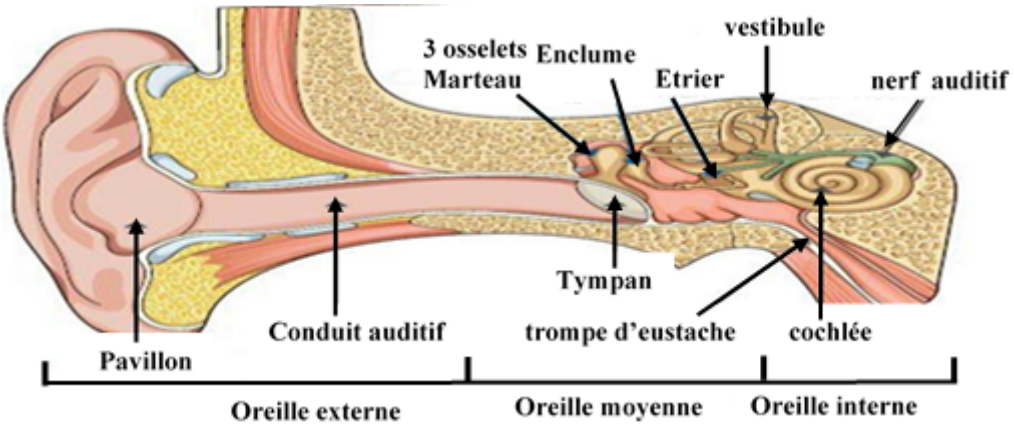


FIGURE 2.5 – *Structure de l'oreille humaine.*

2.3 Les caractéristiques du signal de parole

Dans le traitement du signal, un signal est une mesure observée d'un phénomène complexe [38] . La vitesse d'une voiture ou le prix d'une action sont tous deux des exemples de signaux dans différents domaines. Normalement, un signal est modélisé en fonction d'une variable indépendante. Habituellement, cette variable est le temps, et nous pouvons désigner ce signal par f (t). Cependant, un signal n'a pas besoin d'être fonction d'une seule variable. Par exemple, une image est un signal f (x ; y) qui désigne la couleur au point (x ; y) (voir figure 2.6). Une sinusoïde peut être utilisée pour décrire les sons de la parole. Cette forme pure est très utile lors de la description des propriétés des sons vocaux. Une sinusoïde est simplement un mouvement fluide de haut en bas lorsqu'elle est tracée en fonction du temps. Trois types de mesures peuvent être prises à partir d'une sinusoïde elles définissent complètement la forme de la forme d'onde : l'amplitude, la fréquence et la phase [39]. D'un point de vue mathématiques, il est difficile de modéliser le signal de parole, compte tenu de sa variabilité. Les caractéristiques importantes du signal non stationnaire lors de son traitement [39] :

La redondance : Le signal de parole est extrêmement redondant. Cette grande redondance lui confère une robustesse à certains types de bruits. De nombreuses recherches sont menées afin de rendre les systèmes de reconnaissance robustes aux bruits, mais les performances humaines sont encore loin d'être atteintes.

Variabilité du signal : Le signal de parole possède une très grande variabilité. Une même personne ne prononce jamais un mot deux fois de façon identique. La vitesse d'élocution peut varier, la durée du signal est alors modifiée. Toute altération de l'appareil phonatoire peut modifier la qualité de l'émission (exemple : rhume, fatigue...). De plus, la diction évolue dans le temps. La voix est modifiée au cours des étapes de la vie d'un être humain (enfance, adolescence, âge adulte...). La variabilité interlocu-

teur est encore plus accentuée. La hauteur de la voix, l'intonation et l'accent différent selon le sexe, l'origine sociale, régionale ou nationale. Un exemple pertinent de cette variabilité apparaît lorsque nous comparons la voix d'un locuteur originaire du Nord avec celle d'un locuteur originaire du sud de l'Algérie. Enfin, la parole est un moyen de communication où de nombreux éléments entrent en jeu, tels que le lieu, l'émotion du locuteur, la relation qui s'établit entre les locuteurs (stressante ou amicale). Ces facteurs influencent la forme et le contenu du message. L'acoustique du lieu (milieu protégé ou environnement bruité), la qualité du microphone, les bruits de bouche, les hésitations, les mots hors vocabulaire sont autant d'interférences supplémentaires sur le signal de parole.

Les effets de coarticulation : La production parfaite d'un son suppose un positionnement précis des organes phonatoires. Le déplacement de ces organes est limité par une certaine inertie mécanique. Les sons émis subissent alors l'influence de ceux qui les précèdent où les suivent. Ces effets de coarticulation sont un facteur de variabilité supplémentaire important du signal de parole.

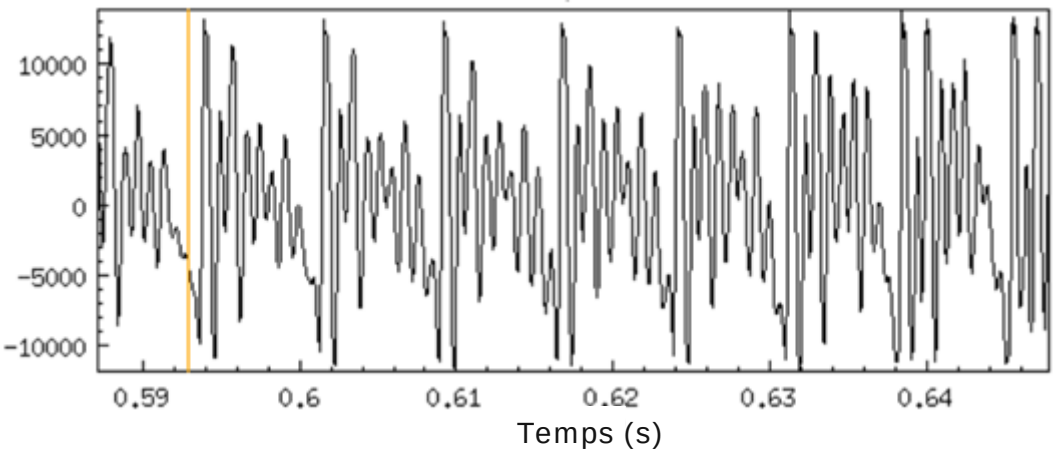


Figure 2.6 – *Signal vocal.*

2.4 La reconnaissance automatique de la parole

Divers processus impliqués dans un système de reconnaissance de la parole sont représentés sur la figure 2.7, Qui montre qu'il existe quatre processus de ce type, à savoir le prétraitement, l'extraction des caractéristiques, le générateur de modèles (apprentissage) et la correspondance de caractéristiques (reconnaissance).

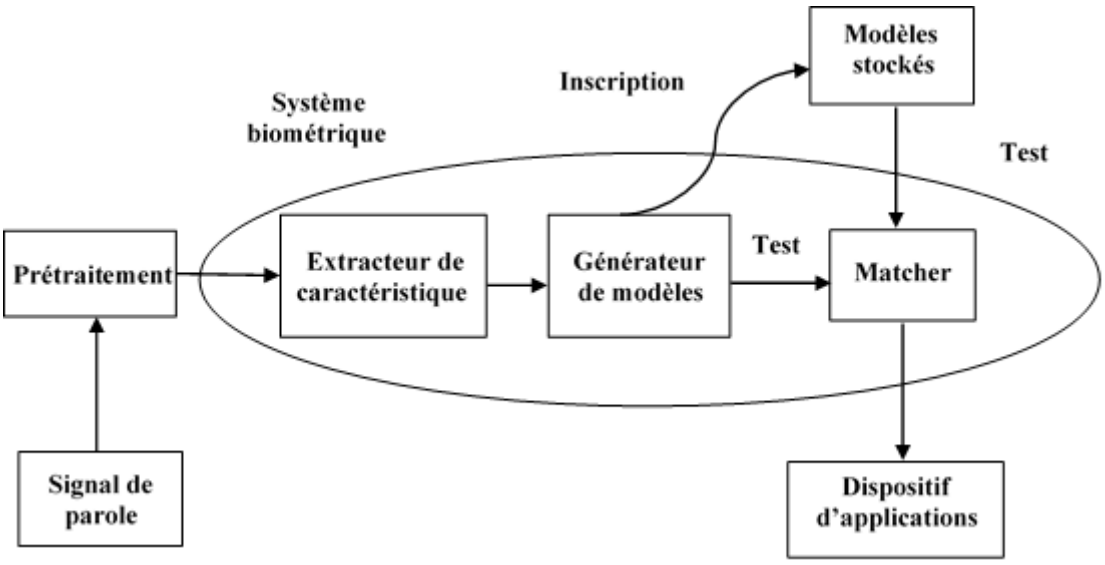


FIGURE 2.7 – *Processus du système de reconnaissance de la parole.*

En fait, le prétraitement et l'extraction de caractéristiques font partie du traitement frontal et le générateur de modèles (apprentissage) enregistre les formes en les modélisant et en générant leurs modèles qui sont stockés dans la base de données. Ensuite, la reconnaissance ou on déclare reconnue la forme la plus probable ou la plus proche de celle représentée au sens d'une distance. Elle correspond à la phase d'identification des formes inconnues par comparaison avec les reconnaissances mémorisées par une méthode de classification

2.4.1 Le prétraitement

Diverses étapes de prétraitement doivent être appliquées au signal avant l'extraction de caractéristiques, parfois considérée comme faisant partie de l'extraction de caractéristiques. Le signal vocal doit être prétraité pour réduire la complexité du calcul. Il permet de réduire la qualité d'information du signal de parole et de facilité la classification. L'échantillonnage permet de passer des données analogiques aux données numériques.

2.4.2 L'extraction des caractéristiques

Le processus d'extraction de certaines informations de la voix d'une personne qui peut être utilisée pour représenter cette dernière en particulier est appelé extraction de caractéristiques. L'objectif de cette phase est d'extraire des coefficients représentatifs du signal de parole. Ces coefficients sont calculés à intervalles temporels réguliers. En simplifiant les choses, le signale de parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censés modé-

liser et doivent extraire le maximum d'informations utiles pour la reconnaissance. Ces caractéristiques devraient contenir des propriétés telles que :

— Stabilité dans le temps
— Fréquent et naturel dans le signal vocal
— Sensibilité minimale au mimétisme
— Facilement mesurable
— Moins de variabilité environnementale
— Grande variabilité inter-locuteur et petite variabilité intra-locuteur.

Il ressort que les seuls types de paramètres vraiment pertinents et utilisables efficacement sont les paramètres de l'analyse spectrale, les paramètres dynamiques et éventuellement les paramètres prosodiques. Combiner ces différentes sources d'information peut conduire à des performances supérieures à celles que l'on obtient en les utilisant séparément [40] : Cette séparation est rendue possible grâce à un filtre déconvolutif.

• **Paramètres de l'analyse spectrale :** La représentation du signal par les coefficients Mel Cepstres (MFCC) est très utilisée dans les tâches de reconnaissance du locuteur. Ces coefficients caractérisent bien la forme du spectre et permettent de séparer l'influence de la source du signal vocal de celle du conduit vocal.

• **Paramètres dynamique :** La prise en compte d'une information de type dynamique est un facteur d'amélioration des performances d'identification du locuteur. Les dérivées du premier et du second ordre, appelées aussi coefficients de delta et delta-delta, permettent d'introduire une information concernant le contexte temporel d'une trame courante. D'après Harb[41] l'utilisation de ces paramètres reste l'approche la plus populaire actuellement en raison de la simplicité pour sa mise en œuvre et l'amélioration des performances que nous pouvons observer.

• **Paramètres prosodiques :** Ces paramètres caractérisent en grande partie le style d'élocution d'un locuteur. L'énergie contient l'information liée au niveau acoustique moyen du signal. Ces paramètres s'avèrent fragiles en pratique et ne permettent pas, à eux seuls, de discriminer de manière fiable les locuteurs. En conséquent, ils sont souvent associés aux paramètres de l'analyse spectrale[42].

Parmi les coefficients les plus utilisés et qui représentent au mieux le signal de la parole en reconnaissance de la parole, nous trouvons les coefficients cepstraux, appelés également cepstres. Les deux méthodes les plus connues pour l'extraction de ces cepstres sont l'analyse spectrale (Mel-Scale Frequency Cepstral Coefficients MFCC) et l'analyse paramétrique (codage prédictif linéaire LPC), le signal de parole est transformé en une série de vecteurs calculés pour chaque trame.Aussi le codage Neuro-Prédictif (Neural Predictive Coding NPC) est introduit et représente une extension du

codage LPC, une méthode de codage temporelle mais qui extrait les caractéristiques non linéaire d'un phonème[43].

Il existe d'autres types de coefficients qui sont surtout utilisés dans les milieux bruités, tel que les coefficients PLP (Perceptual Linear Predictive)[44], ces coefficients permettent d'estimer les paramètres d'un filtre autorégressif en modélisant au mieux le spectre auditif. Dans notre travail, nous nous sommes limités à l'utilisation des coefficients MFCC, ces paramètres ont montrés une bonne représentation des aspects perceptuels du spectre de parole.

2.4.2.1 Les coefficients Cepstraux de Fréquence de Mel MFCC

Pour développer un système de reconnaissance de la parole robuste, un mécanisme est nécessaire non seulement pour représenter avec précision le signal acoustique de la parole donné, mais il doit également être fiable. Heureusement, d'énormes recherches sont effectuées dans ce domaine de l'acoustique du signal. La recherche a conduit à une méthode éprouvée pour extraire des caractéristiques uniques à la parole, les Coefficients Cepstraux Mel-Frequency[45].

Les MFCC sont utilisés dans de nombreuses recherches[46][47] car ils fournissent des caractéristiques robustes qui conduisent à un taux de reconnaissance de la parole de haute précision, mais les performances sont largement affectées par le bruit de fond.

Dans cette sous-section, cette méthode d'extraction de caractéristiques largement utilisée pour la reconnaissance du la parole est décrite. Après le prétraitement du signal de parole, les caractéristiques sont extraites sous la forme de coefficient cepstral de fréquence Mel. MFCC représente le signal vocal, sur la base de la perception. Les caractéristiques peuvent être obtenues en utilisant une transformée de Fourier ou une transformée cosinus discrète du signal de parole. La principale différence est que, dans la première, les plages de fréquences sont placées sur une échelle logarithmique appelée échelle de Mel imitant approximativement la réponse du système auditif mieux que les plages de fréquences positionnées linéairement de FFT DCT. L'objectif le plus important du processeur MFCC est d'imiter le comportement de l'oreille humaine. La procédure typique d'extraction de caractéristiques est illustrée à la Figure 2.8 [48].

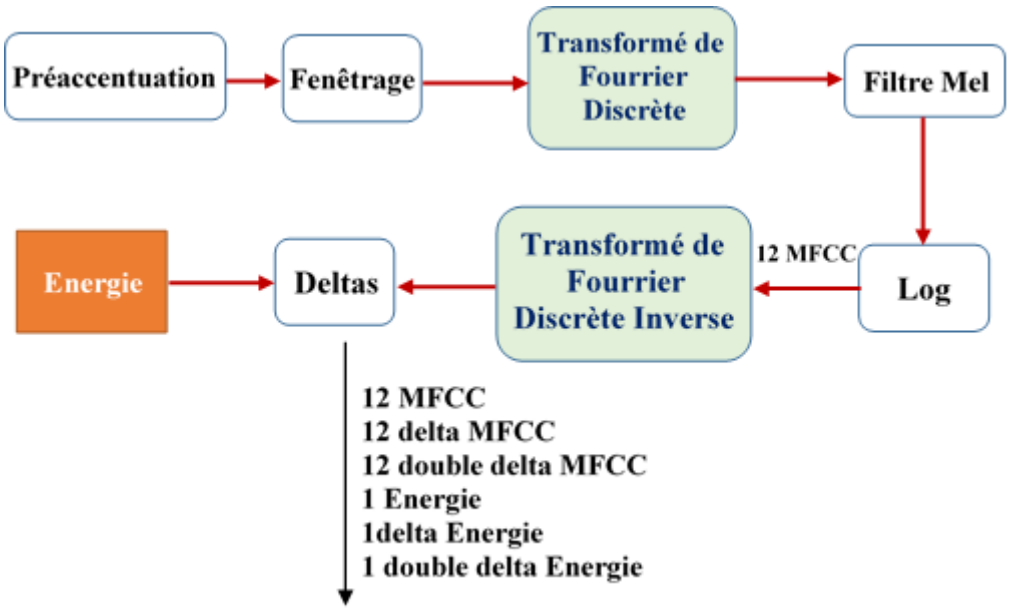


FIGURE 2.8 – *Flux de travail de traitement d'extraction des caractéristiques MFCC.*

2.4.2.1.1 Préaccentuation

L'échantillon de parole est pré-accentué pour compenser la pente spectrale. La pente spectrale est la tendance des signaux audio naturels à avoir moins d'énergie aux hautes fréquences[48].

Le but de la préaccentuation est de compenser la partie haute fréquence qui a été supprimée lors du mécanisme de production sonore des humains. De plus, elle peut également amplifier l'importance des formants haute fréquence. L'exemple suivant montre l'effet de la préaccentuation [49].

Cela réduit en fait la différence des composantes de puissance du signal. Ce filtre peut être appliqué dans le domaine fréquentiel ou temporel. Ce dernier est défini par l'équation suivante :

$$\hat{S}_n = S_n - \alpha S_{n-1} \tag{2.2}$$

Où, α est sélectionné pour être une valeur de 0,95 selon[49]. La fonction de transfert peut être donnée comme suit :

$$H_z = 1 - \alpha Z^{-1} \tag{2.3}$$

La figure 2.9 montre la PSD d'un signal vocal avant et après la préaccentuation. La baisse aux fréquences plus élevées peut être remarquée dans le signal d'origine par rapport au signal préaccentué avec α égal à 0.95, dans lequel la puissance est mieux répartie sur les fréquences relatives. Cette comparaison peut également être vue en utilisant les spectrogrammes de la Figure 2.10. Notez que les hautes fréquences sont plus fréquentes dans le signal pré-accentué.

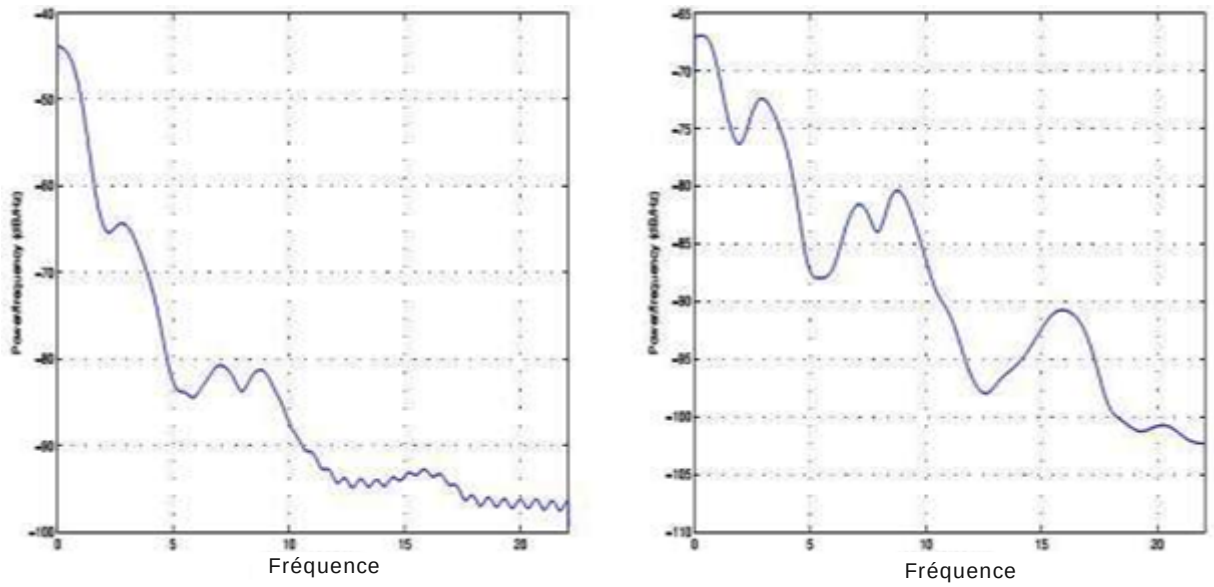


FIGURE 2.9 – *PSD du signal échantillonné à 44,1 KHz avant et après la préaccentuation Power Spectral Density PSD.*

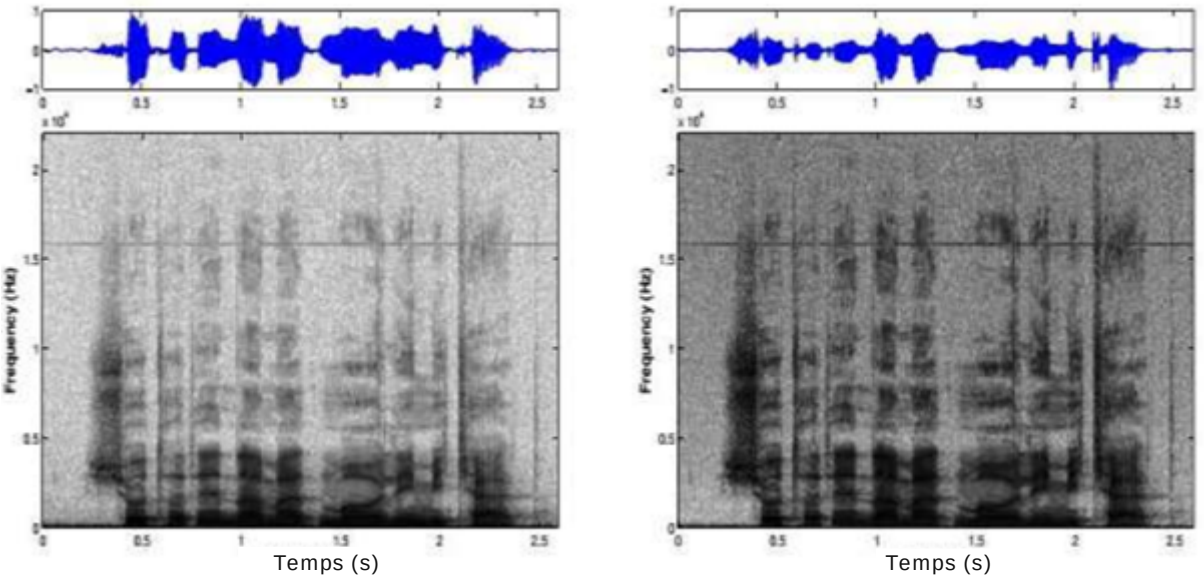


FIGURE 2.10 – *Spectrogramme d'un signal avant et après la préaccentuation.*

2.4.2.1.2 Fenêtrage

Dans le processus de fenêtrage, les échantillons du signal sont multipliés par une fonction de fenêtre. Ceci est fait pour minimiser les discontinuités du signal, découpant efficacement le signal en segments discrets[50]. Un choix populaire est la fenêtre

Hamming. L'équation suivante décrit la fenêtre de Hamming.

$$w(n) = \left\{ \begin{array}{ll} 0.54 - 0.46 * cos(\frac{2\pi n}{N-1}) & si \quad 0 \leq n \leq N-1 \\ \\ 0 & sinon \end{array} \right\} \tag{2.4}$$

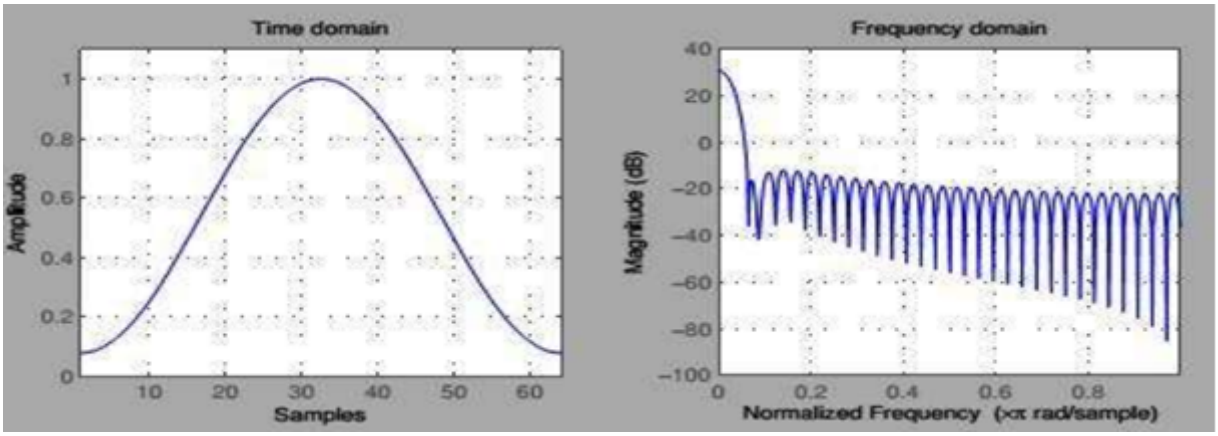


FIGURE 2.11 – *La fenêtre Hamming.*

La Figure 2.11 montre la fenêtre dans le domaine temporel et fréquentiel. L'effet de la fenêtre sur un signal vocal peut être vu sur la Figure 2.12.

2.4.2.1.3 Transformée de Fourier discrète DFT

La DFT est probablement la transformée la plus largement utilisée, en particulier dans le traitement du signal, car elle convertit une séquence de manière interchangeable dans le domaine temporel et fréquentiel[51]. Par conséquent, DFT peut être représenté comme :

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}} \tag{2.5}$$

Une DFT peut être vue sur la Figure 2.13. Elle montre une séquence de 10 échantillons. Cette DFT sera ensuite appliquée au signal vocal après le fenêtrage. Le résultat de cette opération donne l'amplitude et la représentation de phase du signal, comme le montre la figure 2.14.

