

Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance automatique de la parole

Anne Spalanzani

THÈSE

présentée pour obtenir le titre de Docteur
de l'Université Joseph Fourier - Grenoble I
(Spécialité Informatique Systèmes et Communication)

Date de soutenance : 28 Octobre 1999

Membres du jury :

Jacques DEMONGEOT

Président

Régine ANDRÉ-OBRECHT

Rapporteur

Jean CAELEN

Directeur de thèse

Jean-Paul HATON

Examineur

Harouna KABRÉ

Co-directeur de thèse

Michèle SEBAG

Rapporteur

Thèse préparée au sein du laboratoire de Communication Langagière et Interaction Personne
Système (CLIPS-IMAG)

Remerciements

Je voudrais tout d'abord remercier *Jean Caelen* pour m'avoir accueillie au sein de son équipe, ainsi que pour les remarques constructives qu'il a su me donner lors de la rédaction de ce manuscrit. J'exprime toute ma reconnaissance à *Harouna Kabré* qui a été à la source de ce travail et m'a suivie tout au long de cette thèse.

Je remercie l'ensemble du jury pour leurs remarques constructives ainsi que pour leur enthousiasme. *Michèle Sebag* m'a fait l'honneur de rapporter sur ce travail. Je lui suis reconnaissante des discussions que nous avons pu avoir et des remarques très riches en enseignement qu'elle a faites sur ma thèse. Je tiens également à remercier *Régine André-Obrecht* d'avoir accepté d'être rapporteur et d'avoir ainsi pu me faire part de critiques pertinentes. Je remercie *Jean-Paul Haton* et *Jacques Demongeot* d'avoir pu m'accorder une partie de leur temps précieux en ayant accepté de faire partie de mon jury.

Je ne pourrai jamais remercier suffisamment *Sid-ahmed Sélouani* pour son soutien moral et scientifique efficace et constant, ainsi que pour les nombreuses heures passées à discuter de nos projets. J'ai pu également compter sans limite sur le soutien moral et technique de *Mohammad Akbar* et *Dominique Vaufreydaz* qui ont su me rendre cette dernière année plus légère. Je n'oublie pas non plus le temps que *François Bérard* a su me consacrer, ainsi que les connaissances qu'il m'a apportées.

Je remercie *Solange Hollard* et *Eric Castelli* pour avoir relu des parties de ce travail, ainsi qu'un grand nombre de chercheurs de GEOD et du laboratoire CLIPS pour s'être inquiétés régulièrement de l'état d'avancement de mes travaux.

Je remercie *mes parents* pour avoir cru en moi et sans qui je n'aurais pas eu la possibilité d'effectuer cette thèse. Je tiens enfin à faire savoir à ma soeur et mon frère, ainsi qu'à tous mes amis et tout spécialement à Stéphane, combien leur présence a été précieuse durant cette longue aventure. Je les remercie pour la manière dont ils échappent à la routine du quotidien, pour leur esprit d'ouverture et de tolérance. Grâce à eux, *mon futur ne manque pas d'avenir...*

Table des matières

Table des matières	i
Liste des Figures	vii
Liste des tables	xi
Introduction	1
 Chapitre I : L'adaptation : difficultés et solutions ...	7
 I.1 Sources de variabilité du signal de parole	10
I.2 Solutions proposées	13
I.2.1 Traitement acoustique du signal de parole	15
I.2.1.1 Les différents types de bruit	15
I.2.1.2 Amélioration du Rapport Signal à Bruit	17
I.2.1.3 Analyses robustes	19
I.2.2 Traitement du modèle	22
I.2.2.1 Compensation	22
I.2.2.2 Modèles hybrides	23
I.2.3 Multimodalité	24

I.3 Discussion	24
 Chapitre II : Algorithmes évolutionnaires : principes et méthodes	 27
II.1 Algorithmes évolutionnaires	29
II.2 Principes généraux	31
II.3 Algorithmes génétiques	33
II.3.1 Principes généraux	33
II.3.2 Représentation	34
II.3.3 Evolution	34
II.3.3.1 Sélection	34
II.3.3.2 Recombinaison	35
II.3.3.3 Mutation	36
II.4 Stratégies d'évolution	36
II.4.1 Principes généraux	36
II.4.2 Représentation	36
II.4.3 Evolution	37
II.4.3.1 Sélection	37
II.4.3.2 Recombinaison	37
II.4.3.3 Mutation	37
II.5 Nos choix	38
II.5.1 Algorithmes Génétiques	39
II.5.2 Stratégies d'évolution	39
 Chapitre III : EVERA : un outil de simulation pour l'étude de la robustesse des SRAPs	 41
III.1 Principes	43
III.1.1 Généralités	43
III.1.2 Systèmes de Reconnaissance	44

III.2 Plateformes existantes	45
III.3 Architecture du système EVERA	46
III.3.1 Module environnement	48
III.3.2 Module Individu	49
III.3.3 Module évolution	51
III.3.4 Implémentation	53
III.4 Réalisation des corpus	59
III.4.1 Les signaux de parole	59
III.4.2 Les données numérisées.....	59
 Chapitre IV : Adaptation du système : modèle	61
 IV.1 Principes des NN	63
IV.1.1 Rétro-propagation du gradient.....	63
IV.1.2 Initialisation des poids	65
IV.1.2.1 Temps d'apprentissage	65
IV.1.2.2 Qualité du réseau résultant	66
IV.2 Représentations pour les EA	66
IV.2.1 Représentation des poids	67
IV.2.2 Représentation de la structure.....	67
IV.2.3 Architecture du réseau	68
IV.3 Hybridation	68
IV.3.1 Valeur des poids	69
IV.3.2 Topologie du réseau.....	69
IV.4 Méthodes associées aux NN	70
IV.4.1 Apprentissage	70
IV.4.2 Reconnaissance.....	71
IV.4.3 Performance.....	71
IV.4.4 Croisement.....	71
IV.4.5 Mutation.....	72
IV.5 Adaptation aux changements d'environnement	73

IV.6 Confrontation entre le Darwinisme et le Lamarckisme	73
IV.7 Présentation du Modèle	75
 Chapitre V : Adaptation du système : expérimentations	 77
V.1 Plan d'expériences	79
V.2 Détermination de l'architecture des réseaux	81
V.3 Influence des paramètres	83
V.3.1 Influence du type de mutation.....	85
V.3.2 Influence du taux de mutation.....	87
V.3.3 Influence du taux de croisement	88
V.3.4 Influence de la taille de la population	89
V.4 Influence de l'héritage	90
V.4.1 Qualité des performances.....	91
V.4.2 Efficacité de l'évolution	93
V.5 Influence de la stratégie	94
V.6 Différents types d'apprentissage	95
V.6.1 Détermination de l'architecture des réseaux.....	96
V.6.2 Apprentissage local	98
V.6.3 Apprentissage incrémental	99
V.6.4 Apprentissage global.....	100
V.6.5 Population de Réseaux	101
V.7 Influence des environnements	102
V.8 Evolution dans le temps	105
V.9 Application aux digits	108
V.10 Discussion des résultats	109

Chapitre VI : Adaptation des données	113
VI.1 ACP et Reconnaissance de la parole	116
VI.2 Analyse en composantes principales	116
VI.2.1 Principe	116
VI.2.2 Méthode	117
VI.3 Application aux données acoustiques	118
VI.4 Architecture	119
VI.4.1 Représentation	120
VI.4.2 Projection	121
VI.4.3 Performance	121
VI.5 Présentation du modèle	121
VI.6 Préparation des données	123
VI.6.1 Données d'apprentissage	123
VI.6.2 Données de test	123
VI.7 Evolution du système	124
VI.7.1 Algorithme d'évolution	124
VI.7.2 Fonction d'évaluation	125
VI.8 Expérimentations	125
VI.8.1 Plan d'expériences	125
VI.8.2 Influence de la projection par ACP	126
VI.8.3 Influence des paramètres	127
VI.8.4 Algorithmes génétiques	129
VI.8.4.1 Influence de la taille de la population	129
VI.8.4.2 Influence du taux de mutation	130
VI.8.4.3 Influence du taux de croisement	131
VI.8.5 Stratégies d'évolution	132
VI.8.6 Influence de l'environnement	133
VI.8.7 Changements d'environnement	134
VI.9 Discussion des résultats	135

Conclusion et Perspectives	137
Références	145
Annexe	161

Liste des Figures

Figure 1 Organisation du document.	5
Figure 2 Représentation schématique de différentes sources de variabilité.	11
Figure 3 Méthodes pour l'amélioration de la robustesse des SRAPs.	14
Figure 4 Méthodes pour l'amélioration du RSB.	19
Figure 5 Méthodes d'analyse robuste.	21
Figure 6 Méthodes de compensation.	22
Figure 7 Méthodes hybrides.	24
Figure 8 Exemple de convergence locale ou globale d'un réseau de neurones.	32
Figure 9 Différentes branches des algorithmes évolutionnaires.	33
Figure 10 Modèle de la roulette.	35
Figure 11 Opération de recombinaison.	35
Figure 12 Opération de mutation.	36
Figure 13 Représentation schématique des modules de EVERA.	47
Figure 14 Modèle de propagation d'une source sonore.	49
Figure 15 Algorithme général de l'évolution.	51
Figure 16 Représentation schématique des méthodes évolutionnaires implémentées.	52

Figure 17 Classe Environnement.....	53
Figure 18 Classe Individu.....	54
Figure 19 Classe Réseau de neurones.....	56
Figure 20 Classe Projection.....	56
Figure 21 Classe Evolution.....	58
Figure 22 Extrait d'un réseau multicouches.....	64
Figure 23 Architecture d'un perceptron multicouches.....	68
Figure 24 Représentation génotypique du réseau.....	68
Figure 25 Illustration du problème de permutation.....	72
Figure 26 Modèle d'évolution de population de SRAPs.....	75
Figure 27 Algorithme d'évolution des SRAPs.....	76
Figure 28 Individu spécialisé dans la reconnaissance de voyelles.....	81
Figure 29 Influence de l'architecture sur les performances du système pour 200 voyelles.....	82
Figure 30 Evolution du taux de reconnaissance lors de l'apprentissage de 200 voyelles.....	83
Figure 31 Influence du type de mutation.....	86
Figure 32 Influence du taux de mutation pour la mutation multiplicative.....	88
Figure 33 Influence du taux de croisement.....	89
Figure 34 Influence de la taille de la population.....	90
Figure 35 Evolution Darwinienne et taux de reconnaissance.....	92
Figure 36 Evolution Lamarckienne et taux de reconnaissance.....	92
Figure 37 Evolution Darwinienne et nombre d'itérations.....	93
Figure 38 Evolution Lamarckienne et nombre d'itérations.....	94
Figure 39 Comportement d'une population évoluant selon une (5, 20)-ES.....	95
Figure 40 Influence de l'architecture sur les performances du système sur 2000 voyelles.....	97
Figure 41 Evolution du taux de reconnaissance pour l'apprentissage de 10 environnements.....	98
Figure 42 Comparaison des performances de la population GA et d'un NN classique.....	99
Figure 43 Comparaison des performances de la population GA et d'un NN incrémental.....	100

Figure 44 Comparaison des performances de la population GA et d'un NN global.	101
Figure 45 Comparaison des performances des populations GA et NN sans évolution.	102
Figure 46 Influence de l'ordre de présentation des environnements sur les performances maximum.	104
Figure 47 Influence de l'ordre de présentation des environnements sur les performances moyennes.	104
Figure 48 Evolution dans le temps des performances maximum des populations GA et NN.....	106
Figure 49 Evolution dans le temps des performances moyennes des populations GA et NN.....	107
Figure 50 Evolution des performances sur les Digits.....	109
Figure 51 Principe de l'adaptation du système de reconnaissance.....	115
Figure 52 Principe de l'adaptation du système de perception.	115
Figure 53 Direction (u) de projection d'un nuage de points.....	117
Figure 54 Modèle de projection des données bruitées et non bruitées.	119
Figure 55 Représentation d'une base de projection définie par ACP et celle altérée par EA.	121
Figure 56 Modèle d'évolution d'une population de bases de projection.....	122
Figure 57 Algorithme ACP/EA.	124
Figure 58 Influence de la projection dans le sous espace défini par l'ACP.	127
Figure 59 Influence de la taille de la population.	130
Figure 60 Influence du taux de mutation.....	131
Figure 61 Influence du taux de croisement.....	131
Figure 62 Influence du nombre de parents.	132
Figure 63 Influence du type de bruit.....	133
Figure 64 Performances des (25-100)-ES lors de changements d'environnements.	134
Figure 65 Modèle d'hybridation EA/NN pour l'adaptation du système.	142
Figure 66 Modèle d'hybridation EA/ACP pour l'adaptation des données.....	143
Figure 67 Modèle de projection multi-systèmes pour l'adaptation des données.....	144
Figure 68 Evolution de l'erreur : fonction F1.....	164
Figure 69 Evolution de l'erreur : fonction F2.....	165
Figure 70 Evolution de l'erreur : fonction de Shubert.....	166

Figure 71 Evolution de l'erreur lors d'un changement d'environnement.	167
Figure 72 Evolution de l'erreur lors de plusieurs changements d'environnement.	168

Liste des tables

Table 1	Plan d'expériences global pour l'adaptation des NN.	81
Table 2	Plan d'expériences sur l'influence des paramètres de l'évolution.	85
Table 3	Récapitulatif des performances selon le type de mutation.	86
Table 4	Récapitulatif des performances selon la taille de la population.	90
Table 5	Récapitulatif des performances selon le type d'héritage choisi.	92
Table 6	Récapitulatif du temps d'apprentissage selon le type d'héritage.	94
Table 7	Plan d'expériences pour la comparaison de différents types d'apprentissage.	96
Table 8	Récapitulatif des performances selon le type d'apprentissage.	102
Table 9	Plan d'expériences global.	126
Table 10	Plan d'expériences.	129
Table 11	Amélioration des performances par ES.	134
Table 12	Méthodes d'évolution utilisées.	163

Introduction

Les systèmes de reconnaissance automatique de la parole (SRAPs) sont aujourd'hui bien connus dans le monde de l'informatique et suscitent l'intérêt d'un public de plus en plus large (RéfWeb 1). De ce fait, ces systèmes doivent être opérationnels en situation réelle, dans un contexte de fortes variations de conditions acoustiques. L'environnement, l'utilisateur, les microphones utilisés, etc. sont autant de sources de variabilité qui peuvent faire chuter les performances de ces systèmes. Afin de garder des taux de reconnaissance acceptables, il est alors nécessaire d'imposer des contraintes sur leurs conditions d'utilisation. Le choix du microphone, le débit de parole ou la taille du vocabulaire font partie des paramètres à gérer correctement pour une bonne robustesse des SRAPs. Ainsi, un système de reconnaissance de la parole est robuste s'il est capable de garder un bon taux de reconnaissance même si la qualité du signal est dégradée, ou si les caractéristiques acoustiques du signal sont différentes entre la phase d'apprentissage du système et celle d'utilisation.

Cette notion de robustesse est l'un des problèmes majeurs qui limitent l'*utilisabilité* de ces systèmes. Comment rendre les systèmes existants adaptables aux conditions acoustiques et aux caractéristiques changeantes d'élocution en situation réelle, c'est-à-dire hors laboratoire ?

D'une manière générale, la stratégie prédominante adoptée par les chercheurs pour résoudre ces problèmes consiste à enregistrer des corpus d'apprentissage dans des

conditions qui soient les plus proches de celles dans lesquelles les différents systèmes sont utilisés. Cela nécessite de définir des normes et des critères de comparaison des systèmes de reconnaissance, l'objectif étant d'estimer et de garantir leur bon fonctionnement lors de leur utilisation. Malheureusement les sources de variabilité qui peuvent affecter les performances d'un système de reconnaissance sont nombreuses et non prévisibles (changement de locaux, de bruit ambiant, etc.), et ces systèmes, utilisés alors comme des "boîtes noires" statiques (Gong 95), s'adaptent mal aux variations acoustiques.

Toutefois, des travaux mettant l'accent sur des méthodologies nouvelles, visant une meilleure utilisation des données existantes, reçoivent une plus grande attention de la communauté scientifique (Haton 97). L'adaptation aux conditions acoustiques a ainsi fait l'objet de nombreux travaux qui ont débouché sur des techniques de filtrage et d'adaptation au bruit ambiant qui améliorent sensiblement les performances des systèmes de reconnaissance (Junqua et Haton 96). Quant aux problèmes d'adaptation aux locuteurs, ils sont de plus en plus considérés comme cruciaux (Bradford 95) pour les nouvelles technologies vocales. Une multitude d'algorithmes ont ainsi été élaborés afin d'améliorer la robustesse. Nous en faisons une présentation dans le chapitre *L'adaptation : difficultés et solutions*.

Nous sommes conscients de la fragilité de la reconnaissance de la parole. La communication entre un locuteur et son interlocuteur est un mécanisme complexe. Elle implique des échanges de signaux acoustiques dans un certain contexte (l'environnement acoustique) mais aussi des échanges de signes de différents niveaux de signification. Reconnaître signifie chez l'humain, identifier des éléments déjà appris mais aussi posséder des mécanismes qui permettent de généraliser à des éléments nouveaux les processus de reconnaissance acquis pour des éléments anciens. C'est aussi reconnaître qu'il y a des éléments omis ou des éléments redondants, s'adapter à l'environnement, au locuteur, tenir compte de l'évolution de la situation, etc. Pour l'auditeur humain, la reconnaissance n'est en fait que l'un des très nombreux processus qui l'amènent à réagir de manière adéquate à un acte de parole de son interlocuteur. Dans des conditions difficiles, l'auditeur humain utilise encore plus les contraintes offertes par le code linguistique pour réduire les hypothèses et les ambiguïtés, mais utilise aussi les informations provenant de la situation : événements

concomitants, événements redondants, etc. Généralement l'auditeur sait dans quel environnement il est plongé et en connaît les limitations ou les propriétés. Il sait aussi la tâche qu'on lui donne ou qu'il se donne : il est motivé par un but qu'il cherche à atteindre. Il est capable de prendre en compte les contextes sémantique, pragmatique, lexical, etc. (Caelen et al. 96) et d'adapter sa perception au contexte acoustique.

La nature a produit des systèmes extrêmement complexes à l'aide, parfois, de principes très simples. Pourquoi ne pas imaginer que des algorithmes basés sur les capacités d'adaptation et d'évolution des êtres vivants soient un jour capables de gérer l'ensemble des paramètres évoluant au cours d'un dialogue afin de s'approcher des capacités humaines ?

Dans le cadre de cette thèse, nous nous sommes intéressée à un type d'algorithme, inspiré de l'évolution naturelle formulée par Darwin, qui propose de faire évoluer des populations d'individus dans un environnement en favorisant la survie et la reproduction des individus les mieux adaptés à cet environnement (les principes de ces méthodes sont présentés dans le chapitre *Algorithmes évolutionnaires : principes et méthodes*). On connaît l'efficacité de ce type d'approche pour l'optimisation de fonctions complexes, pour la recherche de solutions à des problèmes ayant un nombre de variables très grand. Nous cherchons à savoir si ce type d'algorithmes appelés *algorithmes évolutionnaires* peut constituer une nouvelle approche pour l'adaptation des SRAPs aux changements d'environnement.

Afin d'étudier la possibilité d'incorporer des algorithmes évolutionnaires dans le domaine de la robustesse des SRAPs, nous proposons, dans le cadre de cette thèse, de nous limiter au niveau acoustique du traitement de la parole et plus particulièrement au niveau de l'adaptation acoustique des SRAPs. Souhaitant aborder le problème de la robustesse d'un point de vue global, nous ne faisons aucune supposition sur le type de bruit traité (par exemple téléphone, voiture, etc.) et les méthodes proposées ne dépendent d'aucun domaine spécifique.

Dans ce contexte, notre travail s'est situé à deux niveaux :

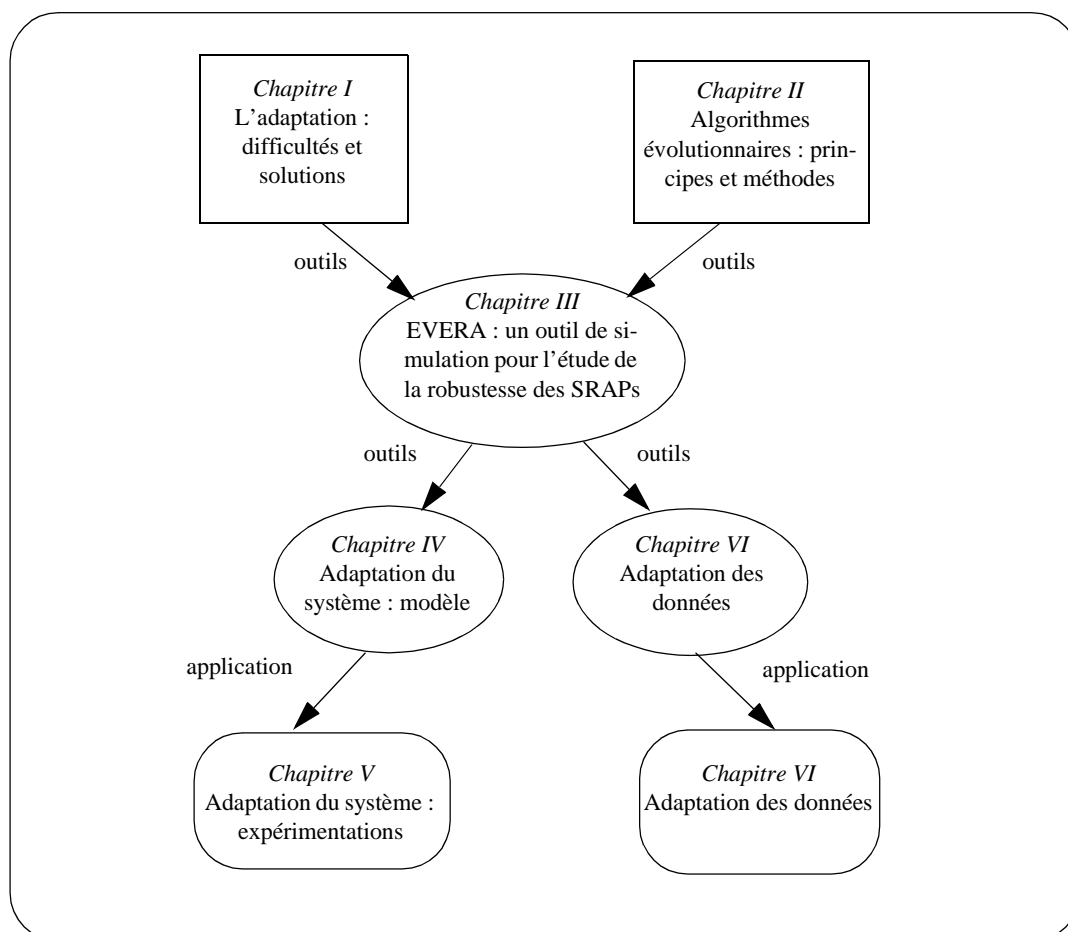
Dans un premier temps, nous avons cherché à adapter le système de reconnaissance lui-même aux différents environnements. Nous avons étudié les capacités du système à s'adapter aux changements de conditions acoustiques, à l'aide d'une approche

locale (par rétro-propagation du gradient) et d'une approche globale (par algorithmes évolutionnaires), en vue de trouver un SRAP optimal. Les résultats sont présentés dans le chapitre *Adaptation du système : expérimentations*.

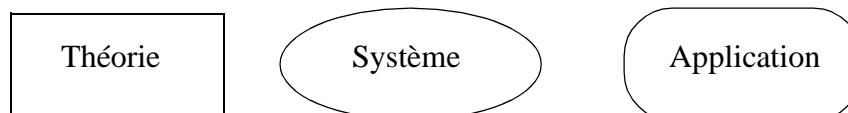
Dans un second temps, nous nous sommes placée dans le cadre du traitement des données en entrée du système. Partant d'une base de projection issue d'une analyse en composantes principales, nous avons cherché à trouver une base de projection adaptée à chaque environnement rencontré et permettant de retrouver les conditions acoustiques connues du système. L'étude que nous avons effectuée s'est alors portée sur l'optimisation de cet espace de représentation par algorithmes évolutionnaires. Nous avons étudié l'influence de ces algorithmes sur l'évolution de l'espace de représentation. Ce travail fait l'objet du chapitre *Adaptation des données*.

En résumé, l'organisation générale du présent document est articulée en 6 chapitres (voir Figure 1) :

- Le chapitre I présente une vue générale des problèmes rencontrés et des solutions proposées dans le domaine de la reconnaissance robuste de la parole.
- Le chapitre II décrit les principes généraux des algorithmes évolutionnaires, leurs capacités et leurs domaines d'application. Nous présentons nos choix concernant les méthodes évolutionnaires utilisées lors des expérimentations.
- Le chapitre III présente les plateformes existantes dans le domaine des algorithmes évolutionnaires et dans celui de la vie artificielle. Nous y justifions nos choix concernant les systèmes de reconnaissance utilisés. Enfin, nous présentons l'outil de simulation que nous avons élaboré et plus particulièrement les méthodes évolutionnaires que nous avons implémentées.
- Les chapitres IV et V présentent une application des algorithmes évolutionnaires pour l'adaptation aux changements d'environnement acoustique des systèmes de reconnaissance de la parole basés sur des réseaux connexionnistes.
- Le chapitre VI propose une application de ces algorithmes dans le domaine du filtrage des signaux de parole.



Légende :

**Figure 1** Organisation du document.

Partant de deux chapitres théoriques sur l'adaptation aux changements de conditions acoustiques (chapitre I) et l'adaptation par algorithmes évolutionnaires (chapitre II), nous avons élaboré un outil de simulation (chapitres III, IV et VI) qui nous a permis d'étudier deux applications en parole : l'adaptation du système (chapitre V) et l'adaptation des données (chapitre VI).

CHAPITRE I

L'ADAPTATION : DIFFICULTÉS ET SOLUTIONS

Ce chapitre traite des problèmes rencontrés lorsqu'un système de reconnaissance automatique de la parole est utilisé en situation réelle, c'est-à-dire hors laboratoire. Nous présentons les différents types de source de variabilité du signal de parole dégradant fortement les taux de reconnaissance, ainsi que les méthodes proposées pour y remédier.

Les systèmes de reconnaissance automatique de la parole (SRAPs) sont de plus en plus nombreux sur le marché. Parmi les plus connus, on pourra citer les systèmes ViaVoice de IBM, ou NaturallySpeaking de Dragon Systems Inc. Moyennant une adaptation, ils sont aujourd'hui pour la plupart robustes aux changements de locuteurs, au faible bruit de fond, etc. Pourtant, l'utilisation de ces systèmes dans des conditions acoustiques difficiles (fort bruit de fond, changement d'environnement rapides, etc.), nécessite toujours d'utiliser un microphone adéquat, d'entraîner le système sur une durée importante, d'effectuer des calculs de bruit de fond, etc. ce qui rend ces systèmes lourds et fastidieux à utiliser.

La différence entre les conditions d'apprentissage et les conditions de test est la principale raison de la chute des performances des systèmes de reconnaissance. De ce fait, un système qui obtient de bonnes performances en laboratoire n'est pas forcément performant en conditions réelles d'utilisation (banque, voiture, avion, etc.) (Néel et al. 96). Les deux principales causes à cela sont l'extrême variabilité acoustique des situations d'usage et la non-exhaustivité des corpus face à la multitude de contextes d'utilisation. Le locuteur et sa manière de produire le signal, l'environnement acoustique et sa manière de le propager, l'interlocuteur et sa façon de le percevoir et de l'interpréter, constituent de multiples sources de variabilité qui limitent *l'utilisabilité* de ces systèmes.

Afin d'assurer une adaptabilité de ces systèmes à ces différentes sources de variabilité, beaucoup de recherches ont été développées dans le but d'améliorer la robustesse des SRAPs. Pour ce faire, des techniques opérant à des niveaux divers ont été élaborées. Par exemple, un traitement peut être effectué de manière précoce au niveau acoustique en filtrant de la meilleure manière le signal altéré, ou il peut être réalisé de manière tardive au niveau sémantique en prenant en compte le contexte dans lequel s'effectue la communication. Enfin, il peut s'opérer au niveau de la prosodie ainsi qu'au niveau multimodal (gestes accompagnant la parole, mouvement des lèvres, etc.) (Caelen et al. 96).

Nous avons fait le choix de nous focaliser sur le traitement acoustique, où deux grandes tendances semblent émerger afin d'améliorer la robustesse des SRAPs. La première s'attache à traiter le signal en entrée pour l'amener le plus possible à des signaux "reconnaissables" par le système. La seconde se focalise sur l'adaptation du système de reconnaissance en lui permettant de réapprendre un corpus adéquat tout en modifiant sa structure.

Dans ce chapitre nous proposons une description des sources de variabilité du signal de parole, ainsi que des méthodes mises en oeuvre pour rendre les SRAPs le moins sensibles possible à ces variations.

I.1 Sources de variabilité du signal de parole

Différentes sources de dégradation du signal ont été répertoriées (cf. Figure 2), qui sont les variations liées au *locuteur*, aux *canaux de transmission* utilisés et à l'*environnement acoustique* (Junqua 97).

Tout d'abord, les variations acoustiques peuvent provenir des caractéristiques physiologiques (taille du conduit vocal par exemple) ou psychologiques (état de stress par exemple) de la personne qui émet le signal. Par ailleurs, les canaux de transmission utilisés, microphone, téléphone, etc. opèrent sur le signal une distorsion qui peut dégrader de façon significative la qualité du signal (Stern 95) (Matrouf 97).

Enfin, l'environnement a une grande influence sur l'intelligibilité du signal de parole : la réverbération de la pièce dans laquelle est utilisé le système par exemple, le bruit ambiant produit par un bruit de fond, un ventilateur, des locuteurs en arrière-plan, etc. peuvent altérer fortement le signal (Stern 95) (Matrouf 97).

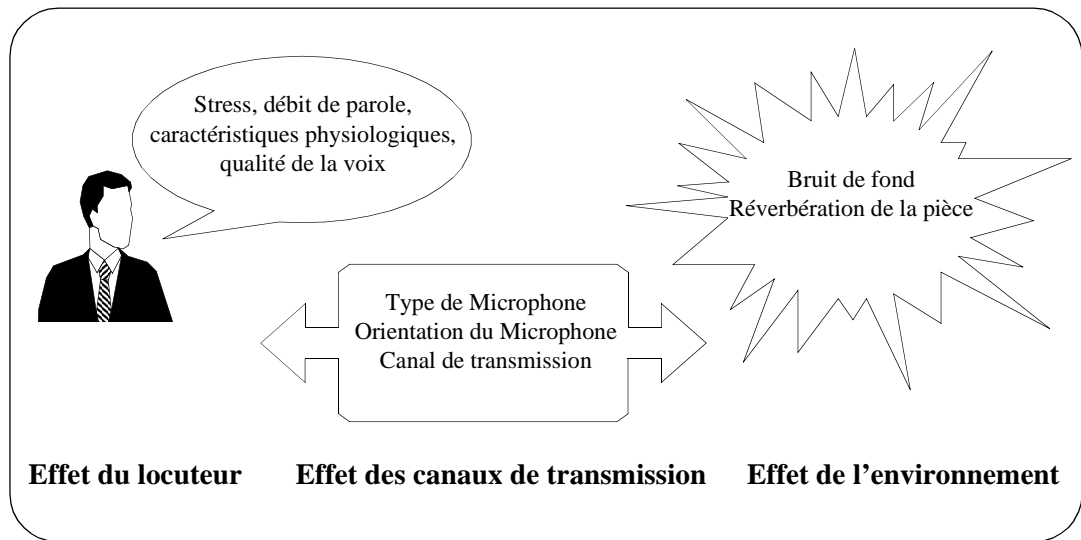


Figure 2 Représentation schématique de différentes sources de variabilité qui peuvent dégrader les performances d'un système de reconnaissance de la parole.

- *Effet du locuteur*

Le même message prononcé deux fois par un même locuteur dans des conditions identiques produit deux formes spectrales différentes. Cette variabilité est une variabilité dite *intra-locuteur*. La qualité de la voix, le débit de parole, le degré d'articulation sont tous des facteurs à la base de variations acoustiques pour un signal donné. Les conditions dans lesquelles le locuteur s'exprime influent sur le signal produit. C'est ainsi que des différences ont été notées entre la parole spontanée et celle issue de la lecture d'un texte. Les conditions de stress, dues à un environnement bruyant, entrent également en jeu. En effet, l'état psychologique du locuteur influe sur la qualité du signal qu'il produit. On a remarqué notamment que lorsque l'environnement est bruyant, le locuteur hausse la voix d'une manière qui change les caractéristiques du signal (Junqua 93) : c'est l'effet Lombard.

Par ailleurs, le même message prononcé par deux locuteurs différents engendre des variations beaucoup plus grandes, classées dans les variations dites *inter-locuteur*. De par la différence physiologique des conduits vocaux des êtres humains (longueur variant entre 17cm et 19cm environ), des différences notables sont constatées entre les locuteurs. La différence la plus évidente est la distinction homme-femme, ou enfant-adulte (Lee et al. 97).

- *Effet des canaux de transmission*

Le type de microphone utilisé, sa distance par rapport au locuteur, la manière dont il est dirigé, sont des facteurs très perturbateurs pour un SRAP et ont donné lieu à un certain nombre de travaux (Chang et Zue 94). De même, beaucoup d'études sont effectuées sur la reconnaissance de parole transmise par téléphone (Gong 97) (Han et al. 97).

- *Effet de l'environnement*

L'environnement dans lequel le locuteur s'exprime a une grande influence sur l'intelligibilité de la parole produite. Les caractéristiques de l'environnement telles la taille de la pièce, les matériaux posés sur les murs de la pièce, leurs facultés d'absorber des fréquences ou de modifier les propriétés du signal, etc. influent sur la qualité du signal perçu. On peut considérer que la pièce constitue un filtre de type convolutif, c'est-à-dire qu'elle agit sur le signal de parole de manière non linéaire dans le domaine *spectral*.

Les bruits ambiants de la pièce (les bruits de bureau, la distorsion due au canal de transmission, les bruits dans des véhicules ou dans le milieu industriel) sont aussi un facteur de variabilité. Ces bruits affectent directement le spectre du signal de parole.

Le spectre résultant du mélange du bruit et du signal de parole a des caractéristiques différentes de celui de la parole propre. Ainsi, on peut voir sur le mélange, l'apparition de certains pics, ou l'aplatissement du spectre par le masquage d'autres pics, la modification de la largeur spectrale de la bande ou la modification de la pente spectrale. Le bruit ambiant de la pièce est souvent considéré comme additif, c'est-à-dire que le bruit agit de manière linéaire dans le domaine *spectral*.

De plus, les bruits ont chacun des caractéristiques propres (périodiques, impulsifs, étendus sur une large bande de fréquences ou non, etc.) et ne peuvent être traités de la même manière. Seule une mesure est commune à tous, qui permet d'avoir une idée sur la quantité de bruit présent dans le signal, le Rapport Signal à Bruit (RSB), mesuré en Décibels (dB).

En résumé, il existe une multitude de problèmes qui limitent l'utilisation des systèmes de reconnaissance de la parole dans le monde réel. Un grand nombre de chercheurs se sont penchés sur ces problèmes. Nous en décrivons les grandes tendances dans la section suivante.

I.2 Solutions proposées

Les SRAPs sont devenus de plus en plus robustes ces dernières années face aux variabilités du locuteur ou aux différents environnements. Les différentes techniques (Junqua et Haton 96) (Gong 95) proposent de traiter le signal entaché par du bruit afin de le rendre le moins bruité possible, d'adapter le système de manière automatique aux petites dégradations du signal, ou d'incorporer de nouveaux média.

En se basant sur l'existence de multiples états de l'art en robustesse (Gong 95) (Stern 95) (Matrouf 97), nous proposons dans la Figure 3 une répartition des méthodes selon deux grandes catégories : soit le signal dégradé est traité afin d'avoir les meilleures caractéristiques possibles (*traitement du signal de parole*), soit le système de reconnaissance s'adapte aux nouvelles conditions acoustiques en intégrant ces nouvelles données (*traitement du modèle de reconnaissance*). Enfin, une catégorie de méthodes ne peut être répertoriée dans ces deux grandes familles : ce sont les méthodes intégrant d'autres sources que celles du signal de parole (*multimodalité*).

Concernant le traitement acoustique du signal de parole, nous pouvons considérer deux grandes tendances : diverses méthodes tentent de réduire l'effet du bruit sur le signal à l'aide de techniques de filtrages ou d'analyses robustes. Ces méthodes sont décrites de manière générale en I.2.1.

Concernant le traitement du modèle lui-même, un certain nombre de méthodes sont proposées pour adapter le système afin qu'il prenne en compte le bruit (*compensation*) ou qu'il utilise plusieurs modèles combinés (*modèles hybrides*).

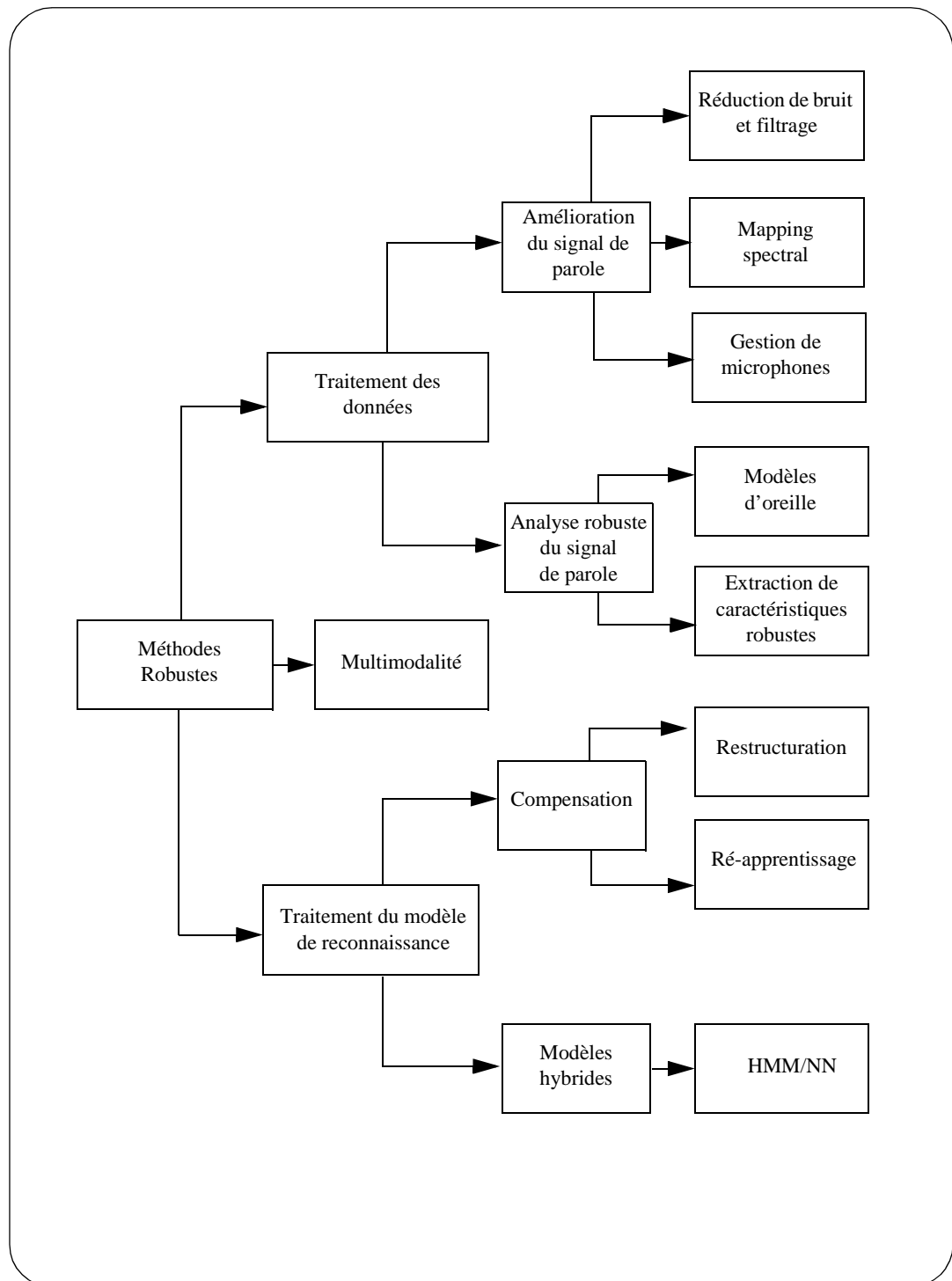


Figure 3 Méthodes pour l'amélioration de la robustesse des SRAPs.

I.2.1 Traitement acoustique du signal de parole

Comme nous l'avons énoncé précédemment, le signal bruité peut être "amélioré" en atténuant le bruit. Différentes techniques ont été élaborées pour pré-traiter le signal en entrée du système de reconnaissance. Nous avons répertorié dans cette catégorie les méthodes traitant le bruit dans le spectre, le mapping, ainsi que la gestion de microphones. Le but commun de toutes ces méthodes étant de rendre le signal, altéré par un bruit ou une distorsion, proche du signal reconnu par le système. Avant d'aborder les méthodes existantes, il est nécessaire de présenter l'effet que produisent les différents types de bruits sur le signal de parole.

I.2.1.1 Les différents types de bruit

Le bruit est défini au sens large comme étant tout signal perturbateur entachant à un degré ou un autre l'intégrité d'un signal utile (véhiculant l'information).

Il est souvent nécessaire d'opérer une distinction entre les bruits dus à des perturbations purement aléatoires et ceux qui apparaissent lors du captage accidentel d'autres signaux.

Plusieurs méthodes de classification des sources de bruits peuvent être opérées (il n'existe pas de classification typique). Ces classifications dépendent le plus souvent de l'application et de l'objectif poursuivis. Ainsi selon l'importance accordée à tel ou tel paramètre du signal, une classification peut apparaître comme étant plus intéressante qu'une autre. Nous donnons dans ce qui suit celles qui sont les plus communément rencontrées.

- *Classification basée sur la source du bruit*

Les bruits peuvent être endogènes ou exogènes au système.

Les sources de bruits externes agissent sur le système par influence : on peut généralement se prémunir facilement des perturbations de ce type de bruit. Leur origine peut être naturelle ou artificielle. Les perturbations artificielles (*man-made noise*) sont principalement générées par les équipements et les parasites créés par les activités humaines : utilisation des appareils électroménagers, imprimantes, etc. Les perturbations naturelles, quant à elles, sont associées aux conditions et au milieu de

propagation (influence des conditions atmosphériques, Fading , etc.). Leur intensité dépend le plus souvent de facteurs spécifiques tels que la localisation, l'environnement électromagnétique, l'architecture et le mode d'utilisation du système, etc. Dans ce cas, une évaluation des performances (du système de reconnaissance par exemple) n'est possible que lors d'un fonctionnement in situ.

Les sources de bruits internes sont très souvent impossibles à éliminer. Le cas du bruit thermique est le plus marquant. Généralement le bon fonctionnement du système n'est assuré que si le niveau du signal utile dépasse de plusieurs ordres de grandeurs le bruit (RSB de quelques dizaines de décibels). Toutefois, dans la réalité, on est souvent confronté au traitement de signaux à très faible RSB, en exploitant au mieux les connaissances *a priori* dont nous disposons à la fois sur ces signaux et sur les bruits les entachant.

- *Classification basée sur la stationnarité*

Un bruit peut être stationnaire ou non stationnaire.

Un processus (bruit) est dit non stationnaire si ses propriétés statistiques (moyenne, écart type, etc.) varient en fonction du temps. Généralement travailler avec une hypothèse de non stationnarité est très difficile. Même si le concept de processus stationnaire n'est qu'un modèle simplifié, il est cependant commode d'emploi et très usité en pratique surtout lorsqu'on se met dans un contexte d'observation limité (fenêtres temporelles, ce qui est le cas en parole) pendant lequel le phénomène (bruit) peut être considéré comme présentant un caractère permanent.

- *Classification basée sur la densité de probabilité*

Ne pouvant disposer ni d'une formulation analytique et ni évidemment d'une représentation mathématique déterministe du processus aléatoire, nous le caractérisons alors par ses propriétés statistiques. La théorie des probabilités est alors utilisée pour la description statistique des bruits et autres signaux aléatoires. On peut alors distinguer au moins trois types de bruits.

Le bruit blanc est un processus dont la densité spectrale de puissance est constante pour toute valeur de la fréquence. Il est ainsi appelé par analogie à la lumière blanche qui est composée des radiations de toutes les longueurs d'onde. Un tel bruit est bien sûr théorique car sa puissance moyenne est infinie. Cependant ce concept est très

utilisé dans les cas où le spectre réel peut être, en première approximation, remplacé par un spectre constant.