2.4.2.1.4 Banc de filtres de Mel retour a la ligne

La plage de perception d'un être humain peut être définie à l'aide de l'échelle de Mel. Les expériences de[52] ont modélisés le système auditif humain en utilisant un banc de filtres à 24 bandes. Le résultat de l'étape précédente, le spectre DFT, ne tient pas compte du fait que le système auditif humain a une moindre sensibilité à une fréquence supérieure à 1 KHz. Les calculs DFT ne concernent qu'une échelle de fréquence linéaire, par conséquent, nous devons appliquer un processus appelé distorsion de fréquence, dans lequel les fréquences du spectre doivent être converties en des nombres plus petits en utilisant l'échelle logarithmique de Mel. Afin de réaliser ceci, un banc de filtres peut être construit comme présenté sur la figure 2.15, et cartographier efficacement les centres de bin de fréquence DFT. Ce filtrage est également appelé Mel-Spectrum défini comme suit :

(2.6)

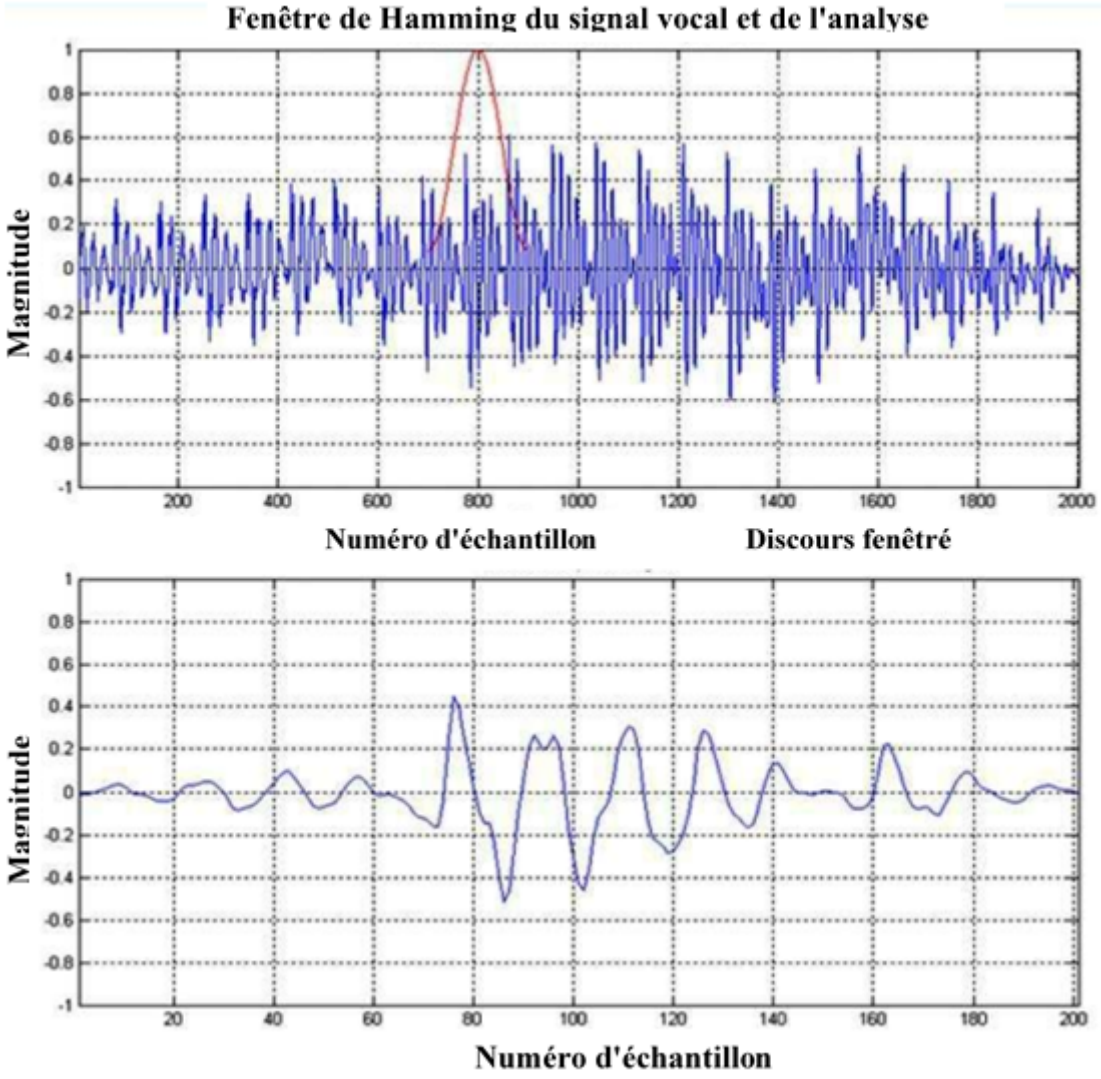


FIGURE 2.12 – *Séquence de N = 10 échantillons DFT.*

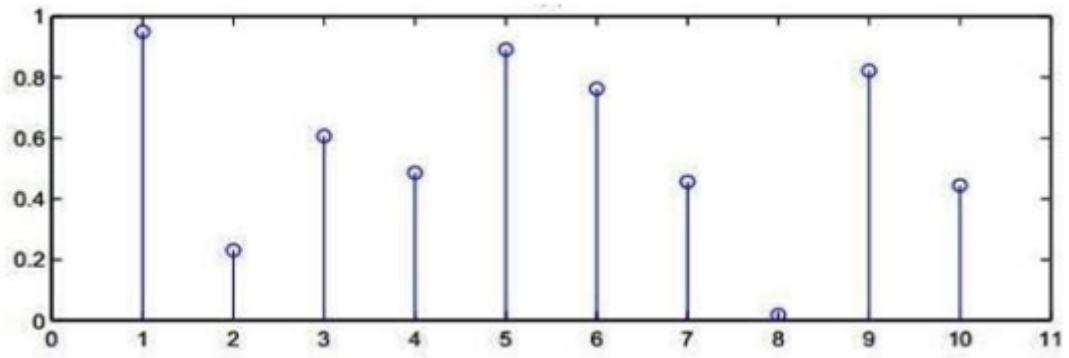


FIGURE 2.13 – *Ampleur du signal vocal fenêtré Banque Mel-Filter.*

2.4.2.1.5 Transformée de Fourier Discrète Inverse IDFT

La transformée de Fourier discrète inverse de Mel-Spectrum est ensuite calculée, donnant les MFCC. L'analyse du signal dans ce nouveau domaine Cepstral s'avère bé-

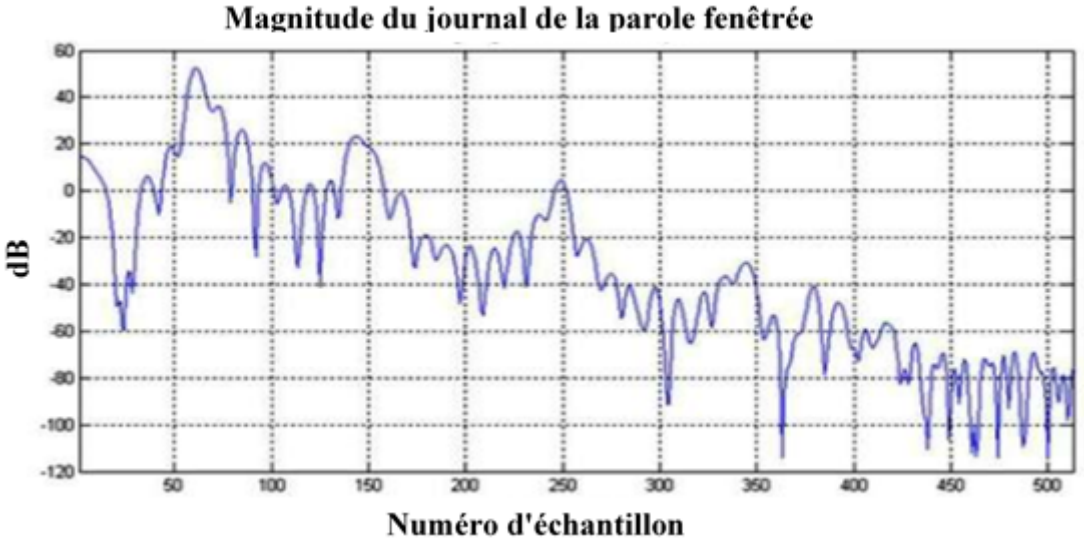


FIGURE 2.14 – *Ampleur du signal vocal fenêtré par banc de filtres de Mel.*

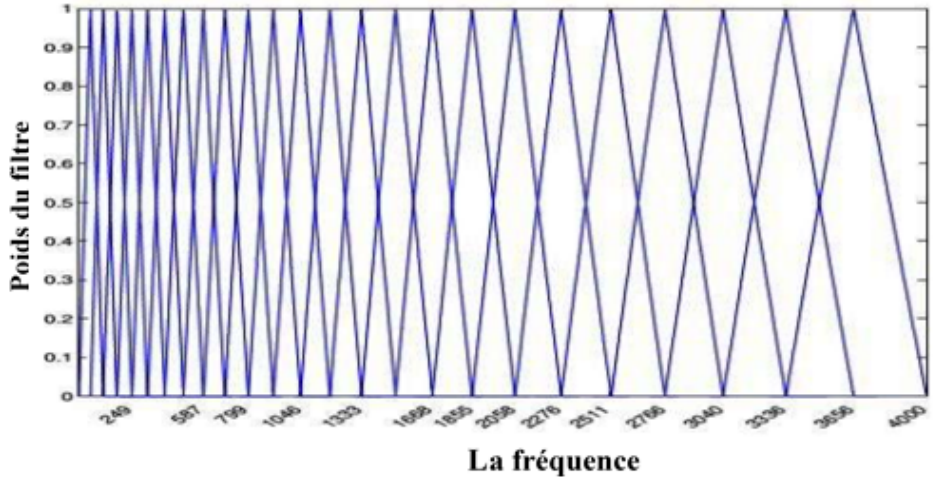


FIGURE 2.15 – *Banc de filtres Mel triangulaires pour un système à 24 filtres.*

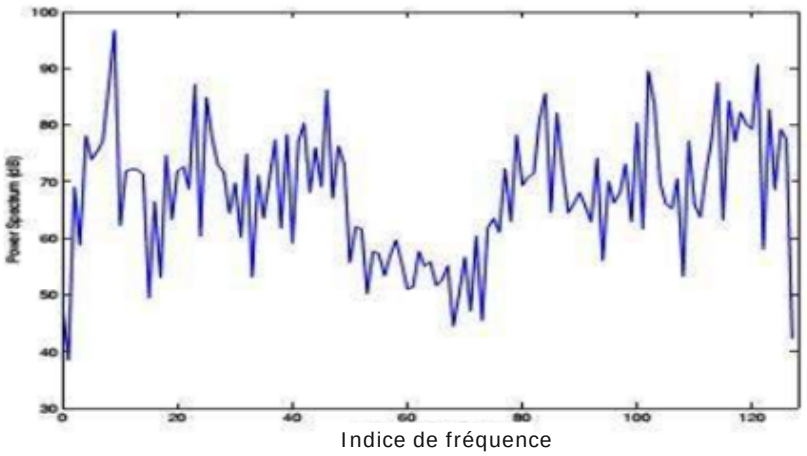


FIGURE 2.16 – *Spectre de puissance de la trame.*

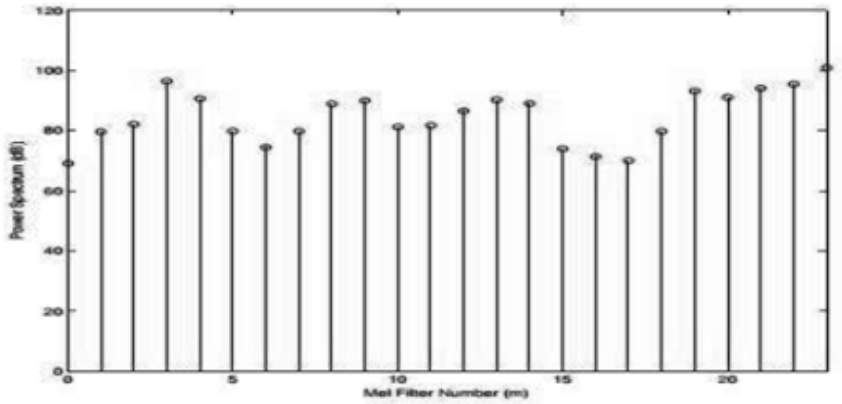


FIGURE 2.17 – *Spectre Mel, spectre de puissance de la trame dans le domaine fréquentiel.*

néfique compte tenu du fait de l'invariance inhérente à la distorsion spectrale linéaire. Les 12 premières valeurs du cepstre contiennent les informations significatives pour fournir des caractéristiques uniques de la forme d'onde.

2.4.2.1.6 Deltas

Pour capturer les changements de trame à trame dans le signal, les premières et deuxième dérivées des coefficients MFCC sont calculées et également incluses.

2.4.2.1.7 Énergie

Cette étape se produit simultanément avec l'extraction des caractéristiques MFCC. L'énergie totale de la trame d'entrée est calculée. Tous ces caractéristiques sont présentés dans la table 2.1

TABLE 2.1 – *Nombre total de caractéristiques résultantes dans un vecteur standard*

Type caractéristique	**Nombre**
coefficients cepstraux	12
coefficients delta cepstraux	12
coefficients double delta cepstraux	12
coefficient d'énergie	1
coefficient d'énergie delta	1

2.4.3 La correspondance des caractéristiques

La correspondance des caractéristiques est une procédure de classification d'objet d'intérêt dans une classe particulière parmi différentes classes. Ces objets d'intérêt, appelés motifs, sont des séquences de vecteurs de caractéristiques extraites du signal d'entrée à l'aide de l'une des techniques d'extraction de caractéristiques expliquée dans la section précédente. Plusieurs techniques de reconnaissance sont disponibles aux fins de la reconnaissance de la parole, telles que : les réseaux neuronal artificiel, la logique floue, la quantification vectorielle (VQ), les machine à vecteurs de support (SVM), ...

2.5 Reconnaissance du locuteur

C'est l'une des tâches que l'ordinateur surpasse les humains. Les scientifiques ont effectué des recherches de régression sur la capacité humaine de reconnaître et de discriminer les signaux vocaux. En déterminant les facteurs détaillant les informations spécifiques aux locuteurs, les chercheurs ont conçu des systèmes de reconnaissance des locuteurs fiables. À l'ère des ordinateurs numériques, les chercheurs ont développé de tels systèmes qui pourraient surpasser les auditeurs humains dans des tâches similaires[53]. De nombreuses limitations et problèmes difficiles restent à surmonter avec ces systèmes. Le signal vocal conserve les informations à plusieurs niveaux. Principalement, la parole transmet un message destiné à transmettre par des mots prononcés. Mais secondairement, la parole a également des informations sur l'orateur. La reconnaissance vocale est liée à l'extraction du message linguistique dans le discours prononcé tandis que la reconnaissance du locuteur identifie une personne qui parle. Cependant, de nos jours, de plus en plus d'attention ont été accordées au domaine de la reconnaissance du locuteur.

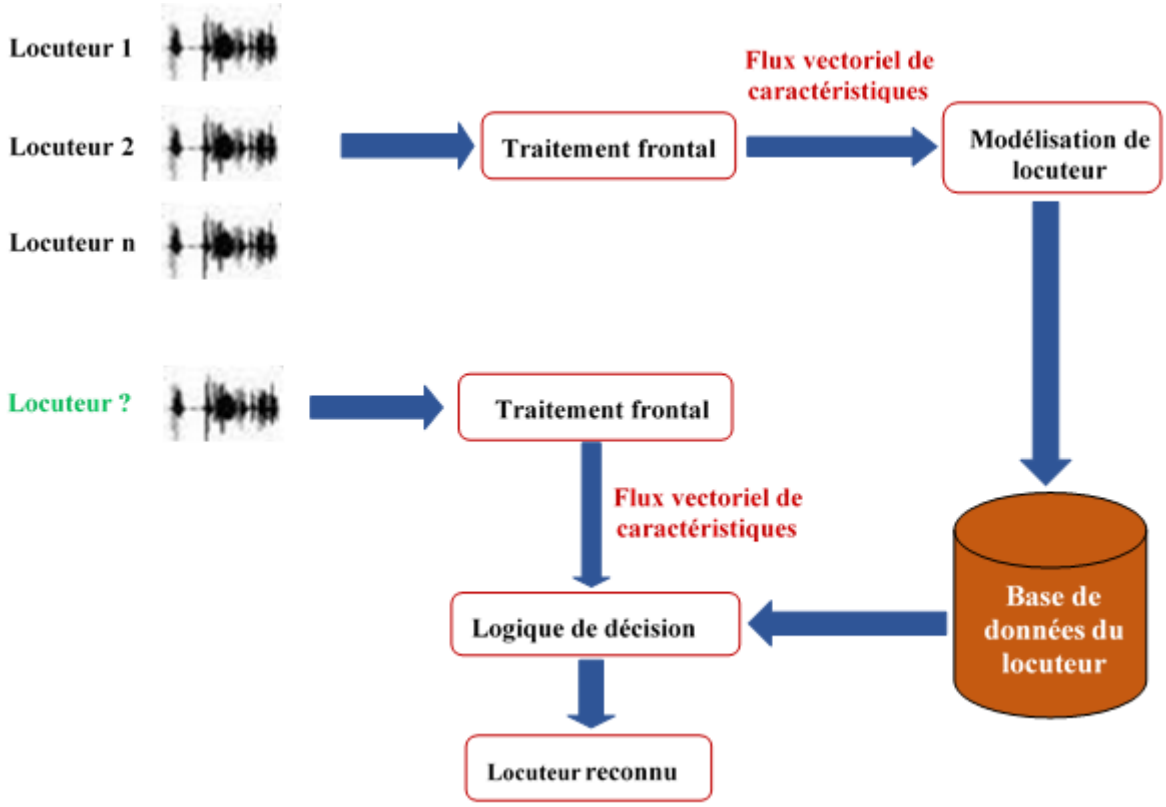


FIGURE 2.18 – *Diagramme schématique du système de reconnaissance du locuteur.*

La reconnaissance du locuteur (qui implique deux applications : l'identification du locuteur et la vérification du locuteur) est le processus qui permet de reconnaitre automatiquement qui parle sur la base d'informations individuelles incluses dans les ondes vocales. Cette technique permet d'utiliser la voix du locuteur pour vérifier son identité et contrôler l'accès à des services.

2.5.1 Domaines d'application de la reconnaissance du locuteur

En plus de la sécurité, la reconnaissance du locuteur a diverses applications :

1) Contrôle d'accès : les emplacements physiques sécurisés ainsi que les bases de données informatiques confidentielles sont accessibles par la voix. L'accès peut également être donné à des sites Web privés et restreints.

2) Transactions en ligne : en plus d'un mot de passe pour accéder à des informations bancaires ou pour acheter un article par téléphone, le signal vocal peut être utilisé comme une couche de sécurité supplémentaire.

3) Application de la loi : les systèmes de reconnaissance des locuteurs peuvent être utilisés pour fournir des informations supplémentaires pour l'analyse médico-légale. La surveillance par appel nominal des détenus peut être effectuée automatiquement en prison.

4) Gestion des données vocales : les services de messagerie vocale, les applications

d'extraction audio et l'annotation des réunions enregistrées ou en direct peuvent utiliser la reconnaissance des orateurs pour étiqueter automatiquement les orateurs.

5) Multimédia et personnalisation : Les pistes sonores et la musique peuvent être automatiquement étiquetées avec des informations sur le chanteur et la piste. Les sites Web et les ordinateurs peuvent être personnalisés en fonction de la personne utilisant le service.

2.5.2 Pourquoi la reconnaissance du locuteur ?

La reconnaissance du locuteur est l'une des techniques les plus facilement disponibles et applicables pour le système de communication humain et machine pour les raisons suivantes :

— La collection de base de données pour la reconnaissance du locuteur est très pratique par rapport à d'autres modèles biométriques.

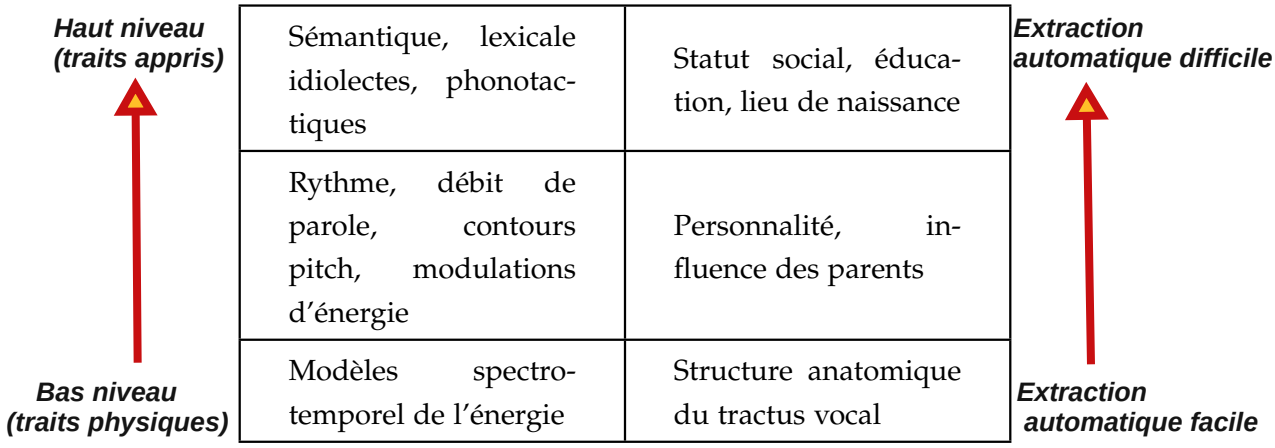
— La parole est le moyen le plus important d'échanger des informations entre les humains. Par conséquent, la parole peut également être le principal moyen d'échange d'informations entre les humains et les machines.

2.5.3 Informations spécifiques au locuteur dans le signal vocal

Le signal vocal transmet des informations sur les caractéristiques physiques, psychologiques et sociales du locuteur[54]. Ces informations sont présentes à différents niveaux. **La table 2.2** montre une classification hiérarchique possible de ces sources d'information ainsi que certains facteurs associés.

Les auditeurs humains utilisent ces sources d'information de manière naturelle pour discriminer les locuteurs[55]. La combinaison idiosyncratique de ces sources (par exemple, faible ton, timbre particulier, rire unique, choix de mots, etc.) facilite une identification précise. C'est la relation unique entre ces caractéristiques qui caractérise la voix d'un individu. En outre, pour les auditeurs humains, il existe une grande différence dans la manière dont l'identification est effectuée en fonction de la familiarité de l'auditeur avec le locuteur (par exemple, parents, conjoint, enfants, etc.)[55]. Cependant, cette distinction n'est actuellement pas applicable aux systèmes de reconnaissance automatique de locuteur. Les facteurs sous-jacents qui conditionnent chacune de ces sources dans le tableau 2.2 sont très divers. Dans une perspective hiérarchique, au niveau le plus bas, les caractéristiques physiques de l'individu, ainsi que les caractéristiques anatomiques du tractus vocal, se reflètent sur la composition spectro-temporelle du signal. Au plus haut niveau, les habitudes et coutumes apprises sur une longue période de temps sont les facteurs primordiaux dans le choix des mots et des structures sémantiques pour véhiculer un message ajouter.

TABLE 2.2 – *Hiérarchie des informations spécifiques au locuteur et facteurs déterminants associés*



Malgré la disponibilité de sources de haut niveau de reconnaissance du locuteur, la grande majorité des systèmes de reconnaissance automatique actuels relaient principalement (sinon uniquement) des informations de bas niveau représentées en termes de schémas spectro-temporels à court terme d'allocation d'énergie. Ceci est principalement dû au fait que la performance des systèmes basés sur des informations spectrales est (au moins) d'un ordre de grandeur meilleure que les systèmes les plus compétitifs basés sur des informations de niveau supérieur (voir[56] par exemple). Aussi, afin d'obtenir un modèle fiable du locuteur basé sur des niveaux d'informations plus élevés, la quantité de parole nécessaire est beaucoup plus importante que dans le cas des informations spectrales[57].

Néanmoins, les améliorations technologiques liées à la reconnaissance automatique de la parole ces dernières années et la disponibilité d'un large éventail de bases de données ainsi que la diversité de représentation apportée par l'utilisation de sources d'informations multiples est un moyen efficace d'obtenir une robustesse au bruit ambiant et aux distorsions de canal[56]. Dans cette thèse, nous nous concentrons sur les informations de bas niveau.

2.5.4 Identification Vs Vérification

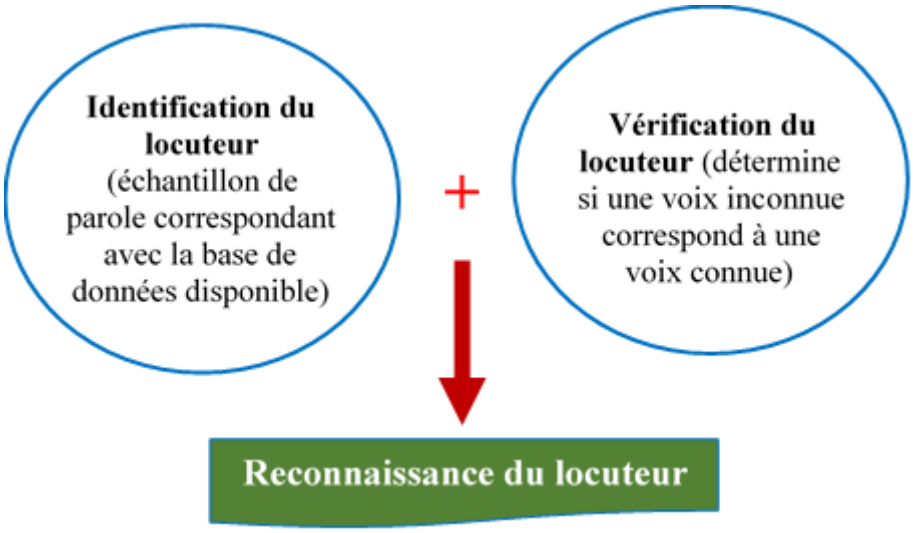


FIGURE 2.19 – *Système de reconnaissance des locuteurs.*

Comme le montre la figure 2.19, la reconnaissance du locuteur comporte deux parties sur la base des applications, à savoir : l'identification et la vérification. Les structures fondamentales pour l'identification du locuteur et la vérification du locuteur sont présentées respectivement sur les figures 2.20 et 2.21. Il peut être clairement observé à partir de la figure 2.20 que, pour l'identification du locuteur, les premières caractéristiques sont extraites de la parole inconnue d'entrée, ensuite ces caractéristiques sont mises en correspondance avec les caractéristiques extraites d'échantillons de parole connus stockés dans le modèle de référence pour chacun des locuteurs un par un et le score de similitude avec chaque locuteur est calculé. Après ce score de similarité maximum est sélectionné par l'un des divers algorithmes disponibles à cet effet. D'après la figure 2.21, il est clair qu'après l'extraction de caractéristiques d'un signal de parole inconnu, celles-ci sont comparées aux modèles de caractéristiques de référence déjà stockés du locuteur revendiqué, puis la décision de vérification (acceptation ou rejet) du locuteur est alors prise sur la base de valeur seuil prédéterminée.

a) Identification du locuteur : est le processus de recherche de l'identité d'un locuteur inconnu en comparant sa voix avec les voix des locuteurs enregistrés dans la base de données. La lacune de ce type d'application en jeu fermé est qu'elle ne peut être utilisée que pour s'identifier à partir d'un groupe de locuteurs déjà connus.

b) Vérification du locuteur : consiste à accepter ou refuser l'identité proclamée par un locuteur, en se basant sur un modèle qui lui est associé. Elle traite le scénario où le système prend en entrée un énoncé de test ainsi qu'une identité proclamée. La tâche consiste alors à prendre une décision binaire qui va confirmer ou infirmer le fait que

l'enregistrement de test est effectivement prononcé par le locuteur proclamé. Cette tâche peut à nouveau être de deux types : in-set et out-of-set. Dans le scénario in-set, le système automatisé suppose que le locuteur de test (le locuteur demandant l'authentification) doit être parmi les locuteurs connus, tandis que dans le scénario out-of-set, le système peut décider que le locuteur de test ne figure pas parmi les locuteurs reconnus.

La combinaison de ces deux systèmes doit être capable d'identifier une personne parmi de nombreux locuteurs connus et de classer un grand nombre de locuteurs inconnus dans une catégorie de locuteurs non valide[58].

2.5.5 Dépendance et indépendance du texte

Les systèmes de reconnaissance des locuteurs sont en outre discriminés sur la base du texte qui est prononcé[59]. Dans les systèmes de reconnaissance de locuteur dépendants du texte, les échantillons de parole utilisés pour l'apprentissage et le test des systèmes sont forcément identiques. Cela peut être illustré par un système de vérification d'autorisation dans lequel une personne doit utiliser à chaque fois un code d'autorisation personnel. De tels systèmes sont avantageux parce que le texte à prononcer est connu mais présentent également l'inconvénient que le système peut être trompé en faisant enregistrer la voix du demandeur et en utilisant cette voix enregistrée pour obtenir une autorisation.

Dans les systèmes indépendants du texte, il n'y a absolument aucune contrainte sur les échantillons de parole utilisés pour former et tester les systèmes de reconnaissance de locuteur. Ils pourraient traiter la parole librement parlée, qui est soit une phrase sélectionnée par l'utilisateur, soit une parole conversationnelle. Ces systèmes ont une grande flexibilité et sont très utiles dans les applications comme la récupération de la messagerie vocale, qui n'ont aucun contrôle sur le texte prononcé par une personne[60], ces systèmes sont plus flexibles, mais plus compliqués..

Il existe également un système intermédiaire entre les deux systèmes détaillés ci-dessus. Ce système est connu sous le nom de système dépendant du vocabulaire, qui impose au texte à prononcer une restriction selon laquelle le texte doit faire partie d'une collection limitée de mots ou de phrases. De ce fait, le système devient plus flexible que le système dépendant du texte et moins complexe que le système indépendant du texte.

Une question importante est celle de la variabilité des caractéristiques mesurées. On parle de variabilité intra-locuteur lorsque l'on s'intéresse à la variation des paramètres mesurés pour un même locuteur et de variabilité inter-locuteur si on considère des locuteurs différents[59][44][61]. Pour la reconnaissance du locuteur, on cherche à extraire des caractéristiques du signal de parole qui présentent une forte variabilité inter-locuteur

(pour pouvoir différencier les locuteurs entre eux) et une faible variabilité intra-locuteur (pour garantir la robustesse du système).