Le bruit Gaussien se dit d'un processus qui possède une densité de probabilité gaussienne. L'importance des processus gaussiens résulte du fait qu'ils sont le modèle asymptotique d'un grand nombre de phénomènes naturels. D'autre part, ils sont entièrement définis par leurs valeurs moyenne et covariance. L'autre propriété fondamentale réside dans le fait qu'un bruit gaussien, lorsqu'il transite par un système linéaire, reste toujours gaussien.

Le bruit poissonien permet l'étude de nombreux phénomènes résultant de la réalisation d'événements aléatoires distribués dans le temps. Ceux-ci peuvent être représentés par une séquence aléatoire d'événements qui sont susceptibles de survenir à n'importe quel instant avec la même probabilité.

- *Classification fréquentielle*

Une classification basée sur la répartition de l'énergie du bruit en fréquence peut également être utilisée.

Le bruit à large bande fait référence à l'étendue du spectre du bruit qui est définie entre deux fréquences de coupures f_1 et f_2 est dont la différence $f_2 - f_1$ qui constitue cette étendue est relativement élevée.

Le bruit à bande étroite est un processus aléatoire dont le spectre constant est circonscrit dans une bande de fréquences B finie relativement étroite définie entre deux fréquences f_1 et f_2 avec $B = f_2 - f_1$. Le spectre est nul en dehors de cette bande.

Cette classification est générale et les différentes catégories se recouvrent. Cependant, il est convenu que lors des expérimentations, on est souvent confronté à des bruits dont les sources sont généralement exogènes et présentent un caractère stationnaire.

I.2.1.2 Amélioration du Rapport Signal à Bruit

Dans cette famille de méthodes, nous incluons toutes les techniques consistant à éliminer au mieux le bruit, qu'il soit endogène ou exogène (cf. Figure 4). Parmi ces

méthodes, nous pouvons répertorier les méthodes de filtrage du bruit, de rehaussement du signal, de soustraction spectrale, mais également les techniques de *mapping* spectral et celles prenant en compte le positionnement et la directivité, ou les caractéristiques filtrantes du ou des microphones.

Différents types de filtres existent pour réduire le bruit. On notera l'utilisation de plus en plus répandue des techniques de séparation aveugle (c'est-à-dire sans connaissance *a priori* sur la statistique du bruit) du bruit et du signal de parole (Cardoso 89) (Jutten et al. 91) (Ngyen-Thi et al. 96). Ces filtres ont montré de grandes capacités à éliminer le bruit dans le spectre mais ils ont l'inconvénient de modifier également certaines bandes de fréquence du signal de parole.

D'autres méthodes proposent de rehausser le signal de parole en effectuant une estimation statistique du bruit environnant et du signal de parole (Ephraim 92a). On note alors un ensemble de techniques basées sur une approche Bayésienne (Ephraim et Malah 83) ou sur des HMM (Ephraim 92b).

Dans le même ordre d'idée, des techniques de soustraction du bruit dans le spectre, après son estimation, sont souvent employées.

Différents types de normalisation du signal provenant d'un environnement bruité (Huerta et Stern 97) ou d'un locuteur inconnu (Gouvêa et Stern 97) sont utilisés, dont la méthode de soustraction de la moyenne cepstrale (Westphal 97). Ces méthodes supposent généralement que les caractéristiques du bruit changent peu dans le temps.

Enfin, la méthode du *mapping* de l'environnement propose, si l'on représente un son dans l'espace des coefficients cepstraux, de réduire la distance entre un signal bruité et son équivalent non-bruité (Juang et Rabiner 87) (Barbier et Chollet 91) (Gong et Haton 94) (Guan et al. 97) (Sagayama 99).

L'utilisation de plusieurs microphones permet d'avoir une notion du lieu d'où provient le signal de parole et de se focaliser dessus en donnant moins d'importance aux sources sonores parasites. Un certain nombre de travaux ont été effectués (Flanagan et Mammone 91) (Tessier et Berthommier 99) traitant jusqu'à 500 microphones en parallèle (Silverman et al. 97).

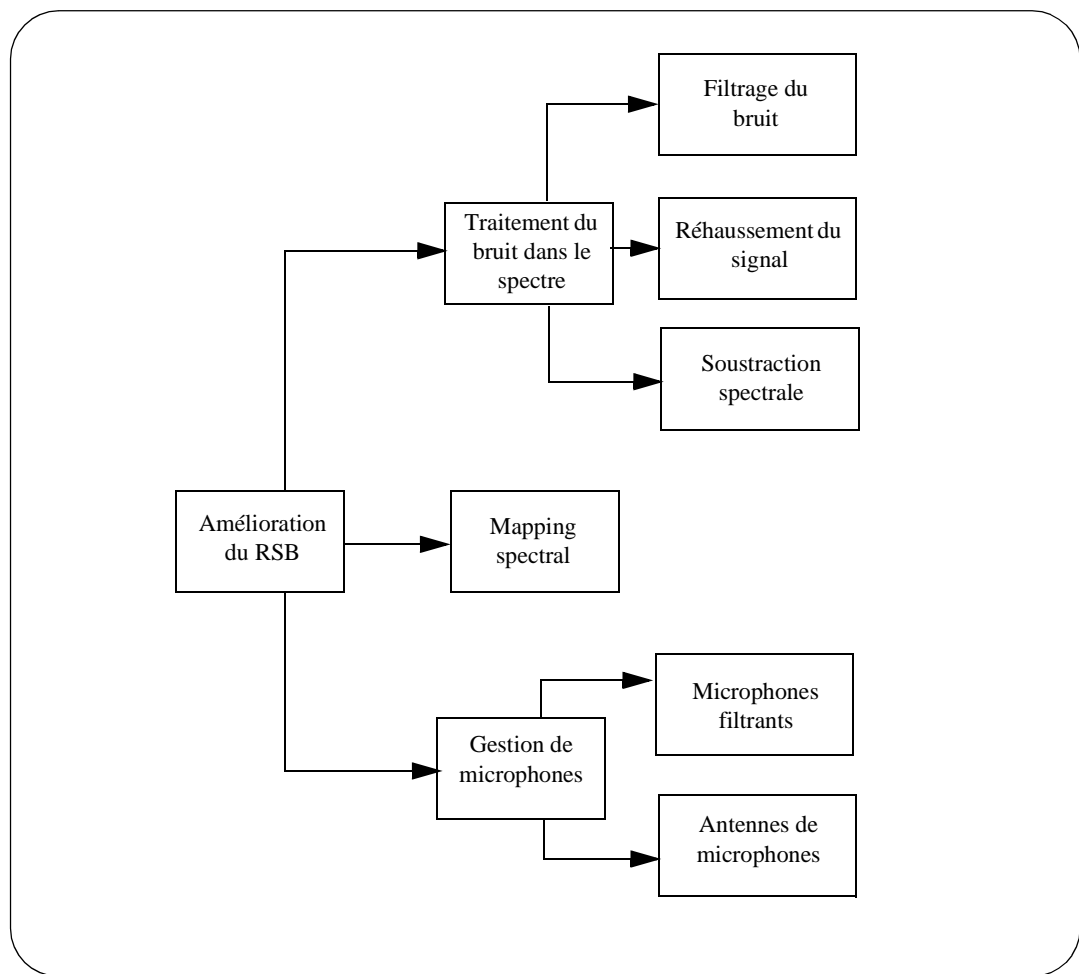


Figure 4 Méthodes pour l'amélioration du RSB.

La section suivante présente les méthodes analysant le signal bruité, de manière à ce que les paramètres découlant de l'analyse soient le moins dépendants du bruit entachant le signal.

I.2.1.3 Analyses robustes

Nous proposons de classer (cf. Figure 5) dans ce type de méthodes les analyses basées sur le *modèle d'oreille*, ou les analyses tentant d'*extraire les caractéristiques* les plus pertinentes du signal.

L'homme a des capacités très efficaces pour percevoir un signal en faisant abstraction du bruit environnant. C'est pourquoi il est naturel qu'un grand effort soit investi afin d'exploiter les connaissances existantes en perception humaine.

Contrairement aux SRAPs, l'être humain possède des capacités d'analyse phonétique du signal remarquablement robustes aux variations acoustiques dépendant du locuteur, des canaux de transmission, de l'environnement, etc. L'homme est capable de se focaliser sur une source sonore en faisant abstraction de l'environnement. Bien que des facteurs de plus hauts niveaux soient en jeu dans ce processus complexe, il est fort probable que le système auditif a un rôle important (Hermansky 97).

Différents modèles d'oreille fonctionnels sont utilisés pour l'analyse acoustique du signal, s'inspirant des capacités du système auditif à percevoir les signaux désirés dont le modèle de (Caelen 85) et les modèles de (Hermansky 90) et (Hermansky 97). Les coefficients PLP (Perceptual Linear Predictive) (Hermansky 90) sont basés sur les concepts psychoacoustiques connus (intégration des bandes critiques définies par Fletcher, préaccentuation par la courbe d'isotonie et loi de Stevens), et combinent plusieurs techniques d'approximation simulant les caractéristiques de l'audition humaine.

Ces méthodes permettent d'améliorer considérablement la robustesse des SRAPs sans nécessiter un calcul aussi coûteux que les autres méthodes.

La méthode RASTA-PLP (Hermansky 97) est une amélioration de PLP. Celle-ci consiste à ajouter un filtre passe-bande afin de réduire l'influence des bruits convolutifs dûs au canal de transmission.

Fletcher (dont les travaux sont reportés dans (Allen 94)) a suggéré que le processus de reconnaissance humain est basé sur des sous-bandes de fréquences qui sont traitées indépendamment les unes des autres. Ainsi le décodage d'un message linguistique se fait en parallèle, et de manière indépendante, dans différentes bandes de fréquences, et la décision finale est effectuée en mettant en commun les informations obtenues par chaque analyse. (Bourlard et Dupont 97) (Tibrewala et Hermansky 97) ont proposé de décomposer le signal en plusieurs bandes de fréquences et d'associer un système de reconnaissance par sous-bande. Cette méthode présente différents avantages : les sous-bandes sont en quelque sorte spécialisées dans la reconnaissance de certaines classes de sons, et le bruit peut être en partie ignoré en donnant moins d'importance aux sous-bandes dans lesquelles il se

trouve (Dupont et al. 97). Une difficulté apparaît en revanche, lors de l'étape de fusion des sorties pour laquelle il est difficile de trouver les bonnes pondérations.

Enfin, l'analyse MFCC (Mel-Frequency scaled Cepstral Coefficient) (Davis et Mermelstein 80) (Junqua 87), ainsi que diverses techniques consistant à tirer partie des caractéristiques du bruit afin d'en limiter l'impact sur le signal de parole (Martinez et al. 97), sont largement étudiées.

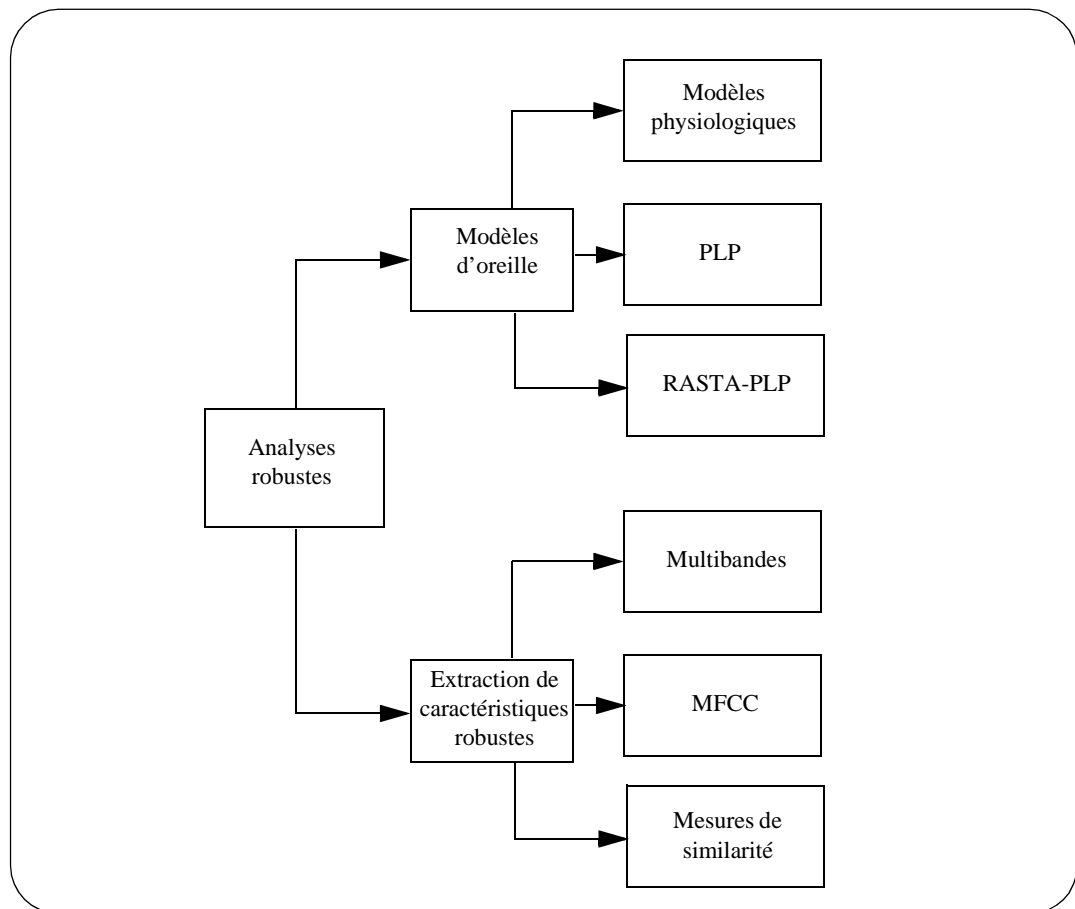


Figure 5 Méthodes d'analyse robuste.

Dans les sections précédentes, nous avons fait un état de l'art des méthodes traitant le signal bruité afin que les paramètres découlant de l'analyse acoustique soient le plus proches de ceux reconnus par le système. Ces méthodes font en sorte que le système de reconnaissance n'ait pas à s'adapter aux nouvelles conditions acoustiques.

Dans la section suivante, nous proposons de parcourir les méthodes permettant au système de prendre en compte le bruit et de le traiter au mieux.

I.2.2 Traitement du modèle

On peut distinguer deux manières d'intégrer le bruit au lieu de l'éliminer. Le système peut intégrer dans ses connaissances les nouvelles données bruitées que l'environnement lui fournit et ceci par ré-apprentissage et restructuration de son modèle. On parle alors de *compensation du modèle*. Mais il est également possible de combiner plusieurs systèmes ou méthodes (on parle alors de méthodes hybrides), ce qui permet de mieux gérer les distorsions entachant le signal.

I.2.2.1 Compensation

Les méthodes faisant partie de l'adaptation des systèmes aux variations acoustiques peuvent être classées en deux catégories (cf. Figure 6) : les méthodes supervisées, dans lesquelles les entités (voyelles, phonèmes, mots, etc.) à reconnaître sont connues, et les méthodes non supervisées, dans lesquelles les entités à reconnaître ne le sont pas. Une autre classification peut être faite en considérant les adaptations en ligne, où les énoncés prononcés par le locuteur sont traités comme de nouvelles données (inconnues), et les adaptations hors ligne, où le locuteur doit prononcer un certain nombre de données choisies par le système à l'avance.

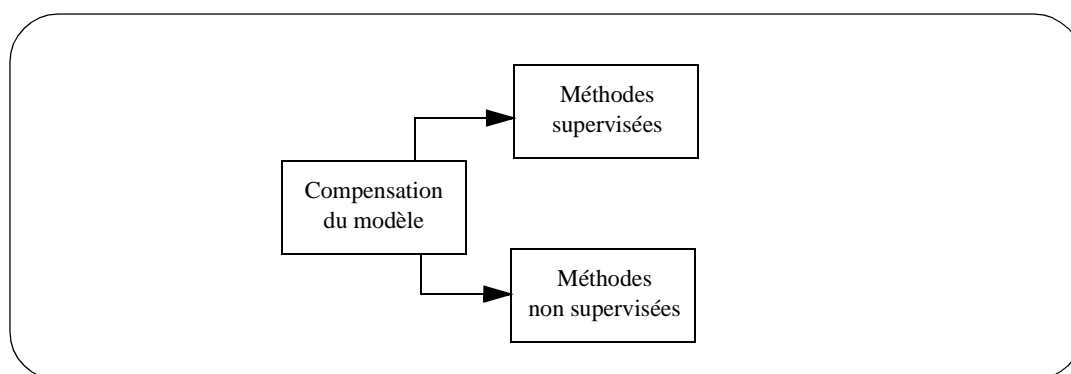


Figure 6 Méthodes de compensation.

D'un point de vue pratique, dans l'impossibilité de répertorier la multitude de conditions acoustiques existant dans la réalité, il est très difficile de construire des bases d'apprentissage exhaustives. Il est en effet fastidieux, voire irréaliste, d'obtenir une base d'enregistrements sonores représentant un grand échantillon de personnes

dans toutes les conditions acoustiques possibles. De plus, lorsque le bruit doit être pris en compte, il est difficile de traiter tous les bruits existants avec toutes les caractéristiques possibles (rapport signal à bruit par exemple). Si l'on mettait au point une base d'apprentissage suffisamment importante pour traiter l'ensemble des conditions acoustiques rencontrées jusqu'à présent, le système aurait certainement des performances médiocres dans toutes les situations, n'ayant pas eu les capacités de se spécialiser pour classifier correctement.

Il est possible de gérer le changement d'environnement en changeant les paramètres du système de reconnaissance. Par exemple, pour les HMM, le nombre d'états, la matrice de transitions, etc. sont modifiés. La méthode consiste à adapter un modèle entraîné dans un environnement non-bruité ou un environnement bruité proche de l'environnement de test (Chiang 97).

De la même manière, les modèles basés sur les réseaux de neurones voient le nombre d'unités, le nombre de couches ou les poids, modifiés.

La plupart des adaptations sont effectuées en mode supervisé car il est très difficile, voire impossible avec les connaissances actuelles, d'adapter le système de manière aveugle. On peut noter cependant les travaux de (Kemp et Waibel 99) sur l'adaptation non supervisée du vocabulaire utilisé par le système.

I.2.2.2 Modèles hybrides

L'hybridation de plusieurs méthodes afin d'exploiter les avantages de chacune, semble constituer une approche intéressante pour la robustesse des SRAPs (cf. Figure 7). Ainsi, on voit apparaître des systèmes combinant plusieurs systèmes de reconnaissance comme, par exemple, un HMM et un réseau de neurones (ou *Neural Network* (NN)).

La combinaison la plus courante de nos jours est l'hybridation HMM et réseaux connexionnistes (ou *Neural Networks* (NN)) (Haton 97). La principale raison est que ces deux systèmes présentent des avantages complémentaires. Les HMM ont une grande capacité à traiter les événements dans le temps, les réseaux de neurones sont experts dans la classification de formes statiques.

Les réseaux de neurones sont souvent utilisés afin de reconnaître l'environnement dans lequel le système se trouve. (Puel 97) propose de combiner un réseau multi-couches capable de fournir des informations sur le signal avec un système de reconnaissance basé sur des modèles de Markov cachés.

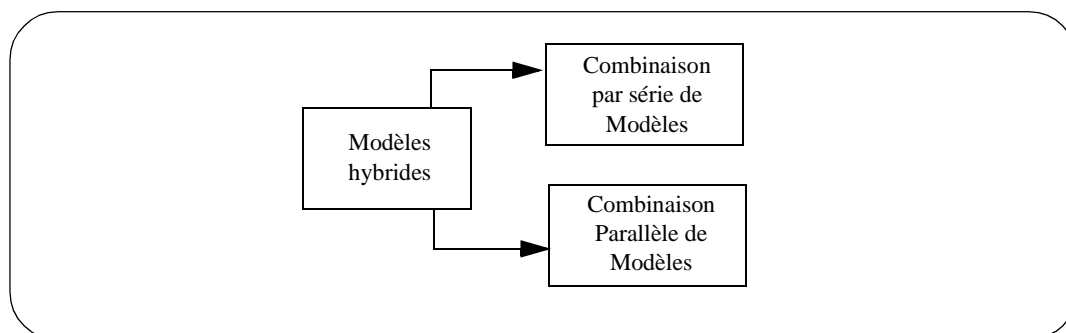


Figure 7 Méthodes hybrides.

I.2.3 Multimodalité

Divers travaux proposent d'incorporer une nouvelle modalité dans les systèmes de reconnaissance de parole. Il est connu que l'homme utilise une information visuelle afin d'améliorer sa capacité de perception sonore en milieu bruité (Kabré 97). (Bregler et König 94) (Jourlin et al. 97) proposent d'utiliser les informations fournies par le contour des lèvres afin d'augmenter le nombre d'indices nécessaires pour une bonne reconnaissance en milieu bruité. (Robert-Ribes et al. 97) (Tessier et al. 98) ont montré que les informations sonores et visuelles sont complémentaires lors de la reconnaissance de voyelles et la combinaison des deux types d'informations améliore la reconnaissance. Pour éviter toute contrainte de positionnement, (Revéret et al. 97) proposent de rendre la détection du contour des lèvres robuste à l'orientation de la tête.

I.3 Discussion

Parmi les nombreuses méthodes abordées dans les sections précédentes, très peu sont utilisées dans les systèmes de reconnaissance opérationnels. En fait, la plupart de ces

méthodes sont encore au stade de la recherche expérimentale et n'ont pas montré de résultats suffisamment convaincants pour être intégrées.

La méthode la plus couramment utilisée est certainement l'accroissement de la taille de la base de données d'apprentissage. Avec l'augmentation de la puissance de calcul et de la taille de la mémoire, il devient possible de fournir une très grande quantité d'information lors de l'apprentissage afin d'avoir un système performant dans de multiples situations de test.

Cependant, malgré la taille de plus en plus considérable des bases d'apprentissage, celles-ci ne sont qu'un échantillon très restreint de l'ensemble des variabilités possibles du signal de parole.

Devant la difficulté de ce problème, différents chercheurs explorent de nouvelles voies de recherche pour une meilleure compréhension des problèmes liés à la parole (Hermansky 97) (Bourlard 96). Nous proposons, dans les prochains chapitres de cette thèse, une étude explorant les possibilités offertes par les algorithmes dits *évolutionnaires*.

CHAPITRE II

ALGORITHMES ÉVOLUTIONNAIRES : PRINCIPES ET MÉTHODES

Ce chapitre présente les principes de l'évolution naturelle et artificielle. Nous détaillons les grandes lignes des algorithmes évolutionnaires et plus particulièrement les algorithmes génétiques et les stratégies d'évolution. Nous présentons ensuite nos choix concernant le type d'évolution que nous utilisons dans la suite de cette thèse.

L'évolution biologique a engendré des systèmes vivants extrêmement complexes. Elle est le fruit d'une altération progressive et continue des êtres vivants au cours des générations et s'opère en deux étapes : la sélection et la reproduction.

- La sélection naturelle est le mécanisme central qui opère au niveau des populations, en sélectionnant les individus les mieux adaptés à leur environnement.
- La reproduction implique une mémoire : l'hérédité, sous la forme de gènes. Ce matériel héréditaire subit, au niveau moléculaire, des modifications constantes par mutations et recombinaisons, aboutissant ainsi à une grande diversité.

Ces principes, présentés pour la première fois par Darwin, ont inspiré bien plus tard les chercheurs en informatique. Ils ont donné naissance à une classe d'algorithmes regroupés sous le nom générique d'*Algorithmes Évolutionnaires* (ou *Evolutionary Algorithms* (EA)). Les sections suivantes en présentent les fondements.

II.1 Algorithmes évolutionnaires

Il existe une catégorie de problèmes pour lesquels il est difficile, voire impossible, de trouver une solution en un temps limité. Il est alors utile de trouver une technique permettant la localisation rapide de solutions sous-optimales, sachant que l'espace de recherche a une taille et une complexité suffisamment importantes pour éliminer toute garantie d'optimalité (De Jong 88). Pour cela, un système capable de s'auto-modifier au cours du temps, tout en améliorant sa performance dans l'accomplissement des tâches auxquelles il est confronté, semble ouvrir la voie à une recherche intéressante.

Les EA sont basés sur des principes simples. En effet, peu de connaissances sur la manière de résoudre ces problèmes sont nécessaires, même si certaines peuvent être exploitées afin de rendre plus efficace l'évolution (en effet, il n'est pas réaliste d'espérer obtenir une méthode d'optimisation raisonnablement efficace si aucune connaissance sur le domaine à traiter). C'est pourquoi, dans de nombreux domaines, les chercheurs ont été amenés à s'y intéresser.

Un certain nombre de travaux ont porté sur l'optimisation de fonctions mathématiques complexes telles celles proposées dans (De Jong 75). Les EA ont été appliqués à différents problèmes issus de la recherche opérationnelle : la coloration de graphes (Eiben et al. 98), le problème du voyageur de commerce (Tao et Michalewicz 98) (Watson et al. 98), la gestion d'emploi du temps (Paechter et al. 98), etc.

On trouve aussi des applications en robotique pour la recherche de trajectoire optimale (Ahuactzin et al. 95) ou l'évitement d'obstacles (Schultz et Grefenstette 92), en vision pour la détection (Heap et Samaria 95) (Baluja 98) ou la reconnaissance d'images (Liu et Wechsler 98) (Brouard et al. 98), en recalage d'images médicales (Roux et Jacq 93), en reconnaissance de formes (Iwata et al. 90), mais aussi en mécanique pour l'optimisation de formes (Kane et Schoenauer 97), en chimie (Geyer et al. 98), économie et gestion de production (Fontanili et Vincent 97), gestion de trafic aérien (Oussedik et Delahaye 98), etc.

Par ailleurs, pour se rapprocher des phénomènes trouvés dans la nature, divers travaux ont porté sur des individus "intelligents" basés sur des réseaux connexionnistes. Ils ont montré l'intérêt d'appliquer les EA à des réseaux de neurones (Belew et al. 90) (Whitley 95), et particulièrement dans l'association avec l'apprentissage par rétro-propagation. Ils permettent entre autres de choisir un bon ensemble de poids initiaux, ce qui permet non seulement d'améliorer la vitesse de convergence des réseaux, mais aussi d'éviter de les faire converger vers des états correspondant à des minima locaux qui ne donneraient pas des solutions optimales (voir le chapitre *Adaptation du système : modèle* pour la bibliographie).

Enfin, un certain nombre de travaux ont été effectués sur l'application des EA dans des environnements changeants (le chapitre *Adaptation du système : expérimentations* en présente la bibliographie).

Concernant le domaine de la parole, il existe peu d'applications, nous en citons quelques unes ci-dessous.

(Nakisa et Plunkett 98) proposent de modéliser les capacités des nouveaux-nés à catégoriser des phonèmes. Afin de se rapprocher le plus possible de la réalité biologique, ils modélisent les capacités cognitives de l'enfant par un réseau à

apprentissage non supervisé de type Hebbien. Les algorithmes génétiques sont utilisés afin d'optimiser la structure du réseau ainsi que les paramètres liés à la fonction d'apprentissage. (Boëffard et Emerard 97) traitent d'une génération automatique de corpus optimisés par algorithmes génétiques pour l'apprentissage d'un réseau de neurones spécialisé dans la prosodie. (Takara et al. 97) proposent d'optimiser, à l'aide d'algorithmes génétiques, la structure d'un système de reconnaissance de mots isolés basé sur des HMM.

Bien qu'aucune application n'ait été réalisée dans le domaine de la robustesse des systèmes de reconnaissance automatique de la parole (SRAPs), ces travaux fournissent une base de travail solide. On se rapproche alors de la vie artificielle (Heudin 94) (Langton 96), où la notion d'interaction entre l'environnement et les individus est très forte. Nous présenterons, dans le chapitre *EVERA : un outil de simulation pour l'étude de la robustesse des SRAPs*, les outils de simulation proposés dans le domaine.

Dans la suite de ce chapitre, nous présentons les principes généraux des systèmes adaptatifs basés sur les EA, ainsi que les méthodes qui en découlent.

II.2 Principes généraux

Les EA sont une classe d'algorithmes d'optimisation par recherche probabiliste basés sur le modèle de l'évolution naturelle. Ils modélisent une *population* d'*individus* par des points dans un espace. Un individu est codé dans un *génotype* composé de *gènes* correspondant aux valeurs des paramètres du problème à traiter. Le génotype de l'individu correspond à une solution potentielle au problème posé, le but des EA est d'en trouver la solution optimale.

La plupart des problèmes peuvent être résolus par une méthode de type recherche locale. C'est le cas, par exemple, de l'apprentissage d'un réseau de neurones par rétro-propagation du gradient. Partant d'une configuration de poids initiale, la méthode va chercher la meilleure solution dans le voisinage de cette configuration. Cette solution est optimale localement mais peut ne pas correspondre à un optimal global car il est possible qu'il existe une meilleure solution qui n'est pas dans le voisinage de la configuration initiale. La Figure 8 illustre ce type de problème.

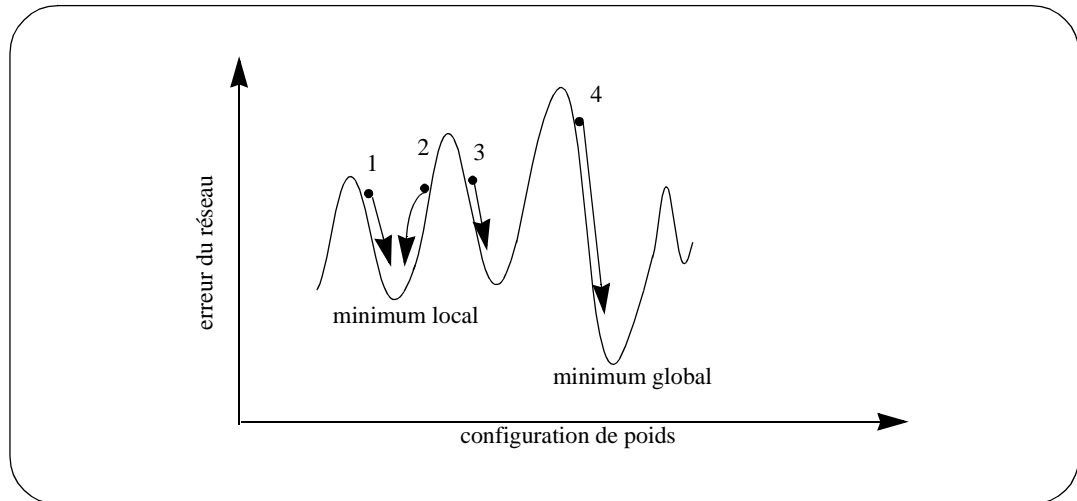


Figure 8 Exemple de convergence locale ou globale d'un réseau de neurones. Convergence vers des erreurs correspondant à des minima locaux (configurations 1, 2 et 3) ou à un minimum global (configuration 4).

Les EA ont montré leur capacité à éviter la convergence des solutions vers des optima locaux, aussi bien lorsqu'ils sont combinés avec des méthodes de recherche locale comme la rétro-propagation du gradient (Belew et al. 90) que lorsqu'ils sont seuls (Goldberg 89).

Quel que soit le type de problème à résoudre, les EA opèrent selon les principes suivants : la population est initialisée de façon dépendante du problème à résoudre (l'*environnement*), puis évolue de génération en génération à l'aide d'opérateurs de *sélection*, de *recombinaison* et de *mutation*. L'environnement a pour charge d'évaluer les individus en leur attribuant une performance (ou *fitness*). Cette valeur favorisera la sélection des meilleurs individus, en vue, après reproduction (opérée par la mutation et/ou recombinaison), d'améliorer les performances globales de la population.

Plusieurs types d'évolution ont été développés, donnant naissance à trois grandes tendances : les Algorithmes Génétiques (ou *Genetic Algorithms* (GA)), les Stratégies d'Evolution (ou *Evolution Strategies* (ES)) et la Programmation Evolutive (ou *Evolutionary Programming* (EP)). Une branche annexe, la Programmation Génétique (ou *Genetic Programming* (GP)) peut aussi rentrer dans ce type de systèmes.

De ces quatre méthodes classiques ont dérivé différentes techniques mélangeant les méthodes d'évolution des unes et des autres. Impossible à classer dans l'une des quatre familles citées ci-dessus, elles sont néanmoins considérées comme des EA.

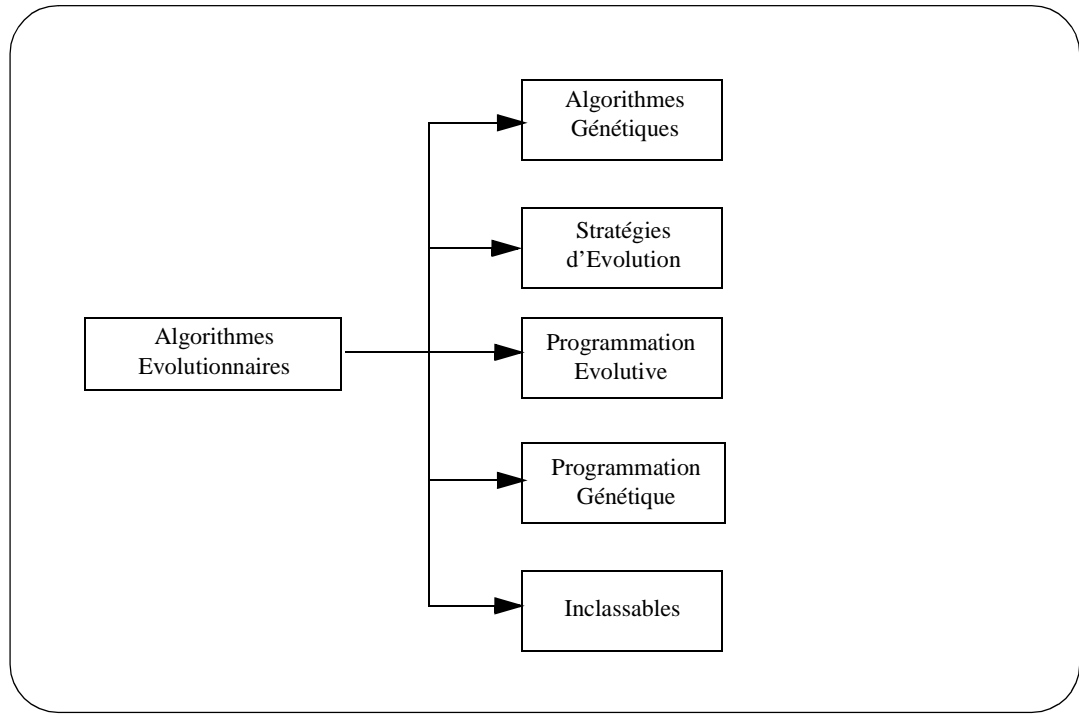


Figure 9 Différentes branches des algorithmes évolutionnaires.

Dans les sections suivantes, nous nous intéressons à deux de ces méthodes et nous décrivons les particularités des GA et des ES, méthodes que nous avons utilisées dans nos expérimentations décrites dans les chapitres suivants.

II.3 Algorithmes génétiques

II.3.1 Principes généraux

Développés dans les années 70 avec le travail de Holland (Holland 75) puis approfondis par Goldberg (Goldberg 89), les GA sont certainement la branche des EA la plus connue et la plus utilisée. La particularité de ces algorithmes est le fait qu'ils font évoluer des populations d'individus codés par une chaîne binaire. Ils utilisent les opérateurs de mutation binaire et de recombinaison de différents types.

II.3.2 Représentation

La représentation des individus est basée sur le codage binaire de l'information. La difficulté de cette représentation est de choisir le codage approprié.

(Michalewicz 96) suggère de trouver une représentation adaptée à l'espace de recherche associé au problème à optimiser et de choisir des opérateurs génétiques adaptés à cette représentation. Dans une telle représentation, deux génotypes donnant des solutions proches doivent différer de peu dans leur représentation. Ce n'est pas vraiment le cas dans la représentation binaire classique. On peut parler des falaises de Hamming traduisant le passage du nombre 7 (0111) au nombre 8 (1000) en 4 mutations. Différentes possibilités ont été proposées, comme le codage de Gray proposant (0100) pour 7 et (1100) pour 8, l'adaptation de la probabilité de mutation des gènes en fonction de leur position dans le génotype (Fogarty 89), ou la représentation de Dedekind (Surry et Radcliffe 96).

II.3.3 Evolution

L'évolution de la population se fait à l'aide des opérateurs de sélection par roulette (Goldberg 89), de mutation binaire et de recombinaison. Les GA favorisent la recombinaison plutôt que la mutation, en donnant de forts taux de recombinaison et de faibles taux de mutation. (Goldberg 89) propose dans son livre 80% de croisement et 3% de mutation.

II.3.3.1 Sélection

La sélection proposée par (Goldberg 89) consiste à sélectionner les individus proportionnellement à leur performance. Un individu ayant une forte valeur d'adaptation a alors plus de chances d'être sélectionné qu'un individu mal adapté à l'environnement. La Figure 10 représente sur un disque l'ensemble des individus et leur répartition selon leur performance. Si l'on considère la somme des performances de la population (le disque entier), la performance de l'individu I représente 9% de la somme, l'individu 2, 40%, etc. Ce qui équivaut à dire que l'individu I a 9% de chance d'être sélectionné, l'individu 2, 40%, etc.

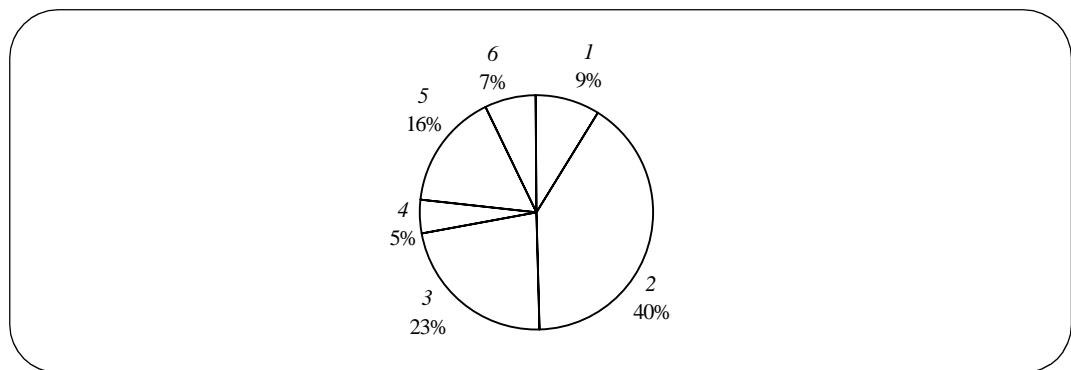


Figure 10 Modèle de la roulette.
Représentation d'une population de 6 individus.

II.3.3.2 Recombinaison

La recombinaison consiste à sélectionner aléatoirement une position de césure (*locus*) et de permuter les parties droites des deux parents. La Figure 11 nous donne un exemple de recombinaison.

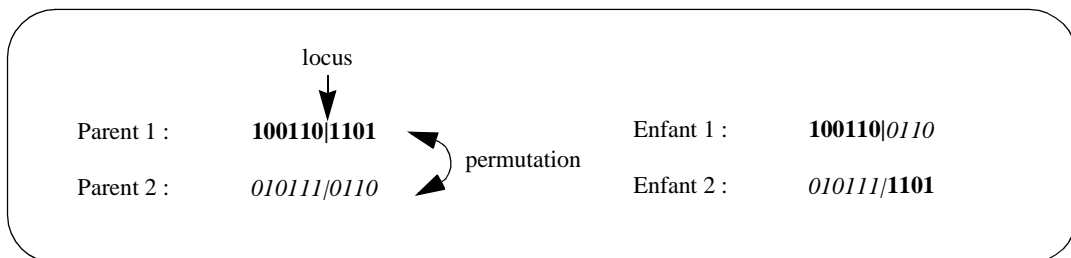


Figure 11 Opération de recombinaison.

D'autres types de recombinaison ont été proposés comme la recombinaison:

- en plusieurs points, qui consiste à sélectionner plusieurs locus et à permuter les différentes parties de la même manière que s'il y avait un seul point de césure,
- discrète, qui consiste à permuter en moyenne une fois sur deux les éléments du génotype,
- intermédiaire qui propose de remplacer le gène de l'individu enfant par une combinaison linéaire des gènes des deux parents,
- géométrique qui prend la racine carrée de la multiplication des deux gènes parents.

Cette liste n'est cependant pas exhaustive et il est possible de trouver des travaux sur l'élaboration d'autres recombinaisons (Sebag et Schoenauer 94) (Spears et De Jong 91).

II.3.3.3 Mutation

La mutation consiste à prendre des gènes aléatoires dans le génotype et à les altérer selon leur type. Un gène binaire i est muté en $\text{non}(i)$. La Figure 12 nous donne un exemple de mutation.

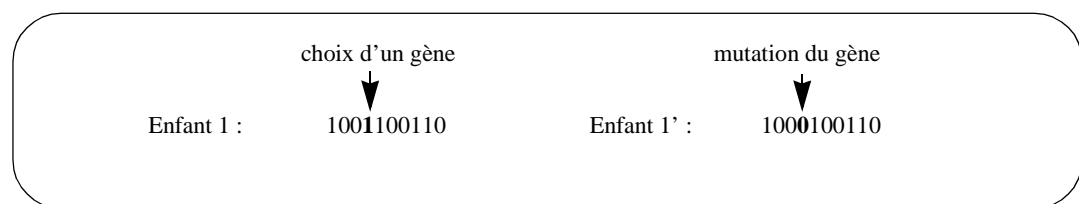


Figure 12 Opération de mutation.

II.4 Stratégies d'évolution

II.4.1 Principes généraux

Les ES sont apparues dans les années 70 avec les travaux de Ingo Rechenberg et ces travaux ont été poursuivis par Hans-Paul Schwefel (Schwefel 84). La première particularité de ces méthodes est de coder les paramètres du problème à résoudre en nombres réels. Ceci implique une méthode de mutation adaptée aux réels. La seconde est d'effectuer une sélection déterministe des individus en ne choisissant que les n ($n \geq 1$) individus classés selon leur performance. La troisième, enfin, est d'encoder les paramètres d'évolution directement dans le génotype afin de les faire évoluer au même titre que les valeurs des paramètres solutions du problème.

II.4.2 Représentation

D'une manière générale, un individu représente donc une solution potentielle au problème \vec{x} (équivalent au génotype des GA). Chaque composant du vecteur \vec{x} est muté selon une loi gaussienne. Un vecteur d'écarts-types $\vec{\sigma}$ est associé au vecteur \vec{x} . Si tous les éléments de \vec{x} sont mutés selon la même loi, $\vec{\sigma}$ ne comporte qu'un seul

élément, sinon $\vec{\sigma}$ est de même taille que \vec{x} . Enfin, il est possible de favoriser une direction de mutation des individus par un vecteur de rotation $\vec{\alpha}$. On parle alors de mutation corrélée. Par la suite, nous ne prendrons pas en compte cette mutation corrélée, pour ne traiter qu'une version simplifiée des ES. Un individu \vec{a} est représenté alors par :

$$\vec{a} = (\vec{x}, \vec{\sigma})$$

II.4.3 Evolution

Les ES favorisent la mutation plutôt que la recombinaison. Travaillant sur des réels, la mutation suit une loi généralement gaussienne avec des écarts-types généralement codés dans le génotype.

II.4.3.1 Sélection

La sélection des individus est déterministe. Deux types de sélection existent, qui sont les sélections (μ, λ) et $(\mu + \lambda)$. La première consiste à sélectionner les μ meilleurs parmi les λ enfants, la seconde sélectionne les μ meilleurs individus parmi les μ parents de la génération précédente et les λ enfants créés (chaque parent créant λ/μ enfants avec $\lambda > \mu$). Cette dernière méthode permet de ne pas perdre les meilleurs individus d'une génération à une autre mais accroît les possibilités que la population converge prématurément vers une solution qui n'est peut être pas optimale mais qui représente un minimum local.

II.4.3.2 Recombinaison

La recombinaison opère ici rarement sur le génotype contenant les variables du problème. Cependant, elle semble très utile pour l'évolution des paramètres de mutation. Les recombinaisons discrètes et intermédiaires sont les plus utilisées. Ainsi, en moyenne un paramètre sur deux est permuté avec celui d'un autre individu.

II.4.3.3 Mutation

Le codage étant réel, le problème se pose quant à la réalisation de la mutation. Les stratégies d'évolution proposent d'utiliser un modèle basé sur des distributions

normales, avec des écarts-types qui peuvent être directement codés dans le génotype. En considérant la représentation simplifiée (sans direction d'évolution) définie plus haut, la mutation est alors $\forall i \in \{1, \dots, k\}$ indice des gènes de \vec{x} et $\vec{\sigma}$:

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1))$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0, 1)$$

où $N_i(0, 1)$ représente la réalisation d'une variable suivant une loi normale d'espérance 0 et d'écart type 1 calculée pour chaque indice i , et $N(0, 1)$ est une variable de même type calculée une seule fois par individu. τ et τ' peuvent être considérés comme des taux d'apprentissage.