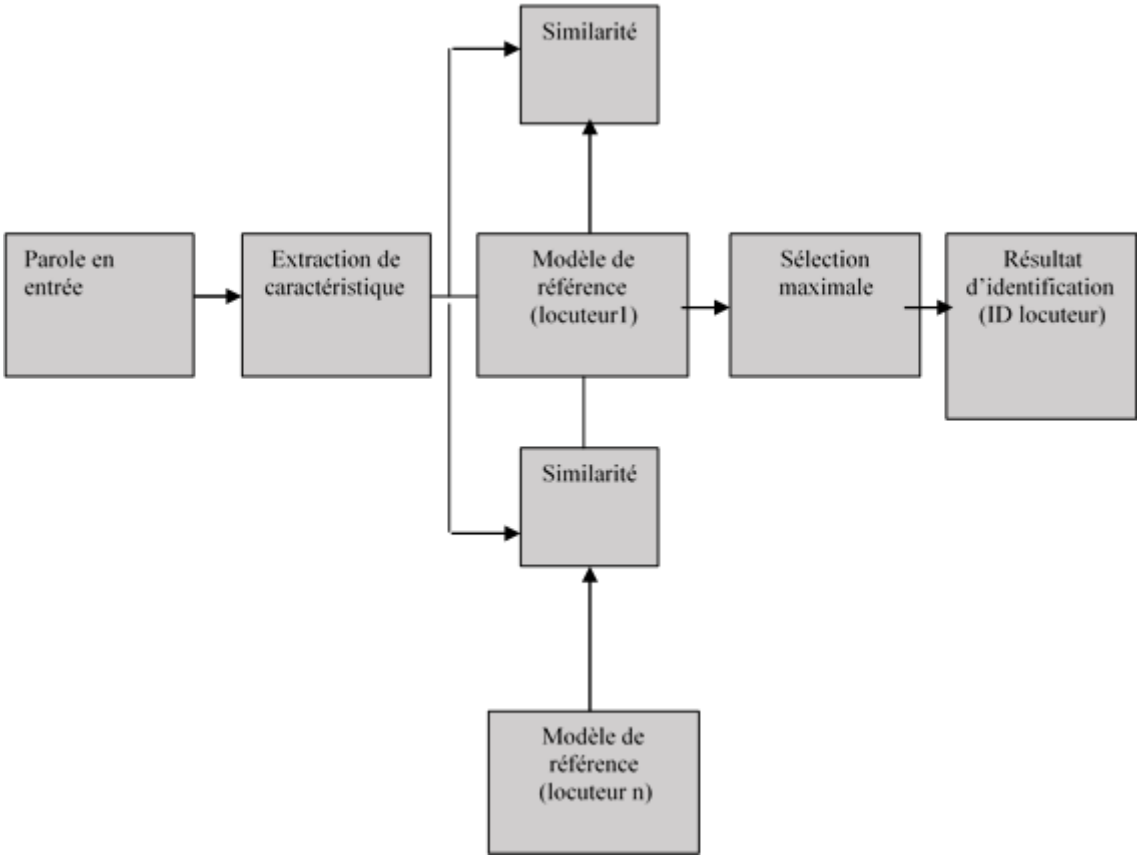


FIGURE 2.20 – *Identification du locuteur.*

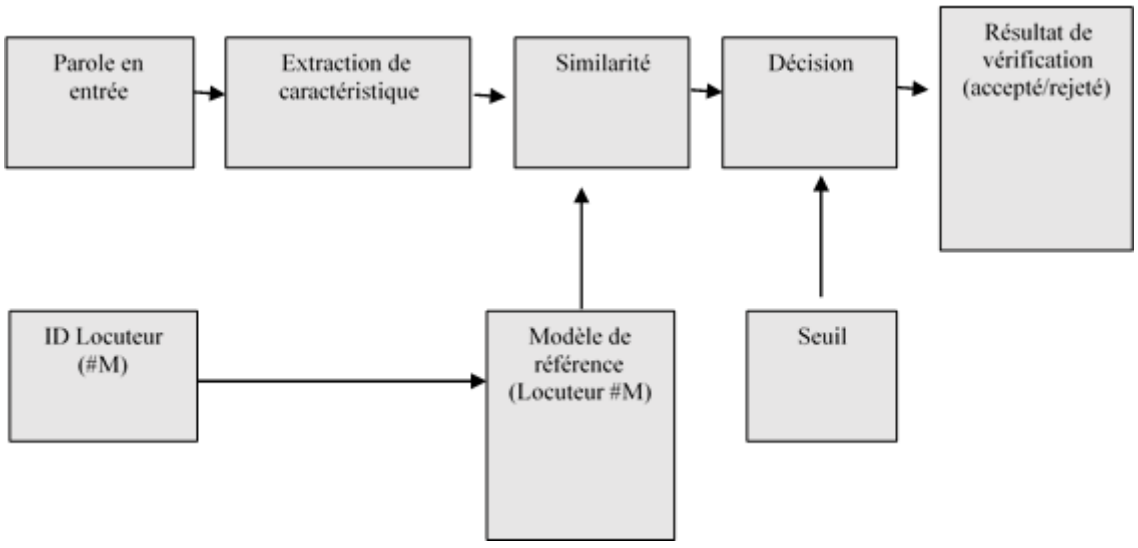


FIGURE 2.21 – *Vérification du locuteur.*

2.5.6 Développement d'un système de reconnaissance du locuteur

Le développement des systèmes de reconnaissance de locuteur peut être classé dans les phases suivantes (illustr2 dans la figure 2.22) :

2.5.7 Identification Vs Vérification

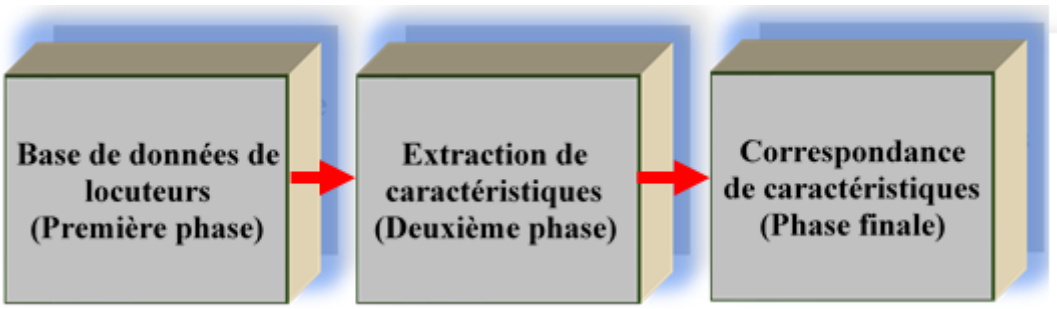


FIGURE 2.22 – *Phases du système de reconnaissance du locuteur.*

a) Collecte de la base de données des locuteurs

La collecte de bases de données est la toute première phase du développement d'un système de reconnaissance des locuteurs. Dans cette phase, des échantillons de parole sont enregistrés à partir de nombreux locuteurs. Après cela, un prétraitement des échantillons vocaux enregistrés est effectué. Ce prétraitement consiste à capturer des paramètres caractéristiques de la parole d'une personne donnée.

b) Extraction des caractéristiques

Les paramètres acoustiques extraits d'un enregistrement donné sont utilisés pour construire un modèle qui résume l'information acoustique correspondante de manière à créer une base de données de référence spécifique à chaque locuteur. Cette base de données de référence sera ensuite utilisée dans la phase suivante du système de reconnaissance du locuteur.

c) Correspondance des caractéristiques

C'est la dernière phase d'un système de reconnaissance du locuteur. Les caractéristiques d'un échantillon de parole inconnu sont d'abord extraites, puis mises en correspondance avec la base de données des caractéristiques de référence à l'aide d'une ou plusieurs techniques disponibles à cet effet, telles que HMM, GMM, VQ, DTW et bien d'autres.

2.6 Etat de l'art

Dans la section suivante, nous présentons quelques travaux ou l'ANFIS a été utilise dans le domaine de la parole. L'ANFIS présente l'avantage d'une bonne applicabilité car il peut être interprété comme une modélisation de linéarisation locale, et même comme des techniques linéaires conventionnelles pour l'estimation d'état et le contrôle d'état qui sont directement applicables. Ce réseau adaptatif a de bonnes capacités et performances dans l'identification, la prédiction et le contrôle du système et a été appliqué dans de nombreux systèmes différents. Comme il n'y a pas beaucoup de travaux de recherche utilisant ANFIS en reconnaissance vocale, il est nécessaire de poursuivre l'exploration et la recherche approfondie sur celui-ci.

ANFIS a été utilisé pour la vérification du locuteur[62] en utilisant les caractéristiques combinatoires de MFCC ; Coefficients de prédiction linéaire LPC et les cinq premiers formants ; Reconnaissance de mots discrets[63], vérification des émotions de la parole[64] basée sur MFCC pour une application en temps réel. Dans[65] ANFIS a été réalisée pour réduire le bruit et améliorer la parole, également pour la reconnaissance de chiffres isolés avec un locuteur indépendant[66]. La reconnaissance du locuteur, de la langue et des mots a été réalisée par l'ANFIS[67], en outre pour la classification du comportement de l'appelant[68]. Tous ces travaux avaient utilisé des techniques de regroupement pour déterminer la structure de l'ANFIS. une classification automatisée par sexe est réalisée par l'ANFIS[69]. Tous ces travaux sont résumés dans le tableau 2.3. La reconnaissance automatique des locuteurs a connu un succès considérable au cours des dernières décennies. De nombreux travaux basés sur différents paradigmes mathématiques ont été proposés afin d'améliorer les taux de reconnaissance. Dans la section suivante nous allons citer quelques travaux (voir la table 2.4)

[72] a appliqué le modèle de Markov caché HMM pour identifier dix arabophones où les (MFCC) ont été sélectionnés pour décrire le signal vocal. Le taux d'identification était de 100 % pendant les expériences dépendant du texte et de 80 % pour les expériences indépendantes du texte.

[73] ont utilisé des formants et l'entropie de paquets d'ondelettes comme entrées d'un réseau neuronal pour la classification. Le taux de reconnaissance a été trouvé de 90,09 % pour les voyelles dépendantes et de 82,50 % pour les voyelles indépendantes. Le réseau neuronal de régression généralisée GRNN et le réseau neuronal de rétropropagation BPNN ont été utilisés comme classificateurs[74] pour classer trente-six locuteurs parlant chinois en utilisant la méthode d'extraction de caractéristiques EMD par décomposition en mode empirique. La première méthode a donné un résultat de 78,16 % et la seconde a donné un résultat de 88,82%. Le modèle NN réseau de neurones artificiels est

TABLE 2.3 – *État de l'art de l'ANFIS dans la parole.*

Auteurs	**Caractéristiques**	**ANFIS pour**	**Résultats (Précision %)**
[70]	transformée en ondelettes	Reconnaissance des mots isolés en anglais	99%
[62]	MFCC ; LPC et les cinq premiers formants	la vérification du locuteur	7,14 % EER
[63]	LPC	Reconnaissance de mots discrets	58 %
[64]	MFCC	vérification des émotions de la parole	93 % pour colère ; 89 % pour triste et 85,3 % pour heureux %)
[65]	MFCC	Reconnaissance des chiffres de la parole malaise	85,24%
[67]	LPC, RC, LPCC, LAR, ARCSIN, LSFs	La reconnaissance du locuteur, de la langue et des mots	83,15%
[71]	MFCC	La reconnaissance des phonèmes	100%

utilisé comme modèle de classification [75] . Les caractéristiques MFCC ont été extraite du signal vocal. Ensuite, la dimension du vecteur propre d'entrée est réduite et les informations redondantes sont éliminées au moyen de l'algorithme Kmean Linde-Buzo-Gray. L'erreur était de 0,015 alors que le nombre de couche cachée est de 950.

[76] ont proposé un cadre NN de réseau neuronal pour la classification et la vérification des locuteurs indépendants du texte, à l'aide de la base de données TIMIT 8K. Les 39 (MFCC) ont été générés à partir du discours prétraité. Le système a atteint une classification à 100 %. Un classificateur ANN avec rétro-propagation (BPNN) est utilisé [77] , les seize caractéristiques MFCC extraites de 50 utilisateurs, la précision d'identification d'environ 92 % à 70 % est obtenue pour 10 à 50 utilisateurs. Une méthode de cluster basée sur SVM multiclasse a été proposée [78]. Des vecteurs de caractéristiques composés de 13 MFCC, 13 delta-MFCC sont ajoutés après l'application de CMS et RASTA aux MFCC. Les expériences effectuées sur le corpus vocal NIST-2002 pour 64 locuteurs avec une précision de 97%.

[78] ont appliqué SVM, GMM, et SVM fusionné avec un GMM pour la reconnaissance du locuteur et de la langue sur la base de données NIST 2003. Où 38 MFCC et 36 LPC ont été utilisés pour décrire le signal vocal. Le taux d'erreur EER était de 6,46% pour SVM, de 7,47% pour GMM et pour SVM / GMM était de 5,55%. De plus, le classificateur discriminant SVM combiné à la classification traditionnelle GMM a été proposé[79] avec un vecteur de caractéristiques cepstrales de 39 MFCC extrait du discours, cette combinaison a été testée sur 64 locuteurs de la base de données TIMIT. Ils ont atteint un taux d'identification de 100% avec la deuxième série d'expériences ne traitant que 3 énoncés par locuteur.

[80] ont construit un noyau de machine à vecteurs de support en utilisant le superviseur GMM. Ils ont effectué des expériences sur le corpus de reconnaissance de locuteur NIST 2005 (SRE) où un vecteur MFCC à 19 dimensions est extrait du signal de parole pré-accentué, le taux d'erreur (EER) était de 5,68% et la valeur minimale du coût de décision (minDCF) était 0,0222.

[81] ont proposé GMM / SVM pour l'identification automatique des locuteurs pour trois caractéristiques acoustiques, à savoir RASTA-MFCC, Gamma Tone Frequency Cepstral Coefficients GFCC et Mean Hilbert Envelope Coefficients MHEC dans différentes conditions de bruit. Des expériences ont été menées sur le corpus de bases de données étiquetées par téléphone TIMIT. La précision était de 70,32% pour RASTA-MFCC, 68,49% pour GFCC et de 73,27% pour MHEC sous le bruit de la rue.

[82] ont présenté un système de reconnaissance des locuteurs arabes pour les applications médico-légales à l'aide de GMM-UBM. Trente-neuf MFCC ont été utilisés pour l'extraction de caractéristiques. Toutes les expériences ont été menées sur 40 locuteurs

de la base de données vocale KSU. Ils ont atteint, un taux de reconnaissance de 97,8% avec un EER égal à 1,98%.

Divers types de caractéristiques profondes ont été présentés pour la vérification du locuteur dépendant du texte[83]. Ces caractéristiques sont toutes implémentées à la fois dans le cadre GMM-UBM et de vecteur d'identité. La base de données RSR2015 a été choisie pour évaluer tous les systèmes. Le meilleur système atteint l'EER 0,1%. Tous ces travaux sont résumés dans le tableau 2.4.

[85] ont présenté le problème de l'identification des locuteurs d'une grande population dans des conditions bruyantes. Ils ont proposé une approche d'arbre de décision basée sur la classification floue où ils d'utilisent un arbre de décision pour partitionner hiérarchiquement l'ensemble de la population en groupes de petite taille, et déterminer à quel groupe de locuteurs au nœud feuille appartient un locuteur testé, puis appliquer MFCC + GMM pour l'identification.

[86]ont discuté diverses techniques d'extraction de caractéristiques leurs avantages et inconvénients. Plusieurs nouvelles techniques sont développées combinant deux ou plusieurs techniques pour une meilleure précision et efficacité dans le domaine de la reconnaissance robuste du locuteur. Selon eux, les formes d'onde temporelles des signaux vocaux véhiculent des informations auditives complètes.

[87]ont discuté de deux algorithmes de reconnaissance du locuteur pour. Les résultats de leur étude ont montré que MFCC et DTW fournissent ensemble une bonne efficacité pour la reconnaissance de locuteurs dépendant du texte.

Table 2.4 – *Quelques travaux sur la reconnaissance des locuteurs.*

Auteurs	**Dataset**	**Paramètres**	**Méthode classification**	**Résultats (Précision %)**
[72]	10 arabophone	MFCC	HMM	100%
[73]	Propre jeu de données	cinq formants et sept entropies	ANN	90,09 % voyelle-dépendante 82,50% Voyelle-indépendant
[74]	base de données de laboratoire 36	EMD	GRNN BPNN	78% 89%
[75]	Ensemble de données propre	MFCC	Erreur Kmeanlbg ANN	Erreur 0,015
[76]	TIMIT	39 MFCC	ANN	100%
[77]		16 MFCC	ANN BPNN	92% 70%
[84]	NIST 2002	MFCC CMS RASTA	SVM	97%
[78]	NIST 2003	38 MFCC 36 LPC	SVM GMM SVM+GMM	6,46% EER 7,47%5,55%
[79]	TIMIT 64 haut-parleurs	39 MFCC	SVM+GMM	100
[80]	NIST 2005	19 MFCC	SVM+GMM	5,68 %
[81]	TIMIT	RASTA MFCC GFCC MHEC	GMM+SVM	70,32% 68,49% 73,278%
[82]	KSU	39 MFCC	GMM- UBM	97,8% EER 1,98
[83]	RSR2015	Caractéristiques profonde	GMM- UBM	0,1% EER

2.7 Conclusion

La reconnaissance automatique de la parole a obtenu un succès substantiel au cours des dernières décennies. De nombreux algorithmes et schémas basés sur différents paradigmes mathématiques ont été proposés pour tenter d'améliorer les taux de reconnaissance. Dans ce chapitre nous avons présentés les principes de base du signal de parole, les différentes techniques d'extraction des paramètres et plus précisément les coefficients cepstraux de Mel (MFCC). Ensuite, nous avons exposés le domaine de la reconnaissance du locuteur, les différentes étapes à suivre ainsi que les travaux réalisés dans ce domaine.

Les Algorithmes évolutionnaires 3

Sommaire

3.1 Introduction . . . 69
3.2 Les Algorithmes Evolutionnaires . . . 69
3.3 Terminologie . . . 71
3.4 Principe de base des algorithmes évolutionnaires . . . 71
3.4.1 Caractéristiques alternatives des algorithmes évolutifs . . . 73
3.4.2 Représentation de la population . . . 73
3.4.3 Fitness et coût . . . 74
3.4.4 La sélection . . . 75
3.4.5 Les opérateurs de reproduction . . . 75
3.5 Stratégies d'évolution . . . 77
3.6 Programmation génétique . . . 78
3.7 Programmation évolutionnaire . . . 78
3.8 L'évolution différentielle . . . 79
3.9 Les algorithmes génétiques . . . 81
3.9.1 Introduction . . . 81
3.9.2 Principe de base de l'algorithme génétique . . . 81
3.9.3 Le fonctionnement d'un Algorithme Génétique . . . 83
3.9.4 Forces et limites des Algorithmes Génétiques . . . 90
3.10 Optimisation par l'essaim de particules . . . 92
3.10.1 Introduction . . . 92
3.10.2 Fonctionnement du PSO . . . 93
3.10.3 Topologies . . . 96
3.10.4 Forces et limites des PSO . . . 97
3.11 Conclusion . . . 98

3.1 Introduction

Les techniques d'optimisation sont des approches dont le but est de trouver les meilleures solutions à des problèmes complexes en minimisant ou en maximisant une ou des fonctions objectifsreposant sur une ou plusieurs variables de décision qui donnent une valeur de la fonction objectif. L'optimisation combinatoire, la programmation non linéaire et la programmation linéaire sont conçues pour résoudre un large éventail de problèmes d'optimisation. Les techniques d'optimisation sont classées en deux familles : les méthodes exactes et les méthodes approximatives (méthodes heuristiques et méta-heuristiques). Un exemple d'uneclassification des techniques d'optimisation est illustré en figure 3.1.

Parmi les propriétés majeures des Algorithmes Evolutionnaires (AE) est que l'espace de recherche n'est pas exploré en partant d'une seule solution possible mais avec toute une population de solutions possibles ; ainsi que, les individus de la population peuvent échanger des attributs de solution entre eux. De cette façon, l'heuristique AE peut surpasser un algorithme de descente du gradient simple ou multi-initialisations, car ils sont plus résistants à une convergence prématurée vers un optimum local dans les espaces de recherche multivariables ;cependant, les AEne garantissentpasde trouver l'optimum global.

La figure 3.1 présente quelques approches parmi les techniques d'optimisation avec plus d'exemplesde la familledes approches évolutives. Dans le domaine des algorithmes d'optimisation évolutive on peut citer les principaux algorithmes suivants : l'algorithme génétique (GA) [1], la programmation génétique (GP) [2], l'évolution différentielle (DE) [3], la stratégie d'évolution (ES) [4], et la programmation évolutive (EP) [5]. Chacune de ces techniques possède de nombreuses variations et est utilisée dans de nombreuses applications industrielles différentes. La section suivante présentes ces techniques d'optimisation en détails.

3.2 Les Algorithmes Evolutionnaires

Le célèbre naturaliste Charles Darwin a défini la sélection naturelle ou la survie du plus fort comme la préservation des différences et des variations individuelles favorables, et la destruction de celles qui sont nuisibles[1].

Les premières analogies entre le mécanisme de la sélection naturelle et un processus d'apprentissage (ou d'optimisation) ont conduit au développement des « Algorithmes Evolutionnaires » (AE)[2], dans lesquels l'objectif principal est de simuler le processus évolutif dans un ordinateur. Les algorithmes évolutifs sont des méthodes globales, pa-

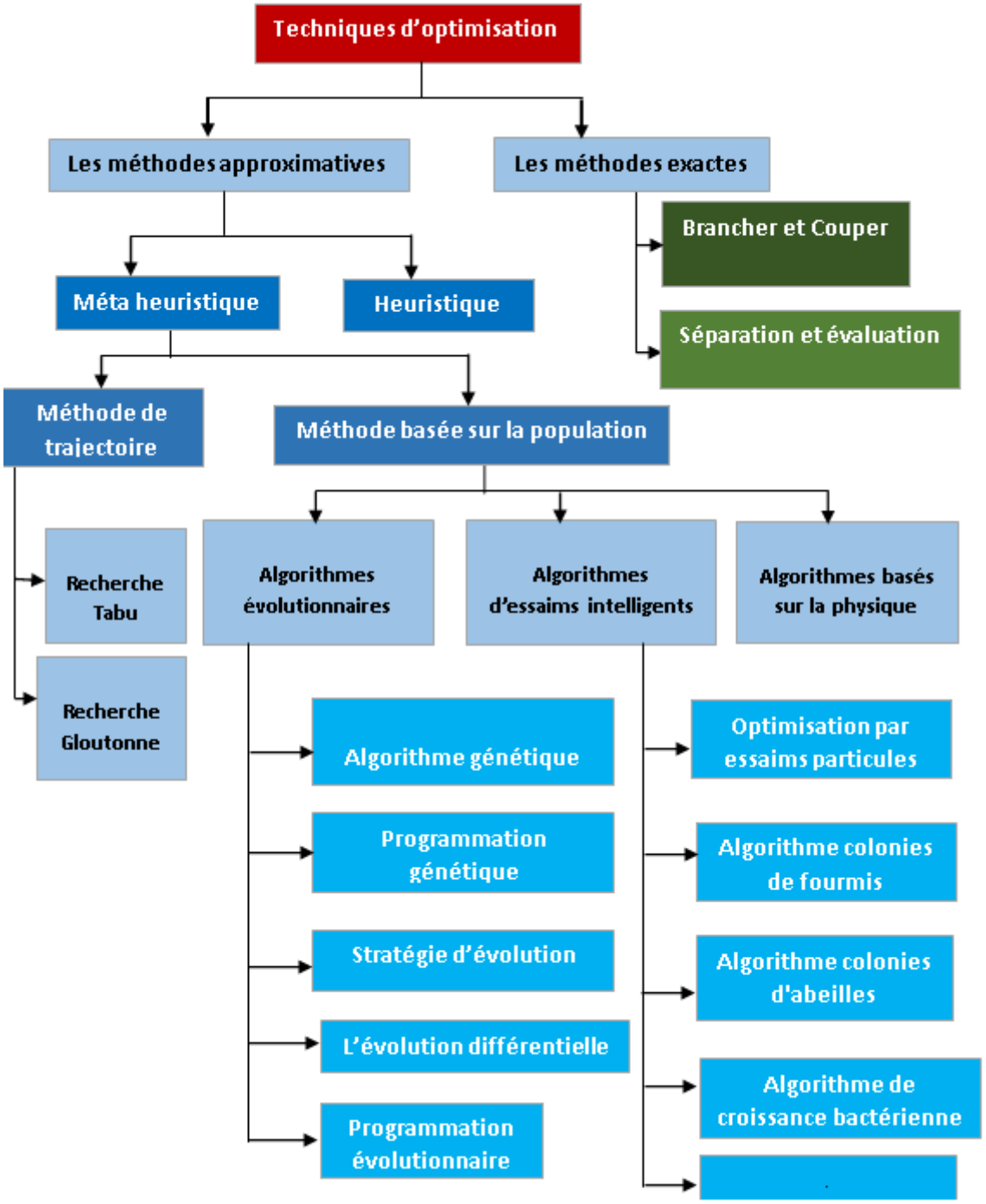


FIGURE 3.1 – *Classification des techniques d'optimisation.*

rallèles, de recherche et d'optimisation, dites bio-inspirées, fondées sur les principes de la sélection naturelle[3]et de la génétique des populations[4].

Les algorithmes évolutifs sont des méthodes de recherche stochastique qui imitent la métaphore de l'évolution biologique naturelle et / ou du comportement social des espèces utilisés pour résoudre des problèmes dans de nombreux domaines de l'ingénierie et de la science.

3.3 Terminologie

Une forte ressemblance avec les processus biologiques ainsi que leurs applications initiales pour la modélisation de systèmes adaptatifs complexes ont inspiré la terminologie utilisée par les chercheurs enAE. Ils empruntent beaucoup à la génétique, à la théorie de l'évolution et à la biologie cellulaire. Ainsi, une solution candidate à un problème est appelée un individu, tandis que l'ensemble complet des solutions actuelles est appelé une population. Pour certains domaines problématiques, une population peut être divisée en plusieurs sous-populations ou espèces. La représentation réelle (codage) d'un individu s'appelle son génome ou chromosome. Chaque génome est constitué d'une séquence de gènes, c'est-à-dire d'attributs qui décrivent un individu. Lavaleur d'un gène est appelée un allèle.

Lorsque des solutions individuelles sont modifiées pour produire de nouvelles solutions candidates, on dit qu'elles se reproduisent, les nouvelles solutions produites sont appelées des fils. Lors de l'évaluation d'une solution candidate, elle reçoit une note appelée fitness, qui indique la qualité de la solution dans le contexte d'un problème donné. Lorsque la population actuelle est remplacée par des descendants, la nouvelle population est appelée une nouvelle génération. Enfin, l'ensemble du processus de recherche d'une solution optimale est appelé évolution[5].

3.4 Principe de base des algorithmes évolutionnaires

Il existe trois processus principaux dans tous les algorithmes évolutifs. Le premier processus est le processus d'initialisation où la population initiale d'individus est générée aléatoirement selon une représentation de solution. Chaque individu représente une solution, directement ou indirectement. Si une représentation indirecte est utilisée, chaque individu doit d'abord être décodé en une solution. Chaque solution de la population est ensuite évaluée pour sa valeur de fitness dans le second processus.

Les valeurs de fitness peuvent être utilisées pour calculer la fitness moyenne de la population ou pour classer une solution individuelle au sein de la population enfin de sélection. Le troisième processus est la génération d'une nouvelle population par perturbation des solutions dans la population existante(voir figure 3.2)[6]

Après l'initialisation (figure 3.2), la population est évaluée et les critères d'arrêt sont vérifiés. Si aucun des critères d'arrêt n'est satisfait, une nouvelle population est générée à nouveau et le processus est répété jusqu'à ce qu'un ou plusieurs des critères d'arrêt soient satisfaits.

Un critère d'arrêt peut être statique ou dynamique. Par exemple, un critère d'arrêt statique peut permettre à un algorithme de s'exécuter pendant un nombre fixe d'itéra-

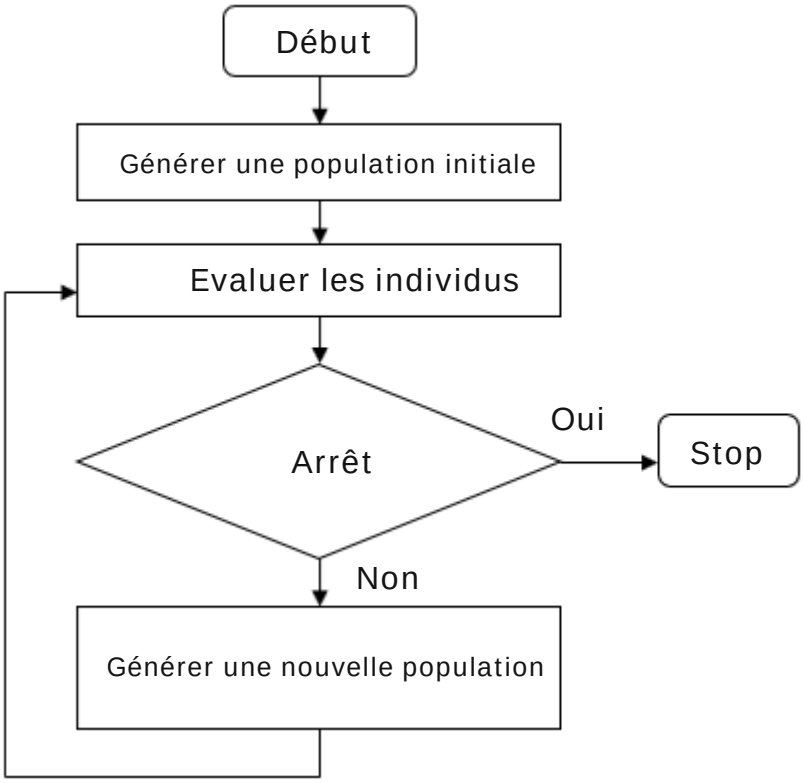


FIGURE 3.2 – *Organigramme de l'algorithme évolutionnaire .*

tions. Un exemple de critère d'arrêt dynamique consiste à répéter le processus jusqu'à ce que les k% des meilleures convergent vers la meilleure solution trouvée. Dans certains cas, une combinaison de plusieurs critères d'arrêt est utilisée.

En général, nous avons besoin des composants de base suivants pour mettre en œuvre un AE afin de résoudre un problème [7] :

1. Une représentation des solutions potentielles au problème.
2. Un moyen de créer une population initiale de solutions potentielles (cela se fait normalement de façon aléatoire, mais des approches déterministes peuvent également être utilisées).
3. Une fonction d'évaluation qui joue le rôle de l'environnement, en notant les solutions en fonction de leur fitness.
4. Une procédure de sélection qui choisit les parents qui se reproduiront.
5. Opérateurs évolutifs qui modifient la composition des fils (communément : croisement et mutation).
6. Valeurs des différents paramètres utilisés par l'algorithme évolutif (taille de la population, probabilités d'application des opérateurs évolutifs, etc . . .

Algorithme 1 : algorithme évolutionnaire

1. Générer la population initiale G(0) aléatoirement, et mettre i=0 .
2. Répéter
   (a) Evaluer chaque individu dans la population.
   (b) Sélectionner des parents à partir de G(0) en se basant sur leur fitness dans G(i).
   (c) Appliquer des opérateurs de reproduction aux parents et produire les fils qui forment G(i+1) .
   (d) i=i+1 .
3. Jusqu'à ce que critère d'arrêt soit satisfait.

3.4.1 Caractéristiques alternatives des algorithmes évolutifs

De nombreuses variantes ont été développées dans le modèle d'algorithme évolutif global, dans le but d'améliorer les performances et l'adaptation à des domaines de problèmes particuliers. Ainsi, il peut être plus utile ou plus approprié de considérer les AE comme une méthodologie générale de résolution de problèmes, plutôt que comme un outil spécifique.Cependant, de nombreuses méthodes en AEs'inspirentdes algorithmes génétiques (SGA) présentés par Goldberg en 1989[8].

Pratiquement les bon paramètres d'un AE sont choisis lors del'expérimentation et sont souvent basés sur des tests empiriques limités. De plus, même si un bon choix de paramètres peut être trouvé pour un problème particulier, cet ensemble sera sous-optimal pour de nombreux autres problèmes, peut-être même non optimalpour des problèmes de même famille. Faire un mauvais choix des paramètres d'un AE peut produire des résultats extrêmement mauvais.

Un bref résumé des principales caractéristiques des algorithmes évolutionnaires est présenté ci-dessous. Pour une discussion complète voir[9].

3.4.2 Représentation de la population

En utilisant unalgorithme évolutif pour résoudre des problèmes d'optimisation, la première tâche importante est de déterminerune bonnereprésentation des solutionsselon les éléments ou la terminologie d'unalgorithme évolutif spécifique. Il est important de choisir une représentation qui est leplus susceptible de produire des solutions réalisables. Ceci est une considération de conception commune pour tous les algorithmes évolutifs.

La représentation d'une solution peut être directe ou indirecte. La principale considération de conception est de s'assurer que chaque individu généré peut toujours être décodé en une solution réalisable. Pour un problème complexe, la représentation indirecte est souvent utilisée avec une procédure de décodage pour convertir une représentation de solution indirecte en une solution réalisable. Une fois la solution décodée, safitness peut être évaluée.

La population initiale peut être soit un échantillon aléatoire de l'espace des solutionspossiblessoit un ensemble construit avec des individus trouvés par des procédures de recherche locales simples, si celles-ci sont disponibles[10].

En plus de la représentation de la solution, deux paramètres communs qui doivent être déterminés initialement sont la taille de la population et le nombre maximum d'itérations. Les choix de ces deux paramètres ont une influence majeure sur la qualité de lasolution et le temps d'exécution, et en pratique, ces valeurs sont presque toujours déterminées de manière empirique par des essais.

3.4.3 Fitness et coût

Chaque individu de la population se voit attribuer une valeur de fitness, qui exprime la capacité de la solution à résoudre le problème. La valeur de fitness détermine de manière probabiliste dans quelle mesure l'individu réussira à propager ses gènes (son code) aux générations suivantes. Les meilleures solutions se voient attribuer des valeurs de fitness plus élevées que les solutions moins performantes.

Un algorithme évolutif cherche à maximiser la fitness moyenne de sa population grâce à l'application itérative des opérateurs génétiques. La valeur de fitness d'une solution dans le domaine des AE correspond à une valeur de coût dans le domaine du problème. « Coût » est un terme couramment associé aux problèmes d'optimisation traditionnels. Il représente une mesure de performances : à savoir, plus le coût est bas, meilleure est la performance. Les optimiseurs cherchent à minimiser les coûts. Par conséquent, il est évident qu'en maximisant la fitness, l'AE minimise effectivement le coût. Les mesures de performance brutes doivent être converties en valeur de coût. Ce processus est généralement simple pour les problèmes à objectif unique, mais devient plus compliqué dans le cas multi-objectif.

Un algorithme évolutionnaire nécessite généralement la définition d'une fonction rendant compte de la pertinence des solutions potentielles, à partir des grandeurs à optimiser. Elle se nomme, fonction d'adaptation f ou fonction objectif (ou fitness function en terminologie anglo-saxonne). Elle est souvent exprimée par la composition de deux fonctions g et h :

$$f = g \circ h \tag{3.1}$$

avec : $\circ$: loi de composition de fonctions.

g : fonction de transformation, pouvant être linéaire, exponentielle, logarithmique, etc. ...

h : fonction objectif ou de coût, elle dépend de l'objectif recherché.

C.-à-d. Une fonction objective évalue l'optimalité des individus et associe à chacun sa fitness qui représente la valeur de sa qualité.

3.4.4 La sélection

Des solutions individuelles doivent être sélectionnées pour l'application des opérateurs de reproduction pour une réinsertion ultérieure dans la population.Deux catégories de procédures de sélection peuvent être distinguées : les procédures déterministes et les procédures stochastiques.

Pour unesélection déterministe, les meilleurs individus sont sélectionnés, un tri de l'ensemble de la population est nécessaire par conséquent le temps de calcul est important pour des populations de grandes tailles. Les individus les moins performants sont totalement éliminés de la population, et le meilleur individu est toujours sélectionné, on parle alors d'élitisme.

Pour une sélection stochastique, les meilleurs individus sontfavorisésde manière probabiliste, ce qui laisse une chance aux individus moins performants. Par contre, il peut arriver que le meilleur individu ne soit pas sélectionné, et qu'aucun des enfants n'atteigne une performance aussi bonne que celle du meilleur parent.La sélection par roulette est la plus commune des méthodes de sélection stochastique[2].

3.4.5 Les opérateurs de reproduction

L'évolution d'une populationest réalisée à l'aide d'un ensemble d'opérateurs de reproduction (génétiques stochastiques), qui manipulent le code génétique. La plupart des algorithmes évolutifs incluent des opérateurs qui sélectionnent des individus pour la reproduction, produisent de nouveaux individus en fonction de ceux sélectionnés et déterminent la composition de la population à la génération suivante. Le croisement et la mutation sont deux opérateurs bien connus. L'opérateur de croisement (recombinaison), également appelé crossover, implique l'échange de matériel génétique entre chromosomes (parents), afin de créer de nouveaux chromosomes (progéniture). Essentiellement, la recombinaison tend à diriger la recherche vers des zones supérieures de l'espace de recherche.

L'opérateur de mutation apporte de petites modifications aléatoires à un chromosome. Historiquement considéré comme un opérateur de fond par la communauté des algorithmes génétiques, son rôle est souvent considéré comme une garantie que la probabilité de rechercher une chaîne donnée ne sera jamais nulle et une opportunité de récupérer un bon matériel génétique qui peut être perdu par l'action de sélection et recombinaison.

Les opérateurs génétiques ont fait l'objet de discussions approfondies, tant sur la composition que sur la finalité des différents opérateurs. Certains chercheurs ont abandonné la recombinaison, tandis que d'autres considèrent l'effet de la mutation comme minime. Le choix des opérateurs génétiques doit être fait avec le choix de la représentation. Les choix des paramètres des opérateurs, tels que le taux de mutation et la taille de la mutationdépendentessentiellementdesperformances de l'algorithme. Notez que les paramètres peuvent varier de manière adaptative au cours d'une analyse afin d'améliorer les résultats d'une recherche et que ces paramètres peuvent eux-mêmes être sujets à uneévolution[6]

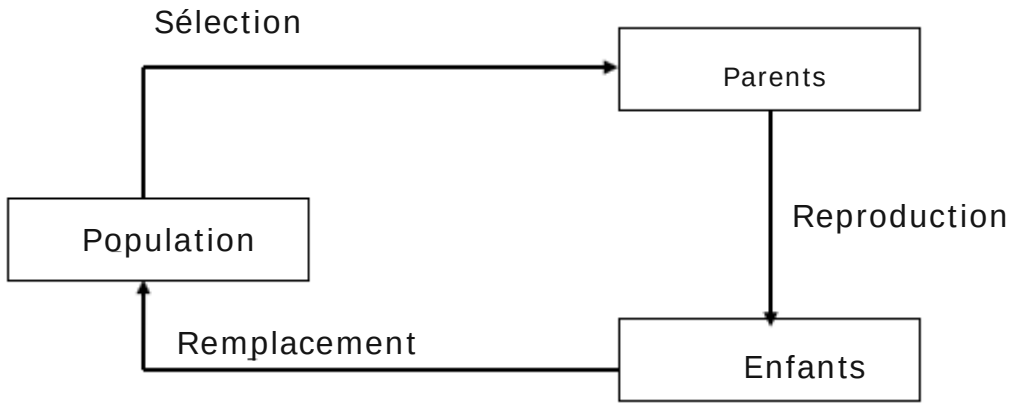


Figure 3.3 – *Organigramme d'un algorithme évolutif.*

Une fois la nouvelle génération construite, les processus qui aboutissent à la génération suivante de la population recommencent (figure 3. 3). L'algorithme évolutif explore et exploite l'espace de recherche pour trouver de bonnes solutions au problème. Il est possible pour un AE de sortir avec plusieurs solutions différentes, mais toutes aussi bonnes, à un problème, en raison de son utilisation d'une population.

Les AE sont des outils robustes, capables de faire face aux discontinuités et au bruit dans le paysage problématique. Ils se sont avérés utiles pour résoudre des problèmes qui ne peuvent être résolus par des moyens conventionnels. L'inclusion d'heuristiques spécifiques à un domaine n'est pas une condition préalable, bien qu'elle puisse améliorer les performances d'une évaluation environnementale.

Malgré les concepts simples impliqués, les algorithmes évolutifs peuvent devenir assez compliqués. De nombreuses variantes ont été proposées.

3.5 Stratégies d'évolution

Les stratégies évolutionnaires (ES) ont été développées initialement par Rechenberg en 1973comme une méthode de résolution des problèmes d'optimisation continus. Les stratégies d'évolution sont différentes par rapport aux Algorithmes Génétiques (AG), principalement dans la procédure de sélection. Dans l'AG, la prochaine génération est créée à partir de la population parente en choisissant des individus en fonction de leur valeur de fitness, en gardant une taille constante de la population. Dans les ESune population temporaire est créée, sa taille est différente de celle de la population parente (en fonction des paramètres supposés k et l). Dans cette étape, les valeurs de fitness ne sont pas importantes. Les individus de la population temporaire subissent des croisements et des mutations. À partir de ces populations, un nombre supposé des meilleurs individus est sélectionné pour la génération suivante de la population (de manière déterministe).

Les ES fonctionnent sur des vecteurs de nombres réels, tandis que les AG classiques opèrent sur des vecteurs binaires. Les principaux types de ES sont $ES(1+1)$, $ES(\mu+\lambda)$et $ES(\mu,\lambda)$[11].

Algorithme 2 : Stratégie évolutionnaire de $ES(\mu+\gamma)$

1. Déterminer la fonction objective (FO)
2. Assigner le nombre de génération à 0 (t=0)
3. Créer d'une manière aléatoire μ individus dans la population initiale P(t)
4. Evaluer les individus dans la population P(t) en utilisant FO
5. **Tant que** le critère de terminaison n'est pas satisfait Faire
6. t=t+1
7. Créer la population T(t) par la reproduction de γ individus de la population P(t-1)
8. Créer la population M(t) par le croisement et la mutation des individus de la population T(t)
9. Evaluer les individus dans la population M(t)
10. Sélectionner μ meilleurs individus dans la population P(t) des populations P(t-1) et M(T)
11. **Fin Tant que**
12. Retourner le meilleur individu de la population P(t)

3.6 Programmation génétique

La programmation génétique (GP)est relativement nouvelle particulièrement adaptée à l'évolution de structures complexes de dimensions variables. Il s'agit d'une forme spécialisée d'AG qui opère sur des types de solutions très spécifiques, utilisant des opérateurs génétiques modifiés. Le GP a été développé par Koza[12]comme une tentative de trouver la voie pour la génération automatique des codes de programmes lorsque les critères d'évaluation pour leur bon fonctionnement sont connus. Les individus de la population sont des arbres d'analyse de programmes informatiques (au lieu de chromosomes linéaires (de bits ou de nombres)) où les valeurs de fitness sont mesurées en les exécutant [13]. Les opérateurs génétiques ont été conçus spécialement pour fonctionner sur des arbres, par exemple, le croisement comme étant unéchange de sous arbres, la mutation comme étant unchangement de nœud ou feuille. Un programme génétique peut être considéré comme une variante desalgorithmes génétiques avec unedifférente façon pour représenter des solutions.

3.7 Programmation évolutionnaire

La programmation évolutionnaire (PE) a été développée par L.J. Fogel dans les années 1960 et reprise par D.B. Fogel et d'autres chercheurs dans les années 1990 [14]. La PE a été développée comme un outil pour découvrir la grammaire d'unlangage. Initialement conçue dans le but de faire évoluer des machines à états finis et a été par la suite étendue aux problèmes d'optimisation de paramètres (Les chromosomes sont représentés sous forme de machines à état fini (MEFs)). L'objectif est d'optimiser ces MEFs afin de fournir une représentation correcte du langage ciblé. Typiquement, la programmation évolutionnaire n'utilise pas d'opérateurs de croisement, le principal processus est l'opération de mutation qui est utilisée pour créer aléatoirement la nouvelle population en mutant chaque individu de la population parentale. Dans la PE, une population de μ solutions potentielles au problème est d'abord générée aléatoirement. Chaque individu i de la population produit λ descendants résultants de mutations. Une opération de sélection naturelle est alors appliquée afin de former une nouvelle population de μ individus. Le processus de mutation-sélection est répété itérativement jusqu''à ce qu'une solution acceptable soit trouvée.

Algorithme 3 : Programmation évolutionnaire

1. Déterminer la fonction objective (FO)
2. Assigner le nombre de génération à 0 (t=0)
3. Créer d'une manière aléatoire les individus dans la population initiale P(t)
4. Evaluer les individus dans la population P(t) en utilisant FO
5. **Tant que** le critère de terminaison n'est pas satisfait Faire
6. t=t+1
7. Créer la population M(t) par la mutation de chaque individu de la population P(t-1)
8. Evaluer les individus dans la population M(t) en utilisant FO
9. Sélectionner les individus dans la population P(t) de la somme des individus de P(t-1) et M(t) utilisant la sélection de classement.
10. **Fin Tant que**
11. Retourner le meilleur individu de la population P(t)

3.8 L'évolution différentielle

L'évolution différentielle (DE) est un type d'algorithmes évolutifs utile principalement pour l'optimisation de fonctions dans l'espace de recherche continu. Bien qu'une version de l'algorithme DE pour les problèmes combinatoires ait également été discutée [15], la version principale de l'algorithme a été discutée par Storn et Price [16]. DE a été conçue pour résoudre des problèmes à variables continues. Sa stratégie consiste à biaiser un opérateur de mutation, appliqué à un individu, en fonction des différences calculées avec d'autres individus sélectionnés aléatoirement. L'idée principale de l'algorithmeDEest liée au calcul de la différence entre deux individus choisis au hasard dans la population. DE détermine le gradient de la fonction dans une zone donnée, pas en un seul point. Par conséquent, l'algorithme empêche la solution de s'en tenir à un extrême local de la fonction optimisée [17].

Algorithme 4 : Evolution différentielle

1. Déterminer la fonction objective (FO)
2. assigner le nombre de génération à 0 (t=0)
3. Créer d'une manière aléatoire les individus dans la population initiale P(t)
4. **Tant que** le critère de terminaison n'est pas satisfait Faire
5. t=t+1
6. **Pour chaque** i^{iem} individu dans la population P(t) Faire
7. Générer d'une manière aléatoire trois nombre entiers
8. $r_1, r_2, r_3 \in$ [1, taille de la population] où $r_1 \neq r_2 \neq r_3 \neq i$
9. Pour chaque j^{iem} gene dans i^{iem} individu $(j \in [1, n])$ faire
10. $v_i, j = x_{r1,j} + F * (x_{r2,j} - x_{r3,j})$
11. d'une façon aléatoire générer un nombre réel $rand_j \in [0, 1)$
12. **Si** randj < CR alors $u_i, j := v_i, j$
13. **sinon**
14. $u_i, j := x_i, j$
15. **Fin Si**
16. **Fin Pour**
17. **Si** l'individu u_i est meilleur que l'individu x_i **Alors**
18. remplacer l'individu x_i par l'individu enfant u_i
19. **Fin Si**
20. **Fin Pour**
21. **Fin Tant que**
22. Retourner le meilleur individu dans la population P(t)

3.9 Les algorithmes génétiques

3.9.1 Introduction

Les algorithmes génétiques (AG) développés par J. Holland[18]présentent des qualités intéressantes pour la résolution de problèmes d'optimisation complexes. Leurs fondements théoriques furent exposés par Goldberg[19]. Ilstentent de simuler le processus d'évolution des organismes vivants. Les AG sont appliqués en biogénétique, en informatique, en ingénierie, en économie, en chimie, en fabrication, en mathématiques, en physique et dans d'autres domaines.

3.9.2 Principe de base de l'algorithme génétique

L'idée principale de l'AG est d'imiter la sélection naturelle et la survie du plus apte. En AG, les solutions sont représentées sous forme de chromosomes. Les chromosomes sont évalués pour leurs valeurs de fitness et ils sont classés du meilleur au pire en fonction de la valeur de fitness.Une solution à un problème donné est représentée sous la forme d'une chaîne, appelée «chromosome», constituée d'un ensemble d'éléments, appelés «gènes», qui contiennent un ensemble de valeurs pour les variables du problème à optimiser[8]. Traditionnellement, la structure de donnéesutilisée dans un algorithme génétique se compose d'une chaîne binaire. Cependant, les AG ne sont pas limités aux représentations en chaînes de bits, mais d'autres codages sont également possibles, par ex. vecteurs de nombres réels ([20][21][22][23]),une suite d'instructions ou des programmes informatiques[12].

Chaque chromosome (chaîne) est une concaténation d'un certain nombre de sous-composants appelés gènes. Les gènes se produisent à diverses positions ou locus du chromosome et prennent des valeurs appelées allèles. Dans la représentation sous forme de chaîne de bits, un gène est un bit, un locus est sa position dans la chaîne et un allèle est sa valeur (0 ou 1). Le terme biologique génotype fait référence à la composition génétique globale d'un individu et correspond à une structure de l'AG. Le terme phénotype fait référence aux caractéristiques extérieures d'un individu et correspond à une structure décodée dans l'AG.

Les AG sont généralement des méthodes de type boîte noire qui utilisent exclusivement les informations de fitness ; ils ne nécessitent pas d'informations sur le gradient ou d'autres connaissances internes du problème. En optimisation, l'objectif est de trouver la ou les meilleures solutions possibles à un problème, par rapport à un ou plusieurs critères.

Le processus de production de nouvelles solutions en AG imite la sélection naturelle des organismes vivants, et ce processus est accompli grâce à des applications répétées

de trois opérateurs génétiques : sélection, croisement et mutation (figure 3.4). Premièrement, les meilleurs chromosomes sont sélectionnés pour devenir parents afin de produire une nouvelle progéniture (nouveaux chromosomes). Afin simuler le principe de la sélection naturelle, les chromosomes avec une meilleure aptitude sont sélectionnés avec des probabilités plus élevées que les chromosomes avec une aptitude moins bonne. Les probabilités de sélection sont généralement définies à l'aide d'untri relatif aux valeurs de fitness. Une fois les chromosomes parents sélectionnés, l'opérateur de croisement combine les chromosomes des parents pour produire une nouvelle progéniture (perturbation des anciennes solutions). Étant donné que des individus plus forts sont sélectionnés plus souvent, les nouvelles solutions ont tendance à devenir très similaires après plusieurs générations et la diversité de la population peut décliner ; et cela pourrait conduire à une stagnation de la population. La mutation est un mécanisme permettant d'injecter de la diversité dans la population pour éviter la stagnation. Des discussions plus détaillées peuvent être trouvées dans la littérature [18], [8][24][25].

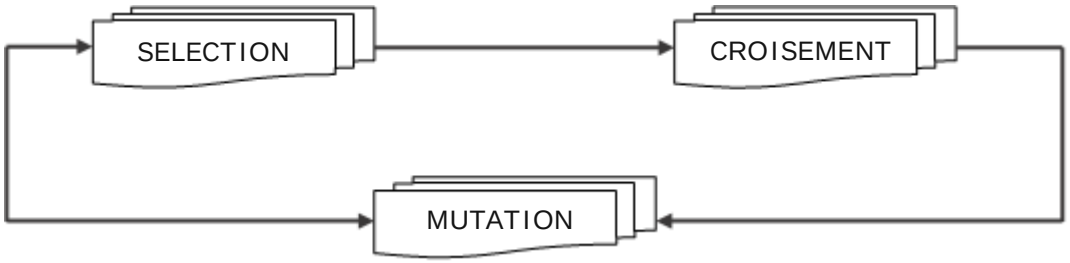


FIGURE 3.4 – *Boucle de base d'un GA.*

Quatre paramètres principaux affectent les performances des AG : la taille de la population, le nombre de générations, le taux de croisement et le taux de mutation. Une plus grande taille de population (c'est-à-dire des centaines de chromosomes) et un grand nombre de générations (des milliers) augmentent les chances d'obtenir une solution optimale globale, mais augmentent considérablement le temps d'exécution.

En plus des 4 paramètres précédents, d'autres choix doivent être pris pour un AG,par exemple, la méthode de sélection et le mécanisme d'attribution des probabilités basé sur l'aptitude (fitness). Différentes méthodes de sélection peuvent nécessiter différents mécanismes d'attribution des probabilités afin d'assurer la diversité de la nouvelle population et l'amélioration des solutions. Il existe de nombreuses méthodes de sélection proposées dans la littérature, les deux plus populaires sont la sélection de la roulette et la sélection par tournoi. La méthode de croisement et la probabilité de croisement sont un deuxième choix à prendre en considération lors de l'exécution des AG.

De nombreuses méthodes de croisement sont rapportées dans la littérature car les méthodes de croisement simples ont tendance à produire des chromosomes irréalisables

ou inutilisables pour de nombreux problèmes d'optimisation complexes. Pour finir, la méthode de mutation et la probabilité de mutation doivent être bien choisies car elles peuvent aider à maintenir une bonne diversité de la population en injectant de nouveaux éléments dans les chromosomes. En général, ces trois choixsont pris de façon empirique à l'aide d'exécutions et de tests.

3.9.3 Le fonctionnement d'un Algorithme Génétique

Le fonctionnement d'un AG est alors basé sur les phases suivantes (illustrés dans la figure 3.5) :

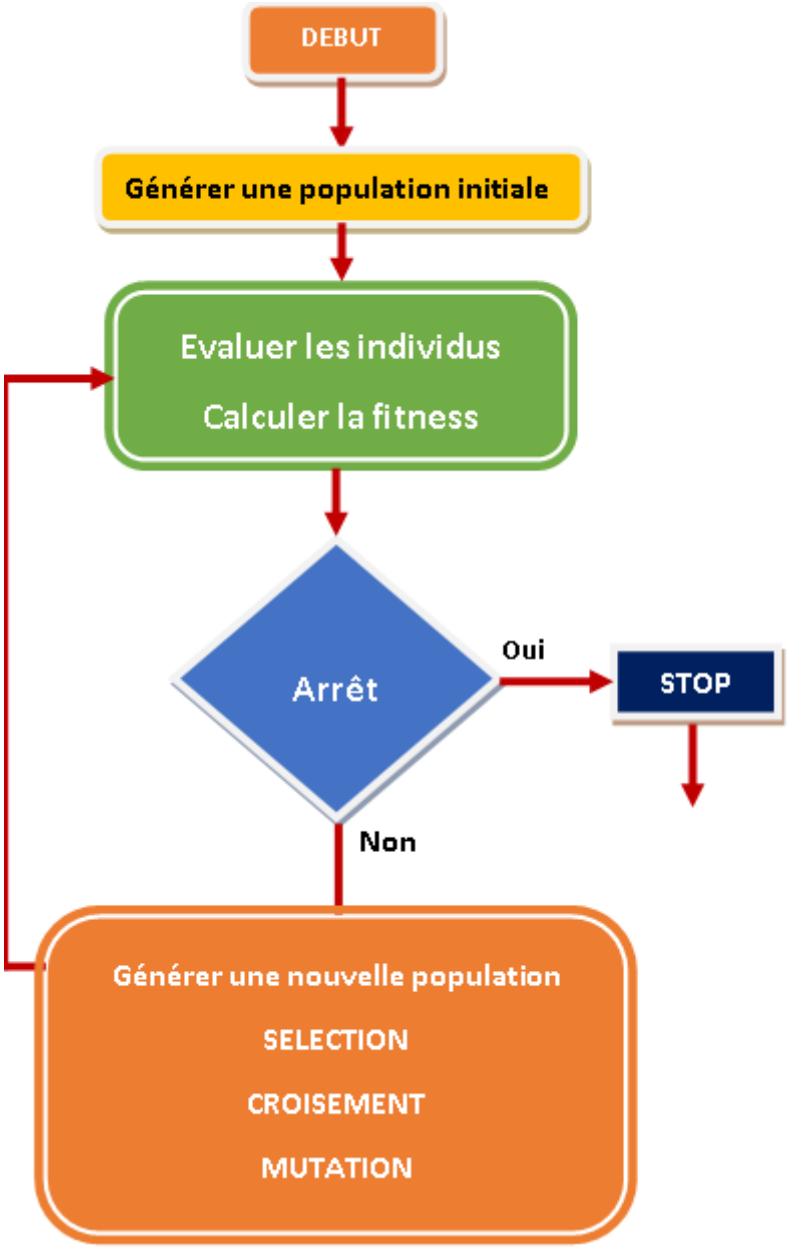


FIGURE 3.5 – *L'organigramme de l'algorithme génétique.*

A) Population initiale

L'algorithme AGdémarreavec une population initiale aléatoire. Cette population peut être générée à partir d'une distribution aléatoire gaussienneou uniforme pour augmenter la diversité. Cette population comprend de multiples solutions, qui représentent les chromosomes d'individus. Chaque chromosome possède un ensemble de variables, qui simule les gènes. L'objectif principal de l'étape d'initialisation est de répartir le plus uniformément possible les solutions dans l'espace de recherche pour augmenter la diversité de la population et avoir de meilleures chances de trouver des régions prometteuses.

Il faut représenter les différents états possibles de la variable dont on cherche la valeur optimale sous uneforme utilisable pour un AG : c'est l'étape de codage. Il existe principalement trois types de codage : le codage binaire (alphabet minimum 0,1), le codage réel et le codage en basen. Les sections suivantes discutent les étapes clés pour améliorer les chromosomes de la population initiale.

B) La sélection

Au cours de chaque génération, une partie de la population existante est sélectionnée pour produire une nouvelle génération. Les individus sont sélectionnés selon un processus basé sur leur fitness, où les solutions les plus performantes sont généralement plus susceptibles d'être sélectionnées. Certaines méthodes de sélection évaluent la fonction de fitness de chaque solution et sélectionnent les meilleures solutions. D'autres méthodes évaluent uniquement un échantillon aléatoire de la population, ce qui rend le processus moins long.

La plupart des méthodes de sélection sont conçues de manière à sélectionner une petite proportion de solutions moins performante afin de conserver une certaine diversité de population, évitant ainsi une convergence prématurée.