Ainsi, une mutation gaussienne est appliquée à chaque individu de la population en ajoutant une variable aléatoire gaussienne centrée en zéro et d'écart-type σ (ajustée au cours de l'évolution).

Hans-Paul Schwefel propose pour des résultats robustes l'initialisation des paramètres τ et τ' suivante :

$$\tau' \propto \frac{1}{\sqrt{2k}}$$

$$\tau \propto \frac{1}{\sqrt{2}\sqrt{k}}$$

II.5 Nos choix

Dans les sections précédentes, nous avons présenté deux méthodes classiques des EA. Nous en avons présenté les principes généraux afin d'en étudier les caractéristiques. Mais il existe une grande quantité de méthodes dérivées de ces principes classiques (Bäck 98). Sans même élaborer d'autres manières de muter ou de recombiner, il existe de multiples manières de combiner ces méthodes : en faisant varier la représentation des individus (binaire, réelle, etc.), le type de mutation (binaire, gaussienne, etc.), de recombinaison (discrète, intermédiaire, etc.), de sélection (roulette, déterministe, etc.). Il n'existe pas de règles concernant le choix d'une stratégie plutôt qu'une autre. Si les GA sont au départ plus spécialisés dans les codages binaires et les ES dans les codages réels, il est aujourd'hui très courant

d'utiliser les GA sur des individus codés en valeurs réelles, des GA auto-adaptatifs donc très proches des ES (Bäck 92), etc.

Parmi toutes les méthodes réalisables, nous avons dû faire des choix, que nous présentons dans la section suivante.

II.5.1 Algorithmes Génétiques

Les systèmes que nous allons utiliser par la suite sont représentés par des paramètres manipulant des valeurs réelles. Alors que deux écoles se confrontent sur le choix du codage de ces paramètres (en réels ou en entiers), nous avons choisi la représentation réelle. (Michalewicz 96) justifie l'utilisation des réels lorsque les paramètres ont de grandes valeurs et nécessitent une grande précision. Le codage de ce type de paramètres nécessite un nombre important de valeurs binaires, ce qui accroît considérablement la taille de l'espace de recherche et donc la complexité du problème à résoudre.

La représentation en réels a bien sûr été utilisée dans les GA, avec une mutation adaptée. Les puristes parlent alors d'EA au sens large, mais pour des raisons de simplicité, nous gardons le terme de GA pour une méthode utilisant la sélection par roulette, la mutation (réelle) et un opérateur de recombinaison.

La première méthode que nous utilisons dans nos expériences consiste alors en un GA de ce type. Nous proposons plusieurs types de mutation réelle (voir "*Module Individu*" page 49) et une recombinaison discrète.

II.5.2 Stratégies d'évolution

Concernant la deuxième méthode que nous avons utilisée, nous avons préféré nous limiter à l'étude d'une version simple des ES ne prenant pas en compte la mutation corrélée. Par souci de garder une certaine diversité dans nos populations, nous avons utilisé la stratégie (μ, λ) -ES. Si l'on considère un individu $\vec{a} = (\vec{x}, \vec{\sigma})$, chaque x_i est muté selon son écart-type associé σ_i . Le vecteur $\vec{\sigma}$ subit une recombinaison discrète.

Ces choix sont fondés sur les constatations trouvées dans la littérature. Nous sommes cependant consciente qu'aucune théorie n'appuie ces préférences. Notre démarche est donc empirique à ce niveau.

CHAPITRE III

EVERA : UN OUTIL DE SIMULATION POUR L'ÉTUDE DE LA ROBUSTESSE DES SRAPS

Ce chapitre présente l'outil EVERA (Environnement de Vie artificielle pour l'Etude de la Robustesse des Apprentis) que nous avons élaboré afin d'étudier l'apport des algorithmes évolutionnaires à la robustesse des systèmes de reconnaissance automatique de la parole.

Nous désirons étudier le comportement de systèmes de reconnaissance automatique de la parole (SRAPs), capables d'évoluer à l'aide d'algorithmes évolutionnaires, lors de changements d'environnements acoustiques. Pour cela, nous avons élaboré une plateforme de simulation permettant d'adapter différents composants d'un SRAP (analyseur acoustique, moteur de reconnaissance), et d'agir sur les différents paramètres des méthodes évolutionnaires. Nous avons intitulé notre plateforme EVERA (Environnement de Vie artificielle pour l'Etude de la Robustesse des Apprentis), que nous décrirons dans la suite de ce chapitre.

III.1 Principes

III.1.1 Généralités

Notre simulateur EVERA permet de faire évoluer des populations de systèmes à l'aide d'algorithmes génétiques ou de stratégies d'évolution (Kabré et Spalanzani 97). Bien qu'il soit possible de simuler des problèmes d'optimisation connus (voir *Annexe*), nous nous sommes plus particulièrement intéressée aux applications dans le domaine de la parole et essentiellement à ceux s'attachant aux problèmes de robustesse. Comme il a été décrit dans le premier chapitre, de nombreux problèmes restent posés en ce qui concerne la robustesse des systèmes de reconnaissance de la parole en environnement acoustique changeant.

Afin de rester fidèle aux approches connues dans ce domaine, nous proposons d'étudier l'évolution de deux types d'individus :

- Nous simulons dans un premier temps des individus ayant des capacités de sélection de capteurs, ce qui signifie des individus capables d'extraire du signal de parole les informations nécessaires et suffisantes pour que le système de reconnaissance soit performant dans une série d'environnements bruités. Nous désirons doter ces individus d'une capacité de filtrage des caractéristiques représentant le bruit.
- Dans un second temps, les individus sont eux-mêmes des systèmes de reconnaissance. Ils ont des capacités d'apprentissage. Nous désirons qu'ils soient

capables de s'adapter de manière efficace aux changements de conditions acoustiques afin d'assimiler les bruits qui apparaissent dans l'environnement.

Comme nous venons de l'évoquer, les deux types d'expériences que nous désirons mettre en place utilisent des systèmes de reconnaissance de la parole. Il s'agit désormais de discuter du choix concernant ces systèmes.

III.1.2 Systèmes de Reconnaissance

Dans le premier chapitre, les systèmes de reconnaissance de la parole ont été classés en trois types. Les modèles basés sur les HMM forment le premier, ils sont aujourd'hui les modèles les plus utilisés (Jouvet 95). Les réseaux de neurones (NN) constituent la seconde classe. Différents types de réseaux sont utilisés et parmi eux, les perceptrons multi-couches (Barnard et al. 95) (Burr 88) (Peeling et Moore 87) ou les réseaux récurrents (Hild et Waibel 93) (Robinson et Fallside 88). Ces méthodes appliquées à la reconnaissance de la parole sont largement détaillées dans (Haton 97) (Tebelskis 95) (Lippmann 89). Enfin, une troisième catégorie consiste à hybrider des HMM avec des réseaux connexionnistes (Deroo et al. 98) (Devillers 92), et sont souvent nommés HMM/NN. Cette voie de recherche a été largement discutée dans (Haton 97).

Nous avons fait le choix d'utiliser un système de reconnaissance de la parole basé sur un réseau connexionniste de type perceptron multi-couches pour plusieurs raisons. Tout d'abord, ces réseaux ont de grandes capacités de classification et ont montré leurs aptitudes en parole, notamment pour les mots isolés (Burr 88) (Peeling et Moore 87). De plus, par rapport à l'ensemble des systèmes connexionnistes, ils ont l'avantage d'être basés sur des principes simples et relativement maîtrisables. Contrairement aux réseaux récurrents par exemple, leur temps de convergence peut être relativement rapide. Il est également possible de détecter le moment où l'algorithme d'apprentissage n'est plus capable d'améliorer les performances, ce qui permet d'optimiser le temps d'apprentissage. Enfin, l'hybridation des NN avec des EA a été largement étudiée depuis une dizaine d'années (voir chapitre *Adaptation du système : modèle* pour l'état de l'art), ce qui est moins le cas des HMM, dont la bibliographie sur l'hybridation est nettement moins développée. On notera quand même dans ce domaine les travaux de (Brouard et al. 98) sur la reconnaissance

d'images, ou quelques travaux en parole sur l'optimisation de structure de chaînes de Markov pour la reconnaissance de mots isolés (Takara et al. 97).

III.2 Plateformes existantes

Il existe un grand nombre de simulateurs d'EA dont un bon échantillon est décrit dans (RéfWeb 2). La plupart de ces systèmes proposent des méthodes génériques basées sur les algorithmes classiques des EA (Schoenauer 95) (Bäck 96). Cependant, il n'est pas encore dans les habitudes d'utiliser ces bibliothèques pré-définies, et la plupart des chercheurs programment leur propre plateforme. La raison la plus probable est la relative simplicité de l'implémentation de ce type d'algorithmes. La seconde raison est la grande diversité des applications des EA dans des domaines très différents. Les contraintes imposées par le domaine d'application (valeurs bornées des gènes, dépendance des variables entre elles, nécessité de parallélisme, etc.) impliquent d'avoir un système ad-hoc au problème traité. Dans ce cas, il est beaucoup plus simple d'implémenter les méthodes désirées avec une architecture, des opérateurs d'évolution et une fonction d'évaluation des individus bien adaptés au problème que l'on désire traiter.

Dans notre cas, la notion d'environnement est très importante. Les individus, "reconnaisseurs" ou "systèmes perceptifs" de signaux sonores, interagissent dans leur environnement acoustique. Bien que les actions qu'ils effectuent pour s'adapter soient très dépendantes de l'environnement, les individus n'exercent aucune modification sur l'environnement qui les entoure. C'est peut-être la raison pour laquelle il n'est pas possible de parler de vie artificielle proprement dite (Langton 96). Cependant, il est intéressant d'étudier les environnements de vie artificielle existants dont le concept est proche du nôtre.

Concernant ces environnements, une multitude d'outils ont été développés pour des applications plus ou moins spécifiques à un problème donné. Généralement, ces plateformes de simulation proposent un environnement qui facilite le développement d'ensembles d'agents interagissant dans un environnement dynamique. Swarm (Hiebeler 94) a permis la simulation de modèles économiques (Weisbuch et al. 96), de sociétés d'insectes (Bonabeau et al. 94) etc., LEE (Menczer et Belew 96) s'est

plus concentré sur la description d'agents pourvus d'un réseau de neurones, de capacités sensori-motrices et d'un réservoir d'énergie, Manta (Drogoul 93) permet la simulation de colonies de fourmis, etc. La grande ouverture de ce genre de système a également permis d'étudier des phénomènes liés à la communication langagière. (MacLennan 91) et (Werner et Dyer 91) ont étudié l'apport et l'évolution de la communication entre des individus lors de l'accomplissement de tâches simples. (MacLennan 91) étudie plus particulièrement l'évolution des performances d'une population d'individus représentés par des machines à états finis, ayant pour but de décrire leur environnement afin que leurs voisins puissent agir en fonction. (Werner et Dyer 91) travaillent sur une population d'individus sexués de type réseaux connexionnistes afin de voir l'évolution du langage de femelles guidant des mâles aveugles pour les rejoindre. Si ces travaux se rapprochent du domaine de la parole, ils sont néanmoins orientés sur l'étude du dialogue plutôt que sur le traitement de la parole au niveau acoustique.

Malgré l'abondance de plateformes existantes, nous n'avons pas trouvé un système permettant d'étudier le problème qui nous intéressait, à savoir la robustesse des SRAPs. Nous avons besoin d'un simulateur d'environnement acoustique suffisamment flexible afin de nous permettre d'agir sur les caractéristiques des conditions acoustiques. Nous désirions ensuite pouvoir simuler différents types d'individus ayant des rôles très différents dans ces environnements acoustiques.

C'est pourquoi nous avons préféré développer notre propre simulateur : EVERA.

III.3 Architecture du système EVERA

Comme nous l'avons précédemment évoqué, nous désirons étudier l'effet des changements d'environnement acoustique sur les systèmes de reconnaissance, ainsi que l'apport de l'évolution sur l'amélioration des performances de ces systèmes. En conséquence, il est nécessaire de pouvoir agir sur trois facettes :

- les caractéristiques de l'environnement,
- les caractéristiques des individus,
- les paramètres de l'évolution.

Notre système se décompose donc en 3 modules principaux :

- Le module *Environnement* permet de gérer les paramètres de l'environnement acoustique, il se charge de fournir aux individus, ainsi qu'au module d'évolution, les signaux de parole provenant d'un environnement acoustique virtuel.
- Le module *Individu* gère des individus de différents types et en particulier ceux de type réseau de neurones ou base de projection. Chaque individu a pour charge de s'évaluer et de fournir sa performance au module d'évolution.
- Le module *Evolution* se charge de sélectionner les individus parmi la population et de créer la génération suivante. Lorsque l'évolution est terminée, l'environnement est averti et peut proposer de nouvelles données.

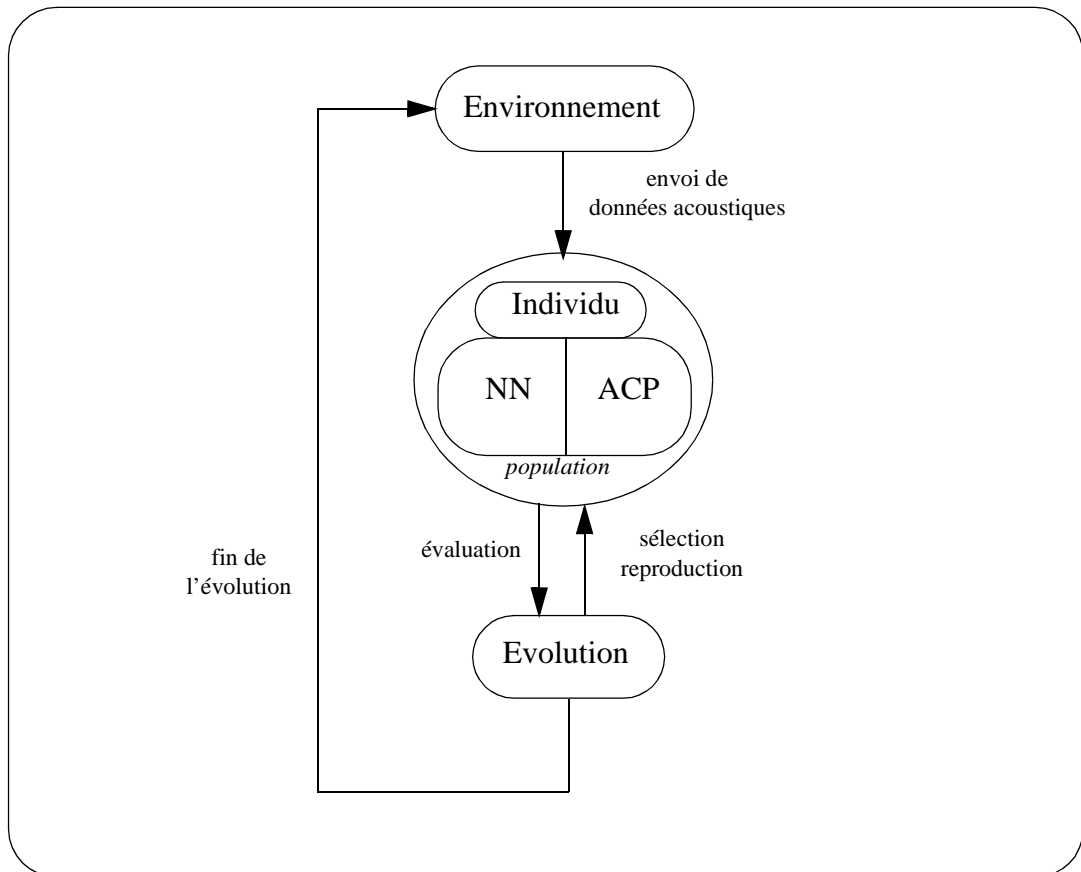


Figure 13 Représentation schématique des modules de EVERA.

III.3.1 Module environnement

Le module Environnement gère les paramètres liés au signal de parole. Il permet de choisir un type de pièce fermée. La taille de la pièce, les objets bruyants meublant la pièce (téléphone, réveil, etc.), la distance entre la source produisant un signal de parole et le microphone le captant, permettent de jouer sur la qualité du signal produit. On peut ainsi modifier le rapport signal à bruit, ainsi que le temps de réverbération du signal.

Cet environnement acoustique virtuel, dans lequel évoluent les individus, est simulé par un modèle mathématique d'une pièce close (quatre murs, un plafond et un sol) basé sur la méthode des images (Allen et Berkley 79) (Tsingos 98). Ce modèle simule la propagation des ondes sonores dans une pièce fermée rectangulaire. Les murs sont considérés comme rigides, ce qui permet de supposer que l'onde réfléchie n'a ni perdu de vitesse, ni changé de fréquence ou de direction.

La Figure 14 représente schématiquement le modèle de Allen. Une pièce est représentée par le triplet <longueur, largeur, hauteur>. Le son perçu en un point de la pièce est calculé en fonction des caractéristiques de la pièce, de l'emplacement du microphone (M) et du lieu où a été produit le signal (X). La réverbération de la source sonore est modélisée par une source-image symétrique par rapport au mur réfléchissant. Ainsi, la source X a six images (pour les quatre murs, le sol et le plafond) et le signal perçu en M est le résultat de la propagation de cette source et ses six images vers M (on néglige les réflexions secondaires). L'ajout de bruit dans la pièce est considéré comme une autre source sonore qui est traitée de la même manière.

En conséquence, cet environnement virtuel laisse le choix sur le type de bruit, le rapport signal à bruit ainsi que sur le temps de réverbération. Ce qui nous permet de générer une grande quantité de conditions acoustiques de manière simple et maîtrisée (Kabré et Spalanzani 97).

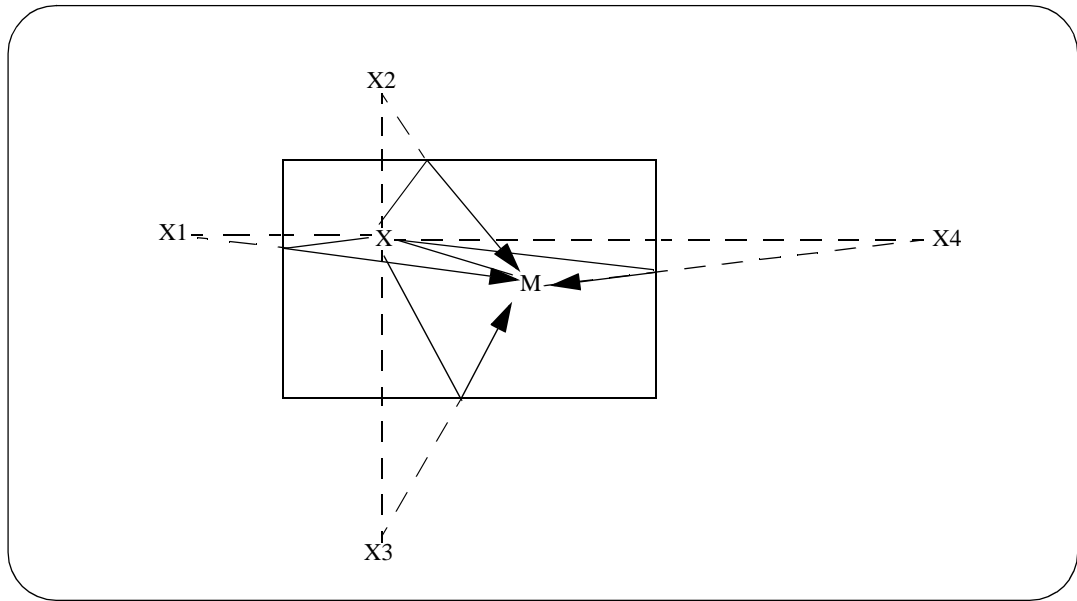


Figure 14 Modèle de propagation d'une source sonore.
Réduction de la représentation à 4 murs.

III.3.2 Module Individu

Les individus ont trois capacités : muter leurs gènes, se croiser avec d'autres individus et évaluer leur adéquation à l'environnement (performance). Si la mutation et le croisement ne dépendent pas du type d'individu mais du type d'évolution choisi, le calcul de la performance, lui au contraire, en est fortement dépendant. C'est pourquoi nous avons fait le choix de gérer cette fonction dans des classes héritées spécifiques aux problèmes que nous désirons traiter (voir III.3.4).

La classe individu gère les paramètres spécifiques à chacune de ces instances. Ainsi, les variables du problème à optimiser sont dans un vecteur constituant le génotype, la valeur d'adaptation au problème est représentée par la variable *fitness*. Selon la stratégie choisie, l'individu a un ou deux parents ayant permis de le générer, qui sont représentés par les variables *Parent1* et *Parent2*. Concernant les stratégies d'évolution, plusieurs paramètres sont nécessaires, à savoir le vecteur d'écart-types $\vec{\sigma}$ associé au génotype, et les constantes d'apprentissage τ_0 et τ_i utilisées dans la mutation des individus ES.

- *Mutations*

Parmi la multitude de mutations existantes, trois mutations sont proposées. Il est possible de multiplier un gène par une valeur aléatoire (mutation multiplicative), d'ajouter une valeur aléatoire (mutation additive) ou d'effectuer une mutation gaussienne.

La mutation multiplicative consiste à réduire la valeur du gène en la multipliant par une valeur aléatoire comprise entre 0 et 1, soit pour un gène x_i :

$$x_i = x_i \cdot \text{rand}(0, 1)$$

La mutation Additive consiste à muter les gènes en ajoutant une valeur aléatoire comprise entre -1 et 1, soit pour un gène x_i :

$$x_i = x_i \pm \text{rand}(0, 1)$$

La mutation Normale consiste à ajouter une valeur calculée selon une loi normale centrée sur 0 et d'écart-type σ :

$$x_i = x_i \pm \text{norm}(\sigma)$$

- *Recombinaison*

Il existe aussi un grand nombre de recombinaisons. Parmi elles, deux sont implémentées dans EVERA. La première est le croisement simple, base des algorithmes génétiques classiques, la seconde est le croisement discret plus spécifique aux stratégies d'évolution (la description de ces méthodes est faite dans le chapitre *Algorithmes évolutionnaires : principes et méthodes*).

- *Performance*

Le calcul de la performance (fitness) est différent selon le type d'individu. La raison est évidente : les individus n'ont pas le même objectif. Dans le cadre des premières expériences, l'objectif des individus est de fournir les données les moins bruitées possibles au système de reconnaissance. Dans les secondes, leur objectif est de reconnaître au mieux les informations bruitées qui leur sont fournies.

Les fonctions objectif utilisées sont présentées de manière plus détaillée lors de la description de l'implémentation (voir III.3.4) et dans les chapitres spécifiques aux deux problèmes (voir chapitres IV et VI).

III.3.3 Module évolution

Ce module gère l'évolution des individus suivant une stratégie déterminée ainsi que les paramètres liés à la population.

La population d'individus est stockée dans cette classe, ainsi que le nombre d'individus de la population. Cette classe gère aussi les paramètres liés à l'évolution de la population. Ainsi, il est possible de choisir le type d'évolution (stratégies d'évolution ou algorithmes génétiques), la probabilité de mutation des gènes, la probabilité de recombinaison des individus. Sont stockés le nombre de mutations, de recombinaisons, la somme des performances de tous les individus, ainsi que la performance moyenne, celles du meilleur individu et du plus faible.

Bien qu'il existe différentes sortes d'algorithmes évolutionnaires, il est possible de généraliser leurs principes par l'algorithme suivant :

```

t = 0                                // initialisation du temps à 0
Initialisation de P(t)                // création de la population initiale
Evaluation de P(t)                    // évaluation de cette population
Tant que non(critère d'arrêt)         // nombre de générations atteint ou
                                      // solution obtenue acceptable
    P'(t) = Sélection(P(t))           // sélection des parents
    P''(t) = Evolution (P'(t))        // recombinaison et mutation des parents
    Evaluation(P''(t))                // évaluation de la nouvelle population
    P(t+1) = P''(t)                   // mise à jour de la population courante
    t = t + 1                          // nouvelle génération
Fin tant que

```

Figure 15 Algorithme général de l'évolution.

La population est initialisée et évaluée une première fois à l'instant $t = 0$. La population évolue alors pendant un certain nombre de générations (fixé empiriquement et/ou à l'aide d'un critère d'arrêt) en sélectionnant les individus parents, en les mutant et/ou en les recombinant. Les enfants créés permettent alors de créer la population qui constituera la nouvelle génération.

L'algorithme présenté Figure 15 ne présente que les grandes lignes du traitement. La sélection dépend de la stratégie choisie.

Afin d'explorer différentes branches des EA, nous avons implémenté les deux stratégies les plus répandues dans le domaine :

La première est directement inspirée de l'évolution par algorithmes génétiques. Si la représentation des organismes n'est pas binaire, les principes d'évolution utilisés correspondent aux principes des GA (sélection par roulette, croisement et mutation). La mutation est adaptée à la représentation réelle. Nous en décrivons plusieurs types par la suite.

La seconde est basée sur les stratégies d'évolution. La mutation adaptative (non corrélée) est utilisée, la sélection est de type (μ, λ) -ES.

La Figure 16 récapitule l'ensemble des méthodes évolutionnaires implémentées dans EVERA.

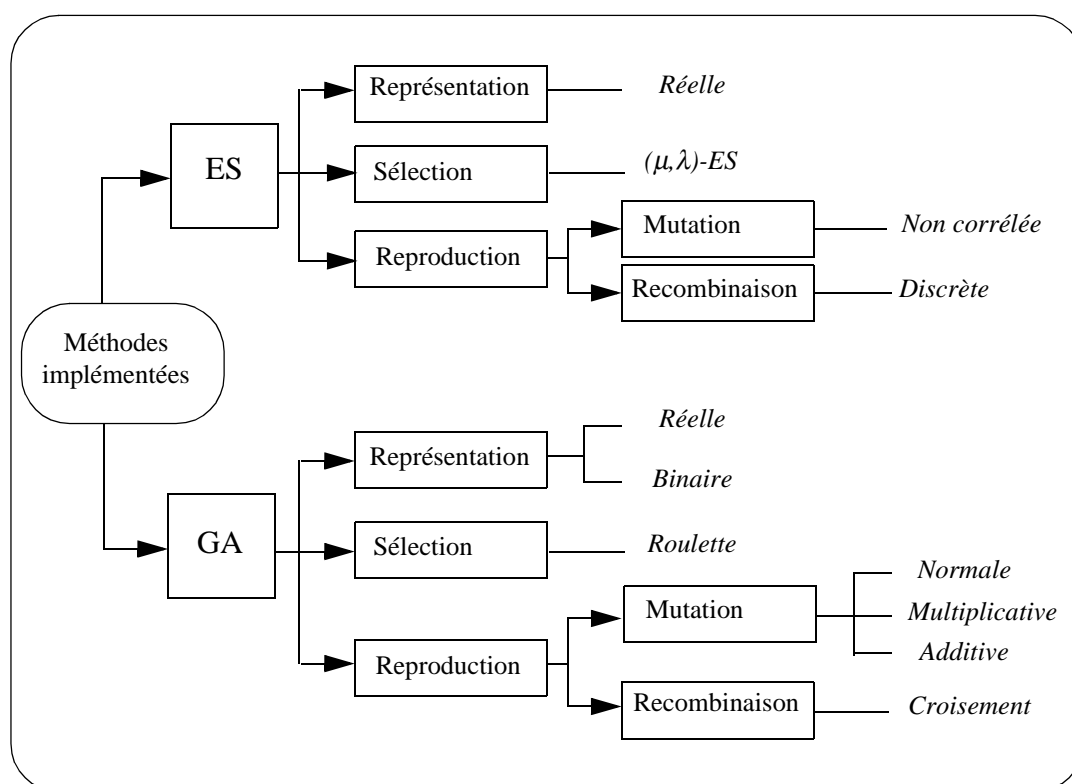


Figure 16 Représentation schématique des méthodes évolutionnaires implémentées.

III.3.4 Implémentation

L'ensemble des modules décrits précédemment sont décrits en C++ et leurs spécifications sont décrites plus en détail ci-dessous.

- *Le module environnement*

Défini dans la Figure 17, le module environnement contient les informations concernant les caractéristiques acoustiques des signaux de parole. Ainsi, l'utilisateur peut choisir le temps de réverbération <reverberation>, le rapport signal à bruit <snr>, le type de bruit entachant le signal de parole <tnoise>, ainsi que le type de signal à fournir au système de reconnaissance <patterns> et son nombre <nbPatterns>.

Par l'intermédiaire d'un code effectuant l'interface entre ces données et le modèle de pièce précédemment décrit, la méthode *CreateCorpus* génère un corpus de signaux de parole bruité.

```

Classe Environnement
{

    Attributs :

        float      reverberation;    // temps de réverbération
        float      snr;              // rapport signal à bruit
        typeNoise  tnoise;           // type de bruit
        typePat    patterns;          // type de signal (voyelles ou digits par exemple)
        int        nbPatterns;        // nombre de signaux envoyés

    Méthodes :

        void      createCorpus (
                        float      reverberation,
                        float      snr,
                        typeNoise  tnoise,
                        typePat    patterns,
                        int        nbPatterns
                    );
        // crée le corpus de signaux de parole bruités selon les paramètres définis par l'utilisateur,
        // à savoir : le temps de réverbération <reverberation>, le rapport signal à bruit <snr>,
        // le type de bruit <tnoise>, le type de signaux à reconnaître <patterns> et
        // le nombre de signaux à reconnaître <nbPatterns>

}

```

Figure 17 Classe Environnement.

- *Le module individu*

Un individu est représenté par la classe **Individu** de la Figure 18.

```

Classe Individu
{

    Attributs :

        genType*   genotype;           // pointeur vers tableau de gènes
        float      currentFitness;      // fitness associée au génotype
        int        lengthGen;           // taille du génotype
        Individu*   mate1;              // parent 1
        Individu*   mate2;              // parent 2
        float*      sigma;              // pointeur vers écarts-types pour la mutation ES
        float       tau0;               // variable associée à la mutation ES
        float       tau1;               // variable associée à la mutation ES

    Méthodes :

        virtual float Evaluate ()=0;
        // associe une fitness en fonction du génotype et du problème à optimiser

        void CrossOverAndMutate (
            Individu*   mate1,
            Individu*   mate2,
            Individu*   child1,
            Individu*   child2,
            MutationType tMut;
            float        pCross,
            float        pMut,
            int*         nCro,
            int*         nMut
        );
        // spécifique à la reproduction GA
        // génère deux individus <child1> et <child2> à partir de deux parents <mate1> et <mate2>
        // selon le type de mutation <tMut>
        // et les probabilités de recombinaison <pCross> et de mutation <pMut>
        // met à jour le nombre de croisements et de mutations effectués

        void CrossEs (Individu* mate1, Individu* mate2, Individu* child);
        // spécifique à la reproduction ES
        // effectue le croisement des sigmas à partir de deux parents <mate1> et <mate2>
        // en vue de créer l'enfant <child>

        void MutateEs (Individu* child);
        // génère un individu <child> par mutation non-corrélée, selon la méthode des ES
    }

```

Figure 18 Classe Individu.

Il est défini par son génotype <genotype> (tableau de réels représentant par exemple l'ensemble des poids des NN, ou la matrice de projection des données), ainsi que par

sa taille <lenghGen>. Il a une performance <currentFitness>, des parents <mate1> et <mate2>. Il contient également des informations nécessaires à la reproduction par ES, à savoir : les écarts-types <sigma> et les paramètres de mutation <tau0> et <taui>.

Un individu est capable de s'évaluer par la méthode *Evaluate*, de se reproduire par la méthode *CrossOverAndMutate* lorsqu'il évolue par GA, et par les méthodes *CrossES* (appliquée à <sigma>) et *MutateES* (appliquée à <génomotype>) lorsqu'il évolue par ES.

Cette classe est constituée des méthodes et des attributs communs à tous les types d'individus. Cependant, l'évaluation est propre au problème et au type d'individu. Ces méthodes sont définies dans des classes héritées que nous allons décrire brièvement. Désirant simuler des individus de type Réseau de neurones ou de type Base de projection, nous avons créé deux classes spécifiques : la classe **NeuralNet** (cf. Figure 19) et la classe **Projection** (cf. Figure 20).

Un individu de type réseau de neurones est défini par son génotype représentant les poids des connexions des neurones. Les attributs <numberOfLayers> et <sizeOfLayers> définissent son architecture en donnant le nombre de couches du réseau, ainsi que le nombre d'unités de chaque couche. L'attribut <maxNumIterations> fournit le nombre maximum d'itérations à effectuer pour l'apprentissage. La méthode *ComputeNopt* effectue l'apprentissage des données de parole, la méthode *RecoNopt* effectue une reconnaissance sur des données test. La méthode *Evaluate*, nécessaire pour l'évaluation de l'individu, propose d'exécuter en série *ComputeNopt* et *RecoNopt*. Le taux de reconnaissance que renvoie cette dernière méthode correspond à la performance de l'individu.

Un individu de type projection est défini par son génotype correspondant aux altérations apportées à la base de projection initiale. Les attributs <vector>, <nbVect> et <vectSize> définissent les coordonnées des vecteurs de la base de projection, leur nombre ainsi que leur taille. La méthode *ProjCorpus* projette le corpus de parole dans le nouvel espace défini par ces vecteurs, altérés par le génotype. L'évaluation, effectuée par la méthode *Evaluate*, consiste à déterminer le

taux de reconnaissance d'un corpus projeté par *ProjCorpus* obtenu par le système de reconnaissance <neuralNet>.

Classe NeuralNet : public Individu

```
{
    Attributs :

    int      numberOfLayers;    // nombre de couches du réseau
    int*     sizeOfLayers;      // nombre d'unités pour chaque couche
    int      maxNumIterations;  // nombre d'itérations maximum pour l'apprentissage

    Méthodes :

    float     ComputeNopt (int numPat, float* input, int* classe);
    // apprentissage de <numPat> patterns représentés par leurs entrées <input>
    // et leurs sorties désirées <classe>.
    // Renvoie le taux de reconnaissance obtenu sur ce corpus d'apprentissage.

    float     RecoNopt (int numPat, float* input, int* classe);
    // reconnaissance de <numPat> patterns représentés par leurs entrées <input>
    // et leurs sorties désirées <classe>.
    // Renvoie le taux de reconnaissance obtenu sur ce corpus de test.

    float     Evaluate ();
    // effectue l'apprentissage du corpus d'apprentissage
    // et renvoie le taux de reconnaissance du corpus de test.
}
```

Figure 19 Classe Réseau de neurones.

Classe Projection : public Individu

```
{
    Attributs :

    float*   vector;            // vecteurs de la base de projection
    int      nbVector;          // nombre de vecteurs définissant la base de projection
    int      vectorSize;        // taille de la base de projection
    NeuralNet* neuralNet;       // système de reconnaissance effectuant l'évaluation

    Méthodes :

    void      ProjCorpus (
        float*   input,        // données de parole
        float*   vector );     // vecteurs de projection
    // projette les données de parole <input> dans la base définie par <vector>+<génotype>

    float     Evaluate ();
    // effectue la reconnaissance des données fournies par ProjCorpus
}
```

Figure 20 Classe Projection.

- *Le module Evolution*

Ce module est défini par la classe **Evolution** de la Figure 21. Il gère l'ensemble des <numInd> individus du tableau pointé par <fInd>. Il gère le type d'évolution <evolType>, ainsi que le type de sélection associé <selType> et le type de mutation choisi par l'utilisateur <mutType>. Il connaît les probabilités de recombinaison et de mutation <pCross> et <pMut> concernant la méthode des GA, ainsi que le nombre d'occurrences de ces deux méthodes <numCro> et <numMut>. Il connaît le nombre de générations <numGenTot> à effectuer et le nombre déjà réalisé <numGen>. Les attributs concernant les statistiques de la population (somme des performances de la population <sumFitness>, performance de l'individu le plus faible <fitMin>, performance de l'individu le plus fort <fitMax> et performance moyenne de la population <fitAve>) sont gérés dans ce module. Le nombre de parents <numMates> sélectionnés pour la reproduction par ES est également un attribut de la classe Evolution. On retrouve aussi le pointeur <corpus> permettant d'accéder aux données de parole à traiter, ainsi que le fichier <resuFile> stockant tous les résultats obtenus par la population.

Concernant les méthodes proposées dans ce module, elles concernent l'évolution en elle-même, la gestion des résultats obtenus, ainsi que la gestion du corpus de parole. *Evolve* permet de créer la génération suivante à l'aide des méthodes de reproduction *ReproduceGA* et *ReproduceES*, de sélection *RouletteWheel* et *EsSel*, et d'évaluation *Evaluate*. *UpdateStat*, *DisplayResults*, ainsi que *StoreResults* permettent respectivement de mettre à jour les statistiques concernant les performances de la population (individu le plus ou le moins performant, moyenne de la population, nombre de mutations effectuées, etc.), d'afficher les résultats et de les enregistrer dans un fichier. Enfin, la méthode *GetCorpus* se charge de récupérer, s'il est nécessaire, les données de parole provenant du module environnement ou d'un corpus issu d'une base de données créée ultérieurement.

Class Evolution

{

Attributs :

Individu*	fInd;	// pointeur vers le tableau d'individus
int	numInd;	// nombre d'individus
SelectionType	selType;	// type de sélection des individus
MutationType	mutType;	// type de mutation des gènes
EvolutionType	evolType;	// méthode évolutionnaire choisie
float	pCross;	// probabilité de recombinaison (stratégie GA)
float	pMut;	// probabilité de mutation (stratégie GA)
int	numCro;	// nombre de recombinaisons (stratégie GA)
int	numMut;	// nombre de mutations (stratégie GA)
int	numGen;	// nombre de générations effectuées
int	numGenTot;	// nombre de générations à effectuer
float	sumFitness;	// somme des fitness de la population
float	fitMin;	// fitness de l'individu le plus faible
float	fitMax;	// fitness de l'individu le plus fort
float	fitAve;	// fitness moyenne de la population
int	numMates;	// nombre de parents sélectionnés (stratégie ES)
typePat*	corpus;	// pointeur vers le corpus de parole
FILE*	resuFile;	// fichier de résultat

Méthodes :

```

void      Evolve (fInd* next);    // crée la population <next> de la génération suivante.

void      Evaluate ();            // évalue l'ensemble des individus.

Individu* RouletteWheel();
// sélectionne un individu par la roulette wheel définie par Golberg.

Individu* EsSel(int numMate);
// renvoie un pointeur vers les <numMate> meilleurs individus.

void      ReproduceGa (
                Individu* mate1, Individu* mate2,
                Individu* child1, Individu* child2 );
// crée deux nouveaux individus <child1> et <child2>
// par reproduction GA des individus parents <mate1> et <mate2>.

void      ReproduceEs (
                Individu* mate, Individu* matecr1, Individu* matecr2,
                Individu* child );
// crée un nouvel individu <child> par mutation ES de l'individu <mate>
// et recombinaison ES des sigmas des individus <matecr1> et <matecr2>

void      UpdateStat (); // calcule les statistiques sur les résultats de la population

void      DisplayResults (); // affiche les résultats de la population à l'écran

void      StoreResults(FILE* f); // enregistre les résultats de la population dans le fichier <f>

void      GetCorpus ();
// récupère le corpus de données de parole issu du module environnement

```

};

Figure 21 Classe Evolution.

III.4 Réalisation des corpus

III.4.1 Les signaux de parole

Les données de parole présentées pour l'évaluation de nos systèmes sont considérées comme des caractéristiques des environnements acoustiques. Elles correspondent à des séries d'entités (voyelles ou digits) prononcées dans une pièce ayant un certain temps de réverbération, et dans laquelle il y a un bruit de fond plus ou moins prononcé.

Les voyelles sont issues d'une base de données de parole mono-locuteur enregistrée sans bruit suivant un protocole défini par l'Institut de la Communication Parlée (Robert-Ribes 95) et analysée dans (Kabré 95).

Les digits (chiffres de 0 à 9) sont issus de 29 locuteurs de la base de données BDSONS (Descout et al. 86).

Les bruits ont été sélectionnés selon leurs caractéristiques. Ainsi, nous avons choisi un bruit impulsionnel (porte qui claque (*PO*)), des bruits à bande étroite de différentes fréquences (alarme (*AL*), réveil (*RE*)), un bruit proche du bruit blanc (feu (*FE*)) et un bruit fortement non stationnaire (changement de fréquence de radio (*RA*)).

Chaque environnement présenté aux systèmes est alors défini par un triplet (Bruit, Temps de réverbération (en secondes), Rapport Signal à Bruit (en décibels)) et un ensemble d'entités (voyelles ou digits).

L'intelligibilité du signal est inversement proportionnelle au temps de réverbération et proportionnelle au rapport signal à bruit. C'est pourquoi un signal avec une réverbération forte et un rapport signal à bruit faible (par exemple (*FE* 0,7 -6)) est plus difficile à reconnaître qu'un signal comme (*RA* 0,4 20).

III.4.2 Les données numérisées

L'analyse acoustique de signaux bruités a été effectuée par des *Perceptual Linear Predictive coefficients* (PLP). (Hermansky 90) propose de travailler sur des

échantillons de signaux supposés stationnaires qu'on appelle des *fenêtres* dont la taille est d'environ 10 msec et d'associer 5 à 7 paramètres acoustiques par fenêtre.

Chaque voyelle dure environ 64 msec. Après l'analyse PLP, un alignement temporel est effectué afin d'extraire les paramètres les plus pertinents du signal et de garder une taille fixe pour le vecteur spectral de sortie, quelle que soit la longueur de son analyse. Pour une bonne représentation des voyelles, nous avons fait le choix empirique d'en extraire 7. Le corpus que nous avons utilisé comporte 500 (50*10) réalisations de voyelles prononcées par un seul locuteur.

Concernant les digits, les signaux ne sont pas forcément de durées identiques. Nous en avons extrait, pour chacun, 150 paramètres acoustiques. Le corpus comporte 8500 réalisations de digits prononcés par 29 locuteurs différents.

Les expérimentations des chapitres qui suivent seront effectuées à l'aide de cet outil de simulation.

CHAPITRE IV

ADAPTATION DU SYSTÈME : MODÈLE

Ce chapitre traite de l'application des algorithmes évolutionnaires à la reconnaissance de la parole. Nous abordons, dans un premier temps, les principes des réseaux de type perceptrons multi-couches. Nous présentons leur méthode d'apprentissage basée sur la rétro-propagation du gradient ainsi que l'impact de l'initialisation de leurs poids sur leur temps de convergence et leurs performances. Nous décrivons ensuite la manière de combiner ces réseaux avec des algorithmes évolutionnaires et les problèmes liés à la représentation de leur structure dans un génotype. Nous traitons ensuite de l'adaptation aux changements d'environnement et de la confrontation entre les méthodes Lamarckienne et Darwinienne. Enfin, nous présentons les principaux résultats que nous avons obtenus dans le cas de l'adaptation aux changements d'environnement acoustique.

Dans la nature, les organismes vivants ont la capacité d'apprendre pour s'adapter aux aléas de leur environnement et cette capacité d'adaptation leur permet de survivre et de se reproduire. Les algorithmes évolutionnaires (EA) sont directement dérivés des facultés de la nature à s'adapter à l'environnement en évoluant par sélection et reproduction. Les réseaux de neurones sont aussi une manière simplifiée de simuler les capacités des organismes vivants à s'adapter à leur environnement par apprentissage. Simplement parce que la nature fonctionne ainsi, et avec succès, elle a été source d'inspiration de beaucoup de travaux sur l'hybridation des réseaux de neurones (NN) avec des algorithmes évolutionnaires, en espérant que cette combinaison pourra permettre de résoudre des problèmes d'une manière plus efficace que les deux méthodes prises indépendamment.

Si l'on se base sur le principe que les EA effectuent une recherche dans l'espace global des solutions potentielles et que les NN se concentrent plutôt vers une recherche locale dans l'espace du voisinage (dont la solution trouvée, supposée optimale, ne l'est peut-être pas si l'on considère l'espace global), on peut supposer que ces deux méthodes sont complémentaires.

IV.1 Principes des NN

Les réseaux auxquels nous nous intéressons sont des perceptrons multi-couches avec apprentissage par rétro-propagation du gradient. En l'occurrence, nous nous sommes attachés à étudier des réseaux à deux couches fortement connectées permettant de reconnaître des sons. Nous présentons les principes de l'apprentissage dans la section suivante.

IV.1.1 Rétro-propagation du gradient

Mise en application dans les années 80 (Rumelhart et al. 86) (Le Cun 87), la méthode de rétro-propagation du gradient est la méthode la plus utilisée aujourd'hui dans les algorithmes d'apprentissage des réseaux multi-couches.

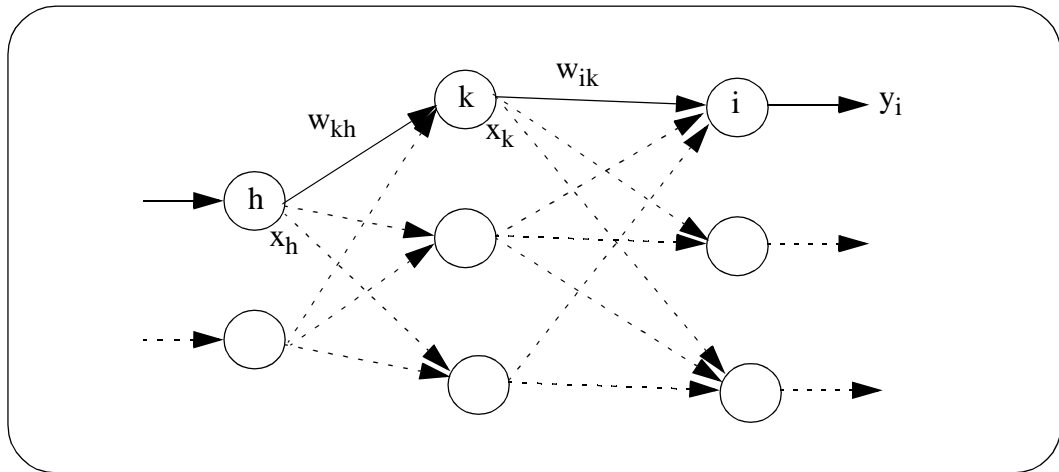


Figure 22 Extrait d'un réseau multicouches.

Le principe, détaillé dans (Hérault et Jutten 94), est basé sur la minimisation de l'erreur quadratique E calculée en fonction des n sorties désirées yd_i et des n sorties effectivement données y_i par le réseau :

$$E = \sum_{i=1}^n (y_i - yd_i)^2$$

Minimiser cette énergie revient alors à modifier les poids des connexions de la manière suivante :

$$\Delta W_{kh}^{(j)} = -a \cdot \delta_k^{(j)} \cdot y_h^{(j-1)}$$

avec a le gain d'adaptation, $\delta_k^{(j)}$ l'erreur du neurone k de la couche j et $y_h^{(j-1)}$ la sortie du neurone h de la couche $j-1$.

Pour les neurones de la dernière couche :

$$\delta_k^{(j)} = y_k^{(j)} - yd_k$$

Pour les neurones des couches internes :

$$\delta_k^{(j)} = \left[\sum_{i \in \text{couche}(j+1)} \delta_i^{(j+1)} \cdot w_{ik}^{(j+1)} \right] \cdot \sigma'(p_k^{(j)})$$

avec :

$$p_k^{(j)} = \sum_i w_{ki} \cdot x_i$$

où x_i correspond à la sortie du neurone i et :

$$\sigma(p_k^{(j)}) = \left(\frac{1}{1 + \exp(-(p_k^{(j)}))} \right)$$

et σ' sa dérivée.

IV.1.2 Initialisation des poids

L'initialisation des poids avant l'application de l'algorithme d'apprentissage par rétro-propagation du gradient est importante. Cette initialisation influe sur la vitesse de convergence du réseau mais aussi sur la qualité du réseau obtenu (Kolen et Pollack 90).

IV.1.2.1 Temps d'apprentissage

Le temps de convergence d'un réseau dépend de l'espace initial de représentation des poids et de l'espace final après convergence. En effet, plus les poids initiaux sont proches de leur valeur finale et plus la convergence est rapide. On peut distinguer deux types de méthodes dans la littérature (Thimm et Fiesler 94) : les méthodes d'initialisation aléatoire dans un intervalle choisi de manière adéquate et les méthodes basées sur des techniques non aléatoires.

(Burel 91) (Fahlman 88) proposent de choisir des poids de manière uniforme dans un intervalle dépendant des données à apprendre. (Fahlman 88) propose des intervalles variant de $[-4.0, 4.0]$ à $[-0.5, 0.5]$ selon les données d'apprentissage.

(Bottou 88) propose d'initialiser les poids dans un intervalle $[-a\sqrt{d_{in}}, a\sqrt{d_{in}}]$, où a est calculé de sorte que la variance des poids corresponde au point où la pente de la tangente de la fonction d'activation est maximum, et d_{in} le nombre d'unités de la couche précédente.

(Lee et al. 93) ont montré théoriquement que la saturation prématurée des neurones (variations des poids trop petites pour avoir un effet sur la sortie des neurones) augmentait avec les valeurs maximales des poids. Ils en concluent que des valeurs

initiales petites accroissent la vitesse de convergence mais aussi que des valeurs trop petites dégradent cette vitesse.

D'autres travaux sont basés sur des schémas d'initialisation et de méthode pseudo-inverse (Chen et Nutter 91) où seule la première couche est initialisée aléatoirement, les autres couches étant initialisées en fonction des sorties produites par la couche qui les précède. (Denoeux et Lengellé 93) proposent une méthode basée sur des données prototypes permettant d'accélérer la vitesse de convergence des réseaux.

IV.1.2.2 Qualité du réseau résultant

L'apprentissage par rétro-propagation du gradient peut être vu comme l'optimisation d'une fonction ayant les poids du réseau pour paramètres. Ceci mène à une convergence de la fonction vers un minimum local qui peut être aussi global. Lorsque c'est le cas, on peut considérer que l'apprentissage a été fait correctement. Lorsque ce n'est pas le cas, le réseau obtenu n'est pas optimal et ne permet pas généralement une bonne classification. Si dans la plupart des cas, la solution trouvée est très proche de la solution optimale et donc acceptable, il reste des cas où la solution est médiocre. Des études ont été faites (Wessel et Barnard 92) pour initialiser les poids de manière adéquate aux données à apprendre. (Denoeux et Lengellé 93) montrent des résultats plus robustes lorsque l'apprentissage est effectué avec des prototypes.

D'autres travaux, issus du domaine des EA, ont été menés concernant l'influence de l'initialisation des poids. Nous y faisons référence dans la section IV.3. En guise de préliminaire, nous présentons les différentes manières de traiter les NN par un EA et en particulier, les choix proposés pour représenter un NN dans un génotype.

IV.2 Représentations pour les EA

La représentation d'un réseau connexionniste dans un génotype a suscité un certain nombre de travaux. A un extrême, le réseau est codé littéralement dans le génotype, chaque poids est codé de manière précise (Miller et al. 89) (Belew et al. 90). Dans ce cas, les EA se réduisent à un problème d'optimisation multi-critères standard. A l'autre extrême, le codage se fait à un plus haut niveau en décrivant l'architecture du réseau à l'aide de règles (Harp et al. 89) (Mandischer 93) (Gruau et Whitley 93).

Le codage littéral est cependant la manière la plus directe de représenter le réseau et, n'ayant pas montré moins d'efficacité que d'autres codages quand il s'agit d'optimiser les poids et non la structure, c'est la représentation que nous utiliserons.

La question se pose alors sur la représentation de ces poids dans le génotype, autant dans le codage de réels représentant ces poids que dans l'ordre dans lequel ils apparaissent.

IV.2.1 Représentation des poids

Un choix doit être fait concernant le codage des poids (utilisés par la rétro-propagation en tant que valeurs réelles) dans le génotype. On est confronté ici au problème du choix entre une représentation binaire et une représentation réelle.

Si l'on choisit une représentation binaire, il est nécessaire de décoder les poids en valeurs réelles avant chaque opération de reconnaissance ou d'apprentissage des NN. (Belew et al. 90) (Whitley et Hanson 89) proposent différentes manières d'effectuer cette conversion.

Afin de limiter le nombre d'opérations (déjà coûteuses) à effectuer lors de l'apprentissage, et sachant qu'il n'existe pas de règles bien définies sur la manière optimale de coder les poids, nous avons fait le choix de représenter les poids des connexions de nos réseaux par des réels sans effectuer de transformations.

IV.2.2 Représentation de la structure

Si l'on désire utiliser le croisement de façon efficace lors de la reproduction des individus de type réseaux de neurones, il peut être intéressant de choisir une bonne représentation des différentes couches de connexions afin d'exploiter la structure du réseau et les interactions entre les neurones. Puisque deux allèles ont plus de chance d'être séparées si elles sont éloignées, il semble intéressant dans le cas des réseaux de regrouper les poids ayant une interaction forte (Belew et al. 90).

IV.2.3 Architecture du réseau

Considérons un réseau ayant trois couches de neurones avec n neurones en entrées, $m+1-n$ en couche cachée et $p+1-m$ en sortie. Soit w_{ij} le poids d'une connexion du neurone i vers le neurone j . La Figure 23 donne une représentation de ce réseau :

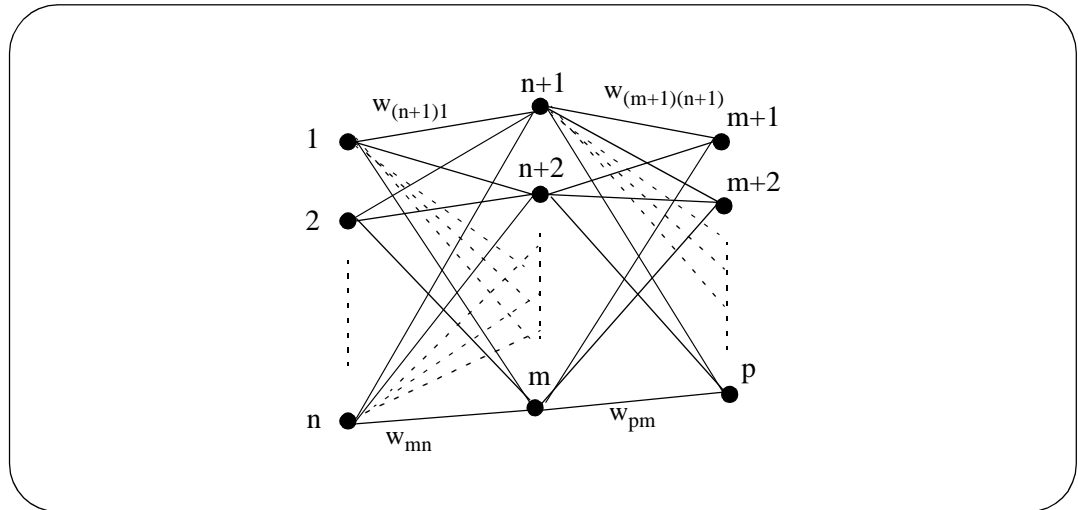


Figure 23 Architecture d'un perceptron multicouches.

Si l'on considère la configuration du réseau représenté dans la Figure 23, sa représentation dans le génotype est la suivante :

$$(w_{(n+1)1}, w_{(n+1)2}, \dots, w_{(n+1)n}, w_{(n+2)1}, w_{(n+2)2}, \dots, w_{(n+2)n}, \dots, w_{(m+1)m}, \dots, w_{pm})$$

Figure 24 Représentation génotypique du réseau.

C'est la représentation que nous utiliserons pour le génotype défini dans la classe **NeuralNet** de EVERA (voir Figure 19 page 56).

IV.3 Hybridation

Un grand nombre de travaux ont été effectués ces dix dernières années sur l'évolution des réseaux de neurones par EA. Trois grands problèmes sont abordés : trouver des poids optimaux (par EA seul ou par la combinaison des EA et de la rétro-propagation du gradient), trouver une structure optimale et ajuster les paramètres de

la fonction d'apprentissage. (Yao 95), (Whitley 95) et (Hancock 92b) proposent une grande bibliographie dans ce domaine.

IV.3.1 Valeur des poids

L'intérêt de l'hybridation des NN avec des EA se base sur l'observation qu'une recherche locale par une méthode de descente de gradient est bien complémentée par une recherche globale effectuée par des EA (Belew et al. 90). Les méthodes de descente de gradient sont sujettes à des variations de performances dues à la position initiale du NN menant parfois à une convergence vers des minima locaux. Les méthodes évolutionnaires, à l'opposé, assurent une recherche dans le domaine complet. Au fur et à mesure des générations, cet espace de recherche est affiné vers des sous-espaces potentiellement performants. Cependant, il est courant pour les EA de trouver une solution proche de la meilleure sans jamais l'atteindre.

La combinaison d'une recherche globale (EA) et d'une recherche locale (rétro-propagation) semble donner des résultats intéressants. Elle suggère que les EA trouvent une bonne configuration des poids initiaux des réseaux afin que la rétro-propagation s'effectue au mieux. En d'autres termes, l'utilisation des EA devrait permettre d'identifier les régions de l'espace qui mènent à une convergence vers une solution optimale. La performance associée à chaque individu est son taux de reconnaissance. Divers travaux traitent de cette approche (Belew et al. 90) (Skinner et Broughton 95).

IV.3.2 Topologie du réseau

Trouver l'architecture adéquate d'un réseau de neurones à un problème donné n'est pas une chose facile. Trouver le nombre optimal de couches cachées ainsi que le nombre de neurones dans chaque couche et leurs connexions se fait plus de manière empirique que par une méthode basée sur un fondement théorique. Depuis quelques années, la plupart des travaux combinant les EA avec les NN se focalisent sur la recherche de topologie optimale pour un problème donné (Yao et Liu 97) (Grönroos 98).

Dans le cadre de cette thèse, notre intérêt s'est porté sur l'hybridation NN/EA pour la recherche de poids optimaux. Dans la suite de ce chapitre, nous présentons les méthodes que nous avons utilisées pour l'adaptation des réseaux de neurones.

IV.4 Méthodes associées aux NN

Les réseaux ont la faculté d'intégrer de nouvelles connaissances à l'aide d'une méthode de recherche locale basée sur l'algorithme d'apprentissage par retro-propagation du gradient. Ils peuvent également évaluer leurs connaissances à l'aide d'un algorithme de reconnaissance et ainsi s'attribuer une performance.

Nous présentons, dans cette section, une manière d'évaluer les individus de type NN, ainsi que les manières de les recombinaisonner et muter.

IV.4.1 Apprentissage

L'algorithme d'apprentissage que nous avons utilisé est adapté de NOPT (Schalkwyk et Fanty 96) qui permet une convergence très rapide vers une solution de bonne qualité. De plus, NOPT utilise un critère d'arrêt basé sur la stagnation de l'erreur, permettant d'arrêter l'apprentissage dès que l'algorithme d'apprentissage n'est plus en mesure de trouver une direction de convergence faisant diminuer l'erreur de reconnaissance.

A l'issue de l'apprentissage, l'algorithme fournit un taux de reconnaissance correspondant au nombre d'entités reconnues divisé par le nombre total d'entités.

$$\text{Taux de reconnaissance} = \frac{\text{Nombre d'entités reconnues}}{\text{Nombre total d'entités}}$$

La fonction d'apprentissage comporte trois paramètres : le nombre d'entités à reconnaître, les caractéristiques associées à ces entités, et les sorties désirées pour chaque vecteur de caractéristiques. Elle fournit le taux de reconnaissance obtenu sur les données d'apprentissage et indique le nombre d'itérations nécessaires pour obtenir ce résultat.

IV.4.2 Reconnaissance

L'entité reconnue est celle qui correspond à la sortie ayant le potentiel maximum. Une entité est bien reconnue si la sortie procurée par le réseau est la même que la sortie désirée.

La fonction de reconnaissance comporte trois paramètres de même sorte que ceux de la fonction d'apprentissage : le nombre d'entités à reconnaître, les vecteurs de caractéristiques des entrées et les sorties désirées associées.

IV.4.3 Performance

Plusieurs choix sont possibles quant au choix de la fonction de performance. La performance utilisée est souvent l'erreur quadratique moyenne calculée lors de l'apprentissage (Miller et al. 89) (Belew et al. 90). (Korning 94) suggère de prendre une autre performance supposant que la performance basée sur l'erreur quadratique est souvent la cause du problème de permutation. (Chalmer 90) utilise l'erreur de reconnaissance du corpus d'apprentissage $(1 - \text{Erreur}) \times 100$.

Nous avons préféré prendre l'erreur de reconnaissance, erreur calculée à partir de données de test.

IV.4.4 Croisement

La plupart des travaux sur l'hybridation de systèmes connexionnistes et d'algorithmes évolutionnaires ont montré des résultats intéressants bien qu'utilisant rarement la recombinaison. La raison est le problème de permutation décrit ci-dessous.

L'exemple proposé dans la Figure 25 (Belew et al. 90) est une illustration des problèmes de permutation (Radcliffe 91). Le problème souligné ici est la quantité de réseaux équivalents que cette architecture peut fournir. Du fait que le réseau soit symétrique, les unités A, B, C et D sont interchangeables. Les génotypes résultants seront différents mais pas le comportement des réseaux associés. On parle alors de représentation dégénérée (Radcliffe 91).

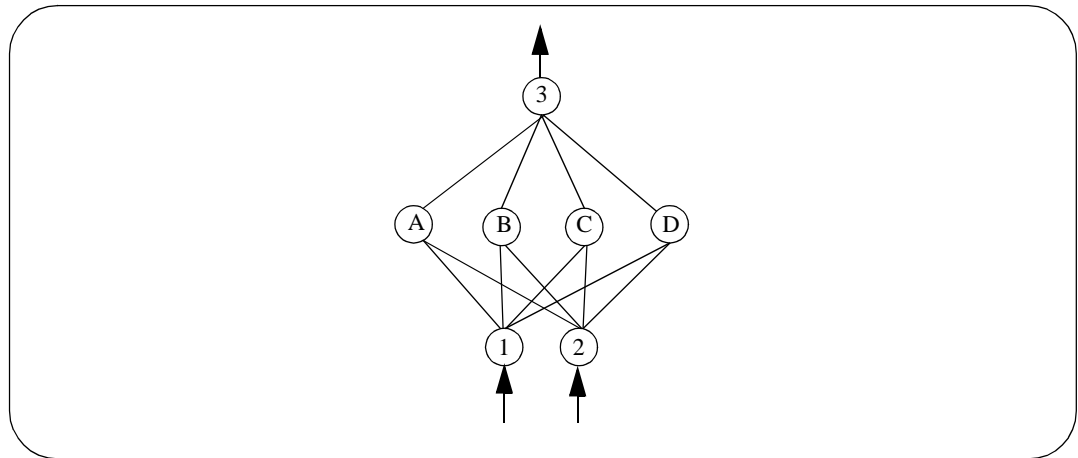


Figure 25 Illustration du problème de permutation.
Il est possible d'interchanger les places des unités de la couche cachée tout en gardant des réseaux équivalents.

Ce problème devient important à prendre en compte lorsque l'on désire utiliser un opérateur de croisement efficace. (Radcliffe 91) propose d'étudier la similarité de la structure des parents et (Hancock 92b) propose de généraliser à l'équivalence de deux réseaux afin de garder dans les enfants les parties que les parents ont en commun et qui semblent donner de bons résultats. Dans (Hancock 92a), il conclut cependant qu'un croisement uniforme classique est souvent aussi efficace qu'un croisement sophistiqué.

Beaucoup de chercheurs ont occulté le problème permutation en n'effectuant aucun croisement. (Menczer et Parisi 90) affirment cependant qu'un faible pourcentage de croisements donne des résultats meilleurs que pas de croisement du tout.

IV.4.5 Mutation

Puisque la représentation des poids que nous avons choisie est basée sur une représentation en nombres réels, nous devons choisir une mutation adaptée. La plupart des travaux utilisent une mutation gaussienne mais diverses variantes ont été proposées (Chellapilla et Fogel 97).

IV.5 Adaptation aux changements d'environnement

Un certain nombre de travaux ont été rapportés concernant l'adaptation aux changements d'environnement par EA. Par exemple, (Nolfi et Parisi 96) proposent une étude de l'adaptation d'un robot dans un environnement changeant contrôlé par une population de réseaux de neurones. (Cobb et Grefenstette 93) proposent d'étudier l'évolution d'une population cherchant l'optimum de fonctions complexes (combinaison de sinusoïdes et gaussiennes par exemple) et la capacité des GA à s'adapter, par exemple, à la translation de telles courbes.

Il résulte de ces travaux que la diversité des individus est certainement la clé pour une bonne adaptation à des environnements changeant rapidement. (Cobb et Grefenstette 93) proposent une comparaison des performances de différentes stratégies (mutation élevée (10%), mutation adaptative, immigration d'individus nouveaux). Il en découle que chacune de ces méthodes présente des avantages et des inconvénients dépendant de la manière dont change l'environnement. La mutation élevée génère une diversité importante, la mutation adaptative fait varier le taux de mutation selon la manière dont l'environnement change. Son taux est faible lorsque les changements sont faibles, élevé lors de brusques changements. L'immigration de nouveaux individus permet d'apporter à la population des individus très différents qui engendrent de la diversité. On peut également trouver dans la littérature des méthodes basées sur les principes thermodynamiques (Mori et al. 96). Par ailleurs, on attribue aux stratégies d'évolution des capacités à s'adapter efficacement aux changements d'environnement (Bäck 96). En effet, l'intégration des paramètres d'évolution dans le génotype permet de faire évoluer les taux de mutation lorsqu'il est nécessaire. On se rapproche ainsi de l'idée de mutation adaptative proposée par (Cobb et Grefenstette 93).

IV.6 Confrontation entre le Darwinisme et le Lamarckisme

Un certain nombre de travaux ont étudié l'influence de l'héritage sur l'évolution des populations (Turney et al. 96). Au début du siècle, deux écoles se confrontaient :

- Le Darwinisme se basait sur l'idée que seules les prédispositions des individus à apprendre sont transmises aux enfants. Les connaissances que les parents ont acquises ne sont alors pas transmises.
- Le Lamarckisme proposait que toutes les connaissances acquises par les parents soient directement transmises aux enfants.

Dans le contexte de ce débat, une sorte d'intermédiaire a été proposé par Baldwin (Turney 96), à savoir qu'il existe une forte interaction entre l'apprentissage et l'évolution. Le principal objectif pour les individus est d'intégrer, au moyen d'un apprentissage individuel, les informations fournies par l'environnement. Les individus ayant le mieux intégré les notions qui leur sont demandées, sont sélectionnés pour la reproduction. De ce fait, ils transmettent à leur descendance leur capacité à apprendre. Si l'on considère qu'un individu artificiel est d'autant plus apte à intégrer des connaissances que la configuration innée de son réseau est proche de celle après apprentissage, on peut alors considérer que la connaissance fait partie de ses gènes. Ainsi, le temps d'apprentissage devient de moins en moins long. Une fois que la population est constituée d'individus étant capables d'acquérir correctement ces notions, on peut considérer que celles-ci sont *assimilées* dans le génotype.

C'est ainsi que (Nolfi et al. 94) montrent que l'apprentissage guide l'évolution (les individus ayant le mieux appris se reproduisent plus souvent que les autres), mais aussi que l'évolution guide l'apprentissage (les individus sélectionnés ont de plus grandes capacités à apprendre et cette capacité s'améliore au cours des générations).

Ainsi, les connaissances instinctives des individus sont transmises et améliorées au cours de l'évolution sans que, et c'est la différence avec la théorie de Lamarck, le génotype ne soit directement affecté par l'apprentissage.

(Mayley 96) propose d'attribuer une pénalité aux individus ayant un apprentissage long. Il affirme aussi que l'assimilation de connaissances ne peut se faire que s'il existe une corrélation de voisinage, c'est-à-dire une corrélation entre la distance de deux génotypes et celle de leurs phénotypes associés.

(Whitley et al. 94) affirment que dans toutes les conditions de test qu'ils ont explorées, l'évolution de Lamarck est beaucoup plus rapide que celle de Darwin et qu'elle donne des résultats souvent plus performants. Ils pensent cependant que leurs

conditions de test sont trop simples pour montrer l'intérêt d'utiliser l'évolution de Darwin.

Concernant le problème qui nous intéresse plus particulièrement, l'adaptation aux changements d'environnement, (Sasaki et Tokoro 97) affirment que le Darwinisme est plus adapté aux changements d'environnement que le Lamarckisme, alors que pour un environnement statique, l'inverse est constaté.

Dans ce chapitre, nous proposons d'étudier les deux méthodes pour notre problème d'adaptation à un environnement acoustique.

IV.7 Présentation du Modèle

Le but de notre expérience est d'adapter le système de reconnaissance de manière automatique aux changements d'environnement acoustique (Spalanzani et Kabré 98).

Le modèle présenté ci-dessous illustre la méthode employée.

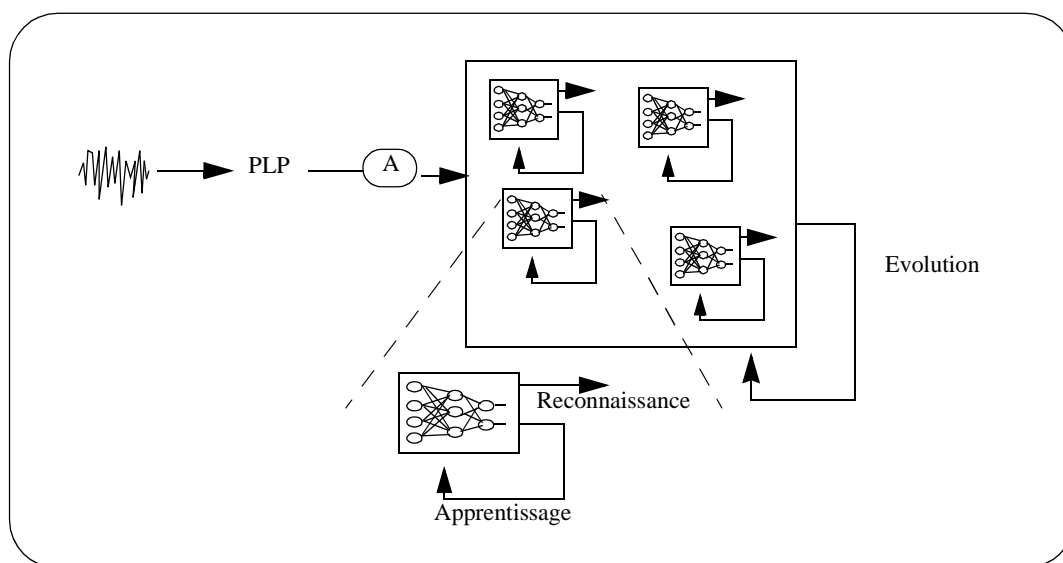


Figure 26 Modèle d'évolution de population de SRAPs.

Une population de SRAPs est créée dans un environnement virtuel acoustique (EAV). Ces SRAPs sont des modèles connexionnistes basés sur les réseaux de neurones de type feed-forward. L'initialisation de la population se fait en créant un certain nombre de SRAPs ayant une architecture identique mais des poids de valeurs différentes.

La population est alors confrontée à une série d'environnements présentés successivement de telle sorte qu'à chaque changement d'environnement, ses individus ne sont plus adaptés. Ils doivent alors effectuer, individuellement, un réapprentissage par rétro-propagation. Les individus résultants les mieux adaptés se reproduiront par algorithmes évolutionnaires. La Figure 27 présente l'algorithme associé au modèle d'évolution.

1. **Initialisation** de la population de SRAPs
2. Si la simulation n'est pas terminée, **changement** de EAV sinon aller à 6
3. **Apprentissage** des SRAPs
4. **Evaluation, Selection and Reproduction** des SRAPs
5. Si l'adaptation n'est pas terminée aller à 2 sinon aller à 3
6. Fin

Figure 27 Algorithme d'évolution des SRAPs.
Adaptation aux changements d'environnements.

Dans le chapitre suivant, nous allons décrire les expériences que nous avons effectuées en utilisant ce modèle.

CHAPITRE V

ADAPTATION DU SYSTÈME : EXPÉRIMENTATIONS

Ce chapitre est consacré à l'application des méthodes présentées dans le chapitre précédent. Nous proposons d'étudier l'apport des EA dans l'amélioration de la qualité des performances des systèmes de reconnaissance soumis à différents changements d'environnement. Nous étudions l'influence de l'architecture des réseaux de neurones, ainsi que celle du choix des paramètres associées aux méthodes évolutionnaires.

Dans ce chapitre, nous allons explorer de multiples manières de faire évoluer des populations de réseaux de neurones spécialisés dans la reconnaissance de parole en environnement changeant. Afin d'avoir une vision globale de l'ensemble des expériences que nous avons effectuées, nous proposons dans la section suivante le plan d'expérience.

V.1 Plan d'expériences

Dans les expériences suivantes, nous proposons d'étudier le comportement d'une population évoluant dans une série de 10 environnements. Chaque environnement est constitué de 160 voyelles pour l'apprentissage et de 40 pour le test.

Dans un premier temps, nous déterminons une architecture adéquate pour le type de problèmes que nous étudions.

Dans un second temps, nous étudions l'influence des paramètres de mutation, de croisement, ainsi que la taille de la population. Pour cela, nous faisons varier le type de mutation, les taux de mutation et de croisement, ainsi que le nombre d'individus dans la population.

Nous procédons ensuite à l'étude de deux types d'héritage : l'évolution selon Darwin et celle selon Lamarck. Nous présentons les performances (en terme de taux de reconnaissance et temps d'apprentissage) d'une population évoluant selon les paramètres que nous avons jugés comme les plus adéquats lors de l'étude précédemment décrite.

Puis nous présenterons la comparaison des performances des deux stratégies évolutionnaires que nous avons choisies : les GA et les ES.

Enfin, nous proposons de comparer la méthode la meilleure trouvée à différentes manières d'appréhender l'apprentissage d'un corpus variant au cours du temps. Comme nous l'avons déjà précisé, les systèmes de reconnaissance sont soumis à de multiples changements de conditions acoustiques lors de leur utilisation. Afin de les rendre robustes, il est possible de les entraîner sur des corpus contenant des données provenant de diverses conditions acoustiques. Le système de reconnaissance est alors

adapté à divers environnements. Lorsqu'un environnement nouveau apparaît, il est possible de refaire un apprentissage du système sur les données nouvelles ainsi que sur celles qu'il a déjà apprises. Ainsi, nous comparons les performances des GA avec :

- un réseau unique ayant à s'adapter, par apprentissage seulement, au cours des différents environnements acoustiques. Nous parlons alors d'*apprentissage local*.
- un réseau unique ayant à intégrer, à chaque changement d'environnement, les nouvelles informations à celles déjà intégrées. Nous parlons alors d'*apprentissage incrémental*.
- un réseau unique ayant été entraîné sur tous les environnements qui lui sont présentés. Nous parlons alors d'*apprentissage global*.
- une population de réseaux ayant à s'adapter, par apprentissage seulement, au cours des différents environnements acoustiques. L'expérience se différencie de celle sur l'*apprentissage local* de par la non unicité des réseaux. Nous parlons alors de *population de réseaux*.

Nous présentons alors une étude de l'influence de l'ordre dans lequel les environnements sont présentés.

Nous étudions ensuite l'évolution des performances des GA dans le temps. Nous présentons les performances obtenues par les GA et les populations de réseaux dans un série de 50 environnements.

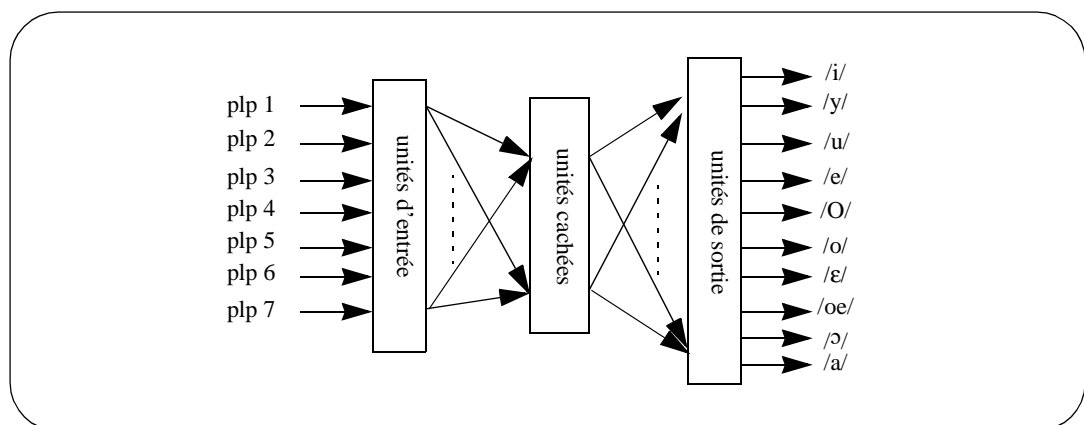
Enfin, nous augmentons la complexité des environnements acoustiques en présentant aux populations des séries de digits dans différents environnements.

Table 1 Plan d'expériences global pour l'adaptation des NN.

Expériences	Nb Environnements	Nb patterns / environnement	type de patterns
Détermination de l'architecture du réseau	1	200	Voyelles
Influence des paramètres d'évolution	10	200	Voyelles
Détermination de la structure du réseau	10	200	Voyelles
Comparaison de différents types d'apprentissage	10	200	Voyelles
Influence de l'ordre de présentation des environnements	10	200	Voyelles
Robustesse dans le temps	50	200	Voyelles
Robustesse dans la complexité	10	600	Digits

V.2 Détermination de l'architecture des réseaux

Les réseaux de neurones que nous utilisons sont spécialisés dans la reconnaissance de voyelles ou digits. Leur architecture est fixe et, dans le cas de la reconnaissance de voyelles, peut être schématisée par Figure 28.

**Figure 28** Individu spécialisé dans la reconnaissance de voyelles.

Le nombre d'unités d'entrée est fixé à 7 (1 entrée par coefficient PLP), celui de sortie à 10 (1 sortie par voyelle). Seul le nombre d'unités en couche cachée est indéterminé. Afin de déterminer une architecture adéquate, nous effectuons une série d'apprentissages en faisant varier le nombre d'unités (neurones) de la couche cachée. Ceci nous permettra d'éviter tout effet de sur-apprentissage et d'architecture non adéquate.

Tous les apprentissages sont effectués jusqu'à la convergence du réseau. La Figure 29 récapitule les taux de reconnaissance résultant d'un apprentissage de 160 voyelles. La phase de test comporte 40 voyelles à reconnaître. 9 simulations ont été effectuées par architecture.

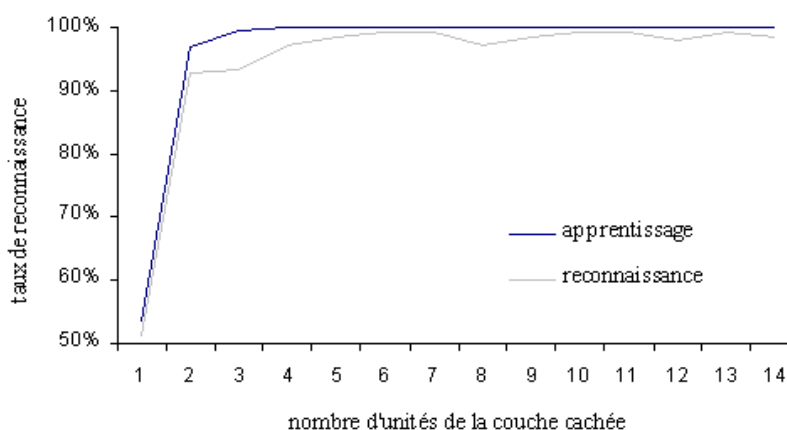


Figure 29 Influence de l'architecture sur les performances du système pour 200 voyelles. Les données proviennent d'un seul environnement acoustique et d'un seul locuteur. 160 sont utilisées pour l'apprentissage, 40 pour le test. Les taux de reconnaissances reportés sont ceux obtenus sur le corpus d'apprentissage (courbe apprentissage) et celui de test (courbe reconnaissance).

La tâche à apprendre est simple (les voyelles) et les taux de reconnaissance dépassent rapidement 97% (à partir de 5 unités). Ainsi, 6 unités en couche cachée fournissent un taux de reconnaissance en apprentissage ou en test de 98,7%. Une architecture plus complexe fournit des résultats équivalents mais il n'est pas nécessaire d'ajouter de la complexité dans l'apprentissage. Afin d'écarter toute ambiguïté sur la possibilité d'apprentissage par coeur, nous avons vérifié que les performances ne chutent pas au cours de l'apprentissage. La Figure 30 récapitule les taux de reconnaissance sur un corpus de test au cours de l'apprentissage. Ainsi, on remarque

que 300 itérations (cycles d'apprentissage) sont nécessaires pour une bonne convergence du réseau et qu'aucune baisse de performance n'apparaît au-delà.

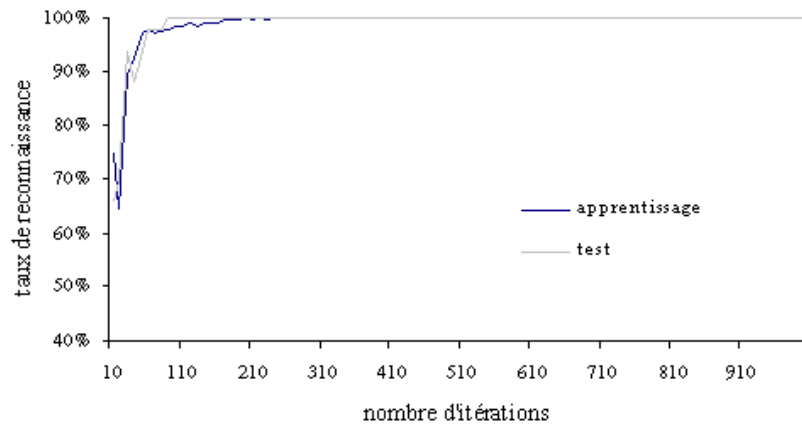


Figure 30 Evolution du taux de reconnaissance lors de l'apprentissage de 200 voyelles. 160 voyelles sont utilisées pour l'apprentissage, 40 pour le test. Les taux de reconnaissances reportés sont ceux obtenus sur le corpus d'apprentissage (courbe apprentissage) et celui de test (courbe test).

Suite aux résultats obtenus, les expériences qui suivent utilisent des individus réseaux ayant une architecture fixe de 7 unités en entrée, 6 cachées et 10 en sortie. Il est maintenant possible d'étudier l'influence des paramètres liés aux EA.

V.3 Influence des paramètres

Il est nécessaire d'effectuer des choix concernant la manière de faire évoluer les individus. Partant d'une méthode basée sur les GA, nous désirons déterminer quelle mutation et quel croisement choisir, quels taux (fréquences d'agissement) leur associer. Nous devons choisir une taille de population, la manière de passer les connaissances d'une génération à l'autre (héritage). Enfin, nous explorons une autre méthode évolutionnaire, à savoir les ES.

Tout d'abord, il faut décider des types de mutation et de croisement, ainsi que de leurs taux. La littérature dans le domaine donne une idée de l'ordre de grandeur des taux de mutation ou de croisement. En effet, celui de la mutation est rarement au-dessus de 10%, celui du croisement souvent supérieur à 60%. Nous allons fixer le

taux de croisement à 60% et celui de mutation à 10%. Par la suite, nous étudierons l'influence de ces taux pour quelques valeurs.

Nous étudions l'influence du type de mutation, à savoir :

- une mutation suivant une loi normale avec différents écarts-types,
- une mutation additive,
- une mutation multiplicative.

Les tests sont effectués avec une probabilité de mutation de 10%, un croisement de 60% et un héritage Lamarckien.

Nous déterminons, ensuite, le taux de mutation (multiplicative) donnant les meilleures performances.

Puis, nous présentons l'influence du taux de croisement sachant que la mutation est multiplicative et opère avec un taux de 10%.

L'influence de la taille de la population est alors étudiée sachant un croisement de 60% et une mutation multiplicative de 10%.

Nous comparons alors les deux méthodes d'héritage (Lamarckienne et Darwinienne) ainsi que les méthodes évolutionnaires ES et GA.

La série d'expériences est récapitulée dans la Table 2.

Table 2 Plan d'expériences sur l'influence des paramètres de l'évolution.

Expérience	Type Stratégie	Type de mutation	Taux de mutation	Taux de croisement	Taille population	Type héritage
Influence Type mutation V.3.1	GA	Norm (0 0,5)	10%	60%	20	Lamarck
	GA	Norm (0 0,1)	10%	60%	20	Lamarck
	GA	Norm (0 0,2)	10%	60%	20	Lamarck
	GA	Norm (0 1)	10%	60%	20	Lamarck
	GA	Norm (0 2)	10%	60%	20	Lamarck
	GA	Norm (0 3)	10%	60%	20	Lamarck
	GA	Mult	10%	60%	20	Lamarck
	GA	Add	10%	60%	20	Lamarck
Influence Taux de mutation V.3.2	GA	Mult	0%	60%	20	Lamarck
	GA	Mult	3%	60%	20	Lamarck
	GA	Mult	6%	60%	20	Lamarck
	GA	Mult	10%	60%	20	Lamarck
	GA	Mult	15%	60%	20	Lamarck
Influence Taux de croisement V.3.3	GA	Mult	10%	0%	20	Lamarck
	GA	Mult	10%	30%	20	Lamarck
	GA	Mult	10%	60%	20	Lamarck
	GA	Mult	10%	80%	20	Lamarck
Influence Taille population V.3.4	GA	Mult	10%	60%	2	Lamarck
	GA	Mult	10%	60%	10	Lamarck
	GA	Mult	10%	60%	20	Lamarck
	GA	Mult	10%	60%	50	Lamarck
	GA	Mult	10%	60%	100	Lamarck
	GA	Mult	10%	60%	200	Lamarck
Influence Héritage V.4	GA	Mult	10%	60%	20	Lamarck
	GA	Mult	10%	60%	20	Darwin
Influence Stratégie V.5	GA	Mult	10%	60%	20	Lamarck
	ES	Mult	10%	60%	20	Lamarck

V.3.1 Influence du type de mutation

Concernant la mutation, il est nécessaire de prendre une mutation adaptée aux valeurs réelles que nous manipulons. Ainsi, comme nous l'avons présenté dans la description de notre système, nous proposons 3 types de mutation, à savoir la mutation gaussienne qui est la plus utilisée pour les valeurs réelles, la mutation multiplicative qui permet de remettre les poids des connexions mutées à des valeurs proches de

celles données à l'initialisation du réseau, et une mutation additive qui est proche de la mutation gaussienne sans respecter la loi normale.

Il s'agit maintenant de comparer ces 3 types de mutation. Nous présentons les performances, sur une moyenne de 9 simulations, de populations de 20 individus. Pour chaque environnement (200 voyelles), la population a 100 générations pour s'adapter selon l'évolution Lamarckienne.

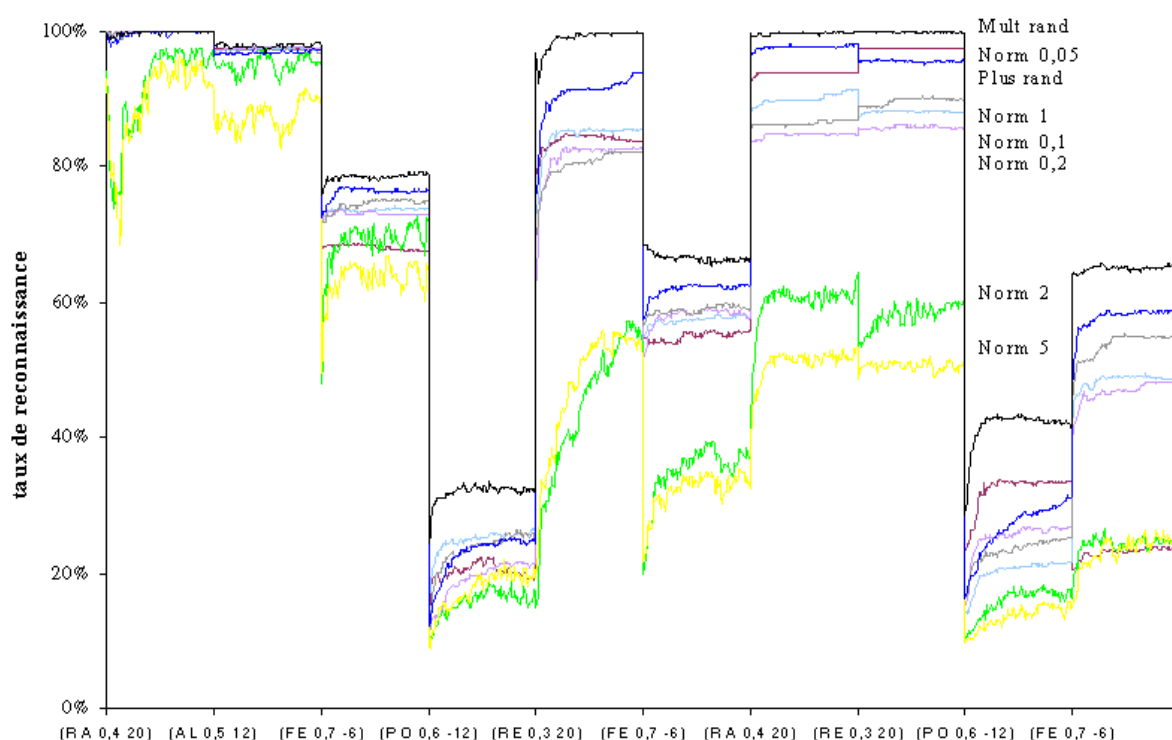


Figure 31 Influence du type de mutation.