Les méthodes de sélection les plus populaires comprennent : La sélection par roue de roulette qui est l'un des nombreux opérateurs de sélection dans la littérature[26][27][28] ; sélection de Boltzmann [29] ; sélection par tournois[30] ; sélection par rang [31] ; sélection de l'état permanent [32] ; sélection par troncature [33] ; sélection locale [34] ; sélection floue [35] ; sélection uniforme par fitness [36] ; sélection proportionnelle [37] ; sélection par rang linéaire [37] ; reproduction en régime permanent [38]

a) Sélection élitiste

Elle garantit que le ou les meilleurs éléments de la population survivent de génération en génération. La stratégie élitiste la plus élémentaire copie le meilleur élément de la population actuelle vers la population suivante, si cet élément n'a pas été transféré par le processus normal de sélection, de croisement et de mutation. Toute méthode de sélection standard peut être rendue élitiste.

b) Sélection de rang

Chaque individu de la population se voit attribuer un rang numérique basé sur sa fitness, et la sélection est basée sur ce classement plutôt que sur la valeur de la fitness. L'avantage de cette méthode est qu'elle peut empêcher des individus très performants de dominer prématurément la population, ce qui réduirait la diversité génétique de la population et pourrait entraver les tentatives de trouver de meilleures solutions.

c) Sélection générationnelle

Les fils des individus sélectionnés à partir de chaque génération deviennent la génération suivante. Aucun individu n'est conservé entre les générations.

d) Sélection hiérarchique

Les individus passent par plusieurs cycles de sélection à chaque génération. Les évaluations de niveau inférieur sont plus rapides et moins discriminantes, tandis que celles qui survivent à des niveaux plus élevés sont évaluées plus rigoureusement. L'avantage de cette méthode est qu'elle réduit le temps de calcul global en utilisant une évaluation plus rapide et moins sélective pour éliminer la majorité des individus qui montrent peu ou pas de promesses, et ne retenir que les individus qui survivent à ce test initial.

e) Sélection du tournoi

Elle simule n tournois pour choisir n individus. Chaque tournoi consiste à échantillonner k éléments de la population pour choisir le meilleur,voir [33]pour une analyse détaillée de la sélection par tournois. Si la taille du tournoi est plus grande, les individus faibles ont moins de chances d'être sélectionnés.

f) Sélection de roue de roulette

Dans la sélection par roulette une probabilité de sélection est calculée pour chaque individu. Cette probabilité est donnée par :

$$P_s(i) = \frac{f(i)}{\sum_{i=1}^{n} f(i)} \tag{3.2}$$

Où $f(i)$ est la fitness de l'individu i et n est le nombre d'individus dans la population. Les solutions candidates avec une meilleure fitness seront moins susceptibles d'être éliminées. Néanmoins, avec une sélection proportionnée par la fitness, il y a une chance que des solutions plus faibles survivent au processus de sélection ; ceci est un avantage car, bien qu'une solution puisse être faible, elle peut inclure certains gènes qui

pourraient s'avérer utiles après le processus de croisement. L'analogie avec une roue de roulette peut être envisagée en imaginant une roue de roulette dans laquelle chaque solution candidate représente une tranche sur la roue. La taille des tranches est proportionnelle à la probabilité de sélection de la solution. La sélection de n chromosomes dans la population équivaut à jouer à n jeux sur la roulette, car chaque candidat est choisi indépendamment (voir figure 3.6).

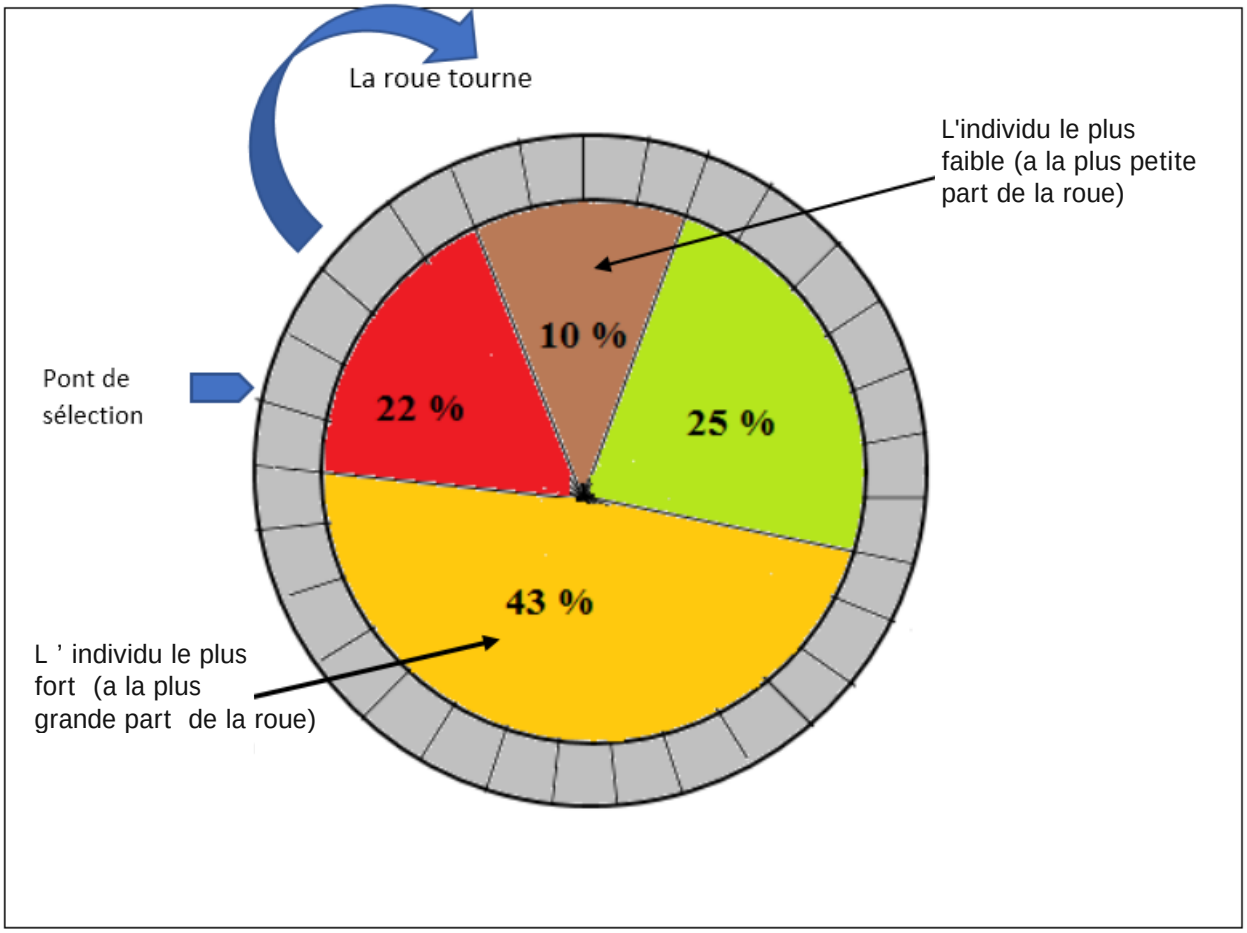


FIGURE 3.6 – *Sélection de la roue de roulette.*

C) Le croisement

Après l'application d'un opérateur de sélection, les individus doivent être employés pour créer une nouvelle génération. Le croisement est un opérateur génétique qui ressembleau croisement naturelet qui combine (accouple) deux chromosomes (parents) pour produire un nouveau chromosome (progéniture). L'idée derrière le croisement est que le nouveau chromosome peut être meilleur que les deux parents s'il prend les meilleures caractéristiques de chacun des parents. Le croisement se produit au cours de l'évolution selon une probabilité de croisement définit par l'utilisateur. De nombreuses

techniques de croisement existent dont : le croisement à un seul locus, à double locus et à multiple locus[39].

Afin d'appliquer l'opérateur de croisement, la population courante est divisée en deux sous populations de même taille (N/2) et chaque couple est formé par un membre provenant de chaque sous population participe à un croisement avec une probabilité (p_c) sa valeur est généralement choisie dans la plage 0.5, ..., 0.8. Cela signifie qu'uncroisement est appliqué aux individus avec une probabilité pcsinon un clonage avec une probabilité de $1 - P_c$. Si le croisement a lieu entre deux chromosomes parents (ch_1etch_2), constitués de l gènes, on tire aléatoirement une ou plusieurs positions de chacun des parents. On échange ensuite les sous chaînes obtenues à partir de ces positions de chacun des chromosomes, ce qui produit deux enfants $ch'_1etch'_2$

• **Croisement en un point :** comme le montre la figure 3.7, dans cette technique de croisement, les chromosomes des parents sont coupés en un point commun choisi au hasard et les sous-chromosomes résultants sont échangés [40].

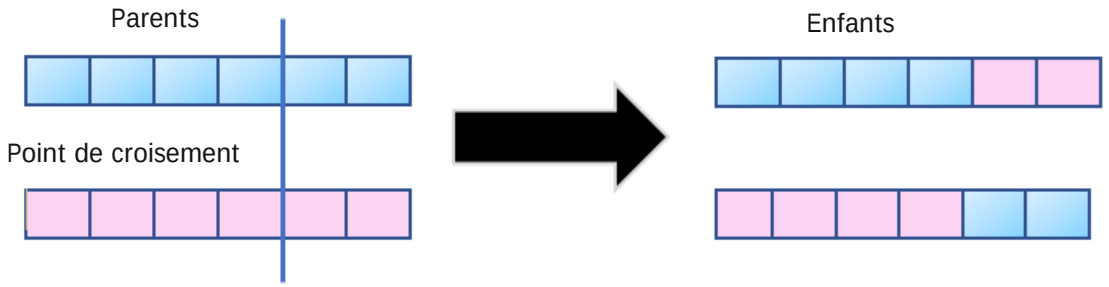


FIGURE 3.7 – *Croisement à un point.*

Croisement à deux points : Deux sites de croisement sont sélectionnés au hasard et les chromosomes parents échangent le segment qui se trouve entre les deux sites de croisement. Voir la figure 3.8

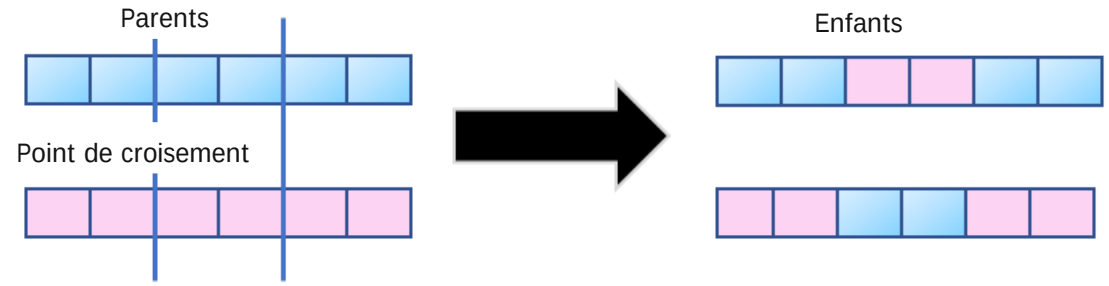


FIGURE 3.8 – *Croisement à deux points.*

• **Croisement uniforme :** Chaque gène de la progéniture est sélectionné au hasard parmi les gènes correspondants des parents comme le montre la figure 3.9. Notez que le croisement à un point et à deux points produit deux descendants, tandis que le croisement uniforme n'en produit qu'un seul.

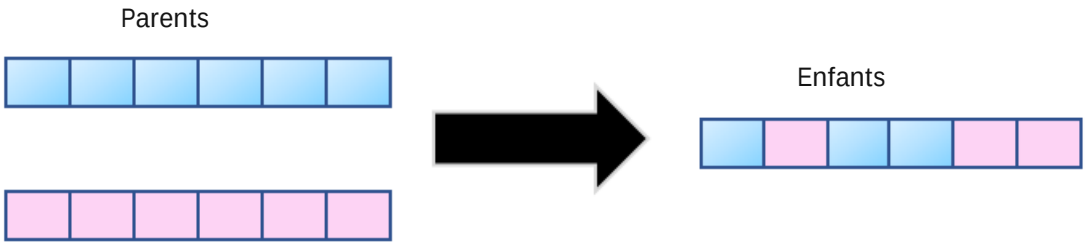


FIGURE 3.9 – *Croisement Uniforme.*

Ces types de croisement sont très efficace uvent s'étendre à n'importe quel type de chaînes (réelles, en base n, etc...). Néanmoins certains auteurs [41] préfèrent utiliser un croisement de type barycentre dans le cas de chaînes réelles, deux gènes $ch_1(i)$ et $ch_2(i)$ sont sélectionnés dans chacun des parents à la même position i. Ils définissent deux nouveaux gènes $ch'_1(i)$ et $ch'_2(i)$ par combinaison linéaire :

$$\begin{cases} ch'_1(i) = \alpha * ch_1(i) + (1-\alpha) * ch_2(i) \\ ch'_2(i) = (1-\alpha) * ch_1(i) + \alpha * ch_2(i) \end{cases} \tag{3.3}$$

Où α est un paramètre de pondération aléatoire qui prend généralement ses valeurs dans l'intervalle [-0.5, 1.5], ceci permet de générer des points entre ou à l'extérieur des deux gènes considérés. D'autres techniques de croisement dans la littérature sont : le croisement uniforme [42] ; le croisement semi-uniforme [43] ; le croisement de trois parents [44] ; le croisement partiellement adapté [45] ; le croisement de cycle [46] ; la commande de croisement [47] ; le croisement heuristique [48] ; le croisement masqué [49]et le croisement multipoint [50].

D) La mutation

La mutation est un opérateur génétique qui modifie une ou plusieurs valeurs génétiques dans un chromosome par rapport à son état initial. Cela ressemble aux accidents naturels qui peuvent survenir lorsque l'ADN est copié. Cela peut entraîner l'ajout de valeurs de gènes entièrement nouvelles à la population. Avec ces nouvelles valeurs génétiques, l'algorithme génétique peut être en mesure d'augmenter la diversité de la population et éviter les optima locaux. La mutation se produit au cours de l'évolution

selon une probabilité de mutation définie par l'utilisateur. Dans un chromosome binaire, il consiste à retourner des bits aléatoires du génotype.



FIGURE 3.10 – *Mutation.*

Pour les chaînes de bits, la mutation par retournement de bits est généralement utilisée ce qui consiste à basculer un bit de 0 à ce versa avec une probabilité définie. Pour les chaines codées en base n, la mutation consiste à remplacer le gène initial par un chiffre en base n tiré au sort (voir figure 3.10). Dans le cas d'un codage réel, on utilise principalement deux opérateurs de mutation : la mutation uniformeet la mutation non uniforme[51], la mutation gaussienne[52] est l'opérateur le plus populaire. La plupart des processus dans la nature suivent une distribution gaussienne. La Probabilité de mutation (Pm) est la probabilité de retourner un bit individuel. Dans les AG, Pm est généralement maintenu très bas, généralement dans la plage [0.001, 0.01]. S'il est réglé trop haut, la recherche se transformera en une recherche aléatoire primitive. Une bonne règle pourrait être $P_m = 1/l$, où l est le nombre de bits dans le chromosome. Certaines des techniques de mutation populaires dans la littérature sont : Mutation de puissance [53] ; Uniforme [54] ; Non uniforme [55] ; Gaussienne[52] ; Réduction de la taille [56] ; Mutation supervisée [57] ; Mutation d'unicité [58] ; Variation de la probabilité de mutation [59].

E) Fonction d'évaluation (fitness function)

Un algorithme génétique nécessitela définition d'une fonction rendant compte de la pertinence des solutions potentiellesà partir des variables à optimiser, cette fonction est nommée fonction d'adaptationf (ou fitness function en terminologie anglosaxonne). Elle est souvent exprimée par la composition de deux fonctions g et h :

$$f = g \circ h \tag{3.4}$$

avec : ∘ : loi de composition de fonctions

g : fonction de transformation, pouvant être linéaire, exponentielle, logarithmique, etc. . . .

h : fonction objectif ou de coût, elle dépend de l'objectif recherché.

Critères d'arrêt

Le critèred'arrêtdéfinitla condition avec laquelle la boucle évolutive principale se termine. Souvent, l'algorithme évolutionnaire s'exécute pendant un nombre prédéfini de générations. Cela peut être raisonnable dans divers contextes expérimentaux. Le temps et le coût des évaluations des fonctions de fitness peuvent limiter la durée du processus d'optimisation. Une autre condition d'arrêt est la convergence du processus d'optimisation. Lors de l'approximation de l'optimum, la progression des améliorations de la fonction de fitness peut diminuer de manière significative, si aucune amélioration significative n'est observée, le processus évolutif s'arrête.

3.9.4 Forces et limites des Algorithmes Génétiques

Dans cette section, les avantages et inconvénients des AG sont décrits

Les forces des algorithmes génétiques

— Les algorithmes génétiques sont intrinsèquement parallèles. De nombreux algorithmes ne peuvent explorer l'espace de solution d'un problème que dans une direction à la fois. Cependant, étant donné que les AG ont plusieurs descendants, ils peuvent explorer l'espace des solutions dans plusieurs directions à la fois.

— Les algorithmes génétiques sont bien adaptés pour résoudre des problèmes où l'espace de toutes les solutions potentielles est extrêmement grand ou trop vaste pour effectuer une recherche exhaustive dans un laps de temps raisonnable.

— Les algorithmes évolutifs, et en particulier les AG, se sont avérés efficaces pour échapper aux optima locaux et s'approcher del'optimum global, même pour une fonction objectif très complexe. Il n'y a généralement aucun moyen de dire si une solution donnée à un problème est l'optimum global ou simplement un optimum local très élevé, mais il est possible de dire qu'un AG peut presque toujours fournir une très bonne solution.

— Les algorithmes génétiques peuvent optimiser de nombreux objectifs simultanément. De nombreux problèmes du monde réel ne peuvent pas être énoncés en termes d'une valeur unique à maximiser, mais doivent être exprimés en termes d'objectifs multiples, généralement avec des compromis : l'un ne peut être amélioré qu'aux dépens d'un autre. Les AG sont très bons pour résoudre de tels problèmes.

— Les AG ne savent rien des problèmes qu'ils doivent résoudre. Au lieu d'utiliser des informations spécifiques au domaine précédemment connues pour guider chaque étape et apporter des changements avec un œil spécifique vers uneamé-

lioration, comme le font les concepteurs humains, ils apportent des changements aléatoires à leurs solutions candidates, puis utilisent la fonction de fitness pour déterminer si ces changements produisent une amélioration.

Les limites des AG

— La première considération lors de la création d'un algorithme génétique est de définir une représentation du problème en question. Ce n'est pas anodin, car il n'existe pas de méthodologie pour le faire. La représentation choisie doit tolérer des changements aléatoires dans les solutions et éviter des résultats absurdes, qui ne satisfont pas aux contraintes du problème.

— Si la fonction de fitness est mal choisie ou définie de manière imprécise, l'algorithme génétique peut être incapable de trouver une solution au problème ou finir par résoudre le mauvais problème.

— En plus de faire un bon choix de fonction de fitness, les autres paramètres d'un AG (la taille de la population, le taux de mutation et de croisement, le type et la force de la sélection) doivent également être choisis avec soin. Si la taille de la population est trop petite, l'algorithme génétique peut ne pas explorer suffisamment l'espace des solutions pour trouver systématiquement de bonnes solutions. Si le taux de mutation génétique est trop élevé ou si le schéma de sélection est mal choisi, le schéma bénéfique peut être perturbé et la population risquede changer trop rapidement pour qu'une sélection puisse aboutir vers une convergence.

— Un problème bien connu qui peut survenir avec un AG est connu sous le nom de convergence prématurée. Si un individu très fortpar rapport à la population émerge tôt, il peut se reproduire si abondamment qu'il réduit trop tôt la diversité de la population. Cela signifie que l'algorithme risque de convergertrès tôt vers un optimum local au lieu de diversifierla recherche vers un meilleur optimum.

3.10 Optimisation par l'essaim de particules

3.10.1 Introduction

L'optimisation par essaims de particules (PSO) est un algorithme de recherche heuristique basé sur le comportement de flocage / essaimage observé chez diverses espèces dans la nature. Développé par Kennedy et Eberhart [60],la PSO s'inspire du comportement social d'une volée d'oiseaux migrateurs essayant d'atteindre une destination inconnue. Dans une PSO, chaque solution est un « oiseau » dans une nuée appelé « particule ». Une particule est analogue à un chromosome (individu de la population) dans les AG.

Physiquement, cela imite une volée d'oiseaux qui communiquent ensemble lorsqu'ils volent. Chaque oiseau regarde dans une direction spécifique, puis lorsque des oiseaux communiquent, ils identifient l'oiseau qui se trouve au meilleur endroit. En conséquence, chaque oiseau se dirige vers le meilleur oiseau en utilisant une vitesse qui dépend de sa position actuelle. Chaque oiseau examine alors l'espace de recherche à partir de sa nouvelle position locale, et le processus se répète jusqu'à ce que la nuée atteigne une destination souhaitée. Il est important de noter que le processus implique à la fois une interaction sociale et une intelligence afin que les oiseaux apprennent de leur propre expérience (recherche locale) et également de l'expérience des autres oiseauxautour d'eux (recherche globale).

Dans une PSO, une solution est représentée comme une particule et la population de solutions est appelée un essaim de particules. Chaque particule a deux propriétés principales : la position et la vitesse. Chaque particule se déplace vers une nouvelle position en utilisant la vitesse. Une fois qu'une nouvelle position est atteinte, la meilleure position de chaque particule et la meilleure position de l'essaim sont mises à jour au besoin. La vitesse de chaque particule est ensuite ajustée en fonction des expériences de la particule. Le processus est répété jusqu'à ce qu'un critère d'arrêt soit satisfait.

Contrairement aux AG, le processus évolutif dans les PSO ne crée pas de nouveaux oiseaux à partir des parents. Au contraire, les oiseaux de la population n'évoluent que leur comportement social et par conséquent leur déplacement vers une destination [61].

Une différence importante entre lesPSO et d'autres algorithmes bio-inspirés, tels que les algorithmes génétiques (GA) ou l'évolution différentielle (DE), est que chaque particule a une composante mémoire. La mémoire est une partie importante de la PSO et permet aux particules de revenir vers de bonnes positions précédemment connues, par rapport aux AG ou à la DE

3.10.2 Fonctionnement du PSO

Les quatre étapes principales du PSO sont(illustrées dans la figure 3.11) :

A) Initialisation

Tout d'abord, les particules de l'essaim sont initialisées. Cela implique de définir leurs positions de départ initiales dans l'espace de recherche et de définir une vitesse initiale. L'initialisation la plus simple consiste à utiliser une distribution uniforme, utilisée dans la définition de la norme PSO 2007 [62], bien que Clerc [63] suggère qu'elle soit un mauvais choix à utiliser. En effet, sur un échantillon d'initialisation uniforme, l'espace de recherche ne sera pas couvert uniformément, en moyenne sur de nombreuses initialisations, mais dans le cas de chaque initialisation individuelle, il peut ne pas fournir une représentation raisonnable. En tant que tel, Clerc [63] suggère d'utiliser une distribution de Hammersley ou une méthode de Tessellations pour fournir une couverture plus uniforme de l'espace de recherche.

Le processus est initialisé avec un groupe de N particules aléatoires (solutions), La ième particule est représentée par sa position en tant que point dans un espace à S dimensions, où S est le nombre de variables. Tout au long du processus, chaque particule i surveille trois valeurs

— Sa position actuelle (X_i) .

— La meilleure position atteinte lors des cycles précédents (P_i) .

— Sa vitesse de vol (V_i).

Ces trois valeurs sont représentées comme suit :

$$\textit{Position actuelle} \qquad X_i = (x_{i1}, x_{i2}, \dots x_{is}) \tag{3.5}$$

$$\textit{Meilleur position } P, \textit{prcdentes} \qquad P_i = (p_{i1}, p_{i2}, \dots p_{is}) \tag{3.6}$$

$$\textit{Vitesse de vol } V \qquad V_i = (v_{i1}, v_{i2}, \dots v_{is}) \tag{3.7}$$

B) Évaluation de la fitness

AAu début de chaque itération, la position de chaque particule est évaluée par la fonction objectif ; la valeur de la fonction à cette position est la valeur de fitness de la particule. L'évaluation de la fitness est généralement la partie la plus coûteuse en calcul, et en tant que telle, il est souhaitable d'avoir un algorithme qui converge rapidement pour réduire le nombre d'évaluations.

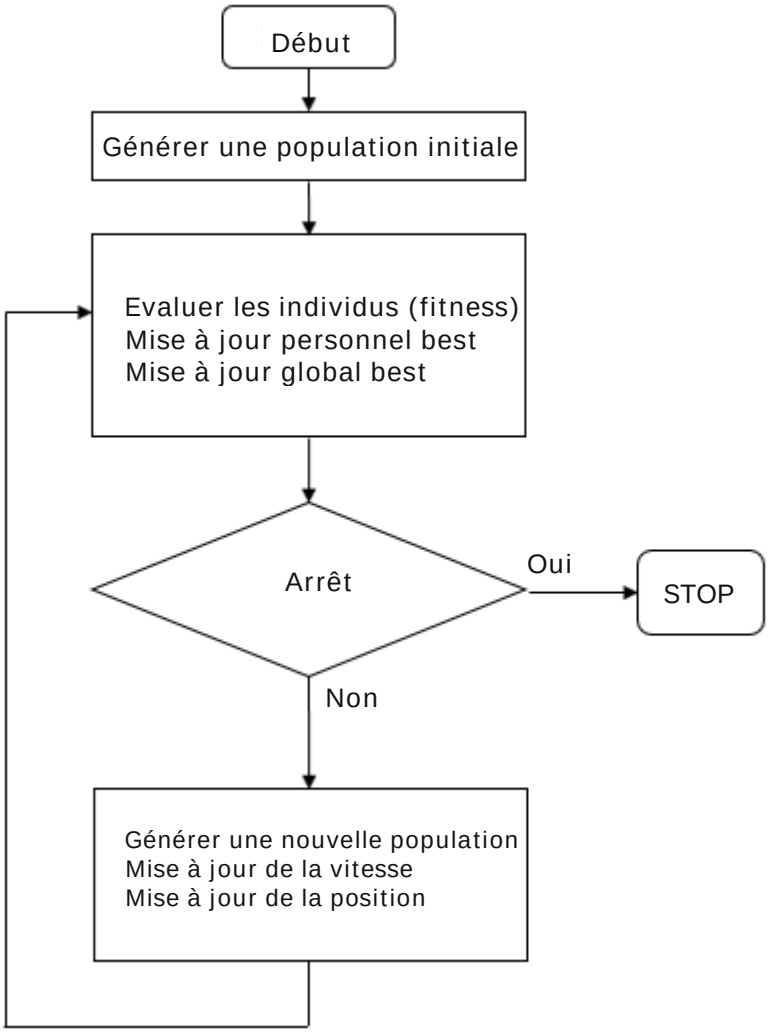


FIGURE 3.11 – *Organigramme de l'algorithme d'optimisation par essaims de particules.*

C) Mise à jour de la position et de la vitesse

Les deux opérations clés dans les PSO sont la mise à jour de la position et la mise à jour de la vitesse. Cette dernière caractérise la dynamique du mouvement de la particule à travers l'espace de recherche. Dans chaque intervalle de temps (cycle), la position (P_g) de la meilleure particule (g) est calculée comme étant la meilleure forme physique de toutes les particules. En conséquence, chaque particule met à jour sa vitesse V_i pour rattraper la meilleure particule g, comme suit : La vitesse est mise à jour en fonction de trois composants :

— L'ancienne vitesse (inertie ou terme d'élan),
— L'expérience d'une particule individuelle (terme cognitif ou d'auto-apprentissage),
— L'expérience de l'ensemble de l'essaim (terme d'apprentissage collectif ou social).

Chaque terme est associé à une constante de poids. Pour l'algorithme PSO de base, le nombre de constantes requises est de trois.

$$v_i(t) = v_i(t-1) + C_1 r_1 * (x_i - y_i) + C_2 r_2 * (x_i - y_i) \tag{3.8}$$

$$x_i(t) = x_i(t-1) + v_i(t) \tag{3.9}$$

y^j est simplement la position la plus connue de toutes les particules de l'essaim.

$$y_{gBest} = argminf(y_i), y \in y_1 ... y_p \tag{3.10}$$

Où alors $y_i = y_{gBest}$ pour tout i.

L'équation de mise à jour de la vitesse se compose de trois parties : partie mémoire ; partie sociale ; et partie cognitive. Il est à noter que le deuxième terme de l'équation (3.4) représente la partie cognitive de la particule en comparant sa position actuelle à sa meilleure position passée. Le troisième terme de l'équation (3.4) représente la collaboration sociale entre les particules, qui compare la position actuelle d'une particule à celle de la meilleure particule [64]. La partie sociale amène l'essaim à se contracter progressivement vers la meilleure solution globale y_{gBest} . La partie cognitive amène la particule à se déplacer vers sa meilleure position passée, y_i.