Comparaison des mutations normales d'écart-type 0.05, 0.1, 0.2, 1, 2 et 5, ainsi que des mutations additives et multiplicatives consistant respectivement à ajouter ou à multiplier une valeur comprise entre -1 et 1. Les courbes représentent les performances moyennes des populations. Taux de croisement à 60%, taille de la population à 20.

Table 3 Récapitulatif des performances selon le type de mutation.

Performances	Norm 0,05	Norm 0,1	Norm 0,2	Norm 1	Norm 2	Norm 5	Add	Mult
minimum	68,85%	66,71%	65,36%	66,93%	38,59%	35,04%	69,83%	73,78%
maximum	71,03%	69,80%	68,55%	70,15%	56,24%	52,68%	74,80%	80,87%
moyenne	70,18%	68,51%	67,25%	68,79%	51,04%	47,81%	72,67%	78,03%

Les résultats montrent que la mutation la plus efficace est la mutation multiplicative. La littérature indique que les poids d'un réseau doivent généralement être initialisés à des valeurs proches de 0 pour que l'apprentissage ait des chances de se faire au mieux. C'est principalement ce que la mutation multiplicative se charge de faire.

Nous avons étudié plusieurs mutations gaussiennes avec des écarts-types différents. Plus l'écart-type est grand et plus la mutation génère des configurations de poids radicalement différentes. Nous avons constaté, lors de nos simulations, que ces poids peuvent prendre des valeurs relativement élevées (compris généralement dans l'intervalle $[-15, 15]$), c'est pourquoi nous nous sommes permis d'aller jusqu'à une mutation gaussienne d'écart-type 5. Cependant, les résultats sont assez mauvais puisqu'en moyenne, avec ce type de mutation, le meilleur individu obtient 52,68% de reconnaissance alors que la mutation multiplicative obtient 80,87%. La mutation additive obtient, quant à elle, des performances se situant entre les deux autres mutations et obtient 74,80% de reconnaissance pour le meilleur individu.

Nous avons donc fait le choix de garder cette mutation multiplicative. Pour la suite, les simulations présentées sont effectuées avec cette mutation.

V.3.2 Influence du taux de mutation

La Figure 32 montre l'influence du taux de mutation sur le comportement des populations de taille 20 soumis à un croisement de 60%.

Les résultats présentés montrent que le taux de mutation n'a pas beaucoup d'influence si l'on compare les taux prenant les valeurs de 3%, 6%, 10% et 15%. Cependant, comme le montrent les résultats obtenus sans mutation (taux égal à 0), il est nécessaire d'altérer les poids des réseaux.

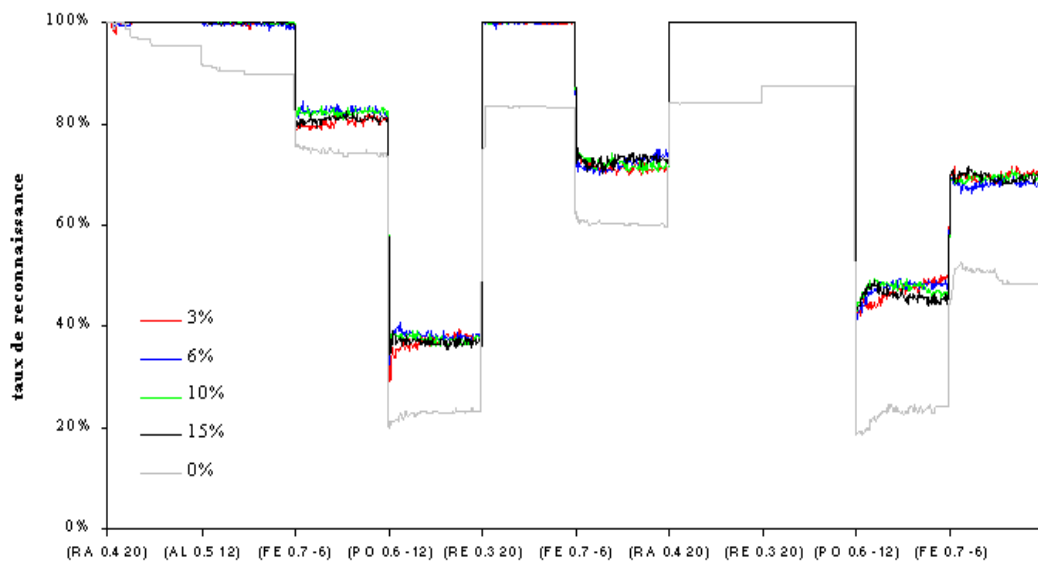


Figure 32 Influence du taux de mutation pour la mutation multiplicative. Comparaison des performances des populations soumises à des mutations de taux 0% (pas de mutation), 3%, 6%, 10% et 15%. Les courbes représentent les performances moyennes des populations. Taux de croisement à 60% et taille de la population à 20.

Nous gardons alors un taux de mutation à 3% pour les expériences suivantes. Il s'agit désormais d'étudier l'influence du croisement.

V.3.3 Influence du taux de croisement

Concernant l'opérateur de recombinaison, un seul type a été implémenté pour cette expérience. Les simulations sont effectuées sur une population de 20 individus soumis à une mutation de 3%. La Figure 33 montre que le choix du taux de croisement n'influe pas sensiblement sur les performances des individus. En effet, les résultats produits par les taux de 0%, 30%, 60% et 80% sont quasiment identiques.

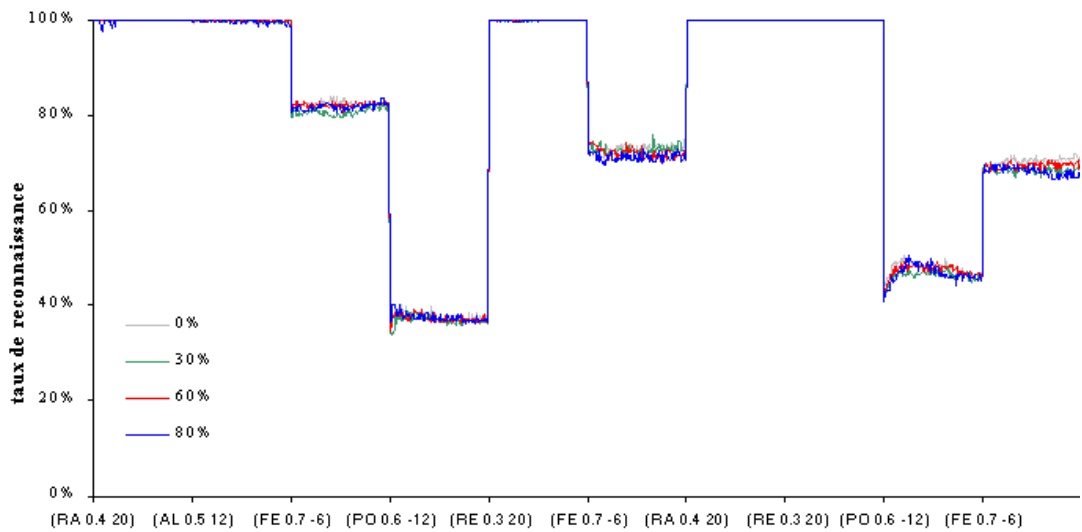


Figure 33 Influence du taux de croisement.

Comparaison des performances des populations soumises à un croisement simple de taux 0% (pas de croisement), 30%, 60% et 80%. Les courbes représentent les performances moyennes des populations. Taux de mutation à 3% et taille de la population à 20.

V.3.4 Influence de la taille de la population

Enfin, toutes ces simulations sont effectuées avec 20 individus. La Figure 34 présente le comportement de populations de tailles différentes. Le taux de mutation est fixé à 3% et celui de croisement à 60%

Nous pouvons constater que les performances des populations augmentent avec leur taille. En effet, une population de 2 individus obtient des performances de 78,16% pour le meilleur individu, alors qu'une population de 200 individus obtient des performances de 82,57% pour le meilleur individu. Cependant, compte-tenu du temps CPU nécessaire pour faire évoluer de telles populations, nous ne pouvons pas nous permettre de simuler de grandes populations. Nous avons choisi de prendre une population de 20 individus car ses performances sont raisonnables (80,87% pour le meilleur individu), tout en gardant des temps de simulations corrects.

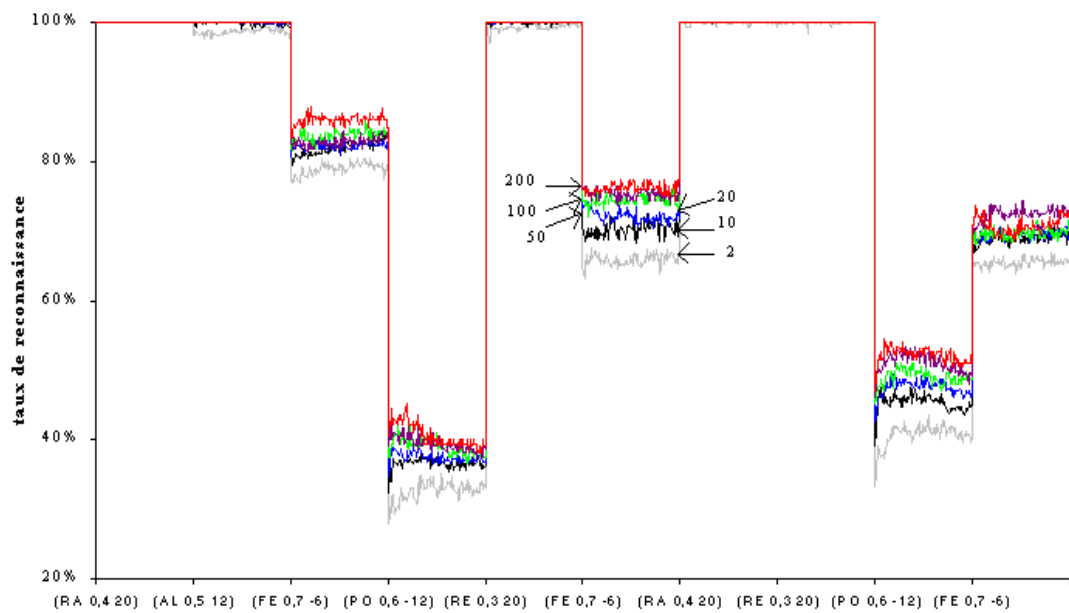


Figure 34 Influence de la taille de la population.
 Comparaison des performances des population selon le nombre d'individus composant chacune d'elle.
 Les performances moyennes des populations sont reportée. Taux de mutation à 3%, taux de
 croisement à 60%.

La Table 4 récapitule les résultats obtenus pour cette expérience.

Table 4 Récapitulatif des performances selon la taille de la population.

Taux de reconnaissance	2	10	20	50	100	200
max	78,16%	80,27%	80,87%	81,53%	82,10%	82,57%
min	76,19%	74,77%	73,78%	71,53%	70,35%	66,92%
moyenne	77,21%	77,82%	78,03%	77,88%	78,27%	77,97%

V.4 Influence de l'héritage

Jusqu'à présent, nous avons considéré que le passage des connaissances d'une génération à une autre se faisait de manière fidèle, à quelques mutations près. On se trouvait ainsi dans un type d'évolution Lamarckien. Comme nous l'avons présenté en IV.6, il existe deux manières de traiter l'héritage des connaissances d'une génération à une autre.

Le but des expériences qui vont suivre est de déterminer la méthode la plus efficace pour notre problème d'adaptation à l'environnement. Puisque les avis sont partagés concernant le type d'héritage à utiliser, nous proposons de tester les performances de nos populations en terme de qualité et d'efficacité des individus. La qualité correspond aux performances de reconnaissance, l'efficacité à la durée d'apprentissage nécessaire pour arriver à ces performances. Dans le cadre de nos expérimentations, ceci consiste à étudier le taux de reconnaissance de notre population ainsi que le nombre d'itérations nécessaires à chaque individu pour que son apprentissage soit optimal (c'est-à-dire que le réseau ait convergé). Les résultats sont une moyenne sur 9 simulations.

V.4.1 Qualité des performances

On peut constater que les performances des individus sont sensiblement équivalentes en moyenne. Les résultats des performances chiffrées sont présentées dans la Table 5. En moyenne, la population évoluant selon la méthode de Lamarck obtient 78% de reconnaissance alors que selon la méthode de Darwin, elle obtient 76%. Concernant le meilleur individu, seulement 1% d'amélioration est constaté entre l'héritage Lamarckien et l'héritage Darwinien, puisqu'en moyenne (sur les 10 environnements, soit 1000 générations), le meilleur individu Darwinien obtient 80% alors que le Lamarckien 81%. Sur l'allure générale des courbes, on peut constater aussi que les résultats fournis par l'évolution Lamarckienne sont plus stables et semblent chuter moins facilement que ceux de la population Darwinienne.

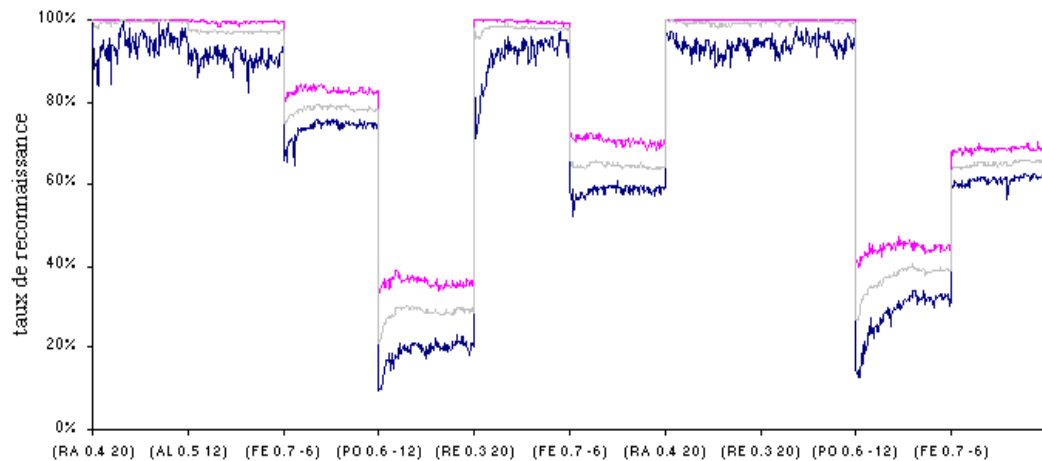


Figure 35 Evolution Darwinienne et taux de reconnaissance.

Taux de reconnaissance d'une population de SRAPs évoluant dans une série d'environnements acoustiques par la méthode de Darwin. Les courbes présentent les performances du meilleur individu, de la moyenne de la population ainsi que du moins performant.

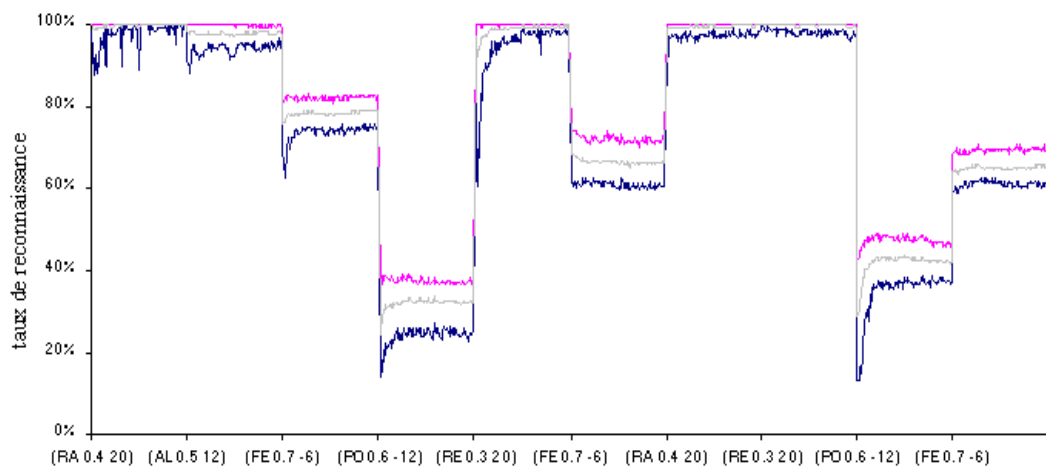


Figure 36 Evolution Lamarckienne et taux de reconnaissance.

Taux de reconnaissance d'une population de SRAPs évoluant dans une série d'environnements acoustiques par la méthode de Lamarck. Les courbes présentent les performances du meilleur individu, de la moyenne de la population ainsi que du moins performant.

Table 5 Récapitulatif des performances selon le type d'héritage choisi.

Taux de reconnaissance	Minimum	Maximum	Moyenne
Evolution Lamarckienne	73,8 %	80,9 %	78 %
Evolution Darwinienne	70,5 %	80,1 %	76,6 %

V.4.2 Efficacité de l'évolution

Concernant maintenant l'efficacité des populations dans la phase d'apprentissage, nous avons reporté, dans la Figure 37 et la Figure 38, le nombre d'itérations nécessaires pour une bonne convergence du réseau de chaque individu.

Bien qu'à chaque changement d'environnement, le nombre d'itérations nécessaires augmente de façon plus significative lors de l'évolution de Lamarck, au cours des générations, la baisse de ce nombre est plus importante et, en moyenne, le nombre d'itérations est plus faible. Il est intéressant de noter que le nombre d'itérations diminue tout au long de l'évolution Darwinienne, ce qui peut signifier qu'il y a bien assimilation des connaissances, et ceci sans avoir utilisé de pénalité comme le propose (Mayley 96).

Dans les deux types d'évolution, le nombre d'itérations nécessaires diminue au cours des générations. Une fois de plus, on peut tout de même constater que l'évolution de Lamarck est plus efficace que celle de Darwin. Comme l'indique la Table 6, il faut en moyenne pour un individu Darwinien 90 itérations pour apprendre correctement, alors qu'il ne faut à un individu Lamarckien que 70 itérations. On peut noter les différences entre les individus les meilleurs (54 pour l'évolution Darwinienne contre 33 pour l'évolution Lamarckienne) et les moins bons (144 pour l'évolution Darwinienne contre 113 pour l'évolution Lamarckienne).

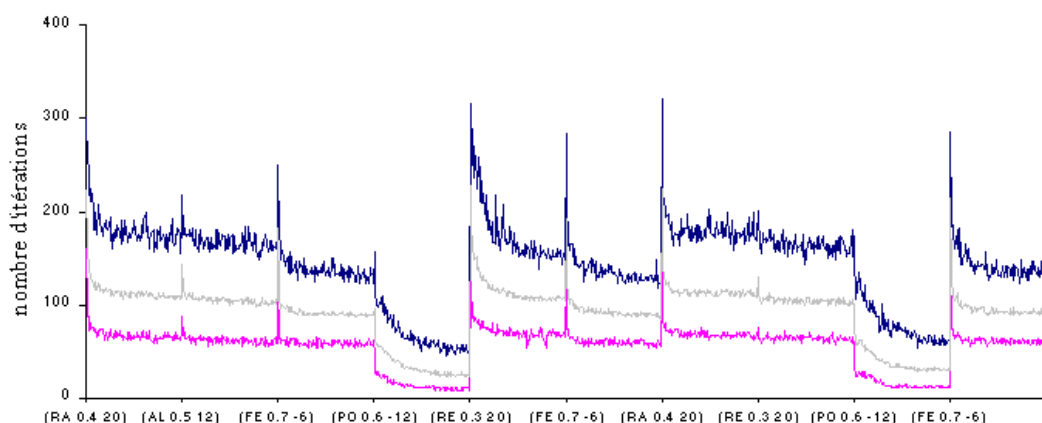


Figure 37 Evolution Darwinienne et nombre d'itérations.

Evolution du nombre d'itérations nécessaires pour l'apprentissage des individus dans différents environnements acoustiques. La population évolue selon la méthode Darwinienne. Les courbes présentent les performances du meilleur individu, de la moyenne de la population ainsi que du moins performant.

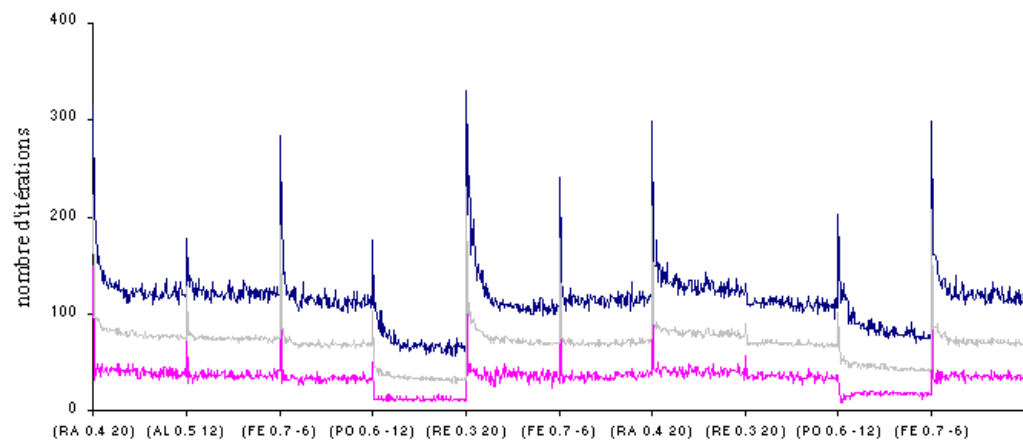


Figure 38 Evolution Lamarckienne et nombre d'itérations.

Evolution du nombre d'itérations nécessaires pour l'apprentissage des individus dans différents environnements acoustiques. La population évolue selon la méthode Lamarckienne. Les courbes présentent les performances du meilleur individu, de la moyenne de la population ainsi que du moins performant.

Table 6 Récapitulatif du temps d'apprentissage selon le type d'héritage.

Nombre d'itérations	Pire	Meilleur	Moyenne
Evolution Lamarckienne	113,6	33,1	68,2
Evolution Darwinienne	144	54,8	91

V.5 Influence de la stratégie

Les résultats présentés dans les sections précédentes ont été obtenus avec une stratégie basée sur les GA. Nous nous sommes cependant intéressés aux ES qui sont, de par leur capacité à faire évoluer leur taux de mutation, réputées pour être efficaces pour les changements d'environnement. Nous avons simulé une population d'individus évoluant selon la stratégie (5, 20)-ES et étudié son comportement sur une moyenne de 9 simulations. Les valeurs de τ et τ' (constantes associées à la mutation des écarts-type σ_i) sont initialisées à 0,07 et 0,21. La Figure 39 présente les résultats obtenus.

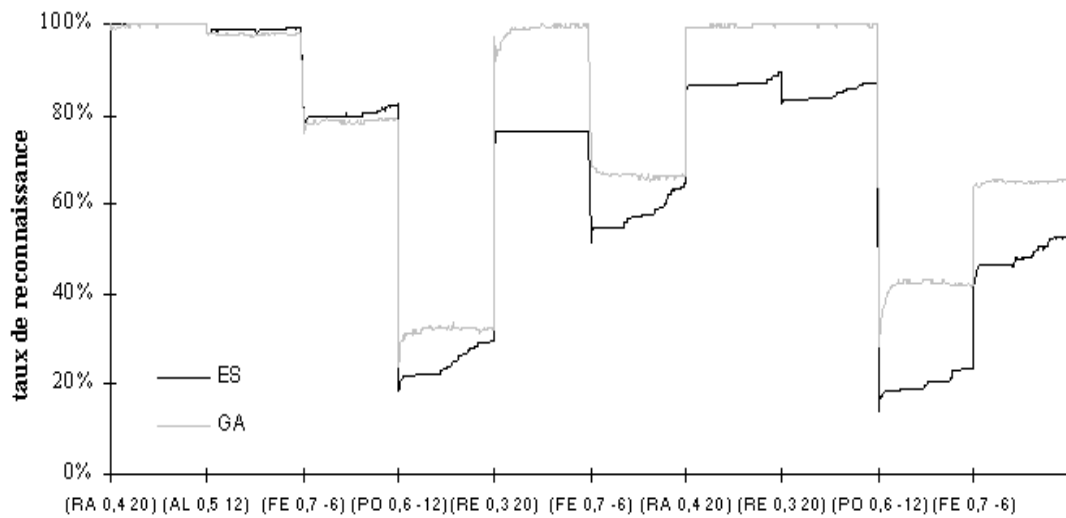


Figure 39 Comportement d'une population évoluant selon une (5, 20)-ES. A chaque changement d'environnement, les σ_i et τ sont ré-initialisés. Les performances moyennes de la population sont reportées.

Nous pouvons constater que les ES que nous avons utilisées donnent de moins bons résultats que les GA. Si, pour les premiers environnements, les ES sont très compétitives, à partir du quatrième environnement, elles semblent perdre de leur efficacité au cours des environnements qui suivent. Afin de comparer les résultats des ES avec ceux des GA, nous avons fixé la taille de la population à 20 individus. Cette taille est peut être trop petite pour permettre aux ES de garder une diversité suffisante lors des multiples changements d'environnement. De plus, comme nous avons pu le constater lors de l'étude de l'influence du type de mutation (Figure 31), la mutation gaussienne (utilisée par les ES) ne semble pas aussi efficace que la mutation multiplicative et ceci, quelque soit l'écart-type associé.

V.6 Différents types d'apprentissage

Alors que nous avons déterminé la méthode la plus efficace pour notre problème, il est intéressant de comparer ses performances à un ou plusieurs réseaux de neurones classiques. Nous proposons alors d'étudier le comportement de ces réseaux face à différentes manières de s'adapter à un nouvel environnement.

Lorsqu'un nouvel environnement se présente, le réseau peut s'adapter de manière efficace aux données qui lui sont présentées sans garder la mémoire sur les données présentées dans l'environnement précédent. C'est la méthode que nous appellerons *apprentissage local*.

Le réseau peut intégrer ces nouvelles données avec les connaissances qu'il a précédemment acquises. Ainsi, la base de données d'apprentissage s'accroît au cours du temps. Nous appelons cette méthode *apprentissage incrémental*.

Enfin, si l'on fait une analogie avec la méthode la plus utilisée pour la robustesse des SRAPs qui construit une base de données gigantesque tentant de répertorier le maximum de conditions acoustiques possibles, nous devons comparer notre méthode à un réseau ayant appris au préalable tous les corpus que nous allons lui présenter. Nous appelons cette démarche *apprentissage global*.

Pour les deux dernières stratégies (apprentissage incrémental et apprentissage global), il est nécessaire d'utiliser un réseau avec une taille plus grande que celle du réseau utilisé dans la première stratégie et toutes les stratégies évolutionnaires. En effet, si nous désirons que ce réseau intègre 10 environnements acoustiques (donc 1600 voyelles), sa taille doit être augmentée. Nous allons déterminer celle-ci dans la section suivante.

Table 7 Plan d'expériences pour la comparaison de différents types d'apprentissage.

Expérience	Nombre de réseaux	Type Apprentissage	Nombre de patterns appris / testés	Evolution
V.6.2	1	Local	160 / 40	Aucune
V.6.3	1	Incrémental	$(nb \text{ env. appris} + 1) * 160 / 40$	Aucune
V.6.4	1	Global	1600 / 40	Aucune
V.6.5	20	Local	160 / 40	Aucune
Comparaison	20	Local	160 / 40	GA

V.6.1 Détermination de l'architecture des réseaux

1600 voyelles prononcées par un seul locuteur dans 10 environnements différents ont servi à l'apprentissage d'un réseau. 400 voyelles provenant de ces mêmes

environnements ont servi au test. Nous avons reporté, dans la Figure 40, les performances de ces réseaux (sur une moyenne de 5 simulations) sur différents nombres d'unités en couche cachée.

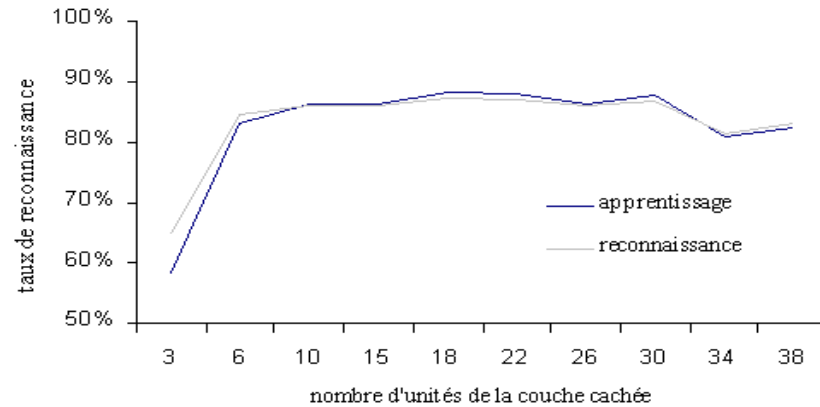


Figure 40 Influence de l'architecture sur les performances du système sur 2000 voyelles. Les voyelles sont sélectionnées dans 10 environnements acoustiques différents. 1600 sont utilisées pour l'apprentissage, 400 pour le test. Les taux de reconnaissances reportés sont ceux obtenus sur le corpus d'apprentissage (courbe apprentissage) et celui de test (courbe reconnaissance).

La première remarque concernant cette figure est qu'il y a très peu de différences entre les performances d'un réseau à 10 unités en couche cachée et celles d'un réseau à 30 unités. Au regard de ces résultats, nous avons été tentés de prendre un réseau à 10 unités en couche cachée mais les performances sont à la limite d'un plateau qui pourrait être gênant suivant les conditions d'utilisation de ce réseau. Nous avons, pour les expériences suivantes, testé un réseau à 15 unités en couche cachée. la Figure 41 montre l'évolution de ses performances au cours d'un apprentissage. On remarque que 300 itérations (cycles d'apprentissage) sont nécessaires pour une bonne convergence du réseau et qu'aucune baisse de performance n'apparaît.

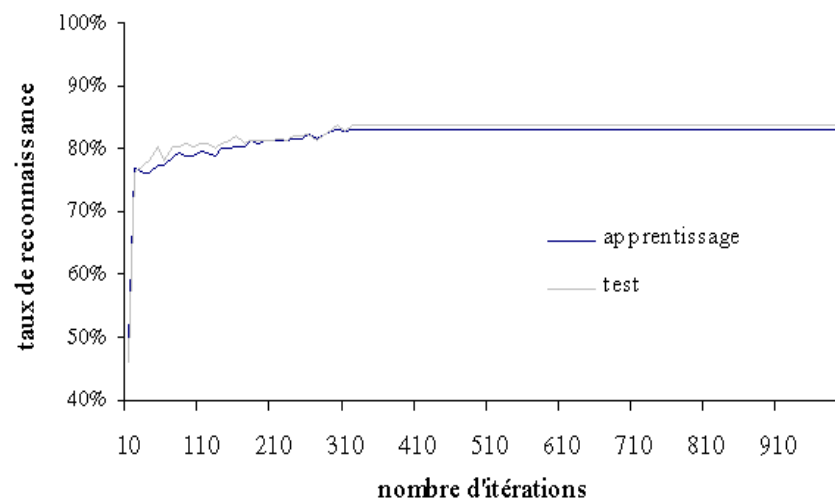


Figure 41 Evolution du taux de reconnaissance pour l'apprentissage de 10 environnements. 2000 voyelles sont sélectionnées dans l'ensemble de ces 10 environnements, 1600 sont utilisées pour l'apprentissage, 400 pour le test. Les taux de reconnaissances reportés sont ceux obtenus sur le corpus d'apprentissage (courbe apprentissage) et celui de test (courbe test).

Ayant fait le choix d'une architecture pour notre réseau, nous allons tester ses performances dans un environnement changeant. Les résultats présentés sont calculés sur une moyenne de 9 simulations.

V.6.2 Apprentissage local

La méthode à apprentissage local consiste à utiliser un réseau de neurones classique en lui permettant, à chaque changement d'environnement, de remettre à jour ses connaissances. Lors de ce nouvel apprentissage, le réseau part de sa configuration de poids qu'il a pour l'adapter afin d'être performant dans le nouvel environnement. Concernant les GA, les résultats présentés par la suite sont ceux du meilleur individu de la population.

La première constatation concernant les résultats présentés dans la Figure 42 est la supériorité des GA sur l'ensemble des environnements. En effet nous obtenons, moyennée sur les 10 environnements, une amélioration de 10,68% des GA par rapport aux performances des NN. De plus, dans les environnements très bruités (par exemple (PO 0,6 -12)), les GA augmentent le taux de reconnaissance jusqu'à 22,6%.

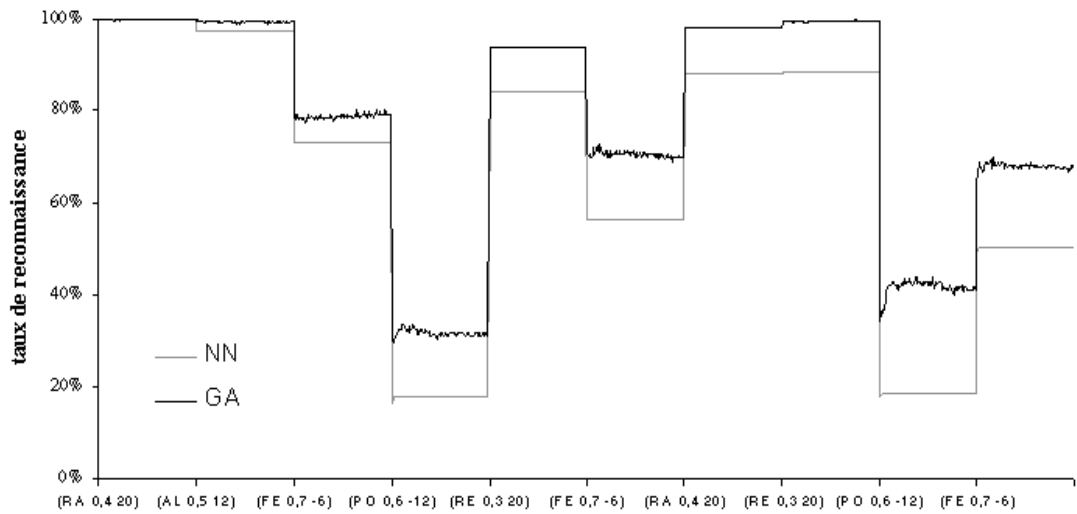


Figure 42 Comparaison des performances de la population GA et d'un NN classique.
Les performances du meilleur individu de la population GA sont reportées.

Concernant les taux de reconnaissance dans des environnements identiques (par exemple (FE 0,7 -6)), nous observons une baisse continue des performances de NN. Les environnements précédant celui qui est testé (c'est-à-dire le contexte) semblent donc jouer un rôle dans la dégradation. De plus, nous pouvons en déduire que les connaissances précédemment acquises sont perdues lorsqu'un nouvel environnement apparaît.

V.6.3 Apprentissage incrémental

A chaque nouvel environnement, le réseau a pour but d'agrandir sa base d'apprentissage avec les nouvelles données. Il effectue alors un apprentissage sur sa nouvelle base de données et teste ses performances dans l'environnement courant. Les GA gardent la même stratégie d'apprentissage que dans les expériences précédentes. Les résultats sont présentés dans la Figure 43.

Les performances du NN incrémental sont un peu meilleures que celles du NN local, elles atteignent 72,28% en moyenne. Cependant, on peut constater de très faibles taux de reconnaissance pour les environnements les plus bruités. Dans ce cas, les GA améliorent les performances de 23,3% et passent par exemple de 20,28% à 43,61% de reconnaissance (environnement (PO 0,6 -12)). De plus, les écarts entre les

performances d'un environnement à un autre sont les plus élevées de toutes les stratégies d'apprentissage utilisées.

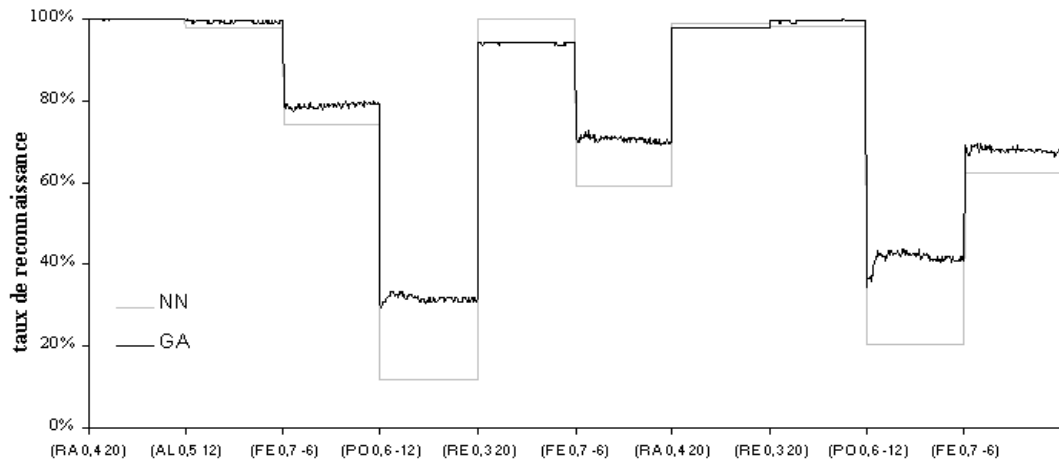


Figure 43 Comparaison des performances de la population GA et d'un NN incrémental. Les performances du meilleur individu de la population GA sont reportées.

V.6.4 Apprentissage global

L'apprentissage global est celui qui est le plus utilisé en parole. Il consiste à faire apprendre, en une seule fois, les données de parole produites dans différentes conditions acoustiques. Le réseau est alors testé dans ces différents environnements sans avoir à s'adapter au changement de conditions. Nous présentons les résultats de ce réseau dans la Figure 44.

Les performances moyennes d'un NN global sont sensiblement identiques à celles d'un NN incrémental. Cependant, la chute dans les environnements difficiles est moindre. Les GA améliorent les taux de reconnaissance de 6% en moyenne et restent plus performants dans la plupart des environnements.

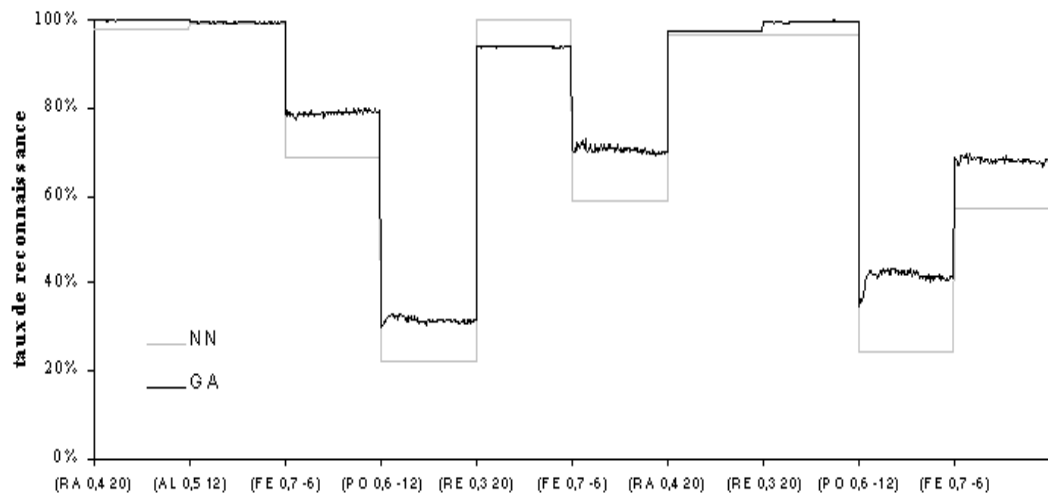


Figure 44 Comparaison des performances de la population GA et d'un NN global. Les performances du meilleur individu de la population GA sont reportées.

V.6.5 Population de Réseaux

La complexité des GA par rapport à celle des différents NN était très différente. En effet, les performances d'un seul réseau sont comparées aux performances d'une population de réseaux. Afin de rétablir un peu d'équité, nous avons étudié les performances d'une population de 20 NN n'ayant pas la capacité d'évoluer par GA, mais ayant l'avantage de constituer 20 réseaux en parallèle. Les résultats sont reportés dans la Figure 45.

Nous observons que le meilleur d'une population de NN sans GA obtient, dans la majorité des environnements, des performances supérieures au meilleur NN issu des GA. Nous obtenons en moyenne 2% de plus lorsque l'évolution n'a pas lieu. Ces résultats sont bien sûr surprenants. Parmi une population de 20 NN initialisés aléatoirement, il semble qu'il existe toujours un individu spécialisé dans l'environnement courant. Ce n'est pas ce que l'on peut affirmer des GA dont la population se spécialise au fil des générations. On peut alors remarquer que notre méthode évolutionnaire n'est pas suffisamment puissante pour trouver un réseau optimal dans toutes les conditions acoustiques.

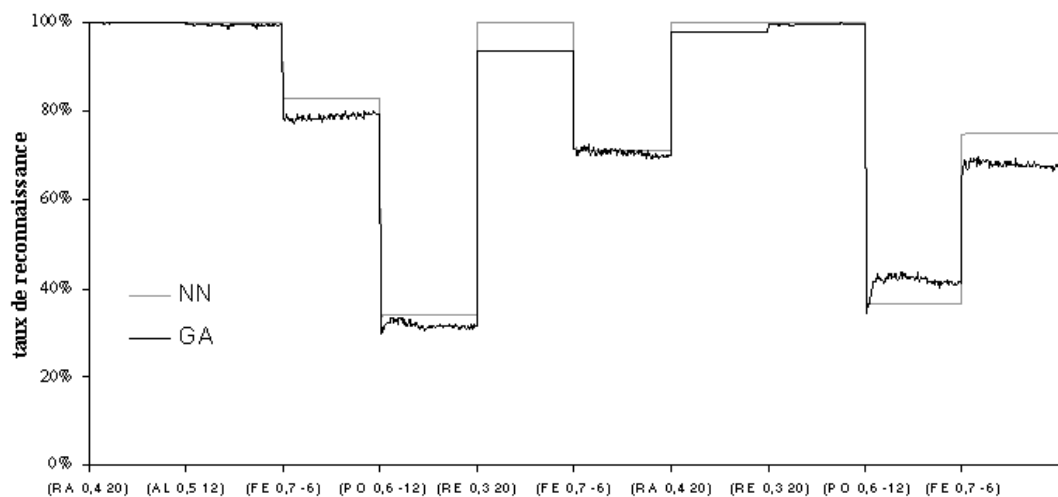


Figure 45 Comparaison des performances des populations GA et NN sans évolution. Les performances du meilleur individu des deux populations sont reportées.

La Table 8 récapitule les résultats des différentes méthodes d'apprentissage. Les taux de reconnaissance présentés sont la moyenne des performances du meilleur individu sur l'ensemble des environnements. On note la supériorité des populations de réseaux avec ou sans évolution.

Table 8 Récapitulatif des performances selon le type d'apprentissage.

NN Local	NN Incrémental	NN Global	NN population	NNGA
67,44%	72,28%	72,17%	79,97%	78,12%

V.7 Influence des environnements

Nous nous sommes intéressée à l'influence de l'ordre des environnements présentés aux systèmes. Dans l'expérience précédente, les deux premiers environnements ne présentaient pas de problèmes majeurs pour la reconnaissance. Ainsi, les réseaux, débutant avec des poids initialisés aléatoirement, donc sans connaissance *a priori*, font un apprentissage optimal, ce qui n'est peut-être pas le cas lorsque qu'ils partent d'une configuration de poids ayant préalablement convergé.

Nous avons alors reproduit la même expérience en ayant effectué une permutation circulaire des environnements de telle sorte que les deux premiers environnements soient ceux qui posent le plus de problèmes aux systèmes, à savoir (Fe 0,7 -6) et (Po 0,6 -12). Nous obtenons alors une supériorité des GA dans la plupart des environnements.

La Figure 46 présente les performances du meilleur individu des populations GA et NN dans une séquence d'environnements commençant par les environnements (Ra 0,4 20), (Al 0,5 12), (Fe 0,7 -6) et (Po 0,6 -12) et une séquence résultant d'une permutation circulaire de celle précédente, commençant par les environnements (Fe 0,7 -6) et (Po 0,6 -12). Afin de comparer avec les résultats précédents, nous avons superposé les résultats obtenus lors de la présentation des deux séries d'environnement.

La première constatation est que les performances d'une population de NN ne sont pas très dépendantes de l'ordre de présentation des environnements. Ceci rejoint l'idée qu'une population de 20 NN comportent toujours, ou presque, un individu adapté ou capable de s'adapter à l'environnement. La population de NN soumis aux GA semble, quant à elle, dépendante de l'ordre de présentation. Ceci vient certainement du fait que les individus sont tous adaptés à l'environnement courant et leur adaptation pour un nouvel environnement n'est pas triviale. On peut confirmer cette idée en regardant l'évolution des moyennes. Si l'on étudie les performances des populations sur l'ensemble de leurs individus, on peut noter la supériorité des GA quel que soit l'environnement choisi.

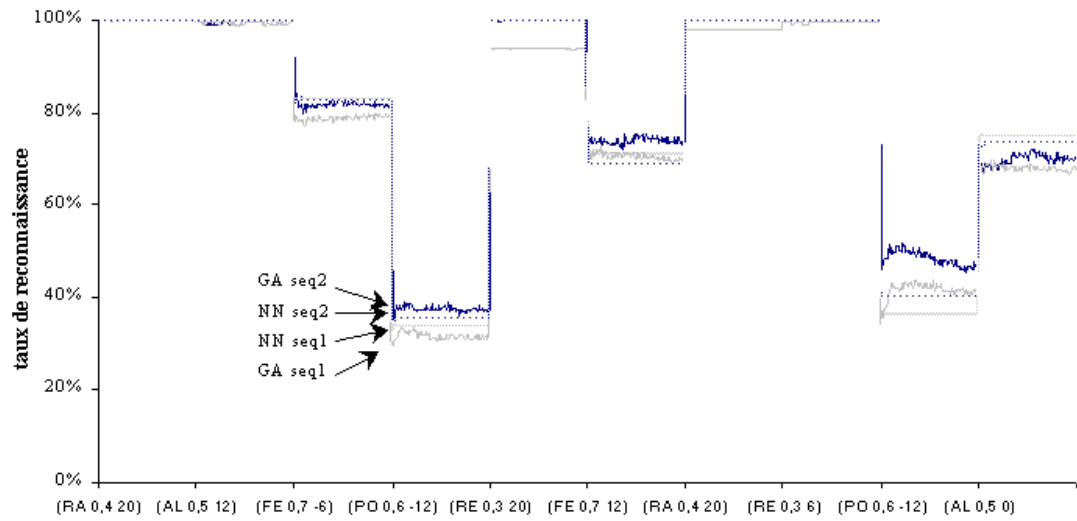


Figure 46 Influence de l'ordre de présentation des environnements sur les performances maximum. Comparaison des performances du meilleur individu d'une population de GA et d'une population de NN sans évolution sur deux séquences d'environnements différant d'une permutation circulaire de deux environnements

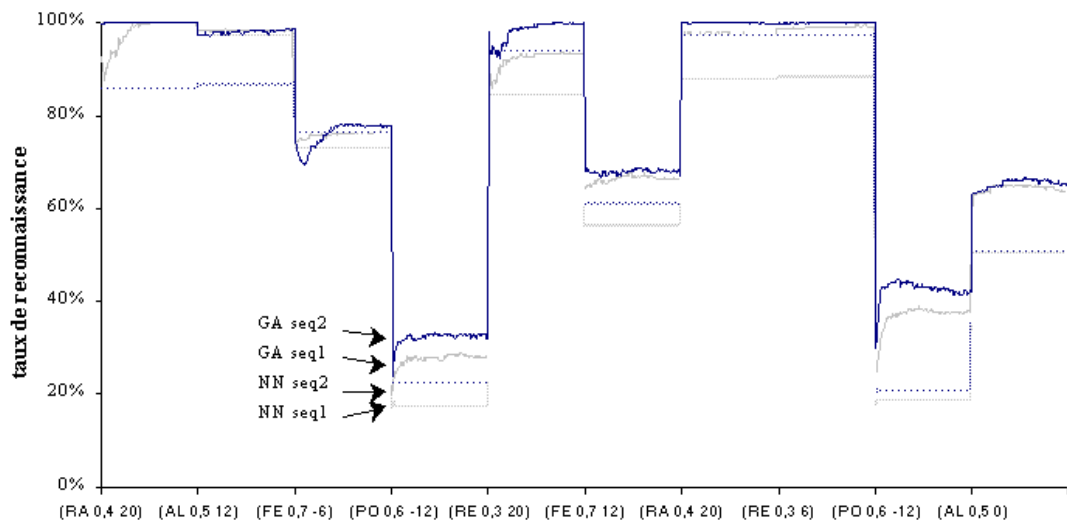


Figure 47 Influence de l'ordre de présentation des environnements sur les performances moyennes. Comparaison des performances moyenne d'une population de GA et d'une population de NN sans évolution sur deux séquences d'environnements différant d'une permutation circulaire de deux environnements.

Notre intérêt s'est alors porté sur l'évolution des performances dans le temps. Ainsi, nous avons effectué la même simulation que celle de la Figure 45 en présentant 5 fois consécutivement la même séquence de changements d'environnement.

V.8 Evolution dans le temps

Le but de cette expérience est tout d'abord d'étudier la constance des GA dans le temps. Pour ne pas dépendre de la séquence d'environnements, nous avons fait le choix de garder la même série revenant régulièrement dans le temps. Ceci nous permet d'étudier les performances de nos systèmes dans des conditions identiques mais à des instants différents.

Nous proposons de comparer les performances de 2 populations dans 5x10 environnements moyennées sur 9 simulations. La séquence de 10 environnements est identique à celle proposée dans les simulations précédentes. 5 occurrences de ces séries sont présentées consécutivement aux populations. La première population est constituée de NN à apprentissage local, la seconde est une population évoluant à l'aide de la méthode basée sur les GA.

Dans un premier temps, nous comparons les performances moyennes (sur les individus) des populations. Nous pouvons noter la régularité des GA dans le temps et la constante décroissance des performances des NN. Les performances des GA sont, de plus, supérieures aux NN, quel que soit l'environnement acoustique.

Si l'on considère, par contre, les performances du meilleur individu, les GA n'obtiennent de meilleures performances que dans les environnements très difficiles où le bruit est très fort. Dans la majorité des autres cas, la population de NN contient un individu adapté ou capable de s'adapter aux conditions acoustiques dans lequel elle est plongée.

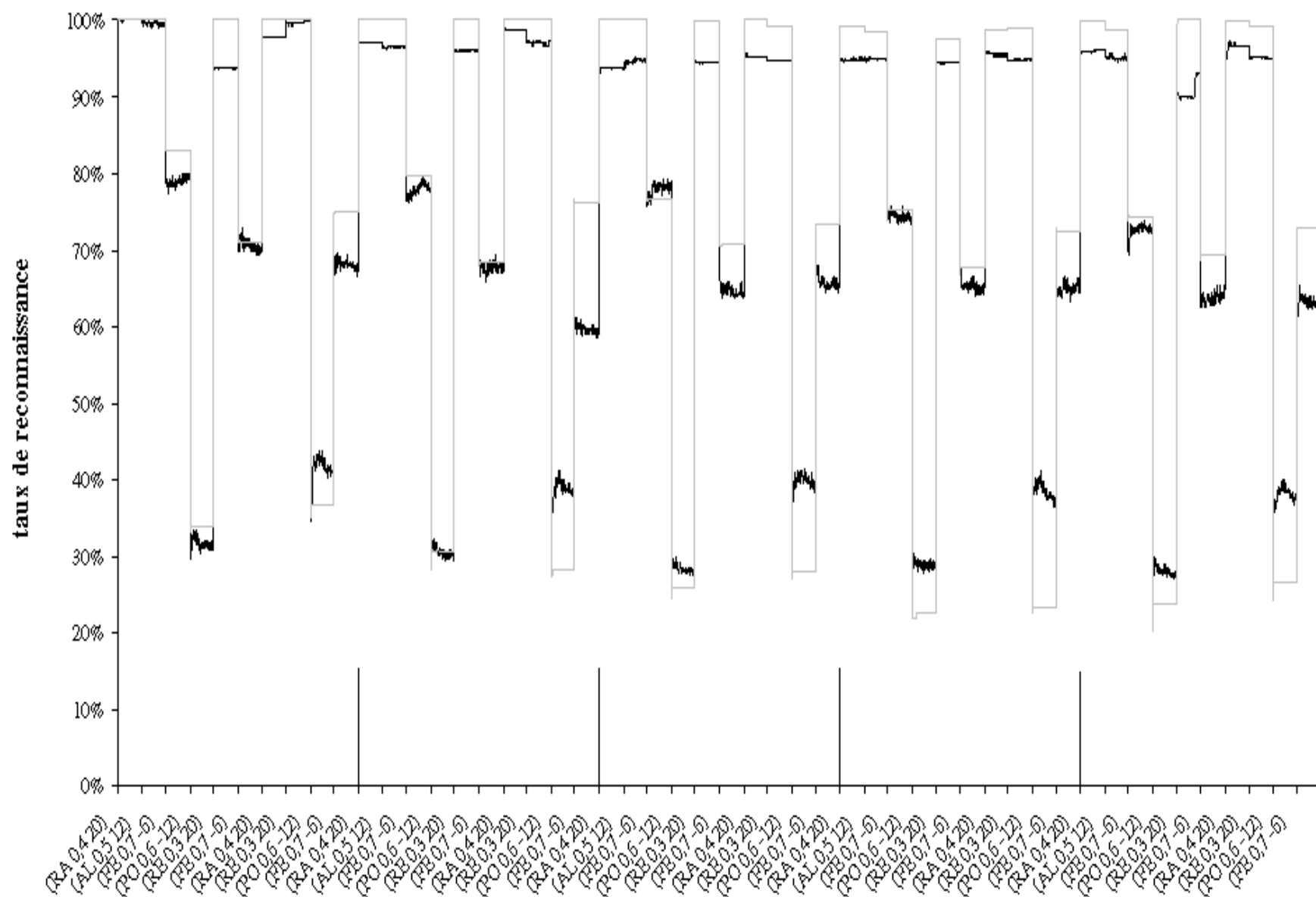


Figure 48 Evolution dans le temps des performances maximum des populations GA et NN. Comparaison des performances sur 5 répétitions de la même séquence d'environnements.

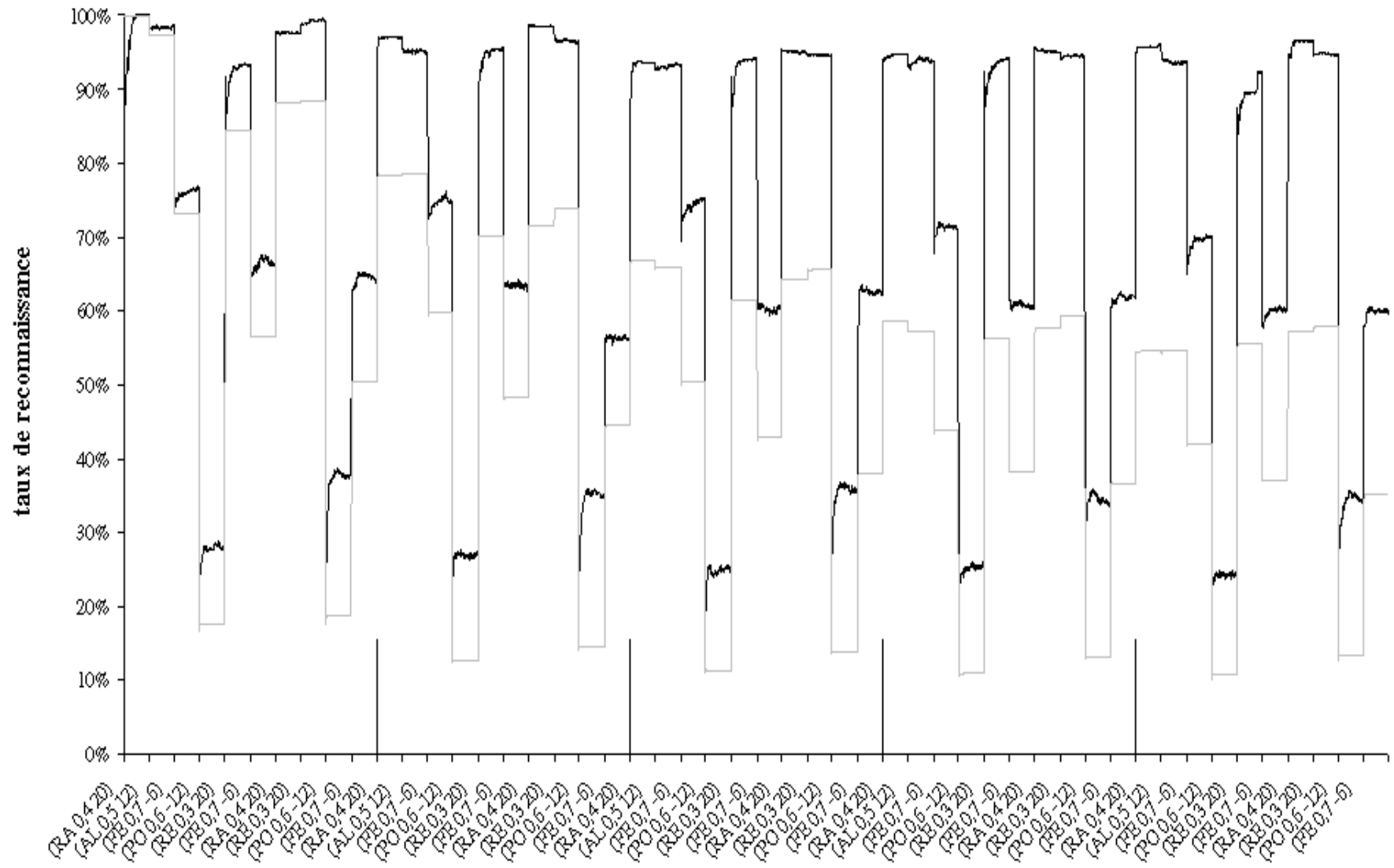


Figure 49 Evolution dans le temps des performances moyennes des populations GA et NN.
 Comparaison des performances sur 5 répétitions de la même séquence d'environnements.

Les expériences que nous venons de mener portaient sur des données considérées comme simples dans le domaine de la reconnaissance de la parole. Elles nous ont permis d'étudier les différents paramètres concernant les méthodes évolutionnaires. Cette étape était cependant nécessaire afin de mettre en place les expériences. Nous allons désormais étudier le comportement des GA dans la reconnaissance de digits de la langue française.

V.9 Application aux digits

Les digits, extraits des 29 locuteurs de BDBS (voir section III.4.1), sont représentés par des vecteurs de 150 composants. Nous avons dû adapter l'architecture des systèmes de reconnaissance. Ils sont désormais composés de 150 unités en entrée, 30 en couche cachée et 10 en sortie.

La complexité de la tâche des GA est alors beaucoup plus grande. En effet, les génotypes associés à chaque individu sont de taille 4800, alors qu'ils étaient de taille 102 pour les voyelles.

Nous avons effectué une simulation en utilisant les mêmes paramètres d'évolution que pour l'apprentissage des voyelles, à savoir 10% de mutation multiplicative, 60% de croisement et 20 individus évoluant sur 100 générations. Les individus sont chargés d'apprendre des paquets de 480 digits et sont testés sur 120. Les résultats sont présentés dans la Figure 50.

La première constatation est que les performances se dégradent plus qu'elles ne s'améliorent au cours des générations. Ceci indique que la complexité est trop grande pour les GA que nous avons considérés. Les performances obtenues par les NN sans évolution étant nettement supérieures, nous savons qu'il existe de bien meilleurs NN que ceux produits par les GA. La complexité de telles simulations engendre des temps de calculs considérables. En effet, une simulation comportant 20 individus évoluant pendant 100 générations sur 10 environnements, dure en moyenne 4 jours sur un PC (PII 350, 192 Mo).

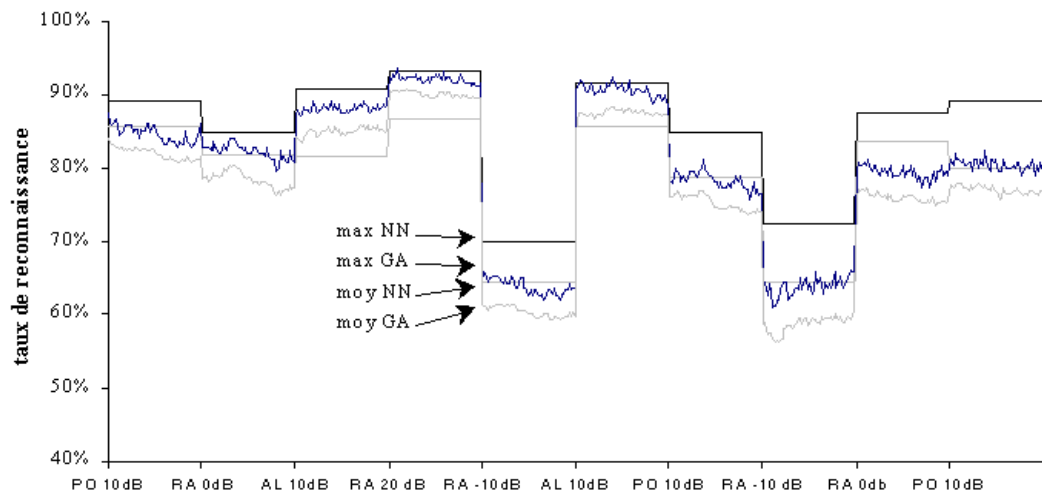


Figure 50 Evolution des performances sur les Digits.
 Comparaison des performances du meilleur individu ainsi que de la moyenne d'une population GA et d'une population NN sans évolution. 480 digits sont utilisés pour l'apprentissage, 120 pour le test.

Par une série de tests sur 2 ou 3 environnements, nous avons vu qu'une population de 200 individus paraît plus performante mais encore insuffisante. Il nous est alors impossible de traiter, dans les conditions actuelles, des populations plus grandes. Notre expérience ne permet donc pas de conclure en faveur des GA.

V.10 Discussion des résultats

Dans cette série d'expériences, nous avons recherché la meilleure manière de faire évoluer une population de systèmes de reconnaissance, considérés comme des individus, à partir de deux méthodes classiques proposées dans le domaine des EA.

Les individus étaient alors de petits systèmes de reconnaissance basés sur des réseaux connexionnistes. Ils avaient pour tâche de reconnaître les dix voyelles de la langue française prononcées par un locuteur dans une série d'environnements acoustiques différents. Nous désirions faire évoluer ces systèmes de telle sorte qu'ils s'adaptent efficacement aux changements de conditions acoustiques. Il est connu qu'en reconnaissance automatique de la parole, la tâche consistant à apprendre et catégoriser les voyelles est une tâche relativement simple. Cependant, elle se complexifie lorsque les paramètres représentant les voyelles sont altérés par du bruit

et dans ce cas, un réseau unique montre de gros problèmes d'adaptation lorsqu'on ne modifie pas sa structure. On peut considérer que celui-ci, lors de l'apprentissage, a convergé vers une configuration de poids correspondant à une configuration stable de laquelle il est très difficile de sortir.

Nous avons fait varier les différents paramètres de l'évolution afin d'en étudier l'influence sur la qualité des résultats. Les EA (GA et ES) que nous avons implémentés sont performants sur des fonctions classiques telles celles de De Jong et de Shubert (voir Annexe). Nous avons testé les capacités des EA à s'adapter à un changement brutal d'environnement, en particulier à la translation de la fonction F1 de De Jong. Nous avons constaté l'efficacité des ES à s'adapter à ce type de changement de manière efficace et robuste dans le temps.

Nous avons ensuite testé les capacités des GA sur une application plus complexe qui est la reconnaissance de voyelles dans des environnements changeants. Passer d'un environnement acoustique à un autre consiste, en quelque sorte, à modifier l'espace de représentation des voyelles d'une manière souvent beaucoup plus complexe qu'une translation (Sagayama 99).

Contrairement aux travaux que l'on peut trouver dans la littérature hybridant NN et GA (Belew et al. 90) (Gruau et Whitley 93), nous n'avons pas toujours obtenu de meilleurs NN en utilisant les méthodes évolutionnaires. Nous pensons que la raison vient de l'algorithme d'apprentissage du NN que nous avons utilisé. En effet, celui-ci a été conçu pour la reconnaissance de la parole et il est fortement optimisé afin d'atténuer la convergence vers des minima locaux. Cependant, l'utilisation d'un réseau plus simple ne nous aurait pas permis de gérer convenablement des corpus aussi importants que ceux que nous avons utilisés. De plus, notre objectif étant d'étudier l'apport de l'hybridation EA/NN pour la robustesse des systèmes de reconnaissance de la parole lors de changements d'environnement, et non l'apport de l'hybridation en tant que telle, il nous a semblé naturel d'utiliser un NN performant pour notre domaine.

Les résultats que nous avons obtenus montrent qu'une population de NN n'ayant pas les capacités d'évoluer est globalement plus performante lorsqu'il s'agit de trouver le meilleur réseau. Cependant, si l'on compare les performances de la population en

moyenne, les GA montrent une nette supériorité. L'interprétation qui en découle est que, parmi une population de NN, il semble exister dans la plupart des cas au moins un individu adapté ou facilement adaptable à l'environnement courant. Ce n'est pas le cas de toute la population puisque l'écart entre les performances du meilleur et de la population moyenne est souvent important. Les GA permettent, quant à eux, d'amener l'ensemble de la population vers des performances acceptables.

Dans le cas de la recherche du meilleur NN, l'évolution n'a montré sa supériorité que dans le cas d'environnements difficiles. C'est principalement dans ce type de conditions que les EA sont nécessaires. Dans de bonnes conditions acoustiques, les méthodes classiques sont très performantes et ne nécessitent pas d'amélioration.

En revanche, dans l'optique de réaliser un système non supervisé, il est important de prendre en compte les performances moyennes de la population qui nous permettront d'effectuer des statistiques.

Nous avons effectué la même étude sur les digits. La méthode évolutionnaire que nous avons choisie ne semble pas engendrer une diversité suffisante pour la complexité du problème que nous désirions résoudre. Il est cependant difficile de tester cette méthode sur une population plus diversifiée puisqu'il serait nécessaire, entre autres, de faire évoluer des populations de plus grande taille. L'évolution d'une population de 20 individus, pendant 100 générations, sur 10 environnements, dure environ 4 jours. Une manière de pallier ce problème est évidemment la parallélisation. En effet, le temps d'apprentissage des individus étant très important (de l'ordre de la minute), distribuer l'apprentissage des individus sur plusieurs machines serait très avantageux.

Une autre manière d'obtenir une population suffisamment diversifiée est d'insérer dans la population des individus nouveaux (Cobb et Grefenstette 93). Il est envisageable de mémoriser les meilleurs individus fournis au cours des environnements et de les ré-injecter lors d'un changement d'environnement.

De plus, s'il existe dans une population de NN sans évolution un bon individu dans la majorité des cas, il semble judicieux de l'exploiter. Dans ce cas, il est certainement

intéressant de faire une sélection élitiste. Nous risquons cependant de perdre la diversité nécessaire à l'adaptation au changement d'environnement.

Ainsi, la combinaison d'une sélection élitiste forte avec une immigration d'individus extérieurs à la population, pourrait être une méthode intéressante à évaluer.

CHAPITRE VI

ADAPTATION DES DONNÉES

Ce chapitre est consacré à la recherche d'espaces de projection optimisés permettant de transformer des données bruitées en des données les moins bruitées possibles. Si l'on considère que l'ajout de bruit consiste en une projection des données, nous désirons trouver la projection inverse permettant de supprimer ce bruit.

Nous proposons, dans un premier temps, de représenter les données acoustiques dans un espace, puis de réduire cet espace à l'aide d'une Analyse en Composantes Principales (ACP). Cette transformation présente deux avantages : non seulement la réduction de l'espace de travail, mais également le fait d'agir sur l'espace de représentation, et non sur les données brutes.

Dans un deuxième temps, nous appliquons les EA sur cet espace de projection afin de réduire au maximum, dans la représentation finale, le bruit entachant le signal.

Dans le chapitre précédent, l'évolution se portait sur le système de reconnaissance en lui-même. Se basant sur les deux grands processus engendrés dans la reconnaissance de la parole chez les humains, nous pouvions alors représenter le principe de manière très schématique de la manière suivante :

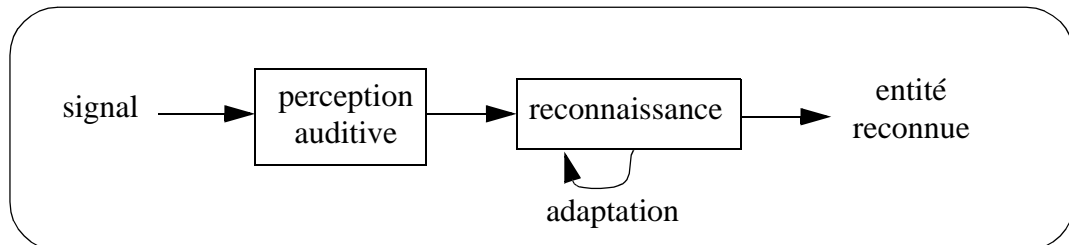


Figure 51 Principe de l'adaptation du système de reconnaissance.

Dans le cadre de ce chapitre, l'adaptation des données consiste alors à faire évoluer le système de perception en fonction de l'environnement. La représentation schématique du principe est alors la suivante :

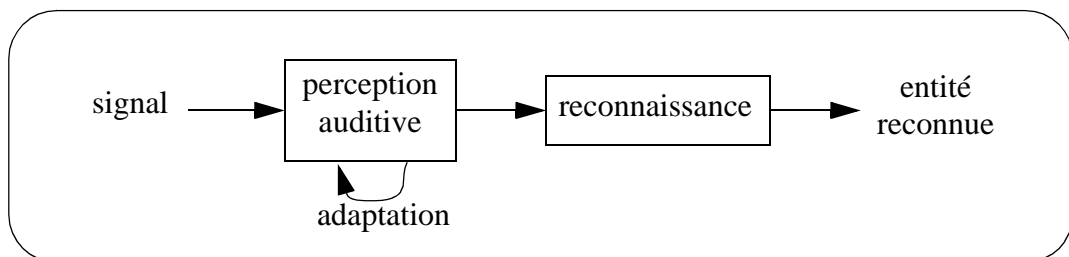


Figure 52 Principe de l'adaptation du système de perception.

Cette fois-ci, le système de reconnaissance est fixe, et seules les données en entrées de celui-ci sont en mesure de s'adapter. Afin de réaliser cette adaptation, nous désirons trouver une base de projection permettant, en quelque sorte, de faire la correspondance (*mapping*) entre les données d'un environnement bruité et celles d'un environnement non bruité. Nous allons, pour cela, combiner l'ACP et les EA.

VI.1 ACP et Reconnaissance de la parole

L'ACP est une méthode de traitement de données permettant de représenter des données de dimension n en des données de dimension $p < n$ par le biais d'une projection dans un sous-espace.

Différents domaines ont expérimenté cette méthode. (Bailer-Jones 96) utilise l'ACP pour classifier les étoiles, (Turk 91) (Liu et Wechsler 98) pour la reconnaissance de visage. Dans le domaine de la parole, cette technique est également utilisée afin de traiter les données de parole. Celles-ci sont souvent constituées d'éléments corrélés, ce qui rend plus difficile la tâche de reconnaissance. (Trompf et al. 93) ont proposé d'utiliser une ACP sur les composantes des données bruitées en amont d'un réseau de neurones spécialisé dans le filtrage de bruit. Il montre que l'association des deux méthodes permet de diminuer fortement l'impact du bruit et fournit des données robustes au système de reconnaissance. Par ailleurs, (Lleida et al. 90) ont utilisé l'ACP afin d'obtenir, pour deux échantillons de parole de durée différentes, un nombre constant de fenêtres (c'est-à-dire d'échantillons de signal supposé stationnaire).

Nous allons désormais présenter le principe de l'ACP dans le cadre de l'application à des données de parole, ainsi que les étapes que nous avons effectuées afin de modifier ces données.

VI.2 Analyse en composantes principales

VI.2.1 Principe

Soit a un échantillon de signal de parole représenté par n coefficients issus d'une analyse acoustique.

$$a = (c_1, c_2, \dots, c_n)$$

m échantillons forment alors une matrice A de dimension $(m \times n)$:

$$A = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

Dans un espace de dimension n muni d'un repère, on place les points $a_i = (c_{i1}, c_{i2}, \dots, c_{in})$. On obtient ainsi une représentation graphique des données : le nuage de points.

L'idée de l'ACP est de déterminer p vecteurs ($p < n$) tels que la somme des distances euclidiennes des points du nuage au sous-espace engendré par ces p vecteurs soit minimale. On montre alors que la projection du nuage de points dans ce sous-espace est la meilleure approximation du nuage en dimension p .

Par exemple, dans le plan (voir Figure 53), l'ACP détermine la droite de direction u qui approche au mieux le nuage de points.

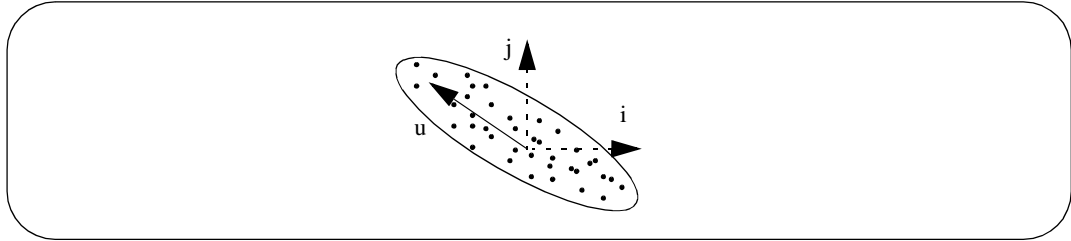


Figure 53 Direction (u) de projection d'un nuage de points.

VI.2.2 Méthode

Pour déterminer le sous-espace, l'ACP opère sur des données centrées sur zéro. De plus, les données n'étant pas forcément de même nature, on normalise les vecteurs afin de s'affranchir des unités de mesure.

Ainsi, si l'on nomme \bar{c}_j et $\sigma(c_j)$, respectivement la moyenne et l'écart-type des données du caractère c_j , on détermine la matrice A_c des données centrées réduites par la formule :

$$A_{c_{ij}} = \frac{A_{ij} - \bar{c}_j}{\sigma(c_j)}$$

L'ACP calcule ensuite la matrice de corrélation suivante :

$$C = {}^t A_c \cdot A_c$$

Cette matrice est diagonalisable. On détermine donc les n valeurs propres que l'on range par ordre décroissant. Les p vecteurs correspondent aux p vecteurs propres associés aux p premières valeurs propres rangées par ordre décroissant. Chacun de ces vecteurs a une capacité de discrimination relatif à sa valeur propre.

Soit P la matrice de passage de l'espace initial dans le sous-espace engendré par les p vecteurs propres, on définit la matrice A_p formée des p coordonnées des projetés des n points par :

$$A_p = P \cdot A_c$$

VI.3 Application aux données acoustiques

Nous désirons appliquer une ACP à des données issues d'une analyse acoustique afin de les présenter à un système de reconnaissance basé sur un réseau connexionniste.

Comme la littérature le suggère, les données résultant de cette ACP sont de taille réduite par rapport aux données initiales et présentées d'une manière telle, qu'elles permettent au réseau de neurones d'effectuer un meilleur apprentissage .

Afin d'obtenir des caractéristiques pertinentes représentant le signal de parole, nous avons utilisé une analyse PLP (Hermansky 90) basée sur les principes psycho-acoustiques. Cette méthode permet d'extraire du signal vocal des indices acoustiques pertinents en éliminant la redondance des informations fournies par le signal de parole.

Les données que nous utilisons peuvent être représentées par des vecteurs composés de caractéristiques ca_i calculées par une analyse acoustique PLP. Elles sont, de plus, centrées réduites :

$$(ca_1, ca_2, \dots, ca_n)$$

Les composantes principales calculées sur ces données représentent un espace de n dimensions. Le choix approprié d'un sous-espace de vecteurs propres de dimension $p < n$ représenté par :

$$(vp_1, vp_2, \dots, vp_p)$$

permet la réduction de la taille des données en les projetant dans ce sous-espace. Ces données sont alors les caractéristiques projetées :

$$(Pca_1, Pca_2, \dots, Pca_n)$$

et représentent le corpus d'apprentissage.

VI.4 Architecture

Le but de notre travail est de trouver une sorte de correspondance entre les données de test bruitées et les données d'apprentissage non bruitées. Le signal non-bruité analysé Anb est projeté dans un espace par la projection Pnb et le vecteur résultant Dnb est mis en entrée du réseau pour l'apprentissage. Le signal bruité analysé Ab est projeté dans un espace par la projection Pb et le vecteur résultant Db est mis en entrée du réseau pour la reconnaissance.

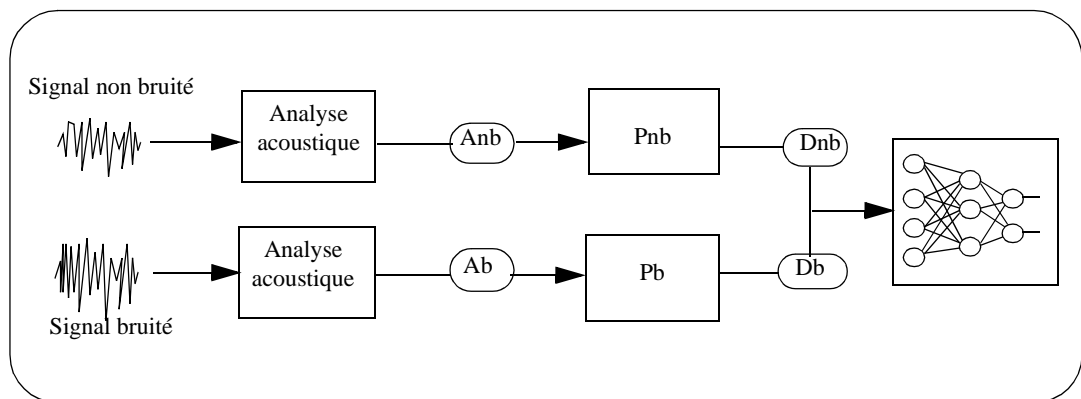


Figure 54 Modèle de projection des données bruitées et non bruitées.

Pour que cette reconnaissance se fasse au mieux, il faut réduire la distance entre le vecteur d'apprentissage et le vecteur de test. Pour cela, nous devons trouver la matrice de projection P_b de sorte que l'on obtienne des données D_b équivalentes aux données D_{nb} .

Dans notre cas, la base de projection P_{nb} des données A_{nb} est une base de vecteurs propres issus d'une ACP. Cette même base ne permet pas une bonne projection des données bruitées A_b . C'est pour optimiser cette base de projection P_b que nous allons utiliser les algorithmes évolutionnaires. Sachant qu'aucune contrainte ne sera posée sur l'optimisation de cette base, l'espace de projection résultant ne sera ni orthogonal, ni normé. Enfin, pour ne pas partir d'une base aléatoire, ce qui rendrait nos simulations plus longues, la base de projection P_b sera optimisée en partant de la base P_{nb} .

VI.4.1 Représentation

Différentes représentations des vecteurs de la base à muter sont possibles. (Liu et Wechsler 98) proposent de travailler sur les angles des vecteurs propres ainsi que sur la sélection des vecteurs les plus pertinents. La représentation du génotype pour m vecteurs est alors la suivante :

$$(\alpha_1, \alpha_2, \dots, \alpha_{(m(m-1))/2}, a_1, a_2, \dots, a_m)$$

Les angles α_i sont codés en binaires, les a_i sont représentés par une valeur binaire indiquant si le vecteur fait partie du sous-espace ou non.

Notre représentation est différente dans le sens où nous ne nous sommes pas intéressée au problème de la sélection des vecteurs et que nous avons préféré travailler non pas sur des angles de rotation mais sur une représentation littérale des modifications ϵ_i pour chacune des valeurs représentant un vecteur. Ce qui donne la représentation suivante :

$$((\epsilon_{p_{1a}}, \epsilon_{p_{1b}}, \dots, \epsilon_{p_{1m}}), (\epsilon_{p_{2a}}, \epsilon_{p_{2b}}, \dots, \epsilon_{p_{2m}}), \dots, (\epsilon_{p_{ma}}, \epsilon_{p_{mb}}, \dots, \epsilon_{p_{mm}}))$$

L'espace altéré par ce génotype est alors :

$$((vp_{1a} + \epsilon_{p_{1a}}, \dots, vp_{1m} + \epsilon_{p_{1m}}), \dots, (vp_{ma} + \epsilon_{p_{ma}}, \dots, vp_{mm} + \epsilon_{p_{mm}}))$$

Ainsi, on peut schématiser l'altération d'un individu par la figure suivante :

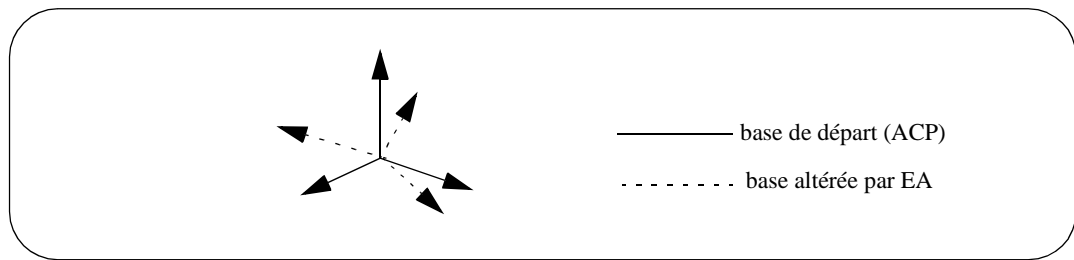


Figure 55 Représentation d'une base de projection définie par ACP et celle altérée par EA. La base de projection ACP sert à initialiser les individus correspondant alors aux bases altérées par EA.

VI.4.2 Projection

Il est alors possible de projeter des données nouvelles :

$$(na_1, na_2, \dots, na_N)$$

dans ce nouvel espace afin, soit de réduire la taille des données, soit d'effectuer une correspondance entre les données de d'apprentissage et les données de test. Les données obtenues peuvent être représentées par :

$$(Pna_1, Pna_2, \dots, Pna_m)$$

VI.4.3 Performance

La performance peut être représentée comme la capacité de l'espace de représentation à projeter les données de telle manière que le système de reconnaissance soit capable de les reconnaître correctement

VI.5 Présentation du modèle

Le système consiste en une population de bases de projections de données évaluée par un système de reconnaissance. Comme nous avons commencé à le décrire précédemment, nous désirons optimiser une base de projection afin de trouver la correspondance entre un environnement bruité (environnement de test) sur un environnement non bruité (environnement d'apprentissage). Le système de

reconnaissance, un système connexionniste, a été entraîné dans un environnement non-bruité. Ce système n'est cependant pas capable de donner des performances acceptables lorsque son environnement change brutalement de conditions acoustiques. Le système décrit dans ce chapitre propose de réduire l'écart entre les conditions d'apprentissage et les conditions de test afin de réduire les chutes de performance du réseau.

Nous aurions pu essayer de travailler directement sur les données mais nous étions confrontés à un problème de contrôle de l'évolution de ces données ainsi qu'à un problème de complexité. En effet, pour garder une certaine cohérence entre les données de même type, il semble logique de muter de la même manière les signaux appartenant à la même catégorie. Ce qui rajoute une forte contrainte sur notre algorithme d'évolution et entraîne un accroissement de la complexité.

C'est pourquoi nous avons choisi de travailler sur un espace de projection identique à toutes les classes de signaux.

Le principe consiste à générer une population de bases de projection à partir d'une ACP non bruitée et de faire évoluer ces espaces afin d'obtenir le meilleur espace de projection de données bruitées.

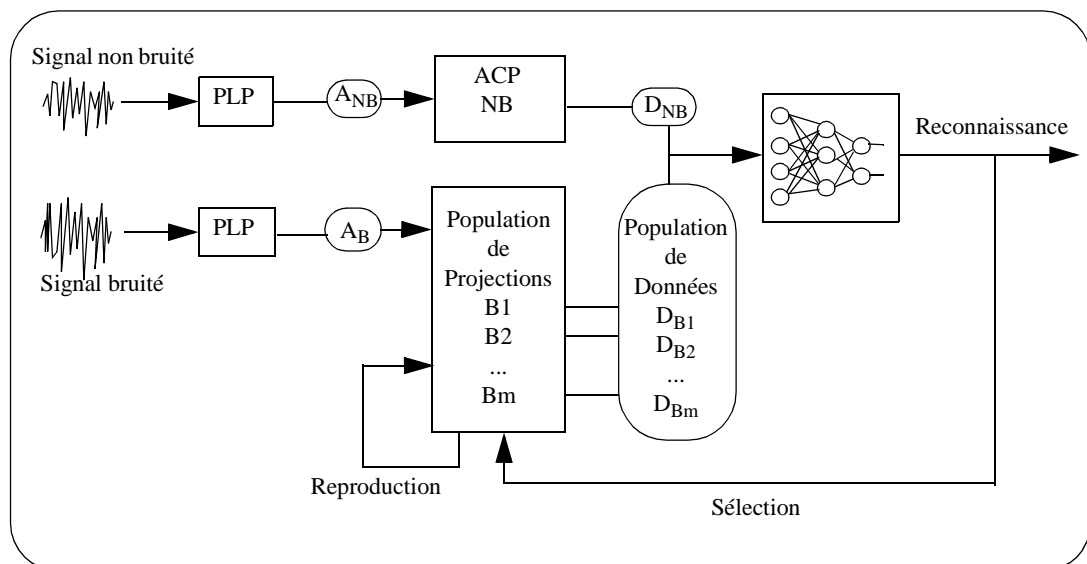


Figure 56 Modèle d'évolution d'une population de bases de projection.

VI.6 Préparation des données

VI.6.1 Données d'apprentissage

Les données pour l'expérimentation sont des digits issus de la base de données BDSons (Descout et al. 86). 5000 digits prononcés par 29 locuteurs sont analysés par la méthode des PLP et sont représentés par des vecteurs de dimension 150 du type :

$$(ca_1, ca_2, \dots, ca_{150})$$

Ces données forment alors une matrice A de taille (5000×150) . Elles sont alors traitées par une ACP qui fournit une base de vecteurs propres constituée de 150 vecteurs :

$$(vp_1, vp_2, \dots, vp_{150})$$

Cette base constitue la base de référence "canonique" qui servira de base de projection des données non bruitées. Les caractéristiques de l'ACP font que l'information véhiculée par les vecteurs propres est proportionnelle à leur rang dans la base. Ainsi, le vecteur vp_1 véhicule plus d'information que vp_2 , vp_2 plus que vp_3 , etc. En étudiant l'apport de chaque vecteur, il est possible d'éliminer les moins importants afin de réduire la base de vecteurs à une dimension $p < 150$ et ainsi, de réduire la taille des données. Chaque digit de la base d'apprentissage est alors projeté dans cet espace. Les données résultantes sont alors de la forme :

$$(Pca_1, Pca_2, \dots, Pca_p)$$

L'ensemble de ces données projetées forment la base d'apprentissage pour le système de reconnaissance.

VI.6.2 Données de test

Un corpus de données bruitées (2500 digits prononcés par 29 locuteurs) est utilisé pour le test. Il est projeté dans les différentes bases de projection issues de la population de bases fournies par les opérateurs génétiques. La base canonique décrite plus haut sert de générateur de la population initiale. Chaque composant de la base est altéré par une petite valeur ϵ_i issue d'un générateur aléatoire de telle sorte que pour un vecteur d'une base donnée on ait : $(vp_1 \pm \epsilon_1, vp_2 \pm \epsilon_2 \dots vpr \pm \epsilon_r)$.

Les données bruitées sont représentées par un vecteur : $(na_1, na_2, \dots, na_{150})$

et une fois projetées sont de la forme : $(Pna_1, Pna_2, \dots, Pna_r)$.

Les corpus résultant de ces projections représentent les corpus de test du système de reconnaissance.

Nous allons maintenant décrire comment la projection de ces données peut être optimisée par les algorithmes évolutionnaires.

VI.7 Evolution du système

La tâche des algorithmes évolutionnaires est de trouver, à partir de la base définie par l'ACP, une base optimisée pour la projection des données bruitées.

VI.7.1 Algorithme d'évolution

L'algorithme d'évolution adapté à cette tâche est le suivant :

```

Calcul de la base canonique par l'ACP sur les données non-bruitées.
Apprentissage du système de reconnaissance sur les données non bruitées
projetées
Initialisation d'une population d'espaces de projection
Pour n générations faire :
    Pour chaque individu (base de projection) faire :
        Projeter les données bruitées dans cette base
        Evaluer cette base en testant les capacités de reconnaissance du
        système sur ces données
    Fin Pour
    Selection et Reproduction des bases
Fin Pour.
Fin Algo.
```

Figure 57 Algorithme ACP/EA.

Après la création d'un corpus non bruité projeté par une ACP, le système de reconnaissance (réseau de neurones) est entraîné jusqu'à ce qu'il converge.

La base utilisée pour la projection de ces données est le générateur de la population qui doit s'adapter aux nouvelles données bruitées. L'adaptation se fait à l'aide

d'opérateurs de sélection, mutation et recombinaison qui permettront, par la suite, de simuler des GA ou des ES.