C_1 et C_2 sont des constantes, r_1 et r_2 sont des vecteurs de nombres aléatoires uniformément distribués dans l'intervalle [0 ; 1] et y_i est la meilleure position du voisinage de l'ième particule. Le voisinage N_i de la particule i est défini comme la topologie d'un réseau par laquelle d'autres particules de l'essaim peuvent partager des informations avec la particule i. La topologie la plus simple est celle où les meilleurs locaux pour toutes les particules sontutilisés pour trouver le meilleur global de l'essaim, $y_i = y_{gBest}$. Dans cette topologie, toutes les particules peuvent partager des informations entre elles, chaque particule est entièrement connectée et y_i est simplement la meilleure position connue de toutes les particules de l'essaim.

Le principal problème rencontré dans l'algorithme PSO est la stabilité / convergence en raison du fait qu'il est possible que la vitesse tende vers l'infini et que les particules s'échappent rapidement de l'espace de recherche.Des limites aux valeurs de vitesse doivent être imposées, $[-v_{max}; v_{max}]$, bien que cela puisse avoir un effet négatif sur la convergence des essaims,ce procédépermet de contrôler la granularité maximale de la recherche,cependant, la valeur maximale reste spécifique au problème. Pour une valeur trop élevés les particules risquent être trop exploratrices ;par contre,une valeur trop petit entraîne un manque d'exploration qui limite la recherche[65].

D'autres améliorations aux PSO ont été apportées en ajoutant un poids d'inertie w au composant de mémoire [15], l'équation de mise à jour de la vitesse modifiée est :

$$v_i(t) = wv_i(t-1) + C_1 r_1 * (x_i - y_i) + C_2 r_2 * (x_i - y_i) \tag{3.11}$$

Un poids d'inertie W est utilisé comme unmoyen pouréquilibrer la capacité de recherche globale et locale des essaims et permet de contrôler le taux de contraction de l'essaim. Un poids d'inertie important (>1.2) favorise une recherche globale en entraînant la particule plus loin dans sa direction actuelle, tandis qu'une petite inertie (<0.8) encourage une recherche locale. En général, une petite inertie est presque toujours favorisée car elle encourage la convergence et la stabilité de l'essaim. Une fois que la nouvelle position d'une particule est calculée à l'aide de l'équation (3.5), la particule vole alors vers elle. A ce titre, les principaux paramètres utilisés dans la technique PSO sont :

— La taille de la population (nombre d'oiseaux).
— Nombre de cycles de génération .
— Le changement maximal d'une vitesse de particule Vmax.
— W(poids d'inertie).

3.10.3 Topologies

La topologie de l'essaim définit la connectivité et la manière dont les informations sont partagées entre les particules de l'essaim. Dans les PSO, la topologie détermine comment le meilleur voisinage y_i (l'équation 3.4) est déterminé.

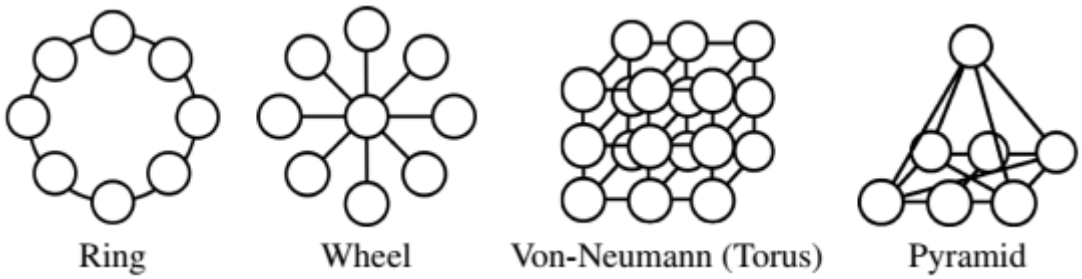


FIGURE 3.12 – *Topologies d'essaim de particules couramment utilisées.*

Les particules dans lesPSO ont trois topologies couramment utilisées [66]connues sous le nom de «gbest (Wheel)», «lbest(Ring)» et «Von-Neumann». Ces topologies sont illustrées à la Figure 3.12

— Topologie en étoile « gbest » : chaque particule est reliée à toutes les autres, l'optimum du voisinage est l'optimum global.
— Topologie en anneau (Ring) : également appelé « lbest », chaque membre de l'essaim n'a que deux voisins, c'est la topologie la plus utilisée.

— Topologie Von-Neumann : chaque particule a quatre voisins (situés au-dessus, en dessous, à gauche et à droite du voisin).
— Topologie en rayon (pyramide) les particules ne communiquent qu'avec une seule particule centrale.

3.10.4 Forces et limites des PSO

a. Les avantages potentiels des PSO sont multiples :

— Robustesse collective : l'échec de composantes individuelles ne peut pas gêner les performances globales.
— Simplicité individuelle : le comportement coopératif rend possible la réduction de la complexité des individus.
— Évolutivité : les mécanismes de contrôle utilisés ne sont pas en fonction du nombre d'agents dans l'essaim.
— Il convient de noter que l'algorithme PSO ne nécessite le tri des valeurs de fitness des solutions dans aucun processus. Cela pourrait être un avantage de calcul significatif par rapport aux AG, en particulier lorsque la taille de la population est importante. Les mises à jour de vitesse et de position dans les PSO ne nécessitent également qu'une simple opération arithmétique sur des nombres réels.

b. Les limites des PSO sont :

— Les nombres aléatoires sont une composante importante de la règle de mise à jour de la vitesse, Wilkes et al [67]montrent que sans le caractère aléatoire, les PSO conduisent à effectuer une recherche sous la forme d'une ligne, tandis que le fait d'avoir des composants aléatoires crée une diversité dans la population.
— Le problème de convergence prématurée, qui peut conduire l'algorithme à stagner dans un optimum local.
— Comme c'est le cas pour d'autres algorithmes en optimisation, l'étape d'initialisation des particules, la quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.
— Le choix des paramètres adéquat est important pour une exécution efficace du PSO. Un travail considérable a été fait pour sélectionner une combinaison de valeurs qui fonctionnent bien dans un large éventail de problèmes.

3.11 Conclusion

Les algorithmes évolutionnaires (AE) constituent une famille d'algorithmes inspirés de l'évolution naturelle. Ces algorithmes sont particulièrement utiles pour la résolution de problèmes où les algorithmes classiques d'optimisation, d'apprentissage ou de conception automatique sont incapables de produire des résultats satisfaisants.

Dans ce chapitre, nous avons décrit le fonctionnement des algorithmes évolutionnaires, nous nous sommes focalisés sur les algorithmes génétiques et l'optimisation par essaim de particule. Nous avons vu que ces algorithmes présentent des avantages importants. Les algorithmes évolutifs, se sont avérés efficaces pour échapper aux optima locaux et découvrir l'optimum global, même pour une fonction objective très complexe. Il n'y a généralement aucun moyen de dire si une solution donnée à un problème est l'optimum global ou simplement un optimum local très élevé.

Experiences et Résultats 4

Sommaire

4.1 Introduction 100
4.2 Problématique 100
4.3 L'approche proposée 101
4.3.1 Le modèle AG-ANFIS proposé : 103
4.3.2 Le modèle PSO-ANFIS proposé 104
4.4 Expérience et Résultats 106
4.4.1 Expérience 1 : La Reconnaissance des phonèmes 107
4.4.2 Expérience 2 : La Reconnaissance des locuteurs 113
4.4.3 Résultats et discussions 115
4.5 Conclusion 120

4.1 Introduction

La reconnaissance de la parole est un domaine qui a connu une évolution impressionnante et d'avancées significatives ont été réalisées en matière d'étude, de recherche et d'exploitation. Ce qui explique notre motivation de travailler dans son domaine.

Dans ce chapitre, on présente notre partie expérimentation et nos contributions à la reconnaissance de la parole. Afin de souligner les apports des méthodes proposées nous avons exploré deux domaines importants de la reconnaissance de la parole qui sont la reconnaissance des phonèmes et la reconnaissance du locuteur. Tout d'abord dans les sections 4.2 et 4.3 on présente la problématique abordée dans cette thèse ainsi que l'approche proposée où on explique en détails les deux modèles d'hybridation AG-ANFIS et PSO-ANFIS proposés. Ensuite, dans la section 4.4 on expose nos expériences et les résultats obtenus, cette dernière est devisée en deux parties ; la première partie consacrée à la reconnaissance des phonèmes de l'ensemble de données TIMIT où l'apprentissage hybride qui combine l'algorithme de rétro-propagation du gradient et la méthode des moindres carrés a été employé. Dans La deuxième partie on a appliqué les deux modèles AG-ANFIS et PSO-ANFIS pour la reconnaissance des locuteurs de l'ensemble de données CHAINS où on explique soigneusement comment ces deux techniques évolutionnaires ont été utilisé pour optimiser les paramètres internes de l'ANFIS.

4.2 Problématique

Le modèle neuro-flou permet de tirer avantages du pouvoir discriminant des réseaux de neurones et de la capacité de raisonnement et de déduction de la logique floue, il peut être formé comme un réseau de neurones avec une interprétation linguistique des variables par la logique floue. Plusieurs architectures ont été proposées, selon le type de règles qu'elles incluent : les règles Mamdani et les règles Sugeno. Dans cette thèse, on a adopté pour le modèle ANFIS (Adaptive Network-based Fuzzy Inference System) l'un des modèles flous les plus influents proposé par Robert Jang [4] qui est une méthode alternative pour la reconnaissance de la parole.

La base de règles de ce modèle contient des règles floues «si-alors» du type Takagi et Sugeno dans lesquelles les parties des conséquents sont des fonctions linéaires des entrées plutôt que des ensembles flous, réduisant ainsi le nombre de règles floues requises.

Lors de la construction de notre modèle ANFIS deux étapes importantes ont été effectuées ; l'identification de la structure qui est la détermination du nombre de règles « si-alors » floues, et des fonctions d'appartenance des ensembles flous des antécédents,

et ceci en utilisant les algorithmes de regroupement. Ensuite, l'apprentissage des paramètres internes des règles floues [8].

L'optimisation de ces paramètres est l'un des principaux enjeux de l'apprentissage de l'ANFIS, l'apprentissage classique est basé sur la descente du gradient, cet algorithme a l'inconvénient de tomber dans de mauvais minimas locaux qui produisent des solutions moins performantes, puisqu'il se limite à un espace de recherche réduit au voisinage d'une solution aléatoire initiale.

4.3 L'approche proposée

Dans cette thèse, on propose une approche alternative d'apprentissage pour optimiser les paramètres du réseau ANFIS plus efficace que la méthode classique. Deux étapes d'apprentissage sont réalisées :

• **Apprentissage de la structure :** permet de déterminer la structure appropriée du réseau, c'est-à-dire le meilleur partitionnement de l'espace d'entrée (le nombre de fonctions d'appartenance pour chaque entrée et le nombre de règles). Si la dimension d'entrée est trop grande ; alors le nombre de règles aura une croissance exponentielle. Ainsi, la détermination d'une structure appropriée devient un problème important.

ANFIS propose trois approches pour identifier les clusters, à savoir le partitionnement de grille, le regroupement soustractif et le regroupement par C moyennes floues. L'approche de partitionnement par grille est utilisée si le nombre d'entrées n'est pas supérieur à 6 ou 7 (ce n'est pas le cas dans cette thèse). Si celui-ci est trop élevé, cette méthode ne pourra pas être utilisée car la mémoire requise sera insuffisante lors de l'utilisation de MATLAB. Pour la méthode des C moyennes floues, le nombre de clusters pour l'ensemble de données doit être spécifié. Puisqu'aucune connaissance préalable sur le nombre de clusters n'est disponible, le regroupement soustractif sera idéal.

Dans notre thèse nous avons utilisé le regroupement soustractif en faisant varier à chaque fois le rayon jusqu'à que la structure la plus optimale soit obtenue. Les paramètres du regroupement soustractifs adoptés sont illustrés dans le tableau 4.1.

• **Apprentissage paramétrique :** effectué pour ajuster les paramètres antécédents et conséquents afin qu'une fonction objectif spécifiée soit minimisée. Il existe plusieurs méthodes d'apprentissage [10] pour mettre à jour ces paramètres, l'apprentissage hybride qui combine l'algorithme de rétro-propagation du gradient GD et la méthode des moindres carrés LSE est l'algorithme utilisé dans cette thèse.

Cet algorithme est réalisé en deux étapes :

— **La passe avant :** où les modèles d'entrée sont propagés vers la sortie et les paramètres conséquents optimaux sont estimés par le LSE, tandis que les paramètres

Table 4.1 – *Les paramètres du FIS utilisé*

Paramètres du FIS	**Valeur**
FIS Type	Sugeno
Type de fonction d'appartenance des entrées	Gaussienne
Type de fonction d'appartenance des sorties	Linéaire
Le facteur d'annulation	1.25
Le taux d'acceptation	0.5
Le taux de rejet	0.15
Le rayon	Par essai

antécédents sont supposés être fixés dans le cycle actuel à travers l'ensemble d'apprentissage.

— **La passe arrière : :** où les modèles sont à nouveau propagés et les paramètres antécédents sont mis à jour par GD, tandis que les paramètres conséquents restent fixes.

Cette procédure hybride est ensuite répétée jusqu'à ce que qu'un nombre maximal de cycles d'apprentissage ait été atteint. Cet algorithme peut rester bloqué dans un optimum local, pour faire face à ce problème, on propose d'utiliser les algorithmes évolutionnaires. Dans cette thèse on a utilisé les AG et les PSO pour optimiser les paramètres antécédents c_{ij} et σ_{ij} le centre et la pente des gaussienne des fonctions d'appartenance) et les paramètres conséquents de l'ANFIS, les deux modèles hybrides AG-ANFIS et PSO-ANFIS sont détaillés par la suite.

4.3.1 Le modèle AG-ANFIS proposé :

Une fois la structure du réseau ANFIS fixée (nombre de fonctions d'appartenance de type gaussienne pour chaque entrée, nombre de règles floues, nombre de paramètres des conséquents) on définit la taille des individus (les solutions) ainsi :

T = Taille de l'individu = paramètres_antécédents + paramètres_conséquents
Paramètres_antécédents =nbr_entrées*nbr_fonction_d'appartenance * gauss
gauss = 2 (paramètres de la fonction gaussienne qui sont le centre et la pente)
Paramètres_conséquents=nombre_classes_sorties*nbr_fonction_d'appartenance*nbr_input+1

Sachant qu'une règle floue du réseau ANFIS est définie comme suit :

Si : $\underbrace{x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } A_2 \text{ ... \& } x_n \text{ est } A_n}_{\text{les antécédents}}$ **alors** $\underbrace{f(x) = r + px_1 + qx_2 + \ldots. + tx_n}_{\text{les conséquents}}$

Le processus évolutif commence, au début, nous avons généré aléatoirement une population initiale de N solutions (chaînes réelles), chaque solution candidate de taille T représente le partitionnement flou de la partie des antécédents des règles et la partie des conséquents. En utilisant les opérations génétiques standards de sélection, de croisement et de mutation, la population entière sera amenée à s'améliorer progressivement jusqu'à que le nombre maximal d'itération soit atteint (1000 itérations) (voir l'organigramme AG-ANFIS proposé (figure 4.2)).

Les paramètres de l'AG utilisés sont déterminés sur la base de plusieurs essais. Les résultats obtenus sont résumés dans le tableau 4.2.

Table 4.2 – *Les paramètres de l'AG utilisé*

Paramètres de AG	**Valeur**
Taille de la population N	[20, 50, 100, 150, 200, 250, 300, 350, 400]
Méthode de sélection	Sélection par roulette
Probabilité de croisement pc	0.4
Probabilité de mutation pm	0.7

4.3.2 Le modèle PSO-ANFIS proposé

Une fois les valeurs des paramètres des antécédents de la fonction d'appartenance et les paramètres conséquents sont fixées, la taille des particules T (les solutions) est connue (voir la section précédente).

Initialement, une population de N particules (de taille T) est générée aléatoirement. Ensuite, l'algorithme PSO distribue ses particules qui constituent un essaim, se déplaçant dans l'espace de recherche. Les particules essayent de trouver la meilleure solution et chaque individu vole dans l'espace de recherche avec une vitesse qui est ajustée dynamiquement en fonction de sa propre expérience de vol et de l'expérience de vol de ses compagnons. Cette vitesse est responsable du mouvement imposé à la particule et sa position spatiale change dans l'espace de recherche pour y trouver une meilleure performance. Les particules convergent vers la position optimale dans l'espace de recherche (voir l'organigramme PSO-ANFIS proposé (figure 4.3)).

Si le critère d'arrêt est satisfait (nombre maximal d'itération), le processus sera arrêté ; sinon, toutes les étapes seront exécutées à nouveau.

Les paramètres du PSO utilisé déterminés par essais et erreurs dans notre thèse. Les résultats obtenus sont résumés dans le tableau 4.3.

TABLE 4.3 – *Les paramètres du PSO utilisé dans notre thèse*

Paramètres du PSO	**Valeur**
Taille de la population	[20, 50, 100, 150, 200, 250, 300]
Poids d'inertie W	1
Accélération cognitive C1	1
Accélération sociale C2	2
r_1 **et** r_2	aléatoire

La fonction à minimiser pour les deux modèles d'hybridation AG-ANFIS et PSO-ANFIS est celle qui calcule le taux de reconnaissance :

$$Accuracy = [\frac{1}{N} Sum(d_i == O_i)] * 100 \tag{4.1}$$

Où d_i est la sortie souhaitée, O_i est la sortie du ANFIS pour l'ième échantillon des données d'apprentissage, et N est le nombre d'échantillons d'apprentissage.

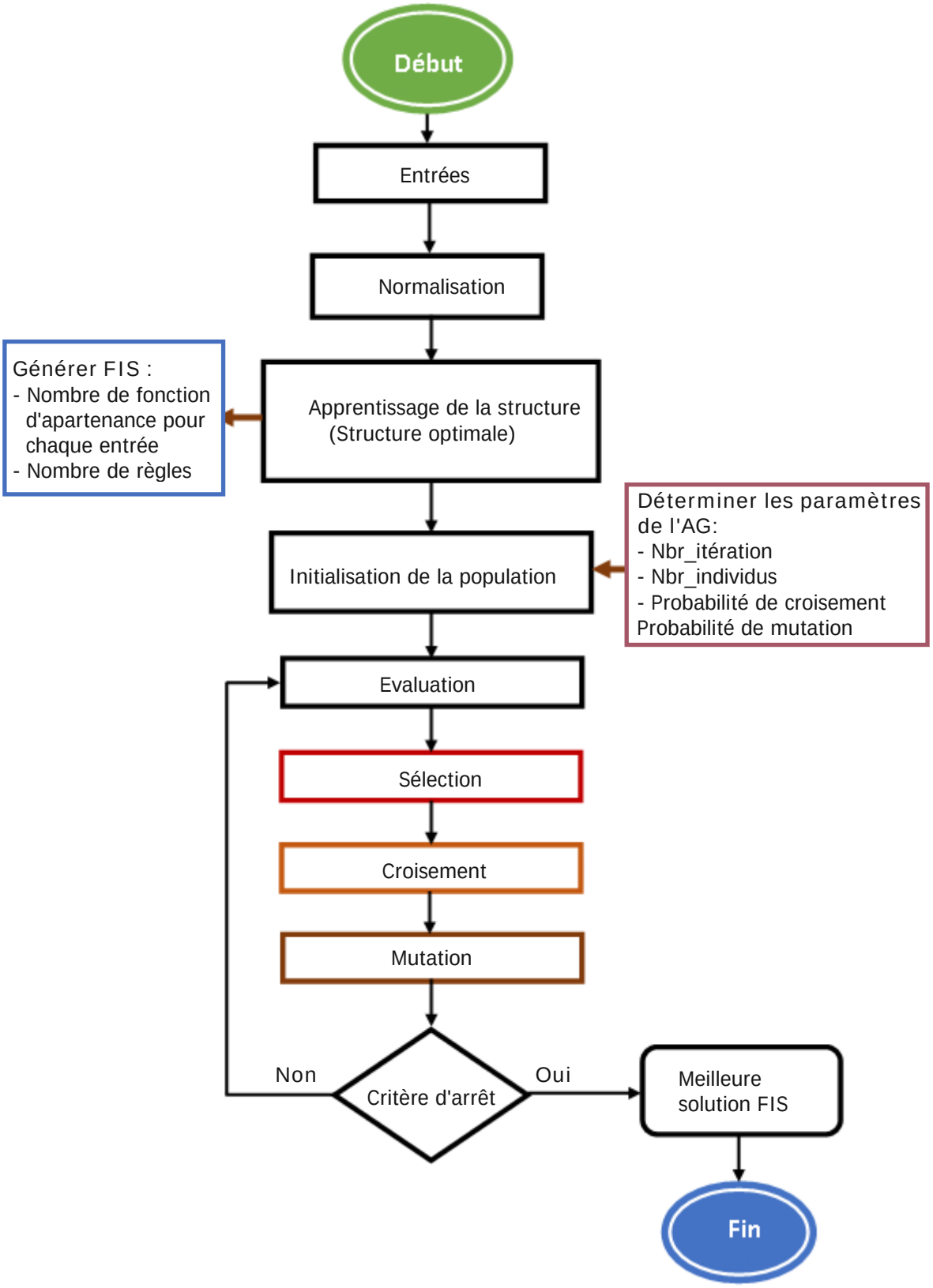


FIGURE 4.1 – *L'organigramme du modèle AG-ANFIS proposé.*

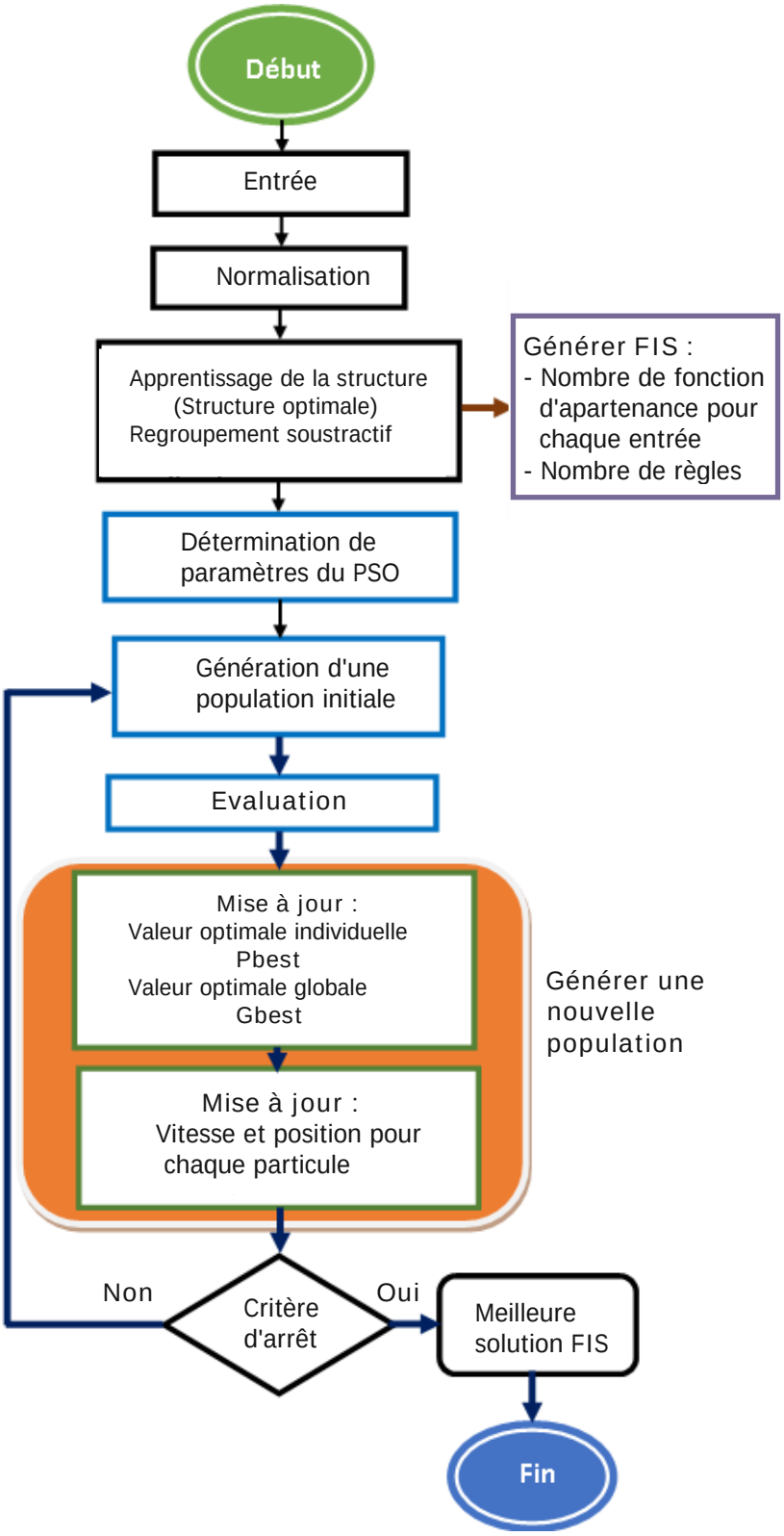


FIGURE 4.2 – *L'organigramme du modèle PSO-ANFIS proposé.*

4.4 EXPÉRIENCE ET RÉSULTATS

Cette thèse explore deux domaines importants de la reconnaissance de la parole qui sont : la reconnaissance des phonèmes et la reconnaissance du locuteur.

Cette partie du manuscrit présente les expériences et les résultats obtenus, afin de mettre en évidence les apports des modèles hybrides proposés. Effectivement, pour comparer plusieurs systèmes de reconnaissance à base d'algorithmes évolutionnaires et de réseaux de neurones, il faut satisfaire un certain nombre de conditions : définir le même critère d'évaluations, utiliser la même base d'apprentissage et la même base de test, les mêmes paramètres d'évolution (nombre d'individus, le nombre de générations, probabilité de croisement, ...).

4.4.1 Expérience 1 : La Reconnaissance des phonèmes

Nous nous sommes intéressés à la reconnaissance vocale basée sur les phonèmes, cet axe de recherche est très attrayant car il est principalement exempté de limitation de vocabulaire. C'est pourquoi les auteurs continuent de développer des systèmes de reconnaissance des phonèmes, afin d'améliorer au maximum leurs performances, d'où notre intérêt à travailler sur cet axe, ainsi nos expériences ont été menées sur la base de données TIMIT.

4.4.1.1 Base de données TIMIT

Le corpus DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT - Texas Instruments (TI) et Massachusetts Institute of Technology (MIT)) [155] contient des enregistrements de discours en anglais transcrits phonétiquement et lexicalement. Enregistré à l'aide d'un microphone de proximité Sennheiser à une fréquence de 16 kHz avec une résolution d'échantillon de 16 bits. TIMIT contient un total de 6 300 phrases (5.4 heures), composées de 10 phrases prononcées par chacun des 630 locuteurs de différents sexes de 8 grandes régions dialectales des États-Unis. Toutes les phrases ont été segmentées manuellement au niveau du phonème.

Un sous-ensemble réduit de TIMIT [11] a été utilisé dans cette thèse, qui est la version abrégée de l'ensemble de test complet, se compose de 192 énoncés, 8 de chacun des 24 locuteurs (2 hommes et 1 femme de chaque région dialectale). En général, les phonèmes peuvent être regroupés en catégories en fonction de traits distinctifs qui indiquent une similitude dans l'articulatoire, l'acoustique et la perception. Le sous corpus utilisé est constitué de 31514 occurrences d'apprentissage et 12055 occurrences pour le test qui sont regroupés en trois classes (tableau 4.4).

4.4.1.2 Le Codage de la base TIMIT par les Coefficients MFCC

Avant l'étape de l'apprentissage, il est nécessaire de transformer le signal de parole en un nombre réduit de caractéristiques pertinentes. Le signal est d'abord filtré par un filtre de préaccentuation pour élever les hautes fréquences. Ensuite, le signal est

segmenté en trames d'une durée fixe N = 25ms (période pendant laquelle le signal de parole est supposé comme stationnaire). Afin de réduire l'effet de bord, une fenêtre de Hamming a été appliquée sur chaque trame du signal.

Après la mise en forme du signal, une transformée de Fourier rapide est appliquée pour passer vers le domaine fréquentiel et pour calculer le signal spectral. Ensuite, un filtrage est effectué en multipliant le spectre obtenu par un banc de filtres (26 filtres triangulaires placés linéairement sur l'échelle Mel) afin de simuler le filtrage effectué par l'oreille humaine. A partir de chaque trame 12 coefficients MFCC sont extraits en utilisant une transformée de cosinus inverse ainsi que l'énergie résiduelle correspondante.

L'utilisation des coefficients MFCC est très intéressant car ils sont plus discriminants, plus robustes et moins corrélés.

4.4.1.3 Apprentissage de la structure de l'ANFIS utilisé pour la reconnaissance des phonèmes

Le rayon des clusters a été utilisé comme base de la formation des clusters. Un intervalle de rayon de 0,0 à 2,5 a été choisi afin de produire un nombre optimal de clusters pour le système d'inférence floue initial (FIS). Le FIS initial a ensuite été transmis à l'ANFIS pour affiner le FIS afin qu'il ajuste l'apprentissage supervisé de manière itérative avec des règles plus détaillées générées.

Chaque centre de cluster représente une règle si-alors floue. La n^{ime}colonne du cième centre de cluster est supposée être la valeur moyenne (c_{in}) de la fonction d'appartenance gaussienne associée définie pour le cième ensemble flou de la n^{ime} variable d'entrée. Ensuite, l'écart type (a_{in}) des fonctions gaussiennes est calculé comme suit :

$$a_{in} = ran(\frac{max(x^n - max(x^n)}{f_b}) \tag{4.2}$$

Par conséquent, les centres de cluster et le facteur d'annulation peuvent être considérés comme des paramètres, dont dépend le nombre de règles floues du FIS initial, avant que les paramètres de base de règles du FIS ne soient réglés par les ANN dans ANFIS.

4.4.1.4 Apprentissage des paramètres de l'ANFIS utilisé pour la reconnaissance des phonèmes

Nous avons utilisé l'algorithme hybride pour ajuster les paramètres internes de l'ANFIS. Les paramètres antécédents sont mis à jour par l'algorithme de rétro-propagation du gradient GD, tandis que les paramètres conséquents sont estimés par la méthode des moindres carrés LSE, Jusqu'à ce que le nombre maximal d'itération soit atteint en calculant à chaque itération le taux de reconnaissance.

TABLE 4.4 – *Sous corpus de la base de données TIMIT utilisé dans notre thèse*

Classe	**Phonème**	**Train**	**Test**
Voyelles	/ ah /	2200	879
	/ aw /	700	216
	/ ax /	3352	1323
	/ ax-h /	281	95
	/ uh /	502	221
	/ uw /	536	170
Fricatives	/ dh /	2058	822
	/ f /	2093	911
	/ sh /	2144	796
	/ v /	1872	707
	/ z /	3574	1273
	/ zh /	146	74
Plosives	/ b /	399	182
	/ d /	1371	526
	/ g /	1337	546
	/ p /	2056	779
	/ q /	3307	1191
	/ t /	3586	1344

4.4.1.5 Résultats et discussion de l'expérience 1

Le corpus utilisé pour la reconnaissance des phonèmes se compose de 6 voyelles, 6 fricatives et 6 plosives : 31514 instances pour l'apprentissage et 12055 instances pour le test. Un regroupement soustractif a été appliqué pour déterminer la structure initiale de l'ANFIS avec un rayon entre [0,0- 2,5], puis l'apprentissage hybride a été réalisé pour trouver un ensemble faisable d'antécédents et de paramètres conséquents, les résultats obtenus sont résumés dans la figure 4.4 et 4.5 :

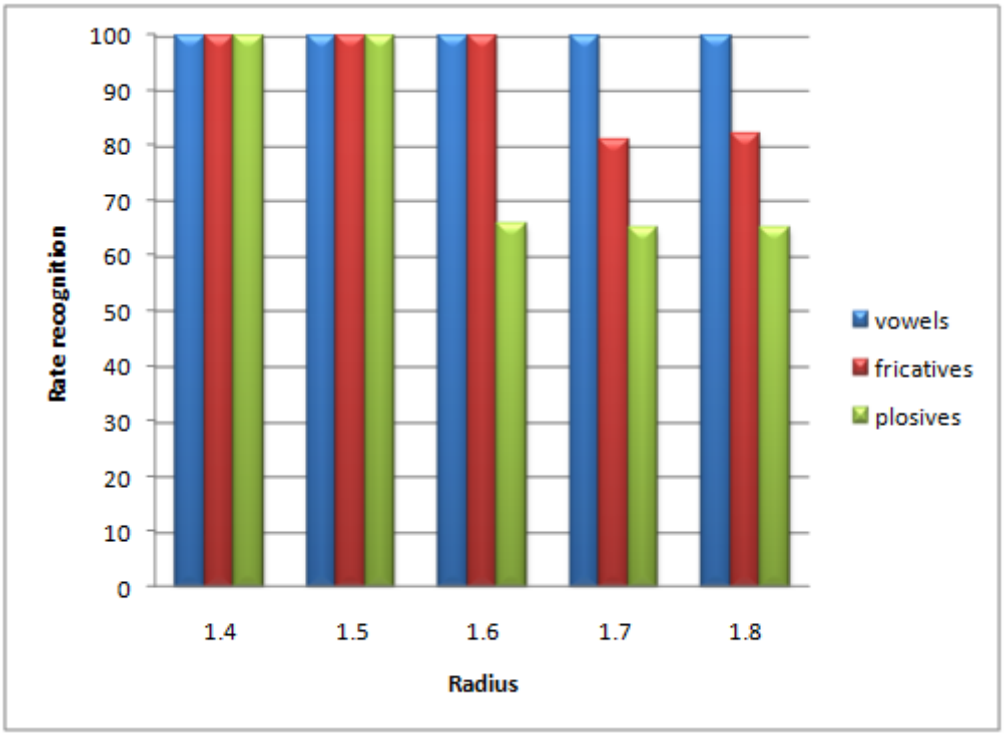


FIGURE 4.3 – *Taux de reconnaissance des phonèmes en fonction du rayon du model ANFIS.*

Avec un rayon entre [0-1.4] un taux de reconnaissance de 100 % a été atteint mais le nombre de règles générées était trop grand, plus de 200 règles ce qui a entraîné un temps d'exécution important et surtout pour les applications en temps réel, et ce pour les trois classes de phonèmes utilisées (figure 4.4 et 4.5).

Pour les voyelles, le meilleur taux de reconnaissance (100 %) a été obtenu avec 6 règles générées pour un rayon de 2.0 et 1.9, le nombre de règles a été réduit de 6 à 4 avec un rayon de 2.1 avec un taux de reconnaissance de 83 % (tableau 4.5).

Pour les fricatives, avec rayon 1.4 un taux de reconnaissance de 100 % a été atteint avec 5 règles floues générées, le nombre de règles peut être réduit à 4 avec un taux de reconnaissance de 80,66 % (tableau 4.6).

Pour les plosives un taux de reconnaissance de 100 % a été atteint avec 5 règles pour

un rayon de 1.4 et 1.5 au-delà de 1.5 3 règles sont obtenues mais réduction du taux de reconnaissance est remarquée (tableau 4.7)

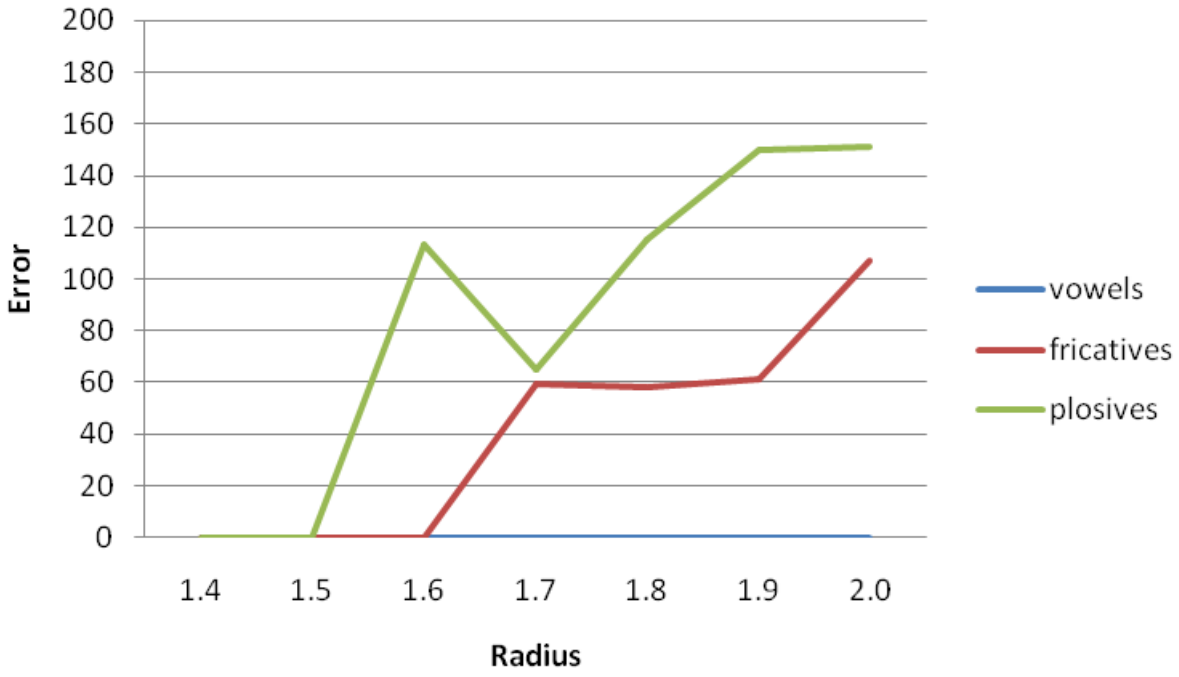


FIGURE 4.4 – *L'erreur Quadratique Moyenne RMSE en fonction du rayon du modèle ANFIS utilisé pour la reconnaissance des phonèmes.*

Un grand nombre de règles sont générées en réduisant le rayon de regroupement afin que la plupart des données soient correctement classées, mais cela prend beaucoup de temps. Au contraire, en augmentant le rayon de regroupement, très peu de règles sont obtenues, de sorte que de nombreuses alternatives ne sont pas considérées par les règles floues, du coup, plusieurs phonèmes ne sont pas correctement classés.

On a pu voir qu'il y avait un compromis entre le taux de reconnaissance et le nombre de règles générées, avec un rayon compris entre [1.5-1.8] de très bonnes performances en termes de taux de reconnaissance ont été obtenues ainsi qu'un nombre réduit de règles générées qui engendrent un temps de calcul très raisonnable.

Pour récapituler, dans cette expérience un réseau adaptatif ANFIS a été appliqué pour la reconnaissance des phonèmes. Un regroupement soustractif a été appliqué pour l'apprentissage d'une structure avec un nombre réduit de règles pour le modèle. L'apprentissage des paramètres du modèle est un apprentissage hybride qui combine la rétro propagation du gradient et l'estimation des moindres carrés LSE pour trouver un ensemble d'antécédents réalisable et les paramètres conséquents.

Les résultats obtenus montrent l'efficacité de la méthode en termes de taux de reconnaissance et du nombre de règles floues générées.

Table 4.5 – *Taux de reconnaissance des voyelles en utilisant le modèle ANFIS*

	Taux-reconnaissance%		**Erreur**		**Nombre - règles**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
1.4	100	100	0	0	37	34
1.5	100	100	0	0	22	23
1.6	100	100	0	0	17	15
1.7	100	100	0	0	11	10
1.8	100	100	0	0	9	8
1.9	100	100	0	0	8	6
2.0	100	100	0	0	7	5
2.1	100	83	0	53	7	4
2.2	78	79	65	62	4	4

Table 4.6 – *Taux de reconnaissance des fricatives en utilisant le modèle ANFIS*

	Taux-reconnaissance%		**Erreur**		**Nombre-règles**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
1.4	100	100	0	0	5	5
1.5	64.33	100	120	0	3	6
1.6	58.66	100	139	0	3	5
1.7	59	80.66	137	59	3	4
1.8	58	81	138	58	3	4

TABLE 4.7 – *Taux de reconnaissance des plosives en utilisant le modèle ANFIS*

	Taux-reconnaissance%		**Erreur**		**Nombre-règles**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
1.4	100	100	0	0	6	5
1.5	100	100	0	0	8	5
1.6	100	65.33	0	113	5	3
1.7	89	65	34	65	4	3
1.8	89	65	34	115	3	2

4.4.2 Expérience 2 : La Reconnaissance des locuteurs

Dans cette deuxième expérience on s'intéresse à la reconnaissance du locuteur. Cette technologie largement utilisée dans diverses situations du monde réel telles que les contrôles d'accès, les applications téléphoniques,...etc.

4.4.2.1 Base de données CHAINS

Le corpus CHAracterizing INdividual Speaker CHAINS [156] est un nouveau corpus de discours collecté dans le but principal de faciliter la recherche sur l'identification du locuteur. Le corpus comprend 36 locuteurs enregistrés dans diverses conditions de parole, ce qui permet de comparer le même locuteur à travers différents styles de parole bien définis. Les locuteurs lisent une variété de textes seuls, en synchronisation avec un co-locuteur correspondant au dialecte, à l'imitation d'un co-locuteur correspondant au dialecte, dans un murmure et à un rythme rapide.

Le corpus CHAINS contient les enregistrements de 36 intervenants. Parmi ceux-ci, 28 sont originaires de la partie orientale de l'Irlande et parlent le hiberno-anglais oriental. Les autres viennent du Royaume-Uni et des États-Unis.

Quatre ensembles de locuteurs disponibles sont utilisés avec des ensembles de 8 locuteurs, 16 locuteurs, 24 locuteurs et 36 locuteurs. Dans cette thèse, les ensembles à 8 et 16 locuteurs ont été utilisés. Les jeux de données sont au format WEKA arff et contiennent de la parole codée à l'aide de 25 MFCC.

4.4.2.2 La structure initiale de l'ANFIS utilisé pour la reconnaissance des locuteurs

Tout d'abord, tous les ensembles de données ont été normalisés dans la plage de (0,1) comme suit :

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.3}$$

Où X_n et X sont respectivement les valeurs normalisées et mesurées. De plus, les valeurs minimale et maximale de l'ensemble de données mesurées sont respectivement X_{min} et X_{max}.

Pour la détermination de la structure ANFIS initiale, l'algorithme de regroupement soustractif a été utilisé pour regrouper l'espace de données afin de déterminer le nombre de règles et les fonctions d'appartenance pour les antécédents et les conséquents.

Pour cette application, la structure ANFIS initiale est générée avec 10 règles. Par conséquent, 10 fonctions d'appartenance de type gaussien pour chaque entrée sont générées. Chaque centre de fonction d'appartenance est initialisé avec le cluster flou correspondant obtenu par le processus de regroupement avec un rayon de 0.5.

4.4.2.3 Apprentissage du réseau ANFIS utilisé pour la reconnaissance des locuteurs

Après la conception de la structure initiale de l'ANFIS, la reconnaissance des locuteurs a été réalisée par trois méthodologies d'apprentissage :

a. L'ANFIS classique :

L'apprentissage hybride qui combine GD et LSE est utilisé, la fonction de référence MATLAB [24]exécute une routine d'entraînement pour le FIS de type Sugeno avec les options d'entraînement suivantes : Nombre d'époques d'entraînement = 10, objectif d'erreur d'apprentissage = 0, taille du pas initial = 0,01, taux de diminution de la taille du pas = 0,9, et taux d'augmentation de la taille du pas = 1,1.

b.L'hybridation AG-ANFIS et PSO-ANFIS

Les entrées au réseau sont les 25 MFCC (20881 instances (apprentissage) et 13626 instances (test)). Après avoir effectué un regroupement soustractif avec un rayon de 0.5 le réseau obtenu est constitué 25 neurones dans la première couche du réseau ; et 10 fonctions d'appartenance pour chaque entrée (25*10) donc 250 neurones dans la deuxième couche.

Tout d'abord, il faut définir les individus de la population (les chromosomes dans le cas des AG et particules dans le cas du PSO), où la taille de l'individu est égale au

nombre de paramètres du système d'inférence flou SIF. Il faut calculer le nombre de paramètres des antécédents et le nombre de paramètres des conséquents :

$$\text{Nbr\_paramètres\_8 locuteurs} = 2580 = \underbrace{2 * 25 * 10}_{\text{les antécédents}} + \underbrace{8 * 10 * (25+1)}_{\text{les conséquents}}$$

$$\text{Nbr\_paramètres\_16 locuteurs} = 2580 = \underbrace{2 * 25 * 10}_{\text{les antécédents}} + \underbrace{16 * 10 * (25+1)}_{\text{les conséquents}}$$

Donc, une population initiale est générée aléatoirement où la taille de la solution est de 2580 (dans le cas de 8 locuteurs) et de 4660 (dans le cas de 16 locuteurs), les paramètres de l'AG et PSO tel que la taille de la population, la technique de sélection... sont résumés respectivement dans le tableau 4.2 et 4.3. Les étapes de la technique évolutionnaire sont déroulées jusqu'à ce que le nombre d'itération soit atteint (1000 itérations).

4.4.3 Résultats et discussions

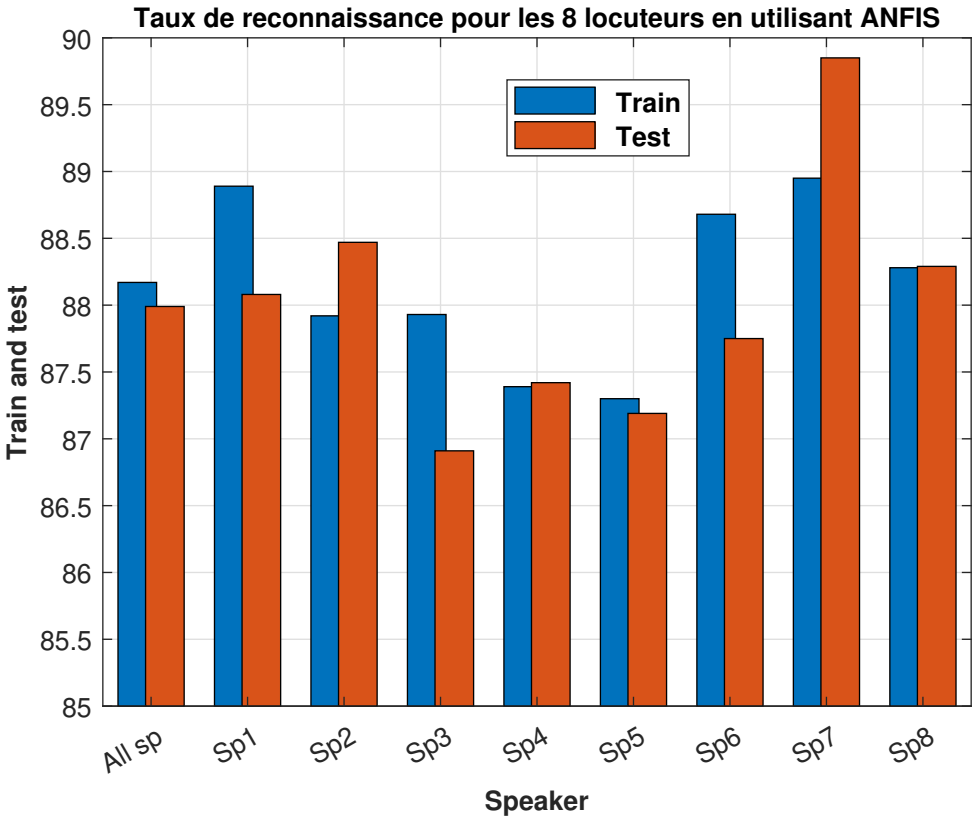


FIGURE 4.5 – *Taux de reconnaissance pour les 8 locuteurs en utilisant ANFIS.*

TABLE 4.8 – *Taux de reconnaissance pour les 8 locuteurs en utilisant AG-ANFIS*

NPop	**20**	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Train	83.09	83.29	84.35	84.02	83.89	84.66	85.10	85.10	85.49
Test	82.77	82.97	83.68	84.31	84.05	83.37	85.06	85.91	85.90

TABLE 4.9 – *Taux de reconnaissance pour les 16 locuteurs en utilisant AG-ANFIS*

Iteration **NPop**	**1000** **100**	**1000** **200**	**1000** **100**	**1000** **200**
Train	89.12	90.54	91.32	91.78
Test	89.98	90.17	91.16	91.20

La figure 4.5 montre les résultats obtenus par l'application de l'ANFIS sur les 8 locuteurs de la base de données CHAINS avec un taux de reconnaissance de 87,99 % ; tandis que la figure 4.6 montre les résultats obtenus par l'application de l'ANFIS sur les 16 locuteurs de la base de données avec un taux de reconnaissance de 93,80

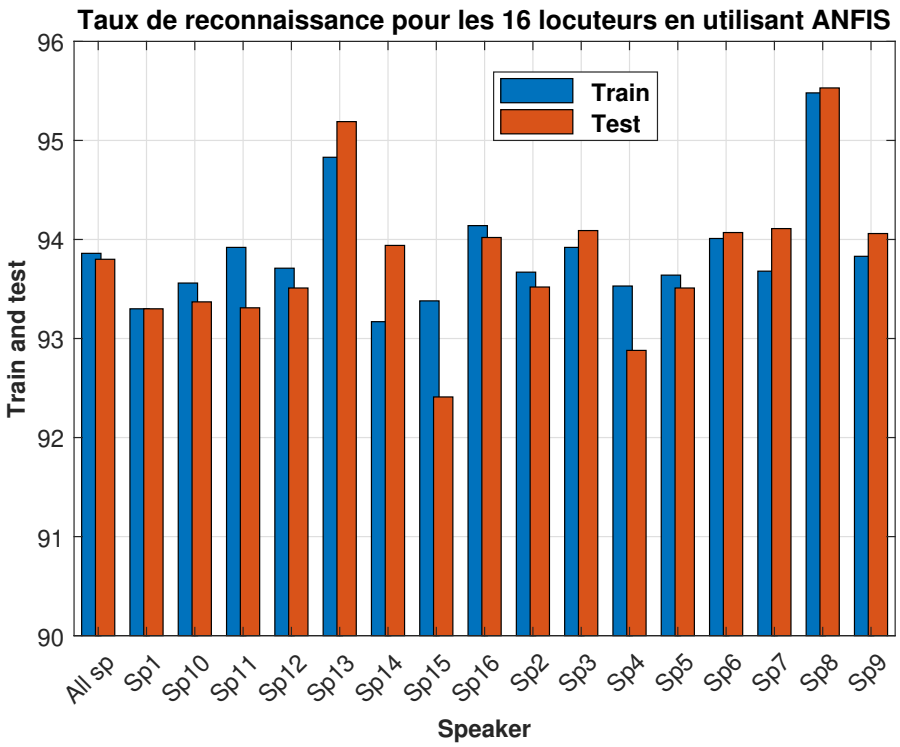


FIGURE 4.6 – *Taux de reconnaissance pour les 16 locuteurs en utilisant ANFIS.*

Table 4.10 – *Taux de reconnaissance pour les 8 locuteurs en utilisant PSO-ANFIS (1000 itérations)*

NPop	**20**	**50**	**100**	**150**	**200**	**250**
Train	88.45	89.29	89.35	89.50	89.87	89.50
Test	88.11	88.79	88.81	88.89	89.17	88.87

Table 4.11 – *Taux de reconnaissance pour les 16 locuteurs en utilisant PSO-ANFIS*

Iteration **NPop**	**500** **100**	**500** **200**	**1000** **100**	**1000** **200**
Train	94.07	94.13	94.21	94.23
Test	93.97	94.01	94.07	94.05

Table 4.12 – *Comparaison ANFIS, GA-ANFIS et PSO-ANFIS pour les 8 locuteurs*

Rate	**ANFIS**	**AG-ANFIS**	**PSO-ANFIS**
Train	88.17	84.50	89.87
Test	87.99	84.31	89.17

Table 4.13 – *Comparaison ANFIS, GA-ANFIS et PSO-ANFIS pour les 16 locuteurs*

Rate	**ANFIS**	**AG-ANFIS**	**PSO-ANFIS**
Train	93.86	91.78	94.21
Test	93.80	91.20	94.07

La figure 4.6 montre les résultats obtenus par l'application de l'ANFIS sur les 8 locuteurs de la base de données CHAINS avec un taux de reconnaissance de 87.99% ; tandis que la figure 4.7 montre les résultats obtenus par l'application de l'ANFIS sur les 16 locuteurs de la base de données avec un taux de reconnaissance de 93.80

Les expériences réalisées par le modèle AG-ANFIS sur les 8 locuteurs sont résumées dans le tableau 4.8. Les résultats obtenus ont montré que le taux de reconnaissance a 84.31% avec 100 et 150 chromosomes puis une légère diminution de 84.05% et 83.37% avec une population de 200 et 250 chromosomes, ensuite une hausse au tour de 85.90% avec 350 chromosomes.

Les résultats obtenus par l'application du modèle AG-ANFIS sur 16 locuteurs après 1000 itérations en faisant varier la taille de la population montre que la meilleure précision obtenue pour 16 locuteurs était de 91.20% avec 200 chromosomes (Le tableau 4.9).

Le tableau 4.10 présente les résultats obtenus par le modèle PSO-ANFIS sur les 8 locuteurs en faisant varier le nombre de particules soit NPop (taille de la population) NPop = 20 ; 50 ; 100 ; 150 ; 200 ; 250 ; 300 avec 1000 itérations. Les expériences réalisées ont montré que plus on augmente la taille de la population plus la précision augmente également jusqu'à atteindre 200 particules résultant soit un taux de reconnaissance de 89.17% qui est le meilleur résultat obtenu par PSO-ANFIS pour 8 locuteurs. En revanche, 300 particules entraînent un taux de reconnaissance moindre de 88.87%.

Le tableau 4.11 montre les résultats obtenus par l'application de PSO-ANFIS sur 16 locuteurs en faisant varier la taille de la population et le nombre d'itérations. La meilleure précision obtenue pour 16 locuteurs était de 94.07% (avec nombre d'itérations = 1000 et une taille de population = 100).

Le tableau 4.12 et le tableau 4.13 montrent une comparaison entre les trois méthodes d'apprentissage pour les 8 locuteurs et les 16 locuteurs, respectivement pour 8 locuteurs, il y a une nette amélioration sur la précision de 87.99% (ANFIS) à 84.31% (AG-ANFIS) et 89.17% (PSO-ANFIS) ainsi que pour les 16 locuteurs de 93.80% (ANFIS) à 91.20%(AG-ANFIS) et 94.07% (PSO-ANFIS). Le modèle PSO-ANFIS offre une amélioration significative du taux de reconnaissance par rapport au modèle AG-ANFIS et aussi à l'ANFIS classique.

Les modèles proposés AG-ANFIS et PSO-ANFIS présentent une amélioration significative par rapport aux résultats trouvés dans la littérature (voir tableau 4.14).

Table 4.14 – *Comparaison des modèles proposés et des modèles existants*

Base de données	**Approche proposée PSO-ANFIS**	**L'pproche proposée AG-ANFIS**	**(Novakovic, 2011)**	**(Fazaki& al., 2015)**	**(Karlos& al., 2016)**	**(Karlos& al. 2018)**
8 locuteurs	89.17	84.31	MLP 71.77	Self SSL 81.45	SVM 87.44	Entropy 83.94
16 locuteurs	94.07	91.20	MLP 58.08	Self SSL 78.19	SVM 83.70	SMargin 83.10

4.5 Conclusion

Le modèle ANFIS a deux types de paramètres, à savoir les paramètres antécédents et les paramètres conséquents, qui sont ajustés tout au long du processus d'apprentissage. L'ajustement de ces paramètres est basé sur l'algorithme de la rétropropagation du gradient. Cependant, à cause des limites de cet algorithme, le modèle est susceptible de tomber dans de mauvais minima locaux. Pour obtenir de meilleures performances, les paramètres internes de l'ANFIS ont été ajustés à l'aide des algorithmes évolutionnaires.

Pour ce faire, tout d'abord, la structure du réseau est obtenue par l'algorithme de regroupement soustractif qui permet de fixer le nombre de fonctions d'appartenance, ainsi les paramètres antécédents et les paramètres conséquents. Par la suite, ces paramètres ont été optimisés en utilisant les algorithmes génétiques et l'essaim de particules. Le taux de reconnaissance est utilisé comme fonction de fitness dans cette expérience. L'avantage de la méthode est clairement mis en évidence en offrant une nette amélioration de la précision par rapport la méthode d'apprentissage classique.

Nous pouvons dire que les algorithmes évolutionnaires constituent une alternative très intéressante pour l'apprentissage du réseau ANFIS, néanmoins, le choix des paramètres adéquats de l'AG et des PSO est important pour une exécution efficace, car il n'y a pas de règle pour déterminer ces paramètres, faire de nombreux essais permet de se doter de l'expérience nécessaire pour les régler.

Conclusion générale

Les systèmes artificiels intelligents tentent de résoudre des problèmes de manière proche de celle de l'être humain. Ces dernières décennies, un grand nombre de techniques de l'intelligence artificielle ont vu le jour, comme les réseaux de neurones, les algorithmes évolutionnaire etc.. . .