VI.7.2 Fonction d'évaluation

Le taux de reconnaissance du système en milieu bruité est la base de notre sélection. A chaque base de projection est associée une valeur d'adaptation dépendant du taux de reconnaissance du système. Les individus ayant effectué les meilleures transformations entre l'environnement bruité et l'environnement non-bruité auront les meilleures performances.

VI.8 Expérimentations

Nous proposons d'étudier l'apport des EA dans l'amélioration des performances d'un système de reconnaissance soumis à différents changements d'environnement acoustique. La section suivante présente la série d'expériences que nous avons menée.

VI.8.1 Plan d'expériences

Les expériences que nous avons menées sont basées sur les performances d'un système de reconnaissance ayant appris 5000 digits dans un environnement non bruité et étant testé dans un environnement bruité dont le corpus est composé de 2500 digits.

Nous avons, dans un premier temps, testé l'effet de la projection par ACP des données bruitées et non bruitées, et notamment l'influence du choix de la dimension de l'espace de projection.

Dans un second temps, nous avons appliqué les EA afin d'améliorer l'espace de projection fourni par l'ACP. Nous avons alors étudié l'influence de différents paramètres des méthodes d'évolution GA et ES.

Partant de la meilleure méthode trouvée, nous avons testé l'effet du bruit lors d'un ou plusieurs changements d'environnement.

Table 9 Plan d'expériences global.

Expériences	Nb environnements	Nb patterns / environnement	Type de patterns
Influence de l'ACP	1	7500	Digits
Influence des paramètres d'évolution	1	2500	Digits
Influence de l'environnement	1	2500	Digits
Robustesse dans le temps	6	2500	Digits

VI.8.2 Influence de la projection par ACP

Les données étant au préalable traitées par l'ACP, nous avons voulu vérifier l'influence que pouvait avoir cette projection sur l'intelligibilité des données et sur les taux de reconnaissance suite à un apprentissage sur différents corpus.

Dans un premier temps, nous avons effectué un apprentissage d'un corpus de 5000 digits non traités par l'ACP. Le réseau est constitué de 150 unités en entrées, 40 en couche cachée et 10 en sortie. Un taux de reconnaissance de 67% a été obtenu (moyenne sur 9 apprentissages).

Dans un second temps, nous avons effectué un apprentissage sur ce même corpus mais préalablement projeté dans des bases issues de l'ACP. Plusieurs bases de tailles différentes sont possibles et nous avons testé les bases de dimension {10, 20,...,150} afin de voir l'influence de la projection mais aussi de la taille des données en entrée. Les réseaux ayant à apprendre ces corpus ne se différencient que par leur nombre d'unités en entrée. Le réseau ayant à apprendre le corpus dont les données sont de dimension 10 a alors 10 entrées, 40 cachées et 10 sorties alors que celui qui est spécifique aux données de dimension 150 a 150 entrées, 40 cachées et 10 sorties.

Les résultats reportés ci-dessous sont le résultat de moyennes sur 9 simulations de taux de reconnaissance sur des corpus de test (soumis au même traitement que le corpus d'apprentissage associé). Le corpus de référence correspond au corpus n'ayant pas subi de traitement par ACP.

Le réseau est entraîné sur un corpus non bruité, puis testé sur un corpus bruité (radio 10db). Le réseau, testé dans l'environnement d'apprentissage, obtient 67%. Ce même réseau, testé dans un environnement bruité, voit ses performances chuter à 11%.

La même expérience a été effectuée avec un corpus d'apprentissage et de test préalablement traité par l'ACP. La Figure 58 montre les performances du réseau pour l'apprentissage et la reconnaissance de données ACP de différentes tailles.

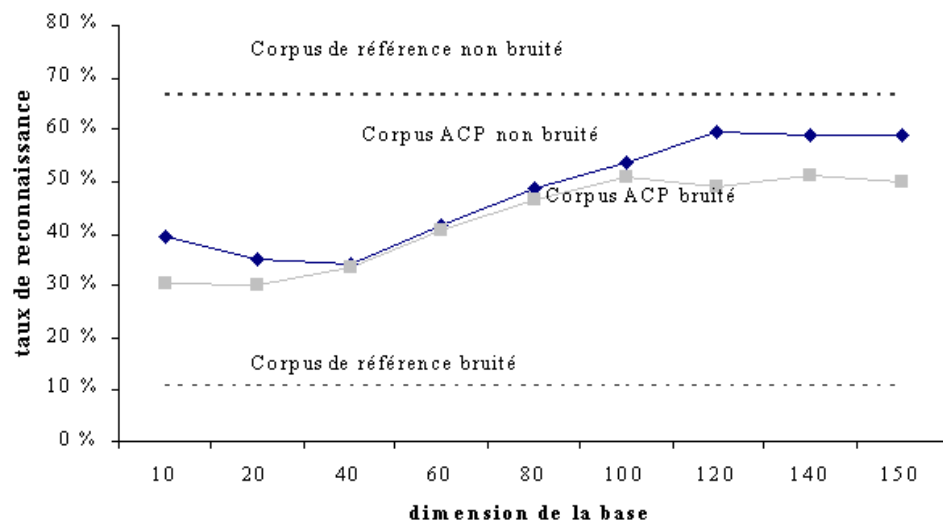


Figure 58 Influence de la projection dans le sous espace défini par l'ACP. Taux de reconnaissance en milieu bruité ou non bruité.

La réduction de la taille des données engendre une chute des performances du système de reconnaissance qui varie de 7% (non réduction) à 27% (réduction d'un facteur 15).

Cependant, la chute des performances entre l'environnement bruité et non bruité est beaucoup moins importante que lorsque les données ne sont pas traitées par l'ACP. Ainsi, les performances en phase de test sont jusqu'à 40% supérieures à celles sans ACP.

VI.8.3 Influence des paramètres

Comme nous l'avons vu dans la figure précédente, le taux de reconnaissance d'un réseau à 10 entrées est de 20% moins bon que celui à 150 entrées. Mais pour des raisons de complexité, nous avons choisi d'étudier les effets des EA sur ces données

de dimension 10. L'amélioration des performances ne sera peut-être pas aussi importante mais le principe reste le même, et c'est principalement lui qui nous intéresse.

Une population d'individus de grande taille (10×150) (nombre de vecteurs de la base * taille d'un vecteur) est générée et les individus ont 1000 générations pour évoluer dans le but d'améliorer leur performance.

L'attribution de cette performance est effectuée par un réseau de neurones pré-entraîné sur le corpus d'apprentissage. Le réseau est constitué de 10 unités en entrée, 40 en couches cachées et 10 en sortie.

Nous avons effectué une série d'expériences afin d'étudier les différents paramètres de l'évolution. Elles sont récapitulées dans la Table 10.

Les expériences ont tout d'abord été effectuées pour un environnement bruité correspondant à 2500 digits prononcés dans un environnement bruité (Radio) avec un RSB de 10dB et aucune réverbération. Afin de s'adapter à cet environnement, nous avons fait varier la taille de la population, le taux de mutation, le taux de croisement d'une population évoluant par GA. La mutation est gaussienne d'écart-type 0,1. Nous avons ensuite étudié la méthode ES et l'influence de la taille de la population et plus particulièrement le nombre d'enfants sélectionné μ .

Nous avons ensuite étudié le comportement d'une population évoluant par la meilleure méthode que nous ayons trouvée dans différents environnements acoustiques bruités par de la radio à -20dB, -10dB, 10dB, 20db, une alarme à 10dB et une porte qui claque à 10dB.

Enfin, nous avons testé cette meilleure méthode sur une série de changements d'environnements consécutifs.

Table 10 Plan d'expériences.

Stratégie	Taux de mutation	Taux de croisement	Taille population	Environnement
GA	3%	80%	100	Ra 10dB
GA	3%	80%	50	Ra 10dB
GA	3%	80%	100	Ra 10dB
GA	6%	80%	100	Ra 10dB
GA	10%	80%	100	Ra 10dB
GA	3%	0%	100	Ra 10dB
GA	3%	30%	100	Ra 10dB
GA	3%	60%	100	Ra 10dB
ES			(2,100)-ES	Ra 10dB
ES			(5,100)-ES	Ra 10dB
ES			(10,100)-ES	Ra 10dB
ES			(25,100)-ES	Ra 10dB
ES			(25,100)-ES	Aucun
ES			(25,100)-ES	Ra -10dB
ES			(25,100)-ES	Ra 10dB
ES			(25,100)-ES	Ra 20db
ES			(25,100)-ES	Al 10dB
ES			(25,100)-ES	Po 10dB
ES			(25,100)-ES	 multiples

VI.8.4 Algorithmes génétiques

Afin de trouver le meilleur ensemble de paramètres dans la population, nous avons fait varier le nombre d'individus dans la population, le taux de mutation ainsi que le taux de croisement.

VI.8.4.1 Influence de la taille de la population

Dans un premier temps, nous avons fixé le taux de mutation (gaussienne) à 3% et le taux de croisement à 80%. La Figure 59 montre le comportement de populations différant par leur taille. Les résultats montrent que 50 individus ne sont pas en nombre suffisant pour faire évoluer les performances du meilleur individu, alors que 100 individus permettent d'améliorer les performances de 13% en 1000 générations et 200 individus atteignent 50,5% de reconnaissance, ce qui est une amélioration de 20%.

La complexité du problème est très importante. En effet, la taille des génotypes atteint 1500, ce qui est considérable pour les EA, ce qui justifie l'emploi de population de grandes tailles.

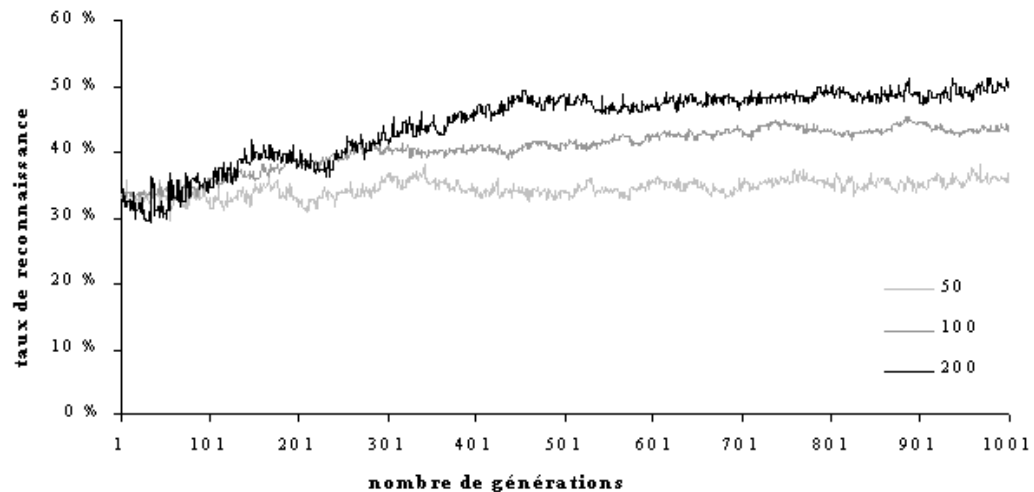


Figure 59 Influence de la taille de la population.
Evolution effectuée par GA avec mutation gaussienne d'écart-type 0,1, taux de mutation de 3% et taux de croisement de 80%.

Par soucis de garder des temps de simulation acceptables, nous décidons de fixer le nombre d'individus à 100 afin d'étudier désormais, l'influence du taux de mutation.

VI.8.4.2 Influence du taux de mutation

Nous avons fixé le taux de croisement à 80% et la taille de la population à 100. La Figure 60 montre le comportement de populations différant par leurs taux de mutation.

Les résultats montrent que le taux de mutation, lorsqu'il est fixé dans une fourchette de 3 à 10%, ne semble pas influencer sur les performances du système car on obtient environ 13% d'amélioration. Cependant, les résultats obtenus, lorsque le taux de mutation est à 0%, montrent que cette mutation reste nécessaire afin d'éviter une convergence prématurée des performances de la population.

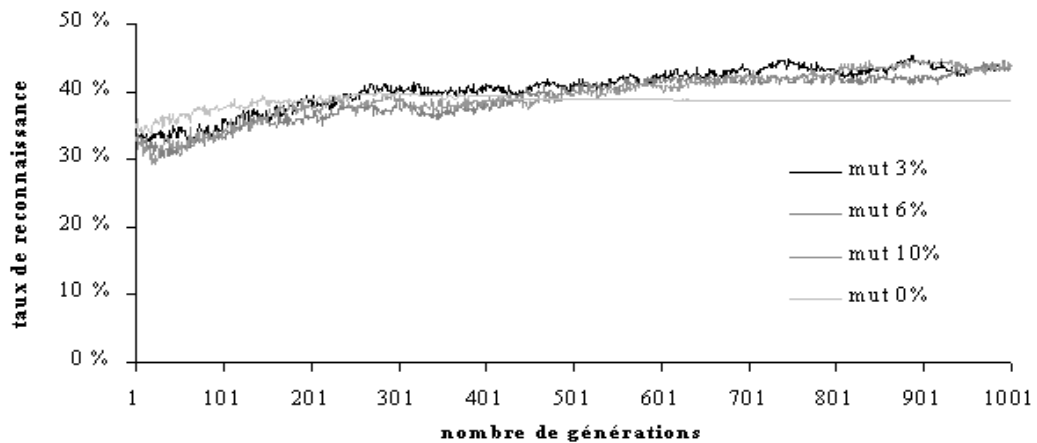


Figure 60 Influence du taux de mutation.
Evolution d'une population de 100 individus, effectuée par GA, avec mutation gaussienne d'écart-type 0,1 et taux de croisement de 80%.

Fixant alors le taux de mutation à 3%, il reste à étudier l'influence du taux de croisement.

VI.8.4.3 Influence du taux de croisement

Nous avons comparé les performances des systèmes pour différents taux de croisement. Nous avons étudié le comportement des populations pour des taux de 0%, 30%, 60% et 80%. La Figure 61 montre les résultats obtenus.

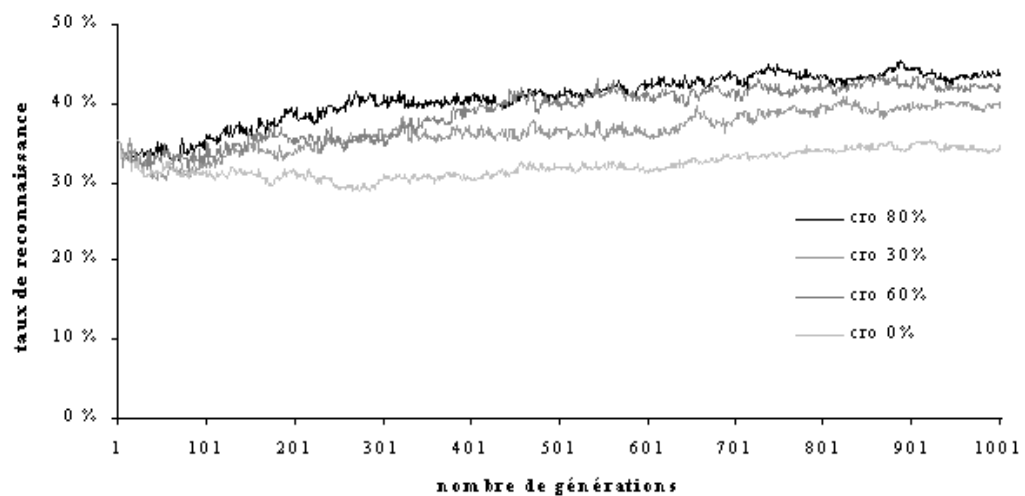


Figure 61 Influence du taux de croisement.
Evolution d'une population de 100 individus effectuée par GA avec mutation gaussienne d'écart-type 0,1 et taux de mutation de 3%.

Nous pouvons constater que le croisement joue un rôle important dans l'amélioration des performances des GA. Il semble nécessaire de croiser fortement les individus afin que les performances s'améliorent.

VI.8.5 Stratégies d'évolution

Concernant l'étude de la méthode ES, nous avons décidé de fixer le nombre d'individus à 100 car les GA ont montré que ce nombre était raisonnable. Nous allons alors étudier l'influence du nombre de parents (μ) sélectionnés pour la reproduction.

Nous avons fait varier μ de 2 à 25 afin de déterminer le degré de sélection des parents pour que l'évolution soit efficace. Sachant que μ est inversement proportionnel au degré d'élitisme, fixer μ à 2, par exemple, consiste à effectuer une sélection très forte puisque seulement 2 individus sont sélectionnés parmi les 100 de la population et ils génèrent chacun 50 enfants.

Nous pouvons constater que les populations évoluant par ES convergent très rapidement vers des solutions acceptables. Cependant, les performances moyennes semblent être proportionnelles au μ choisi. En effet, les performances atteignent 49,2% pour un (25,100)-ES alors qu'elles stagnent à 40% pour (10,100)-ES et (5,100)-ES, ou à 33,89% pour les (2,100)-ES.

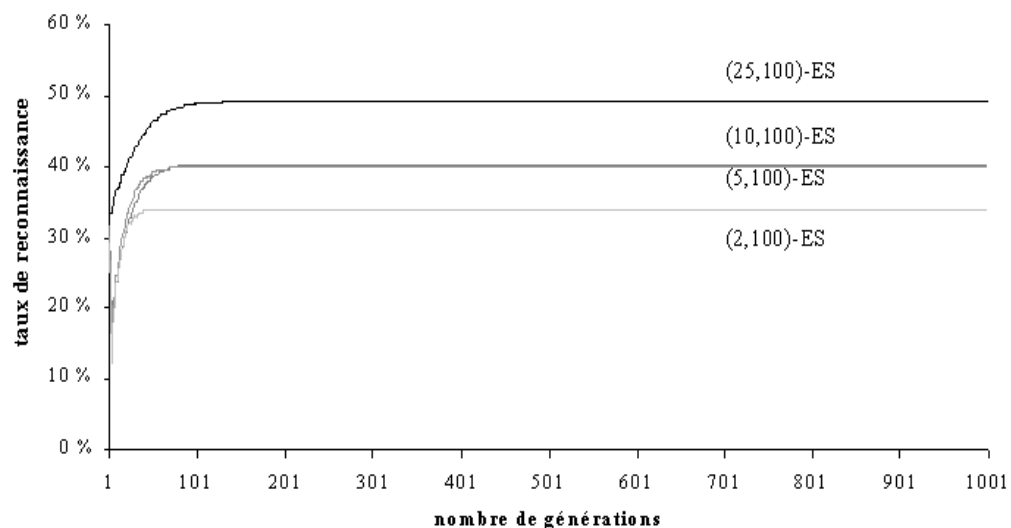


Figure 62 Influence du nombre de parents.
Evolution d'une population de 100 individus à l'aide des ES.

Il apparaît que, dans notre cas, le rapport de 4 (25 parents pour 100 individus) permet que l'évolution se fasse au mieux.

VI.8.6 Influence de l'environnement

La série d'expériences que nous avons effectuées dans les sections précédentes s'est portée sur un seul bruit : Radio 10 dB. Nous avons alors cherché à valider la meilleure méthode trouvée sur des corpus comportant différents bruits ainsi que sur un corpus non bruité. Nous avons fait varier le rapport signal à bruit de la radio, ainsi que le type de bruit : alarme, porte avec un rapport signal à bruit de 10dB.

Les résultats montrent que les ES permettent d'améliorer les performances du système dans tous les bruits que nous avons testés. Cependant, le taux d'amélioration diffère selon le bruit testé et ainsi, on peut noter une amélioration de 10,24% pour la radio à -20 dB et 20,02% pour la radio à 20dB. On peut noter que l'ACP de départ n'est pas optimale puisque les ES permettent d'augmenter de 10,04% le taux de reconnaissance du système.

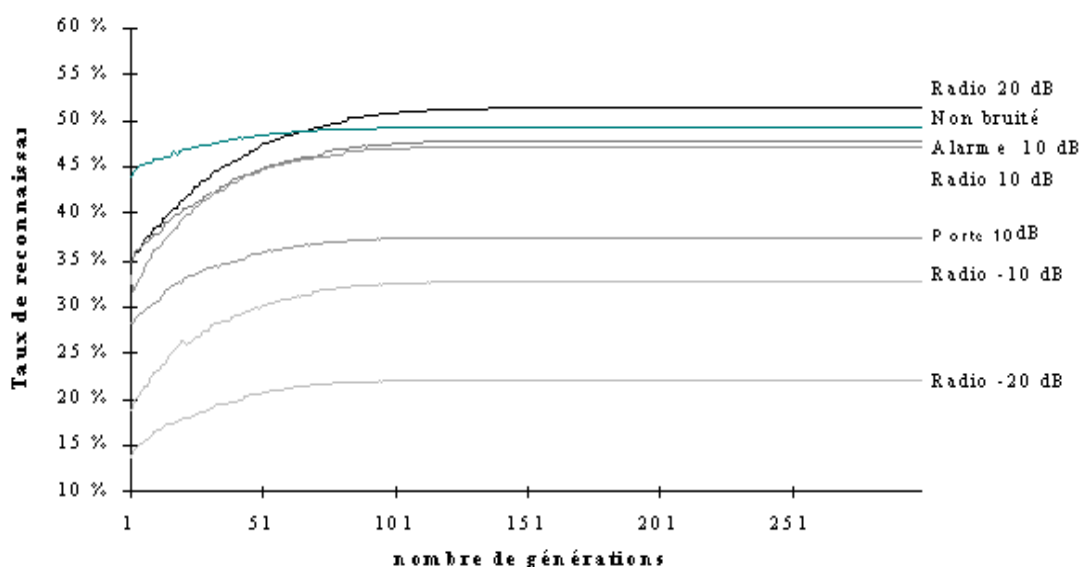


Figure 63 Influence du type de bruit.
Evolution d'une population de (25,100)-ES

La Table 11 montre les résultats chiffrés fournis par l'ACP seule et ceux fournis par les ES.

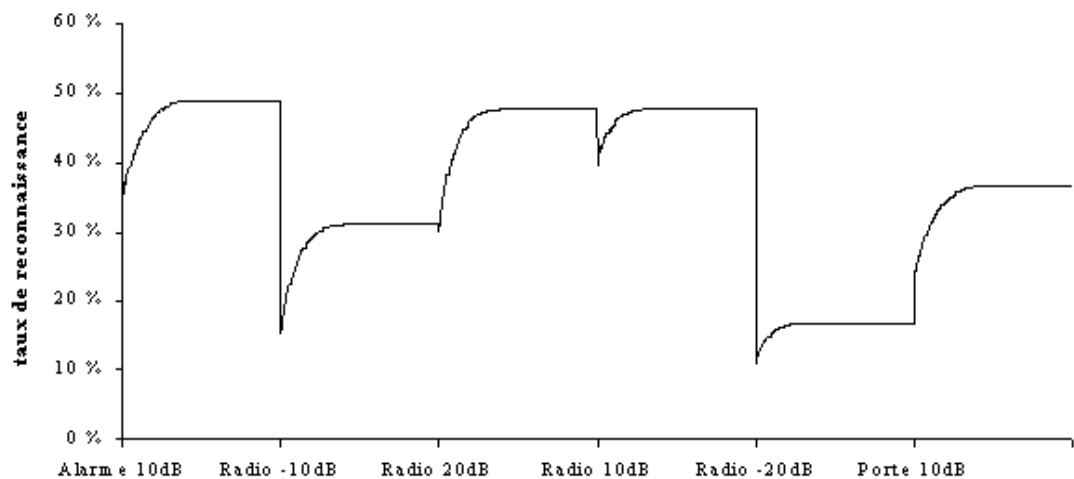
Table 11 Amélioration des performances par ES.

Type de bruit	ACP seule	ACP / ES	Amélioration
Aucun	39,28%	49,32%	10,04%
Ra -20 dB	11,64 %	21,98 %	10,34%
Ra -10 dB	15,55 %	32,66 %	17,11%
Ra 10 dB	32,28 %	47,16 %	14,88%
Ra 20 dB	31,4 %	51,42 %	20,02%
Po 10 dB	22,71 %	37,46 %	14,75%
Al 10 dB	34,88	47,83 %	12,95%

Nous allons désormais tester cette méthode lors d'une séquence de changement d'environnement.

VI.8.7 Changements d'environnement

L'expérience qui suit est effectuée sur 6 environnements différents comportant chacun la réalisation de 2500 digits. La Figure 64 montre le comportement d'une population évoluant lors de cette séquence de changements d'environnement.

**Figure 64** Performances des (25-100)-ES lors de changements d'environnements.

Tout d'abord, nous pouvons remarquer que les ES sont capables d'améliorer les performances du système de reconnaissance dans l'ensemble des environnements rencontrés. De plus, les performances obtenues après convergence de la population ne sont pas très différentes de celles obtenues lors de l'adaptation à un seul

environnement. Par exemple, partant de l'environnement de radio à -10dB, l'adaptation à l'environnement de radio à 20 dB amène le système à une performance de 47,75% de reconnaissance, alors que cette même population n'ayant pas été, au préalable, adaptée à un environnement particulier, obtient 51,42% de reconnaissance. On peut de la même manière comparer, pour chaque environnement, les performances obtenues lors de la séquence à celles lors d'un seul environnement pour remarquer que, bien que pas très différentes, les performances sont globalement plus faibles. Ainsi on obtient, lors de la séquence, 30,95% de reconnaissance au lieu de 32,66% pour la radio à -10dB, 47,84% au lieu de 47,16% pour la radio à 10dB, 36,55% au lieu de 37,46% pour la porte, etc.

VI.9 Discussion des résultats

Dans cette série d'expériences, nous avons élaboré une méthode permettant de modifier des données bruitées afin que leurs caractéristiques soient les plus proches des caractéristiques de ces mêmes données non bruitées. Utilisant une base de projection déterminée par une ACP, nous avons tenté de l'optimiser par différentes méthodes évolutionnaires. Nous avons comparé les GA et ES en faisant varier leurs paramètres d'évolution. Les deux méthodes ont montré leurs capacités à optimiser de telles bases de projection. Cependant, les ES semblaient plus performantes en temps et en qualité. Cette constatation s'est vérifiée sur différents environnements acoustiques étudiés indépendamment ou dans le contexte d'une série d'environnements.

Limitée par la complexité du problème, nous avons travaillé sur des données réduites d'un facteur 15. Notre système de reconnaissance obtenait alors un taux de reconnaissance en milieu non bruité de 40% (traité par ACP). Les résultats ont montré qu'il est possible de remonter ce taux à presque 50% lorsque l'on optimise la projection des données non bruitées. Ceci peut s'expliquer par le fait que l'ACP ne fournit pas forcément une représentation optimale des données puisqu'elle suppose que celles-ci sont linéairement indépendantes. Par notre méthode, nous avons trouvé une meilleure manière de les représenter en occultant la notion d'orthonormalité de l'ACP.

Limitée par la durée d'une simulation (300 générations ES en 6 heures), et la complexité du problème à résoudre (taille d'un génotype de 1500), nous n'avons pas tenté d'appliquer cette méthode sur des données permettant d'obtenir de meilleurs taux de reconnaissance lors de l'apprentissage en milieu non bruité. Ceci pourra faire l'objet de futures expériences. D'une manière générale, les résultats d'adaptation obtenus par l'application du GA comme méthode de pré-traitement des données de parole montrent une amélioration de la population. La méthode ACP/EA est en outre moins coûteuse en temps de calcul que la méthode d'évolution du système discutée dans le chapitre V.

Conclusion et Perspectives

Dans le cadre de cette thèse, nous avons tenté de travailler aux frontières de deux domaines très différents que sont la reconnaissance automatique de la parole et les algorithmes évolutionnaires. Les travaux que nous avons effectués sont partis de l'idée que si les systèmes de reconnaissance étaient capables de s'auto-modifier dans le temps, afin de s'adapter aux changements d'environnement acoustique, ils pourraient être beaucoup plus robustes. L'idée était de s'inspirer des capacités des êtres vivants à s'adapter à leur univers pour être les plus performants possibles dans le but de survivre. Dans le cadre de la reconnaissance de la parole, nous avons considéré les systèmes de reconnaissance automatique de la parole, ou de filtrage, comme des individus ayant à s'adapter à leur environnement acoustique changeant. Il nous paraissait intéressant d'imaginer un système s'adaptant aux changements de conditions acoustiques (caractéristiques du locuteur ou de la pièce par exemple) afin de rester performant quelles que soient ses conditions d'utilisation.

Les êtres vivants sont capables d'adapter leur manière de percevoir les signaux en se concentrant sur le signal qui les intéresse. Ils sont aussi capables de remettre à jour leurs connaissances de l'environnement lorsque cela est nécessaire. Considérant que la reconnaissance de signaux de parole chez les êtres vivants peut se résumer en deux

grandes phases, à savoir la perception des signaux et l'attribution d'entités à ces signaux, nous avons suggéré d'en adapter l'une ou l'autre de manière indépendante. Nous gardons en tête qu'il existe une forte interaction entre ces deux processus mais leur adaptation par algorithmes évolutionnaires de manière parallèle semble, pour l'instant, impossible à maîtriser. En effet, comment adapter notre système de reconnaissance sur des données qui s'adaptent simultanément à celui-ci ?

Ce que nous avons fait...

Dans un premier temps, nous avons expérimenté différentes méthodes évolutionnaires basées sur les principes des algorithmes génétiques et des stratégies d'évolution afin d'adapter des systèmes de reconnaissance basés sur des réseaux connexionnistes. Nous avons déterminé la méthode qui nous paraissait la plus adéquate au problème de l'adaptation des systèmes au changement d'environnement acoustique, à savoir les algorithmes génétiques avec mutation multiplicative et héritage Lamarckien. Cette méthode a montré de bons résultats, stables dans le temps et dans la répétition d'évènements similaires. Le type de réseau que nous avons utilisé était très optimisé afin d'éviter toute convergence prématurée, et particulièrement adapté à la reconnaissance de la parole. Ceci nous a permis d'obtenir de très bons taux de reconnaissance dans les environnements peu bruités. Cependant, lors d'environnements difficiles, ce réseau a montré certaines faiblesses que les algorithmes évolutionnaires ont pu fortement améliorer.

De plus, lorsque l'on a considéré les performances moyennes de la population, les algorithmes évolutionnaires ont fourni de meilleurs résultats que ceux d'une population n'ayant pas de capacité d'évolution. Mais lorsque les meilleurs individus de chaque population étaient comparés, les algorithmes évolutionnaires ne fournissaient de performances meilleures que dans les environnements difficiles.

En augmentant la complexité des environnements à reconnaître, lorsque les entités n'étaient plus des voyelles mais des digits, nous avons dû augmenter la complexité des réseaux connexionnistes, et les algorithmes évolutionnaires ont montré leur limite quant à la capacité d'améliorer de tels systèmes. Nous avons remarqué que

l'évolution n'était pas un processus si simple et qu'elle pouvait nuire aux performances du système. Les capacités des machines que nous avons utilisées n'ont pas permis d'augmenter suffisamment la diversité des individus, ce qui aurait peut-être abouti à de meilleures performances.

Dans un second temps, nous nous sommes placée au niveau de la perception. Nous avons tenté d'adapter les données bruitées afin que leurs caractéristiques soient le moins possible perturbées par le bruit. Le travail s'est articulé autour de la recherche d'une base de projection permettant de passer d'un espace bruité à un espace non bruité. Nous avons comparé différentes alternatives des algorithmes génétiques et des stratégies d'évolution pour préférer cette dernière méthode. Nous avons montré qu'il est possible d'améliorer sensiblement les performances d'un réseau de neurones entraîné dans un environnement non bruité lorsque la base de projection est optimisée par les algorithmes évolutionnaires, et ceci pour l'ensemble des bruits que nous avons testés. Nous avons, de plus, montré que notre système peut s'adapter à une séquence de changements d'environnement.

Ce qui reste à faire...

Si, de par l'extrême complexité de notre problème et la limitation des capacités des machines actuelles, les résultats que nous avons obtenus ne sont pas toujours à la hauteur de nos espérances, ils ont montré des caractéristiques intéressantes qui nous poussent à approfondir nos recherches dans plusieurs directions.

Tout d'abord, afin d'espérer rendre de tels systèmes exploitables, il est nécessaire d'avoir un système totalement autonome et d'obtenir ces résultats de manière non supervisée. Mais pour cela, les systèmes doivent être capables de s'auto-évaluer. Si l'on désire les adapter en arrière-plan lors de l'utilisation du système de reconnaissance global, il n'est pas envisageable de demander à l'utilisateur de valider chaque mot ou chaque phrase afin d'évaluer chaque individu de la population. C'est un problème extrêmement difficile qui commence à susciter quelques travaux en parole (Kemp et Waibel 99). Nous comptons sur des méthodes statistiques pour nous

aider à avancer sur ce problème. Les recherches dans l'utilisation de plusieurs systèmes de reconnaissance adaptés à des environnements différents, l'apport des informations fournies par un système multi-sources, multi-bandes, etc., semblent ouvrir des voies intéressantes dans le domaine. Par ailleurs, il pourrait être envisageable d'utiliser un modèle de plus haut niveau (lexical ou sémantique par exemple) pour l'aide à la décision.

Par ailleurs, il nous faut trouver une méthode évolutionnaire plus adaptée au problème de la reconnaissance de parole afin de rendre l'évolution des populations plus efficace en temps et meilleure en qualité. Cette méthode devra être suffisamment performante pour gérer des individus de grande complexité en un temps raisonnable.

La première manière, et certainement la plus simple, est d'augmenter la taille de la population. Ainsi, plus le nombre d'individus générés est grand et plus la chance de trouver un bon individu est grande. Nous n'avons pas été en mesure d'effectuer ce genre de test du fait des durées trop élevées des simulations. Dans l'avenir, il sera nécessaire de distribuer l'exécution des apprentissages sur différentes machines.

La seconde manière est de travailler plus en finesse sur la manière de sélectionner les individus, celle de les muter ou de les croiser. En effet, le but étant d'explorer des nouvelles voies de recherches encore vierges et de tester l'éventuel apport des algorithmes évolutionnaires pour la robustesse des systèmes de reconnaissance automatique de la parole, nous avons préféré valider les principes de base de l'hybridation avant de nous attaquer à l'affinement des méthodes. Il est évident aujourd'hui qu'un grand travail reste à faire dans cette voie. Nous pourrions alors ouvrir une voie de recherche tendant vers la coévolution des données et des populations (Juillé et Pollack 98), permettant d'étudier l'interaction entre les données et les systèmes de reconnaissance en intégrant progressivement des données de plus en plus complexes au fur et à mesure que les systèmes s'adaptent. Nous pourrions également envisager de faire collaborer les individus permettant d'élaborer une sorte de "super individu" dont les performances résulteraient du vote de l'ensemble de la population.

Enfin, il est tout à fait envisageable de mettre en place des populations multiples adaptées à des environnements différents, évoluant sur des plateformes différentes, avec des individus migrant d'une population à une autre (Cantu-Paz 99). Dans le même ordre d'idée, il serait intéressant de faire évoluer des populations adaptées, non pas à un environnement, mais à un type de vocabulaire. La recherche dans le domaine de la robustesse en reconnaissance de la parole sur petits vocabulaires (de l'ordre de 10 à 100 mots) a un intérêt pour de multiples applications (par exemple, la reconnaissance de digits en voiture (Mokbel et Chollet 91) pour composer un numéro de téléphone ou composer un code). Dans ce type d'application, si l'on se borne à la reconnaissance de digits, on pourrait envisager de créer 10 populations, chacune adaptée à la reconnaissance d'un digit.

Si tout cela était fait...

Nous pourrions alors envisager l'intégration d'un tel système dans un système de reconnaissance classique.

- **Pour l'adaptation du système**

Ainsi, nous pourrions disposer de plusieurs systèmes entraînés dans des environnements différents et d'un système capable de reconnaître de tels environnements afin de diriger l'information du signal de parole vers le système de reconnaissance le plus adapté aux conditions acoustiques. Si aucun des systèmes ne convient, le système combinant réseaux de neurones et algorithmes évolutionnaires serait alors mis en route afin de définir un nouveau système adapté à un nouvel environnement.

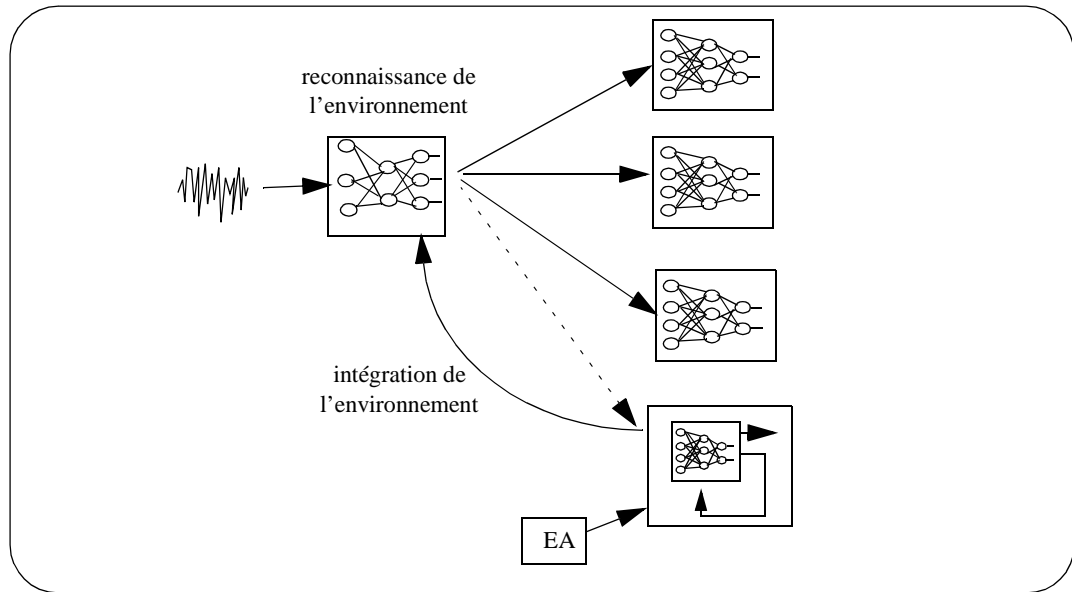


Figure 65 Modèle d'hybridation EA/NN pour l'adaptation du système.

- **Pour l'adaptation des données**

Le modèle présenté ci dessous permet d'encapsuler notre système (algorithmes évolutionnaires + Base + réseaux de neurones) dans un système plus général permettant de garder les bases de projection des environnements déjà rencontrés et de créer une base pour un nouvel environnement.

Comme dans les expériences que nous avons menées (voir chapitre "*Adaptation des données*"), le système de reconnaissance est entraîné dans un environnement non bruité. Un ensemble de bases de projection de données bruitées provenant d'environnements connus lui permet de garder des performances correctes.

- *Un système de reconnaissance*

Nous proposons d'entraîner un système de reconnaissance de types d'environnement afin qu'il choisisse la base de projection adéquate. Si ce système n'a pas reconnu l'environnement, une nouvelle base de projection est créée et optimisée à l'aide des algorithmes évolutionnaires. Le système de reconnaissance d'environnement est alors mis à jour pour intégrer le nouvel environnement.

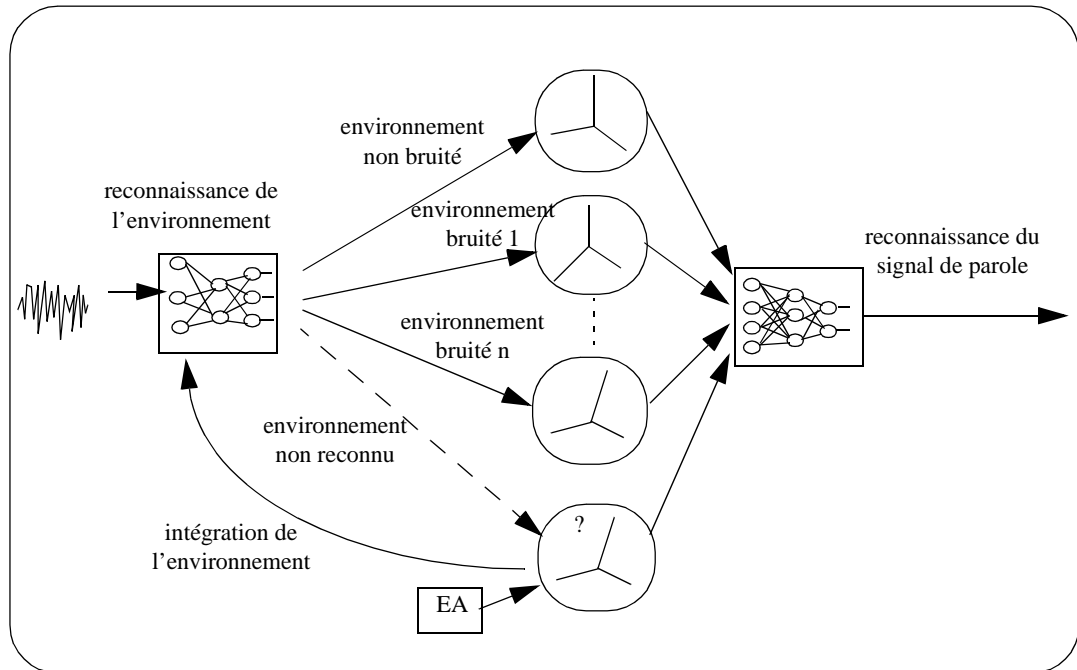


Figure 66 Modèle d'hybridation EA/ACP pour l'adaptation des données.

- *Plusieurs systèmes de reconnaissance de la parole*

Pour ne pas nous restreindre à un seul système de reconnaissance de la parole entraîné dans un environnement non-bruité, nous proposons d'associer à chaque base de projection un système de reconnaissance. Ainsi, sans connaissance *a priori* sur l'environnement, un test de reconnaissance est effectué pour chaque doublet (base-réseaux de neurones) et le meilleur système est sélectionné. On se rapproche alors des solutions proposées par les méthodes basées sur l'analyse multi-bandes (Bourlard et Dupont 97) (Tibrewala et Hermansky 97).

Si aucun des systèmes ne fournit de résultats satisfaisants, une nouvelle base est créée et une recherche du meilleur système de reconnaissance parmi ceux existants lui est associé.

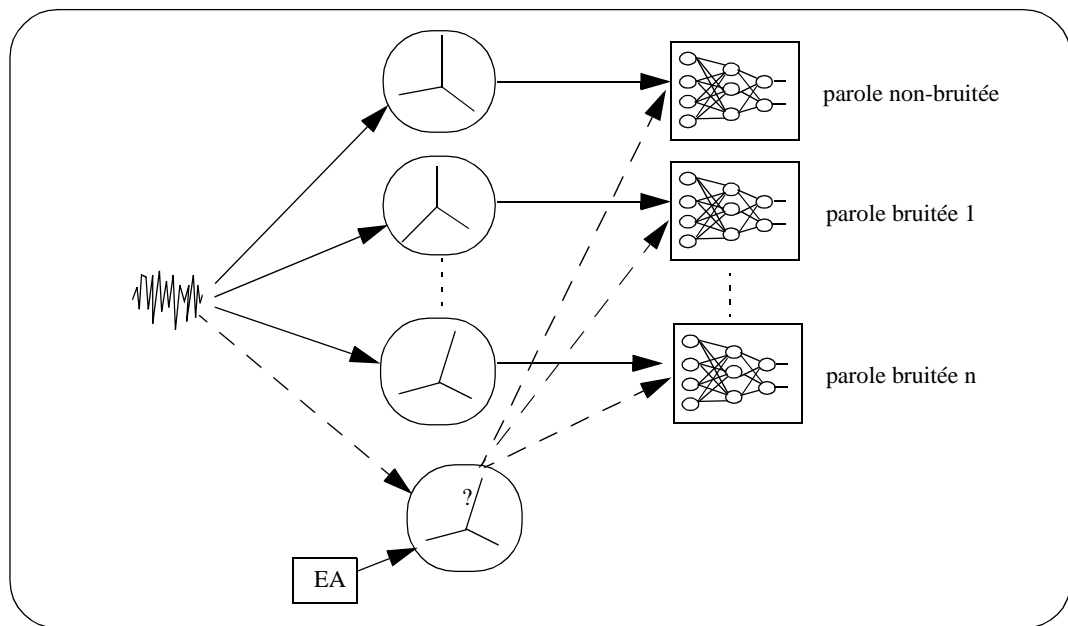


Figure 67 Modèle de projection multi-systèmes pour l'adaptation des données.

Nous pensons que ce type de systèmes pourrait permettre d'élaborer des méthodes statistiques. Par exemple, il est possible d'évaluer le degré de l'impact du bruit en comparant le résultat de la projection d'un signal de parole sur chacune des bases avec la moyenne des signaux projetés en phase d'apprentissage.

Enfin, nous désirons élargir notre recherche à l'application des algorithmes évolutionnaires aux modèles markoviens puisque ces derniers semblent de plus en plus faire l'unanimité dans le monde de la reconnaissance automatique de la parole.