Les systèmes neuro-flous ont prouvé leur efficacité dans une grande variété de problèmes du monde réel et dans l'industrie. Ces systèmes sont créés afin de synthétiser les avantages et de surmonter les inconvénients des réseaux neuronaux et des systèmes flous tel que la capacité d'apprentissage de la première et la lisibilité et la souplesse de la seconde. Plusieurs combinaisons de ces deux méthodes ont été développées, l'un des systèmes neuro-flou dans lesquels un système flou de type Sugeno de premier ordre est implémenté ; est le réseau adaptatif à système inférence flou (ANFIS).

Le déroulement de l'apprentissage du réseau ANFIS passe par deux étapes importantes. En premier, la détermination de la structure appropriée du réseau, c'est-à-dire le meilleur partitionnement de l'espace d'entrée ; ensuite, l'ajustement des paramètres antécédents et conséquents afin qu'une fonction objectif spécifiée soit minimisée. L'apprentissage hybride très utilisé pour ajuster les paramètres internes du réseau ANFIS qui combine l'algorithme de rétropropagation du gradient et la méthode des moindres carrés LSE. Cet algorithme peut rester bloqué aux mauvais minimas locaux. Pour remédier à ce problème, les algorithmes évolutifs peuvent être une solution idéale.

Dans notre thèse, l'algorithme de regroupement soustractif est utilisé pour avoir la structure optimale du réseau ANFIS. Une fois la structure est fixée le nombre des paramètres internes est connu, l'algorithme évolutionnaire essaie de trouver la solution optimale dans cet espace de recherche. Dans les deux modèles proposés AG-ANFIS et PSO-ANFIS, chaque membre (solution) peut être codé comme un vecteur unidimensionnel constitué de variables réelles. La longueur de ce vecteur est basée sur le nombre de paramètres conséquents et de paramètres de prémisse dans le modèle ANFIS. Le processus évolutif commence par générer aléatoirement une population initiale de chaînes réelles, chaque solution candidate représentant la partie prémisse des règles et la partie conséquente. Ensuite, la population entière sera amenée à s'améliorer progressivement jusqu'à que le nombre maximal d'itération est atteint.

L'ensemble de ces hybridations a été appliqué, dans le dernier chapitre de ce mé-

moire, pour la reconnaissance du locuteur. L'avantage de la méthode proposée est clairement mis en évidence en offrant une nette amélioration de la précision par rapport la méthode d'apprentissage classique.

La reconnaissance automatique de la parole reste une tâche difficile, un grand nombre de voies de recherche restent ouvertes pour traiter tous les aspects de RAP. Les résultats que nous avons obtenus sont très intéressants. Cependant, tous les problèmes liés aux approches proposés sont loin d'être résolus. De nombreuses études peuvent être envisagées afin de les améliorer.

— Réduction des entrées en incluant une étape de sélection d'entrée. Des méthodes telles que l'analyse en composantes principales (ACP) peuvent être utilisées pour sélectionner les meilleurs paramètres d'entrée.
— De plus, des algorithmes d'optimisation plus avancés pourraient être utilisés afin d'accélérer le taux de convergence en recherchant les variables internes optimales de l'ANFIS et d'optimiser le processus d'apprentissage.
— Aussi le traitement de benchmarks plus larges contenant plus de locuteurs.

BIBLIOGRAPHIE

[1] C. Touzet, Les Réseaux De Neurones Artificiels, Introduction Au Connexionnisme, 1992.

[2] L. A. Zadeh, « Fuzzy sets », Information and Control, vol. 8, no 3, p. 338-353, Juin 1965.

[3] C.C. Lee, « Fuzzy logic in control systems : fuzzy logic controller. I », IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no 2, p. 404-418, Mar/Apr 1990.

[4] H. Bühler, Réglage par logique floue. Infoscience, 1994.

[5] L. X. Wang, Adaptive Fuzzy Systems and Control : Design and Stability Analysis, 1994.

[6] J. Jantzen, Foundations of Fuzzy Control : A Practical Approach. John Wiley & Sons, 2013.

[7] K. Passino and S. Yurkovich, Fuzzy Control, MA, Reading :Addison-Wesley, 1998.

[8] J.-R. Jang, « ANFIS : adaptive-network-based fuzzy inference system », IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no 3, p. 665-685, Mai 1993.

[9] J. Godjevac, Neuro-fuzzy Controllers : Design and Application. PPUR presses polytechniques, 1997.

[10] S. G. Tzafestas et K. D. Blekas, « Hybrid soft computing systems : a critical survey with engineering applications », Soft Comput. Syst. Control Technol. Singap. Word Sci., p. 119–168, 1999.

[11] D. Nauck et R. Kruse, « What are neuro-fuzzy classifiers », in Proc. Seventh International Fuzzy Systems Association World Congress IFSA, 1997, vol. 97, p. 228–233.

[12] D. Racoceanu, « Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l'Intelligence Artificielle », Habilitation à diriger des

recherches, Université de FRANCHE COMTÉ de Besançon, France, 2006.

[13] J.-S. Wang et C. G. Lee, « Efficient neuro-fuzzy control systems for autonomous underwater vehicle control », in Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), 2001, vol. 3, p. 2986–2991.

[14] W. A. Farag, V. H. Quintana, et G. Lambert-Torres, « A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems », IEEE Trans. Neural Netw., vol. 9, no 5, p. 756–767, 1998.
[15] C.-S. Ouyang, W.-J. Lee, et S.-J. Lee, « A TSK-type neurofuzzy network approach to system modeling problems », IEEE Trans. Syst. Man Cybern. Part B Cybern., vol. 35, no 4, p. 751–767, 2005.

[16] M. Sugeno, « On stability of fuzzy systems expressed by fuzzy rules with singleton consequents », IEEE Transactions on Fuzzy Systems, vol. 7, no 2, p. 201-224, Avr. 1999.

[17] C.-T. Lin, C.-J. Lin, et C. S. G. Lee, « Fuzzy adaptive learning control network with on-line neural learning », Fuzzy Sets and Systems, vol. 71, no 1, p. 25-45, Avr. 1995.

[18] H. R. Berenji et P. Khedkar, « Learning and tuning fuzzy logic controllers through reinforcements », IEEE Transactions on Neural Networks, vol. 3, no 5, p. 724-740, Sept. 1992.

[19] D. Nauck et R. Kruse, « Neuro-fuzzy classification with NEFCLASS », in Operations Research Proceedings 1995, Springer, 1996, p. 294–299.

[20] D. Nauck et R. Kruse, « A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation », in IEEE international conference on neural networks, 1993, p. 1022–1027.

[21] D. Nauck et R. Kruse, « Neuro-fuzzy systems for function approximation », Fuzzy sets and systems, vol. 101, no 2, p. 261-271, 1999.

[22] J.-S. Wang et C. G. Lee, « Self-adaptive neuro-fuzzy inference systems for classification applications », IEEE Transactions on Fuzzy systems, vol. 10, no 6, p. 790-802, 2002.

[23] T. Takagi et M. Sugeno, « Fuzzy identification of systems and its applications to modeling and control », IEEE transactions on systems, man, and cybernetics, no 1, p. 116-132, 1985.

[24] « Fuzzy Logic Toolbox Documentation ». https ://www.mathworks.com/help/fuzzy/.

[25] J.-S. Jang et C.-T. Sun, « Neuro-fuzzy modeling and control », Proceedings of the IEEE, vol. 83, no 3, p. 378-406, 1995.

[26] S. L. Chiu, « A cluster estimation method with extension to fuzzy model identification », in Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, 1994, p. 1240-1245.

[27] H. Li, C. P. Chen, et H.-P. Huang, Fuzzy neural intelligent systems : Mathematical foundation and the applications in engineering. CRC Press, 2018.

[28] J. C. Dunn, « A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters », 1973.

[29] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.

[30] A. K. Jain, M. N. Murty, et P. J. Flynn, « Data clustering : a review », ACM computing surveys (CSUR), vol. 31, no 3, p. 264-323, 1999.

[31] J.-S. R. Jang, C.-T. Sun, et E. Mizutani, « Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence », IEEE Trans. Autom. Control, vol. 42, no 10, p. 1482–1484, 1997.

[32] R. Stuckless, « Developments in real-time speech-to-text communication for people with impaired hearing », Communication access for people with hearing loss, p. 197-226, 1994.

[33] J. B. Allen, « How do humans process and recognize speech ? », in Modern methods of speech processing, Springer, 1995, p. 251-275.

[34] M. Honda, « Human speech production mechanisms », NTT Tech. Rev., vol. 1, no 2, p. 24–29, 2003.

[35] G. Fant, Acoustic theory of speech production. Walter de Gruyter, 1970.

[36] R. Munkong et B.-H. Juang, « Auditory perception and cognition », IEEE Signal Process. Mag., vol. 25, no 3, p. 98–117, 2008.

[37] J. B. Allen, « How Do Humans Process and Recognize Speech ? », IEEE Transactions on Speech and Audio Processing, vol. 2, no 4, p. 567-577, 1994.

[38] H. Beigi, Fundamentals of Speaker Recognition. Springer US, 2011.

[39] J.-P. Haton, C. Cerisara, D. Fohr, Y. Laprie, et K. Smaïli, Reconnaissance Automatique de la Parole Du signal à son interprétation. DUNOD, 2006, p. 392

[40] J. P. Campbell, D. A. Reynolds, et R. B. Dunn, « Fusing high- and lowlevel features for speaker recognition », 2003.

[41] H. Harb, « Classification du signal sonore en vue d'une indexation par le contenu des documents multimédia », These de doctorat, Ecully, Ecole centrale de Lyon, 2003. Disponible sur : http ://www.theses.fr/2003ECDL0039

[42] M. Arcienega et A. Drygajlo, « A Bayesian network approach for combining pitch and reliable spectral envelope features for robust speaker verification », in International Conference on Audio-and Video-Based Biometric Person Authentication, 2003, p. 78-85.

[43] M. Chetouani, A. Hussain, B. Gas, et J.-L. Zarader, « New Sub-band Processing Framework Using Non-linear Predictive Models for Speech Feature Extraction », in Nonlinear Analyses and Algorithms for Speech Processing, Berlin, Heidelberg, 2005, p. 284-290.

[44] H. Hermansky, « Perceptual linear predictive (PLP) analysis of speech », The Journal of the Acoustical Society of America, vol. 87, no 4, p. 1738-1752, Avr. 1990.

[45] X. Huang, A. Acero, et H.-W. Hon, Spoken Language Processing : A Guide to Theory, Algorithm and System Development, 1er édition. Upper Saddle River, NJ : Prentice Hall, 2001.

[46] N. S. Nehe et R. S. Holambe, « DWT and LPC based feature extraction methods for isolated word recognition », EURASIP Journal on Audio, Speech, and Music Processing, vol. 2012, no 1, p. 1-7, 2012.

[47] M. Sahidullah et G. Saha, « Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition », Speech communication, vol. 54, no 4, p. 543-565, 2012.

[48] D. B. Fry, The physics of speech. Cambridge University Press, 1979.

[49] H. P. Combrinck et E. C. Botha, On The Mel-scaled Cepstrum. 1996.

[50] D. J. Thomson, M. F. Robbins, C. G. Maclennan, et L. J.
Lanzerotti, « Spectral and windowing techniques in power spectral analyses of geomagnetic data », Physics of the Earth and Planetary Interiors, vol. 12, no 2, p. 217-231, Août 1976.

[51] K. R. Rao, D. N. Kim, et J. J. Hwang, « Fast Fourier transform : algorithms and applications », 2010.

[52] E. Zwicker, « Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen) », The Journal of the Acoustical Society of America, vol. 33, no 2, p. 248-248, Févr 1961.

[53] A. E. Rosenberg, « Automatic speaker verification : A review », Proc. IEEE, vol. 64, no 4, p. 475–487, 1976.

[54] F. Nolan, The Phonetic Description of Voice Quality. JSTOR, 1982.

[55] J. Kreiman et D. Sidtis, Foundations of Voice Studies : An Interdisciplinary Approach to Voice Production and Perception. Wiley-Blackwell, 2011.

[56] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, et N. Scheffer, « Towards noise-robust speaker recognition using probabilistic linear discriminant analysis », in 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2012, p. 4253–4256.

[57] D. Reynolds et al., « The SuperSID project : Exploiting high-level information for high-accuracy speaker recognition », in 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., 2003, vol. 4, p. IV–784.

[58] D. A. Reynolds et W. M. Campbell, « Text-Independent Speaker Recognition », Springer Handbook of Speech Processing, p. 763-782, 2008.

[59] G. R. Doddington, « Speaker recognition—Identifying people by their voices », Proc. IEEE, vol. 73, no 11, p. 1651–1664, 1985.

[60] D. A. Reynolds, « Automatic speaker recognition using Gaussian mixture speaker models », 1995.

[61] B. L. Tseng, F. K. Soong, et A. E. Rosenberg, « Continuous probabilistic acoustic map for speaker identification », in [Proceedings] ICASSP-92 : 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, Mars 1992, vol. 2, p. 161-164.

[62] V. Srihari, R. Karthik, R. Anitha, et S. D. Suganthi, « Speaker verification using combinational features and adaptive neuro-fuzzy inference systems », in Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia, 2010, p. 98-103.

[63] N. Helmi et B. H. Helmi, « Speech recognition with fuzzy neural network for discrete words », in 2008 fourth international conference on natural computation, 2008, vol. 7, p. 265-269.

[64] Kamaruddin, N. and Wahab, A. «Speech Emotion Verification System (SEVS) Based On MFCC For Real Time Applications». In : Proceeding of the 4th International Conference on Intelligent Environments (IE '08). p. 1-7. 21-22 Jul 2008. Seattle, Washington, USA.

[65] R. Sabah et R. N. Ainon, « Isolated digit speech recognition in Malay language using neuro-fuzzy approach », in 2009 third Asia international conference on Modelling & Simulation, 2009, p. 336-340.

[66] J. Thevaril et H. K. Kwan, « Speech enhancement using adaptive neuro-fuzzy

filtering », in 2005 International Symposium on Intelligent Signal Processing and Communication Systems, 2005, p. 753-756.

[67] B. Pandey, A. Ranjan, R. Kumar, et A. Shukla, « Multilingual speaker recognition using ANFIS », in 2010 2nd International Conference on Signal Processing Systems, 2010, vol. 3, p. V3-714.

[68] P. B. Patel et T. Marwala, « Adaptive neuro fuzzy inference system, neural network and support vector machine for caller behavior classification », in 2011 10th International Conference on Machine Learning and Applications and Workshops, 2011, vol. 1, p. 298-303.

[69] S. Lakra, J. Singh, et A. K. Singh, « Automated pitch-based gender recognition using an adaptive neuro-fuzzy inference system », in 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, p. 82-86.

[70] M. El-Wakdy et al., « Speech Recognition Using a Wavelet Transform to Establish Fuzzy Inference System Through Subtractive Clustering and Neural Network(ANFIS) », in WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering, 2008, no 12.

[71] S. Silarbi, B. Abderrahmane, et A. Benyettou, « Adaptive Network Based Fuzzy Inference System for Speech Recognition Through Subtractive Clustering », International Journal of Artificial Intelligence & Applications, vol. 5, no 6, p. 43, 2014.

[72] H. Tolba, « A high-performance text-independent speaker identification of Arabic speakers using a CHMM-based approach », Alexandria Engineering Journal, vol. 50, no 1, p. 43-47, 2011.

[73] K. Daqrouq et T. A. Tutunji, « Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers », Applied Soft Computing, vol. 27, p. 231-239, 2015.

[74] J.-D. Wu et Y.-J. Tsai, « Speaker identification system using empirical mode decomposition and an artificial neural network », Expert Systems with Applications, vol. 38, no 5, p. 6112-6117, 2011.

[75] Y. Chen, L. Wang, H. Lin, et J. Li, « Design of speaker recognition system based on artificial neural network », in 6th International Symposium on Advanced Optical Manufacturing and Testing Technologies : Optical System Technologies for Manufacturing and Testing, 2012, vol. 8420, p. 84200U.

[76] Z. Ge, A. N. Iyer, S. Cheluvaraja, R. Sundaram, et A. Ganapathiraju, « Neural network based speaker classification and verification systems with enhanced features », in 2017 Intelligent Systems Conference (IntelliSys), 2017, p. 1089-1094.

[77] S. Nandyal, S. S. Wali, et S. M. Hatture, « MFCC based text-dependent speaker identification using BPNN », International Journal of Signal Processing Systems, vol. 3, no 1, p. 30-34, 2015.

[78] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, et P. A. Torres-Carrasquillo, « Support vector machines for speaker and language recognition », Computer Speech & Language, vol. 20, no 2-3, p. 210-229, 2006.

[79] R. Chakroun, L. B. Zouari, M. Frikha, et A. B. Hamida, « A hybrid system based on GMM-SVM for speaker identification », in 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015, p. 654-658.

[80] W. M. Campbell, D. E. Sturim, et D. A. Reynolds, « Support vector machines using GMM supervectors for speaker verification », IEEE signal processing letters, vol. 13, no 5, p. 308-311, 2006.

[81] R. Dhineshkumar, A. B. Ganesh, et S. Sasikala, « Speaker identification system using gaussian mixture model and support vector machines (GMM-SVM) under noisy conditions », Indian Journal of Science and Technology, vol. 9, no 19, p. 1-6, 2016.

[82] M. Algabri, H. Mathkour, M. A. Bencherif, M. Alsulaiman, et M. A. Mekhtiche, « Automatic speaker recognition for mobile forensic applications », Mobile Information Systems, vol. 2017, 2017.

[83] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, et K. Yu, « Deep feature for text-dependent speaker verification », Speech Communication, vol. 73, p. 1-13, 2015.

[84] V. R. Apsingekar et P. L. De Leon, « Support vector machine based spea-

ker identification systems using GMM parameters », in 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, 2009, p. 1766-1769.

[85] Y. Hu, D. Wu, et A. Nucci, « Fuzzy-clustering-based decision tree approach for large population speaker identification », IEEE transactions on audio, speech, and language processing, vol. 21, no 4, p. 762-774, 2012.

[86] U. Shrawankar et V. M. Thakare, « Techniques for feature extraction in speech recognition system : A comparative study », arXiv preprint arXiv :1305.1145, 2013.

[87] S. Shah et A. Bhise, « Fast Speaker Recognition using Efficient Feature Extraction Technique 1 », 2013.

[88] D. E. Golberg, « Genetic algorithms in search, optimization, and machine learning », Addion Wesley, vol. 1989, no 102, p. 36, 1989.

[89] J. R. Koza et J. R. Koza, Genetic programming : on the programming of computers by means of natural selection, vol. 1. MIT press, 1992.

[90] R. Storn et K. Price, « Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces », Journal of global optimization, vol. 11, no 4, p. 341-359, 1997.

[91] E. Mezura-Montes et C. A. C. Coello, « A simple multimembered evolution strategy to solve constrained optimization problems », IEEE Transactions on Evolutionary Computation, vol. 9, no 1, p. 1-17, Févr. 2005.

[92] L. J. Fogel, A. J. Owens, et M. J. Walsh, « Artificial intelligence through simulated evolution », 1966.

[93] F. Hidayat, « Materi Kuliah Kedelapan - Charles Darwin », 2007.

[94] D. B. Fogel, Evolutionary algorithms in theory and practice. John Wiley & Sons, Inc. New York, 1997.

[95] C. Darwin, On the origin of species, 1859. Routledge, 2016.

[96] R. A. Fisher, The genetical theory of natural selection. , 1958.

[97] J. R. Sampson, Adaptation in natural and artificial systems (John H. Holland). Society for Industrial and Applied Mathematics, 1976.

[98] X. Yu et M. Gen, Introduction to evolutionary algorithms. Springer Science & Business Media, 2010.

[99] C. Blum et al., « Evolutionary optimization », in Variants of evolutionary algorithms for real-world applications, Springer, 2012, p. 1-29.

[100] Z. Michalewicz et D. B. Fogel, How to Solve It : Modern Heuristics. Springer Science & Business Media, 2013.

[101] A. Tettamanzi et M. Tomassini, Soft Computing : Integrating Evolutionary, Neural, and Fuzzy Systems. Springer Science & Business Media, 2013.

[102] F. Karray, F. O. Karray, et C. W. De Silva, Soft computing and intelligent systems design : theory, tools, and applications. Pearson Education, 2004.

[103] H. Li, L. Zhang, et Y. Jiao, « Discrete differential evolution algorithm for integer linear bilevel programming problems », Journal of Systems Engineering and Electronics, vol. 27, no 4, p. 912-919, 2016.

[104] A. Slowik, « Application of an adaptive differential evolution algorithm with multiple trial vectors to artificial neural network training », IEEE Transactions on Industrial Electronics, vol. 58, no 8, p. 3160-3167, 2010.

[105] J. H. Holland, Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[106] D. E. Goldberg, « Genetic and evolutionary algorithms come of age », Commun. ACM, vol. 37, no 3, p. 113-119, mars 1994.

[107] L. Davis, « Handbook of genetic algorithms », 1991.

[108] L. J. Eshelman et J. D. Schaffer, « Real-coded genetic algorithms and interval-schemata », in Foundations of genetic algorithms, vol. 2, Elsevier, 1993, p. 187-202.

[109] D. E. Goldberg, Real-coded genetic algorithms, virtual alphabets and blocking. Citeseer, 1990.

[110] D. E. Goldberg, « The theory of virtual alphabets », in International Conference on Parallel Problem Solving from Nature, 1990, p. 13-22.

[111] M. Gen et R. Cheng, Genetic algorithms and engineering optimization, vol. 7. John Wiley & Sons, 1999.

[112] M. Gen, R. Cheng, et L. Lin, Network models and optimization : Multiobjective genetic algorithm approach. Springer Science & Business Media, 2008.

[113] J. Genlin, « Survey on genetic algorithm [J] », Computer Applications and Software, vol. 2, p. 69-73, 2004.

[114] E. Cantú-Paz, « A survey of parallel genetic algorithms », Calculateurs paralleles, reseaux et systems repartis, vol. 10, no 2, p. 141-171, 1998.

[115] D. E. Goldberg et K. Deb, « A comparative analysis of selection schemes used in genetic algorithms », in Foundations of genetic algorithms, vol. 1, Elsevier, 1991, p. 69-93.

[116] D. Goldberg, « A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing | Semantic Scholar », Complex Systems, p. 445-460, 1990.

[117] B. L. Miller et D. E. Goldberg, « Genetic algorithms, tournament selection, and the effects of noise », Complex systems, vol. 9, no 3, p. 193-212, 1995.

[118] R. Kumar, « Blending roulette wheel selection & rank selection in genetic algorithms », Int. J. Mach. Learn. Comput., vol. 2, no 4, p. 365-370, 2012.

[119] G. Syswerda, « A study of reproduction in generational and steady-state gene-

tic algorithms », in Foundations of genetic algorithms, vol. 1, Elsevier, 1991, p. 94-101.

[120] T. Blickle et L. Thiele, « A comparison of selection schemes used in evolutionary algorithms », Evolutionary Computation, vol. 4, no 4, p. 361-394, 1996.

[121] R. Collins et D. Jefferson, « Selection in Massively Parallel Genetic Algorithms », undefined, 1991.

[122] H. Ishibuchi et T. Yamamoto, « Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining », Fuzzy Sets and Systems, vol. 141, no 1, p. 59-88, janv. 2004.

[123] M. Hutter, « Fitness uniform selection to preserve genetic diversity », in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), 2002, vol. 1, p. 783-788.

[124] J. J. Grefenstette et J. E. Baker, How genetic algorithms work : a critical look at implicit parallelism. The proceedings of the third. international conference on genetic algorithms. Morgan Kaufmann Publishers, San Mateo, California, 1989.

[125] G. Syswerda, « Uniform crossover in genetic algorithms », in Proceedings of the 3rd international conference on genetic algorithms, 1989, p. 2-9.

[126] M. Srinivas et L. M. Patnaik, « Genetic algorithms : A survey », computer, vol. 27, no 6, p. 17-26, 1994.

[127] R. Poli et W. B. Langdon, « Genetic Programming with One-Point Crossover », in Soft Computing in Engineering Design and Manufacturing, London, 1998, p. 180-189.

[128] C.-J. Wu et G.-Y. Lin, « Design of fuzzy logic controllers using genetic algorithms », in IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028), 1999, vol. 6, p. 104-109.

[129] E. Semenkin et M. Semenkina, « Self-configuring genetic algorithm with modified uniform crossover operator », in International Conference in Swarm Intelligence, 2012, p. 414-421.

[130] X.-B. Hu et E. Di Paolo, « An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem », in Multi-objective memetic algorithms, Springer, 2009, p. 71-89.

131] S. Tsutsui, M. Yamamura, et T. Higuchi, « Multi-parent recombination with simplex crossover in real coded genetic algorithms », in Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1, 1999, p. 657-664.

[132] T. Bäck, D. B. Fogel, et Z. Michalewicz, « Introduction to evolutionary algorithms », Evol. Comput., vol. 1, p. 59-63, 2000.

[133] I. M. Oliver, Dj. Smith, et J. R. Holland, « Study of permutation crossover operators on the traveling salesman problem », 1987. [134] L. Davis, « Applying adaptive algorithms to epistatic domains. », in IJCAI, 1985, vol. 85, p. 162-164.

[135] J. Grefenstette, R. Gopal, B. Rosmaita, et D. Van Gucht, « Genetic algorithms for the traveling salesman problem », in Proceedings of the first International Conference on Genetic Algorithms and their Applications, 1985, vol. 160, no 168, p. 160-168.

[136] S. J. Louis et G. J. Rawlins, « Designer Genetic Algorithms : Genetic Algorithms in Structure Design. », in ICGA, 1991, p. 53-60.

[137] L. J. Eshelman, R. A. Caruana, et J. D. Schaffer, « Biases in the crossover landscape », in Proceedings of the third international conference on Genetic algorithms, San Francisco, CA, USA, Déc. 1989, p. 10-19.

[138] Z. Michalewicz, Genetic algorithms+ data structures= evolution programs. Springer Science & Business Media, 2013.

[139] R. Hinterding, « Gaussian mutation and self-adaption for numeric genetic algorithms », in Proceedings of 1995 IEEE International Conference on Evolutionary Computation, nov. 1995, vol. 1, p. 384-.

[140] K. Deep et M. Thakur, « A new mutation operator for real coded genetic algorithms », Applied Mathematics and Computation, vol. 193, no 1, p. 211-230, Oct. 2007.

[141] M. Srinivas et L. M. Patnaik, « Adaptive probabilities of crossover and mutation

in genetic algorithms », IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no 4, p. 656-667, 1994.

[142] A. Neubauer, « A theoretical analysis of the non-uniform mutation operator for the modified genetic algorithm », in Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), 1997, p. 93-96.

[143] S. Tsutsui et Y. Fujimoto, « Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA). », in ICGA, 1993, p. 206-215.

[144] G. D. Oosthuizen, « SUPERGRAN : a connectionist approach to learning, integrating genetic algorithms and graph induction », Genetic algorithms and their applications : proceedings of the second International Conference on Genetic Algorithms : July 28-31, Cambridge, MA, 1987.

[145] M. L. Mauldin, « 1984 - Maintaining Diversity in Genetic Search », In : Proceedings of the National Conference on Artificial Intelligence ; 1984, p. 247–250

[146] Carol A.Ankenbrandt, « An Extension To the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms », vol. 1, p. 53-68, 1991.

[147] J. Kennedy et R. Eberhart, « Particle swarm optimization », in Proceedings of ICNN'95-international conference on neural networks, 1995, vol. 4, p. 1942-1948.

[148] R. Poli et D. Broomhead, « Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation », in Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007, p. 134-141.

[149] M. Caudill, « Evolutionary neural networks », AI Expert, vol. 6, no 3, p. 28-33, mars 1991.

[150] R. Dawkins et N. Davis, The selfish gene. Macat Library, 2017.

[151] J. Kennedy, « The particle swarm : social adaptation of knowledge », in Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97),

1997, p. 303-308.

[152] Y. Shi et R. Eberhart, « A modified particle swarm optimizer », in 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), 1998, p. 69-73.

[153] J. Kennedy et R. Mendes, « Population structure and particle swarm performance », in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), 2002, vol. 2, p. 1671-1676.

[154] D. N. Wilke, S. Kok, et A. A. Groenwold, « Comparison of linear and classical velocity update rules in particle swarm optimization : Notes on scale and frame invariance », International journal for numerical methods in engineering, vol. 70, no 8, p. 985-1008, 2007.

[155] V. Zue, S. Seneff, et J. Glass, « Speech database development at MIT : TIMIT and beyond », Speech communication, vol. 9, no 4, p. 351-356, 1990.

[156] F. Cummins, M. Grimaldi, T. Leonard, et J. Simko, « The chains corpus : Characterizing individual speakers », in Proc of SPECOM, 2006, vol. 6, p. 431-435.