Nous souhaitons, à travers ce travail, avoir contribué à l'exploration de nouvelles voies de recherche dans la robustesse des systèmes de reconnaissance de la parole. Nous sommes encore loin de comprendre les capacités cognitives humaines qui permettent à l'homme de s'adapter à de multiples conditions acoustiques, mais nous sommes toujours persuadée que la voie de l'évolution reste une idée qu'il faut approfondir.

Références

- (Ahuactzin et al. 95) Ahuactzin J-M., Mazer E. et Bessière P., *Fondements mathématiques de l'algorithme "fil d'Ariane"*, Revue d'Intelligence Artificielle, 1995.
- (Allen 94) Allen J.B., *How Do Humans Process and Recognize Speech ?* IEEE Transactions on Speech and Audio Processing, vol. 2, n. 4, pp. 567-577, 1994.
- (Allen et Berkley 79) Allen J.B. et Berkley D.A., *Image Method for Efficiently Simulating Small-Room Acoustics*. Journal of Acoustical Society of America, 65(4):943-950, 1979.
- (Bailer-Jones 96) Bailer-Jones C., *Neural Network Classification of Stellar Spectra*. Ph-D thesis, Institute of Astronomy, University of Cambridge, Chapitre 4, pp 77-120, 1996.
- (Bäck 92) Bäck T., *Self-Adaptation in Genetic Algorithms*. Proceedings of the 1st European Conference on Artificial Life, pp. 263-271, ed. F.J. Varela and P. Bourguin, The MIT Press, Cambridge MA, 1992.
- (Bäck 96) Bäck T., *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
- (Bäck 98) Bäck T., *An Overview of Parameter Control Methods by Self-Adaptation in Evolutionary Algorithms*. Fundamenta Informaticae 34, IOS Press, pp 1-15, 1998.
- (Baluja 98) Baluja S., *Finding Regions of Uncertainty in Learned Models : An Application to Face Detection*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, pp. 663-671, Amsterdam,

- Pays-Bas, 1998.
- (Barbier et Chollet 91) Barbier L. et Chollet G., *Robust Speech Parameters Extraction for Word Recognition in Noise Using Neural Networks*. ICASSP'91, pp. 145-148, 1991.
- (Barnard et al. 95) Barnard E., Cole R., Fanty M. et Vermeulen P., *Real-World Speech Recognition With Neural Networks*. International Workshop on Applications of Neural Networks to Telecommunications, Stockholm, Sweden May 22-24, 1995.
- (Baxter 92) Baxter J., *The Evolution of Learning Algorithms for Artificial Neural Networks*. Complex Systems, ed. Green D. et Bossomaier T. IOS Press, 1992.
- (Belew 89) Belew R., *When Both individuals and populations search : Adding simple learning to the genetic algorithm*. Proceedings of the Third Conference on Genetic Algorithms. Morgan Kaufmann (San Mateo), 1989.
- (Belew et al. 90) Belew B., McInerney J. et Schraudolph N., *Evolving Networks : Using the Genetic Algorithms with Connectionist Learning*. CSE Technical Report CS90-174, Computer Science, UCSD, 1990.
- (Boëffard et Emerard 97) Boëffard O. et Emerard F., *Application-dependent Prosodic Models for Text-to-Speech Synthesis and Automatic Design of Learning Database Corpus Using Genetic Algorithm*. EuroSpeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Bonabeau et al. 94) Bonabeau E., Théraulaz G., Arpin E. et Sardet E., *The building behavior of lattice swarms*. Artificial Life IV, édité par Brooks R. et Maes P., MIT Press, 1994.
- (Bottou 88) Bottou L-Y., *Reconnaissance de la parole par réseaux multi-couches*. Proceedings of the International Workshop on Neural Networks and Their Applications, pp 197-217, 1988.
- (Bourlard 96) Bourlard H., *Reconnaissance Automatique de la Parole : Modélisation ou Description ? XXI^{èmes} Journées d'Etude sur la Parole (JEP'96)*, Avignon, France, pp. 263-272, 1996.
- (Bourlard et Dupont 97) Bourlard H. et Dupont S., *Subband-based Speech Recognition*. ICCASP'97, pp. 1251-1254.
- (Bradford 95) Bradford J.H., *The human factors of speech-based interfaces*, SIGCHI Bulletin, vol. 27, n°2, pp. 61-67, 1995.
- (Bregler et Konig 94) Bregler C. et Konig Y., *"Eigenlips" for Robust Speech Recognition*.

- Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Adelaide, Australia, 1994.
- (Brouard et al. 98) Brouard T., Slimane M., Venturini G. et Asselin De Bauville J-P., *Apprentissage génétique hybride de chaînes de Markov cachées*. Apprentissage : des principes naturels aux méthodes artificielles, ed. Ritschard G., Berchtold A., Duc F. et Zighed D., éditions Hermès, pp.241-256, 1998.
- (Burel 91) Burel G., *Réseaux de neurones en traitement d'images : des modèles théoriques aux applications industrielles*. Thèse de doctorat de l'Université de Bretagne Occidentale, 1991.
- (Burr 88) Burr D.J., *Speech Recognition Experiments with Perceptrons*. Neural Information Processing Systems, ed. D. Anderson, 144-153, New-York : American Institute of Physics, 1988.
- (Caelen 85) Caelen J., *Space/Time Data-Information in the A.R.I.A.L. Project Ear Model*. Speech Communication 4 (1985) 163-179, Elsevier Science Publishers B.V., North-Holland, 1985.
- (Caelen et al. 96) Caelen J., Kabré H. et Delemar O., *Reconnaissance de la Parole : vers l'Utilisabilité*. XXI^{èmes} Journées d'Etude sur la Parole (JEP'96), Avignon, France, pp.325-329, 1996.
- (Cantu-Paz 99) Cantu-Paz E., *Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms*. IlliGAL Report No. 99015, Juin 99.
- (Cardoso 89) Cardoso J.F., *Source Separation Using Higher Order Moments*. ICASSP'89, Glasgow, Ecosse, vol. 4, pp. 2109-2112, Mai 1989.
- (Chalmer 90) Chalmer D.J., *The Evolution of Learning : An Experiment in Genetic Connectionism*. Proceedings of the 1990 Connectionist Summer School Workshop, 1990.
- (Chang et Zue 94) Chang J. et Zue V., *A Study of Speech Recognition System Robustness to Microphone Variations : Experiments in Phonetic Classification*. ICSLP, pp. 995-998, 1994.
- (Chen et Nutter 91) Chen C.L. et Nutter R.S., *Improving the Training Speed of Three-Layer FeedForward Neural Nets by Optimal Estimation of the Initial Weights*. International Joint Conference on Neural Networks, IEEE, 1991.
- (Chellapilla et Fogel 97) Chellapilla K. et Fogel D., *Two New Mutation Operators for Enhanced Search and Optimization in Evolutionary Programming*, in Applications of Soft Computing, B. Bosacchi, J.C. Bezdek, and D.B. Fogel (eds.), Proc. SPIE Vol. 3165, pp. 260-269, 1997.

- (Chiang 97) Chiang T., *Speech Recognition in Noise Using On-line HMM Adaptation*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Christiansen 99) Christiansen M.H., *Connectionist Natural Language Processing : The State of the Art*. Special Issue of Cognitive Science on Connectionist Models of Human Language Processing : Progress and Prospects, édité par Christiansen M.H, Chater N. & Seidenberg M.S., 1999.
- (Cobb et Grefenstette 93) Cobb H.G. et Grefenstette J.J., *Genetic Algorithms for Tracking Changing Environments*. International Conference on Genetic Algorithms, Fifth International Conference on Genetic Algorithms (ICGA 93), 523-530, Morgan Kaufmann, July 1993.
- (Davis et Mermelstein 80) Davis S. et Mermelstein P., *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. IEEE Transaction ASSP, ASSP-28(4):357-366, 1980.
- (De Jong 75) De Jong, K., *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. thesis, University of Michigan, 1975.
- (De Jong 88) De Jong K.A., *Learning with Genetic algorithm : An Overview*. Machine Learning, vol. 3, pp. 121-138, 1988.
- (Denoeux et Lengellé 93) Denoeux T. et Lengellé R., *Initializing Back Propagation Networks With Prototypes*. Neural Networks, Vol. 6, pp 351-363, 1993.
- (Deroo et al. 98) Deroo O., Ris C., Malfrere F. et Dutoit T., *Modèles hybrides et reconnaissance de la Parole Indépendante du Locuteur en Français*. Proc. XXIIèmes Journées d'Etudes sur la Parole, Martigny, pp. 401-404, 1998.
- (Descout et al. 86) Descout R., Serignat J.F., Cervantes O., Carre R., *BDSOONS : Une base de données des sons du français*. 12th International Congress on Acoustic, Toronto, Canada, Vol. A, pp. 4 -7, 1986.
- (Devillers 92) Devillers L., *Reconnaissance de la parole continue avec un système hybride neuronal et markovien*. Thèse de doctorat de l'Université PARIS XI, 1992.
- (Drogoul 93) Drogoul A., *De la Simulation Multi-Agents à la Résolution Collective de Problèmes*. Thèse de doctorat de l'Université Paris VI, 1993.
- (Dupont et al. 97) Dupont S., Boulard H. et Ris C., *Robust Speech Recognition Based on Multi-Stream Features*. Robust Speech Recognition For Unknown Communication Channels. Pont-à-Mousson, France, 1997.
- (Eiben et al. 98) Eiben A.E., van der Hauw J.K. et van Hemert J.I., *Graph Coloring with Adaptive Evolutionary Algorithms*. Journal of Heuristics, vol. 4:1, pp. 25-46, 1998.

- (Ephraim 92a) Ephraim Y., *Statistical-Model-Based Speech Enhancement Systems*. Proceeding IEEE, 80(10):1526-1555, 1992.
- (Ephraim 92b) Ephraim Y., *A Bayesian Estimation Approach for Speech Enhancement Using Hidden Markov Models*. IEEE Transaction on Signal Processing, 40(4):775-735, 1992.
- (Ephraim et Malah 83) Ephraim Y. et Malah D., *Speech Enhancement Using Optimal Non-Linear Spectral Amplitude Estimation*. ICASSP'83, pp. 1118-1121, 1983.
- (Fahlman 88) Fallhman S.E., *An Empirical Study of Learning Speed in Backpropagation Networks*. Technical report CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1988.
- (Flanagan et Mammone 91) Flanagan J.L. et Mammone R., *Autodirective Microphone Systems for Communication with Recognizers*. DARPA workshop, pp. 1-6, Février, 1991.
- (Fogarty 89) Fogarty T.C., *Varying the probability of mutation in the genetic algorithm*. In Schaffer, J. D. Ed. Proceedings of the Third International Conference on Genetic Algorithms (Morgan Kaufmann): 104-109, 1989.
- (Fontanili et Vincent 97) Fontanili F. et Vincent A., *Comment optimiser le fonctionnement d'un atelier avec la simulation de flux ?* Revue Française de Gestion Industrielle, vol. 16, n. 3, 1997.
- (Geyer et al. 98) Geyer H., Ulbig P. et Schulz S., *Encapsulated Evolution Strategies for the Determination of Group Contribution Model Parameters in Order to Predict Thermodynamic Properties*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, pp. 663-671, Amsterdam, Pays-Bas, 1998.
- (Goldberg 89) Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Massachusetts, 1989.
- (Gong 95) Gong Y., *Speech Recognition in Noisy Environments : Survey*. Journal of Speech Communication 16, pp 261-291, 1995.
- (Gong 97) Gong Y., *Source Normalization Training for HMM Applied to Noisy Telephone Speech Recognition*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Gong et Haton 94) Gong Y. et Haton J.P., *Stochastic Trajectory Modeling for Speech Recognition*. ICASSP'94, pp. 57-60, 1994.
- (Gouvêa et Stern 97) Gouvêa E.B. et Stren R.M., *Speaker Normalization Through Formant-Based Warping of the Frequency Scale*. Eurospeech'97, 5th

- European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Grönroos 98) Grönroos M., *Evolutionary Design of Neural Networks*. Master Thesis in Computer Science, Department of Mathematical Sciences, Université de Turku, Finlande, 1998.
- (Gruau et Whitley 93) Gruau F. et Whitley D., *Adding Learning to the Cellular Development of Neural Networks : Evolution and the Baldwin Effect*. *Evolutionary Computation* 1(3): 213-233, 1993.
- (Guan et al. 97) Guan C-T., Leung S-H. et Lau W-H., *A Space Transformation Approach for Robust Speech Recognition in Noisy Environments*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Han et al. 97) Han J., Han M., Park G-B., Park J. et Gao W., *Relative Mel-Frequency Cepstral Coefficients Compensation for Robust Telephone Speech Recognition*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Hancock 92a) Hancock P.J.B., *Genetic Algorithms and Permutation Problems : a Comparison of Recombination Operators for Neural Structure Specification*. *Combinations of Genetic Algorithms and Neural Networks*. Ed. D. Whitley et J.D. Schaffer. IEEE Computer Society Press, 1992.
- (Hancock 92b) Hancock P.J.B., *Coding Strategies for Genetic Algorithms and Neural Nets*. Ph.D. Thesis, University of Stirling, 1992.
- (Harp et al. 89) Harp S.A., Samad T. et Guha A., *Toward the Genetic Synthesis of Neural Networks*. *Proceedings of the Third Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1989.
- (Haton 95) Haton J-P., *Les Modèles Neuronaux et Hybrides en Reconnaissance Automatique de la Parole : Etats des Recherches*. Ecole Thématique sur les Fondements et Perspectives en Traitement Automatique de la Parole, Centre de Formation du CNRS de Marseille-Luminy, 1995.
- (Haton 97) Haton J-P., *Proceeding of the ESCA-NATO Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*. Pont-à-Mousson, France, 1997.
- (Heap et Samaria 95) Heap T. et Samaria F., *Real-Time Hand Tracking and Gesture Recognition Using Smart Snakes*. Technical Report 95.1 AT&T Laboratories Cambridge, 24a Trumpington Street, Cambridge CB2 1QA, England, 1995.
- (Hérault et Jutten 94) Hérault J. et Jutten C., *Réseaux neuronaux et traitement du signal*. Ed. Hermès, Paris, 1994.

- (Hermansky 90) Hermansky H. *Perceptual Linear Predictive (PLP) Analysis of Speech*. Journal of Acoustic Society Am. 87(4) 1738-1752, 1990.
- (Hermansky 97) Hermansky H. *Should Recognizer Have Ears ? Robust Speech Recognition For Unknown Communication Channels*. Pont-à-Mousson, France, 1997.
- (Heudin 94) Heudin J-C., *La Vie Artificielle*, Edition Hermès, 1994.
- (Hiebeler 94) Hiebeler D., *The Swarm Simulation System and Individual-based Modeling*. Decision Support 2001: Advanced Technology for Natural Resource Management, Toronto, 1994.
- (Hild et Waibel 93) Hild H. et Waibel A., *Speaker-Independent Connected Letter Recognition with a Multi-State Time Delay Neural Network*. 3rd European Conference on Speech, Communication and Technology (EuroSpeech' 93), Berlin, Germany, pages 1481 - 1484, September 1993.
- (Holland 75) Holland J.H., *Adaptation in Natural and Artificial Systems*. Ann Arbor : The University of Michigan Press, 1975.
- (Huerta et Stern 97) Huerta J.M. et Stern R.M., *Compensation for Environmental and speaker Variability by Normalization of Pole Locations*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Iwata et al. 90) Iwata M., Kitani I., Yamada H., Iba H. et Higuchi T., *A Pattern Recognition System Using Evolvable Hardware*. 1st Conference on Parallel Problem Solving from Nature, Berlin, Allemagne, 1990.
- (Jourlin et al. 97) Jourlin P., Luetin J., Genoud D. et Wassner H., *Integrating Acoustic and Labial Information for Speaker Identification and Verification*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Jouvet 95) Jouvet D., *Modèles de Markov pour la Reconnaissance de la Parole*. Ecole Thématique sur les Fondements et Perspectives en Traitement Automatique de la Parole, Centre de Formation du CNRS de Marseille-Luminy, 1995.
- (Juang et Rabiner 87) Juang B. et Rabiner L., *Signal Restoration by Spectral Mapping*. ICASSP'87, pp 2368-2371, 1987.
- (Juillé et Pollack 98) Juillé H. et Pollack J.B., *Coevolutionary Learning : a Case Study*. Proceedings of the Fifteenth International Conference on Machine Learning, Madison, Wisconsin, July 24 - 26, 1998.
- (Junqua 87) Junqua J-C., *Evaluation of ASR Front-ends in Speaker-dependent and Speaker-independent Recognition*. Journal of Acoustical Society of

- America, 81 S1:S93, 1987.
- (Junqua 93) Junqua J-C., *The Lombard Reflex and its Role on Human Listeners and Automatic Speech Recognizers*. Journal of Acoustical Society of America, 93(1):510-524, 1993.
- (Junqua 97) Junqua J-C., *Impact of the Unknown Communication Channel on Automatic Speech Recognition : A Review*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Junqua et Haton 96) Junqua J-C et Haton J-P, *Robustness in Automatic Speech Recognition, Fundamentals and Applications*. Kluwer Academic Publishers, 1996.
- (Jutten et al. 91) Jutten C., Héroult J., Comon P. et Sorouchyari E., *Blind Separation of Sources*. Signal Processing, vol. 24, pp. 1-29, 1991.
- (Kabré 95) Kabré H., *Audio-visual Speech Recognition Using the Fuzzy Shape Filter Model*. Eurospeech'95, pp. 307-312, 1995.
- (Kabré 97) Kabré H., *Audio-visual Talker Localization for Hand-free Speech Recognition*. Robust Speech Recognition For Unknown Communication Channels. Pont-à-Mousson, France, 1997.
- (Kabré et Spalanzani 97) Kabré H. et Spalanzani A., *EVERA : A System for the Modeling and Simulation of Complex Systems*. Proceeding of the First International Workshop on Frontiers in Evolutionary Algorithms, pp 184-188, North Carolina, 1997.
- (Kabré et al. 98) Kabré H., Caelen J. et Spalanzani A., *Modèles Evolutifs pour la Perception Artificielle des Sons et des Images*. Apprentissage : des principes naturels aux méthodes artificielles, édité par G. Ritschard G., Berchtold A., Duc F. et Zighed D., éditions Hermès, 1998.
- (Kane et Schoenauer 97) Kane C. et Schoenauer M., *Optimisation Topologique de Formes par Algorithmes Génétiques*. Revue Française de Mécanique 4, pp 237-246, 1997.
- (Kemp et Waibel 99) Kemp T. et Waibel A., *Unsupervised Training of a Speech Recognizer : Recent Experiments*. Proceedings of the Eurospeech 99, Budapest, Hongrie, September 1999, à paraître.
- (Kolen et Pollack 90) Kolen J.F. et Pollack J.B., *Back-propagation is sensitive to initial conditions*. Technical Report 90-JK-BPSIC, CIS Dept., Ohio St Univ., Columbus, Ohio, 1990.
- (Korning 94) Korning P.G., *Training of Neural Networks by Means of Genetic Algorithm Working on Very Long Chromosomes*. Technical Report, Com-

- puter Science Department, Aarhus C, Danemark, 1994.
- (Langton 96) Langton C.G., *Artificial Life, an Overview*. MIT Press, Cambridge, Massachusetts, 1996.
- (Le Cun 87) Le Cun Y., *Modèle connexionniste de l'apprentissage*. Thèse de doctorat de l'Université de Paris VI, 1987.
- (Lee et al. 93) Lee Y., Oh S.-H. et Kim M.W., *An Analysis of Premature Saturation in BackPropagation Learning*. Neural Networks, vol. 6, pp. 719-728, 1993.
- (Lee et al. 97) Lee S., Potamianos A. et Narayanan S., *Analysis of Children's Speech Duration, Pitch and Formants*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Lippmann 89) Lippmann R. P., *Review of Neural Networks for Speech Recognition*. Neural Computation 1, pp. 1-38, MIT Press, Massachusetts, 1989.
- (Liu et Wechsler 98) Liu C. et Wechsler H., *Face Recognition Using Evolutionary Pursuit*. Fifth European Conference on Computer Vision, ECCV'98, Université de Freiburg, Allemagne, juin 1998.
- (Lleida et al. 90) Lleida E., Nadeu C., Monte E., Mariño J.B., *Statistical Feature Selection for Isolated Word Recognition*, ICASSP'90, pp. 757-760.
- (MacLennan 91) MacLennan B., *Synthetic Ethology : An Approach to the Study of Communication*. Artificial Life II, ed. Langton C.G., Taylor C., Farmer J.D. et Rasmussen S., Addison-Wesley, 1991.
- (Mandischer 93) Mandischer M., *Representation and Evolution in Neural Networks*. Artificial Neural Nets and Genetic Algorithms Proceedings of the International Conference at Innsbruck, Austria, pages 643-649, Albrecht R.F., Reeves C.R. and Steele N.C. editors, Springer, Wien and New York, 1993.
- (Martinez et al. 97) Martinez R., Alvarez A., Nieto V., Rodellar V. et Gomez P., *ASR in Highly Non-Stationary Environment using Adaptive Noise Cancelling Techniques*. Robust Speech Recognition For Unknown Communication Channels. Pont-à-Mousson, France, 1997.
- (Matrouf 97) Matrouf D., *Adaptation des modèles Acoustiques pour la Reconnaissance de la Parole*. Thèse de doctorat de l'Université Paris-Sud, 1997.
- (Mayley 96) Mayley G., *Landscapes, Learning Costs and Genetic Assimilation*. Special Issue of Evolutionary Computation on the Baldwin Effect, vol. 4, n. 3, 1996.
- (Menczer et Parisi 90) Menczer F. et Parisi D., *"Sexual" Reproduction in Neural Networks*.

- Technical Report PCIA-90-06. C.N.R. de Rome, 1990.
- (Menczer et al. 96) Menczer F., Cecconi F. et Belew R.K., *From Complex Environments To Complex Behaviors*. Adaptive Behavior 4:3-4, Special Issue on Environment Structure, édité par Todd P. , Winter-Spring 1996.
- (Menczer et Belew 96) Menczer F. et Belew R.K., *Latent Energy Environments*. In Adaptive Individuals in Evolving Populations: Models and Algorithms, édité par Belew R.K. et Mitchell M., SFI Studies in the Sciences of Complexity, vol. XXIII, Reading, MA: Addison Wesley, 1996.
- (Michalewicz 96) Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Ed Springer-Verlag, 1996.
- (Miller et al. 89) Miller G.F., Todd M. et Hegde S.U., *Designing Neural Networks using Genetic Algorithms*. Proceedings of the Third Conference on Genetic Algorithms, San Mateo, Morgan Kaufmann, 1989.
- (Mokbel et Chollet 91) Mokbel C. et Chollet G., *Word Recognition in the Car. Speech Enhancement / Spectral Transformations*. ICASSP'91, pp. 925-928, 1991.
- (Mori et al. 96) Mori N., Kita H. et Nishikawa Y., *Adaptation to a Changing Environment by Means of the Thermodynamical Genetic Algorithm*. 4th Conference on Parallel Problem Solving from Nature, Berlin, Allemagne, 1996.
- (Nakisa et Plunkett 98) Nakisa R.C. et Plunkett K., *Evolution of a Rapidly Learned Representation for Speech*. Language and Cognitive Processes. Special issue on Language Acquisition and Connectionism, edited by Kim Plunkett. pp 105-128, 1998.
- (Néel et al. 96) Néel F., Chollet G., Lamel L., Minker W. et Constantinescu A., *Reconnaissance et Compréhension de la Parole : Evaluation et Applications*. Fondement et Perspectives en Traitement Automatique de la Parole, Ed. H. Méloni, AUPELF-UREF, 1996.
- (Ngyen-Thi et al. 96) Ngyen-Thi H.L., Kabré H., Jutten C., et Caelen J., *Séparation of sources : A Method for Speech Enhancement*. Journal of Applied Signal Processing, 3:177-190, 1996.
- (Nolfi et al. 94) Nolfi S., Elman J.L. et Parisi D., *Learning and Evolution in Neural Networks*. Technical Report 94-08, Department of Neural Systems and Artificial Life, Rome, Italie, Juin 94.
- (Nolfi et Parisi 96) Nolfi S. et Parisi D., *Learning to Adapt to Changing Environments in Evolving Neural Networks*. Technical Report 95-15, C.N.R. de Rome, Italie, Septembre 1996.

- (Oussedik et Delahaye 98) Oussedik S. et Delahaye D., *Reduction of Air Traffic Congestion by Genetic Algorithms*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, Amsterdam, Pays-Bas, 1998.
- (Paechter et al. 98) Paechter B., Rankin R.C., Cumming A. et Fogarty T.C., *Timetabling the Classes of an Entire University with an Evolutionary Algorithm*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, Amsterdam, Pays-Bas, 1998.
- (Peeling et Moore 87) Peeling S.M. et Moore R.K., *Experiments in Isolated digit Recognition Using the Multi-layer Perceptron*. Technical Report 4073, Royal Speech and Radar Establishment, Malvern, Worcester, Angleterre, 1987.
- (Puel 97) Puel J-B., *Neural Networks Contribution to Robust HMM Systems*. Robust Speech Recognition For Unknown Communication Channels. Pont-à-Mousson, France, 1997.
- (Radcliffe 91) Radcliffe N.J., *Genetic Set recombination and its application to neural network topology optimization*. Technical report EPCC-TR-91-21. University of Edinburg, Ecosse, 1991.
- (Radcliffe et Surry 97) Radcliffe H.J. et Surry P.D., *Real representations*. Foundations of Genetic Algorithms 4, p 51-72. Ed. Whitley L.D. et Belew R.K., Morgan Kaufmann, 1997.
- (Revéret et al. 97) Revéret L., Garcia F., Benoît C. et Vatikiotis-Bateson E., *An Hybrid Image Processing Approach to Liptracking Independent of Head Orientation*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Robert-Ribes 95) Robert-Ribes J., *Modèles d'intégration audiovisuelle de signaux linguistiques : de la perception humaine à la reconnaissance*. Thèse de doctorat de l'Institut National Polytechnique de Grenoble, Institut de la Communication Parlée, 1995.
- (Robert-Ribes et al. 97) Robert-Ribes J., Lallouache T., Escudier P. et Schwartz J-L., *Integrating Auditory and Visual Representations for Audiovisual Vowel Recognition*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Robinson et Fallside 88) Robinson M. et Fallside F., *Static and dynamic error propagation networks with application to speech coding*. Neural Information Processing Systems, Anderson Ed., 632-641, New York, American institute of physics, 1988.
- (Roux et Jacq 93) Roux C. et Jacq J.J., *Registration of successive DSA images using a simple genetic algorithm with a stochastic performance function*. Pro-

- ceeding of 19th IEEE Northeast Bioengineering Conf., Newark NJ, 223-224, 1993.
- (Rudnick 90) Rudnick M., *A Bibliography of the Intersection of Genetic Search and Artificial Neural Networks*. Technical Report CS/E 90-001, Oregon Graduate Institute, Beaverton, 1990.
- (Rumelhart et al. 86) Rumelhart D.E., Mc Clelland J.L. et PDP research group. *Parallel Distributed Processing. Exploration in the microstructure of cognition*. Bradford Book, MIT Press, Cambridge (MA), 1986.
- (Sagayama 99) Sagayama S., *Differential Approach to Acoustic Model Adaptation*. Workshop on robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finlande, Mai 1999.
- (Sasaki et Tokoro 97) Sasaki T. et Tokoro M., *Adaptation toward Changing Environments : Why Darwinian in Nature ?* Fourth European Conference on Artificial Life, 1997.
- (Schalkwyk et Fanty 96) Schalkwyk J. et Fanty M., *The CSLU-C Toolkit for Automatic Speech Recognitions*, Technical Report no. CSLU-012-96, 1996.
- (Schoenauer 95) Schoenauer M., EvolC, <http://www.eark.polytechnique.fr/EvolC.html>.
- (Schultz et Grefenstette 92) Schultz A.C. et Grefenstette J.J., *Using a Genetic Algorithm to Learn Behaviors for Autonomous Vehicles*. AIAA Guidance, Navigation and Control Conference, Hilton Head, SC, 1992.
- (Schwefel 84) Schwefel H.-P., *Evolution Strategies : A Family of Non-Linear Optimization Techniques Based on Imitating Some Principles of Organic Evolution*, Annals of Operations Research, vol. 1, pp. 165-167, 1984.
- (Sebag et Schoenauer 94) Sebag M. et Schoenauer M., *Controlling crossover through inductive learning*. Proceedings of PPSN94, the third Conference on Parallel Problems Solving from Nature, Schwefel H.-P. Eds., Springer Verlag, Octobre, 1994 .
- (Silverman et al. 97) Silverman H.F., Patterson W.R., Flanagan J.L. et Rabinkin D., *A Digital Processing System for Source Location and Sound Capture by Large Micro-phone Arrays*. ICASSP 97, Volume 1, Page 251, 1997.
- (Skinner et Broughton 95) Skinner A. et Broughton J.Q., *Neural Networks in Computational Materials Science : Training Algorithms*. Modelling and Simulation in Materials Science and Engineering, 3:371-390, 1995.
- (Spalanzani 97) Spalanzani A., *Application des Algorithmes Génétiques pour l'Etude de la Robustesse des Systèmes de RAP en Environnement Changeant*. 2^{ème} Rencontres Jeunes Chercheurs en Parole, La Rochelle, France, 1997.

- (Spalanzani 99) Spalanzani A., *Lamarckian vs Darwinian evolution for the adaptation to acoustical environment change*. Artificial Evolution 99, Dunkerque, à paraître.
- (Spalanzani et Kabré 97) Spalanzani A. et Kabré H., *Perception Artificielle de la Parole par l'Utilisation des Algorithmes Génétiques*. Journées Internationales d'Orsay sur les Sciences Cognitives, Orsay, France, 1997.
- (Spalanzani et Kabré 98) Spalanzani A. et Kabré H., *Evolution, Learning and Speech Recognition in Changing Acoustic Environments*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, pp. 663-671, Amsterdam, Pays-Bas, 1998.
- (Spalanzani et Sélouani 99) Spalanzani A. et Sélouani S., *Improving Robustness of Connectionist Speech Recognition Systems by Genetic Algorithms*, International Conference on Information, Intelligence and Systems, Washington, à paraître.
- (Spalanzani et Sélouani 99) A. Spalanzani et S. Sélouani, *Improving speech data projection by genetic algorithms*, Conference on Soft Computing and their Applications, Alger, pp. 140-145, 1999.
- (Spears et De Jong 91) Spears W.M. et De Jong K.A., *On the Virtues of Parameterized Uniform Crossover*. Proceedings of the International Conference on Genetic Algorithms, pp. 230-236, 1998.
- (Stern 95) Stern R.M., *Robust Speech Recognition*. Survey of the State of the Art in Human Language Technology, Edited by Cole et al., 1995.
- (Stroustrup 89) Stroustrup B., *Le langage C++*, InterEdition, 1989.
- (Surry et Radcliffe 96) Surry P.D. et Radcliffe N.J., *Real Representation*. Foundation of Genetic Algorithms IV, Ed. Morgan Kaufmann, San Mateo, 1996.
- (Takara et al. 97) Takara T., Higa K. et Nagayame I., *Isolated Word Recognition Using the HMM Structure Selected by the Genetic Algorithm*. Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97), vol. 1, pp. 459-462, Munich, Allemagne, April 21-24, 1997.
- (Tao et Michalewicz 98) Tao, G. and Michalewicz Z., *Inver-over Operator for the TSP*, Proceedings of the 5th Parallel Problem Solving from Nature, Springer-Verlag, Lecture Notes in Computer Science, Amsterdam, Septembre 1998.
- (Tebelskis 95) Tebelski J., *Speech Recognition using Neural Networks*. PhD-Thesis, Carnegie Mellon University, Pittsburgh, USA, May 1995.
- (Tessier et Berthommier 99) Tessier P. et Berthommier F., *Switching of Azimut and Elevation for Speaker Audio-Localisation*. ICSP'99, à paraître.

- (Tessier et al. 98) Tessier P., Guerin-Dugue A. et Schwartz J.-L., *Models for Audiovisual Fusion in a Noisy-Vowel Recognition Task*. Journal of VLSI Signal Processing 20, 25-44, 1998.
- (Thierens et al. 93) Thierens D., Suykens J., Vandewalle J. et De Moor B., *Genetic Weight Optimization of a Feedforward Neural Network Controller*. Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, Innsbruck, Austria, p.6, Avril 1993.
- (Thimm et Fiesler 94) Thimm G. et Fiesler E., *High Order and Multilayer Perceptron Initialization*. IDIAP technical report 94-07, 1994.
- (Tibrewala et Hermansky 97) Tibrewala S. et Hermansky H., *Multi-band and Adaptation Approachs to Robust Speech Recognition*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Trompf et al. 93) Trompf M., Richter R., Eckhardt H. et Hackbarth H., *Combination of Distorsion-Robust Feature Extraction and Neural Noise reduction for ASR*. Eurospeech'93, pp. 1039-1042, Berlin, Allemagne, Septembre 1993.
- (Tsingos 98) Tsingos N., *Simulation de Champs Sonores de Haute Qualité pour des Applications Graphiques Interactives*. Thèse de doctorat de l'Université Joseph Fourier de Grenoble, Laboratoire IMAGIS, 1998.
- (Turk 91) Turk M.A., *Interactive-time Vision: Face Recognition as a Visual Behavior*. Ph-D Thesis, Massachusetts Institute of Technology, Carnegie Mellon University, 1991.
- (Turney et al. 96) Turney P., Whitley D. et Anderson R., *Evolution, Learning, and Instinct : 100 Years of the Baldwin Effect*. Special Issue of Evolutionary Computation on the Baldwin Effect, vol. 4, n. 3, 1996.
- (Turney 96) Turney P., *The Baldwin Effect : A Bibliography*. <http://ai.iit.nrc.ca/baldwin/bibliography.html>, 1996.
- (Watson et al. 98) Watson J.P., Ross C., Eisele V., Denton J, Bins J., Guerra C., Whitley D. et Howe A., *The Traveling Salesrep Problem, Edge Assembly Crossover, and 2-opt*. 5th Conference on Parallel Problem Solving from Nature, Springer Verlag, Amsterdam, Pays-Bas, 1998.
- (Weisbuch et al. 96) Weisbuch G., Gutowitz H.A. et Nguyen G., *Information Contagion and the Economics of Pollution*. Journal of Economic Behavior and Organization, vol 29, pp. 389-407, 1996.
- (Werner et Dyer 91) Werner G.M. et Dyer M.G., *Evolution of Communication in Artificial Organisms*, Artificial Life II, ed. Langton C.G., Taylor C., Farmer J.D. et Rasmussen S., Addison-Wesley, 1991.

- (Wessel et Barnard 92) Wessels L.F.A. et Barnard E., *Avoiding False Local Minima by Proper Initialization of Connections*. IEEE Transaction on Neural Networks, vol. 3, N° 6, Novembre 1992.
- (Westphal 97) Westphal M., *The Use of Cepstral Means in Conversational Speech Recognition*. Eurospeech'97, 5th European Conference on Speech Communication and Technology, Rhodes, Grèce, 1997.
- (Whitley et Hanson 89) Whitley D. et Hanson T., *Optimizing Neural Networks Using Faster, More Accurate Genetic Search*. Proceedings of the Third Conference on Genetic Algorithms. Morgan Kaufmann (San Mateo), 1989.
- (Whitley et al. 94) Whitley D., Gordon S. et Mathias K., *Lamarckian Evolution, the Baldwin Effect and Function Optimization*. Parallel Problem Solving from Nature (PPSN III). pp. 6-15. Springer-Verlag, 1994.
- (Whitley 95) Whitley D., *Genetic Algorithms and Neural Networks*. Genetic Algorithms in Engineering and Computer Science. Ed. J. Periaux et G. Winter, 1995.
- (Yao 95) Yao X., *Evolutionary Artificial Neural Networks*. Encyclopedia of Computer Science and Technology. Ed.A. Kent et al., vol. 33, pp137-170, Marcel Dekker Inc., NewYork, NY 10016, 1995.
- (Yao et Liu 97) Yao X. et Liu Y., *A new evolutionary system for evolving artificial neural networks*. IEEE Transactions on Neural Networks, 8(3):694-713, May 1997.
- (RéfWeb 1) <http://svr-www.eng.cam.ac.uk/comp.speech/Section6/Q6.5.html>
- (RéfWeb 2) http://www.wior.uni-karlsruhe.de/Bibliothek/Title_Page2.html

Annexe

VALIDATION DES ALGORITHMES SUR QUELQUES BENCHMARKS

Afin d'évaluer la validité de nos programmes, nous avons testé les performances des GA et des ES sur différentes fonctions proposées dans la littérature. Nous avons testé sur trois fonctions classiques parmi celles répertoriées dans (Michalewicz 96), à savoir : les fonctions F1 et F2 de De Jong, ainsi que la fonction de Shubert.

A. Méthodes d'évolution

Nous avons testé deux méthodes évolutionnaires classiques, à savoir :

- les GA utilisent des paramètres codés en valeurs réelles. Les individus sont au nombre de 100. En moyenne 3% de leur gènes x_i sont mutés selon une loi gaussienne centrée sur x_i et d'écart-type 0,1. Ils sont recombinaés par un croisement simple selon une probabilité de 80%.
- Les ES sont testées avec la stratégie (10,100)-ES. Chaque individu est muté par la méthode non-corrélée. Les écart-types sont recombinaés.

Table 12 Méthodes d'évolution utilisées.

Evolution	Codage des paramètres	NB individus	Type mutation	%mutation	% croisement
GA	réel	100	Norm(x,0.1)	3%	80%
ES	réel	(10,100)-ES	Non-corrélée		

B. Fonction de De Jong F1

La fonction F1 de De Jong est définie de la manière suivante :

$$\sum_{i=1}^3 x_i^2, \text{ où } -5, 12 \leq x_i \leq 5, 12$$

Chaque individu est codé avec trois gènes (x_1, x_2, x_3).

La Figure 68 montre les performances moyennées sur 10 runs d'une population évoluant d'un côté par GA, de l'autre par ES.

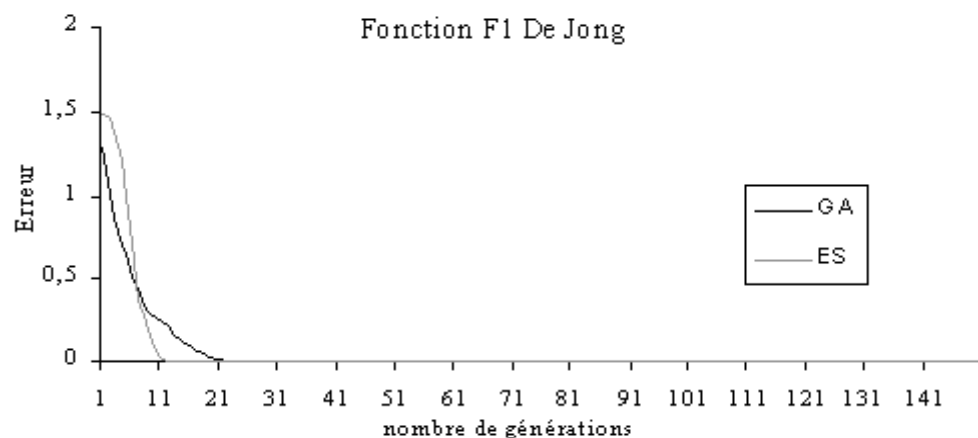


Figure 68 Evolution de l'erreur : fonction F1.

Les deux méthodes obtiennent, après 20 générations, des performances sensiblement équivalentes. Les GA sont plus efficaces lors des toutes premières générations. Cependant, les ES ont une meilleure accélération de convergence qui leur permettent, au bout de 10 générations, d'obtenir des performances supérieures aux GA.

C. Fonction de De Jong F2

La fonction F1 de De Jong est définie de la manière suivante :

$$100(x_1^2 - x_2^2)^2 + (1 - x_1)^2, \text{ où } -2,048 \leq x_i \leq 2,048$$

Chaque individu est codé avec deux gènes (x_1, x_2).

La Figure 69 montre les performances moyennées sur 10 runs d'une population évoluant d'un côté par GA, de l'autre par ES.

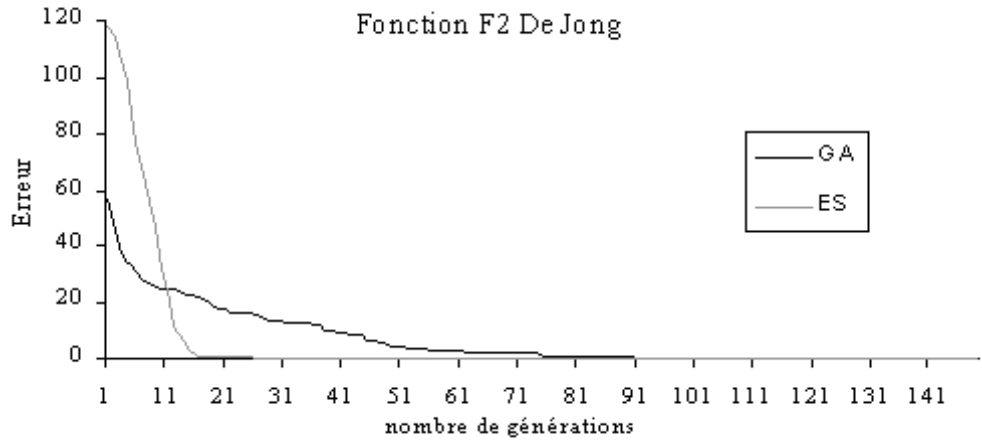


Figure 69 Evolution de l'erreur : fonction F2.

De même que pour la fonction F1, les GA et ES arrivent à des performances équivalentes. Cependant, la vitesse de convergence des ES est nettement plus élevée que celle des GA, ce qui permet aux ES de trouver une solution optimale en 4 fois moins de temps que les GA, à savoir 90 générations au lieu de 25.

D. Fonction de Shubert

La fonction de Shubert est définie de la manière suivante :

$$\sum_{i=1}^5 i \cdot \cos[(i+1) \cdot x_1 + i] \cdot \sum_{i=1}^5 i \cdot \cos[(i+1) \cdot x_2 + i],$$

où $-10 \leq x_i \leq 10$ pour $i = 1, 2$.

Cette fonction comporte 760 minima locaux dont seulement 18 sont globaux.

Sur 40 runs effectués, 70% ont abouti à une erreur proche de 0 pour les deux méthodes. La Figure 70 montre les performances moyennées sur les runs aboutissant à une erreur proche de 0 d'une population évoluant d'un côté par GA, de l'autre par ES.

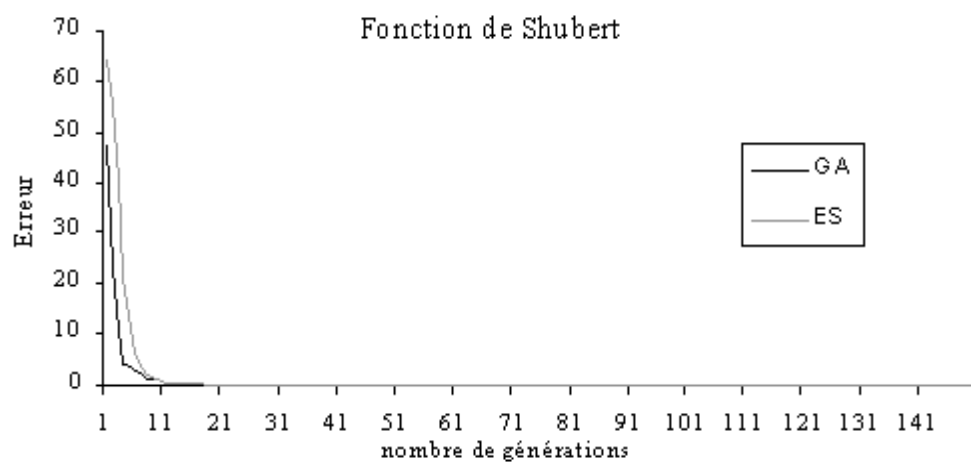


Figure 70 Evolution de l'erreur : fonction de Shubert.

Pour la recherche du minimum de la fonction de Shubert, les GA et ES sont à peu près de même efficacité.

E. Changement d'environnement

Nous avons alors étudié les performances des GA et des ES lors d'un changement d'environnement. Les résultats suivants rapportent les performances des populations lors d'une translation de la fonction F1 de De Jong. On peut alors définir un environnement 1 correspondant à F1 et un environnement 2 correspondant à F2, soit :

Environnement 1 :

$$\sum_{i=1}^3 x_i^2, \text{ où } -5, 12 \leq x_i \leq 5, 12$$

Environnement 2 :

$$\sum_{i=1}^3 (x_i + 2)^2, \text{ où } -5, 12 \leq x_i \leq 5, 12$$

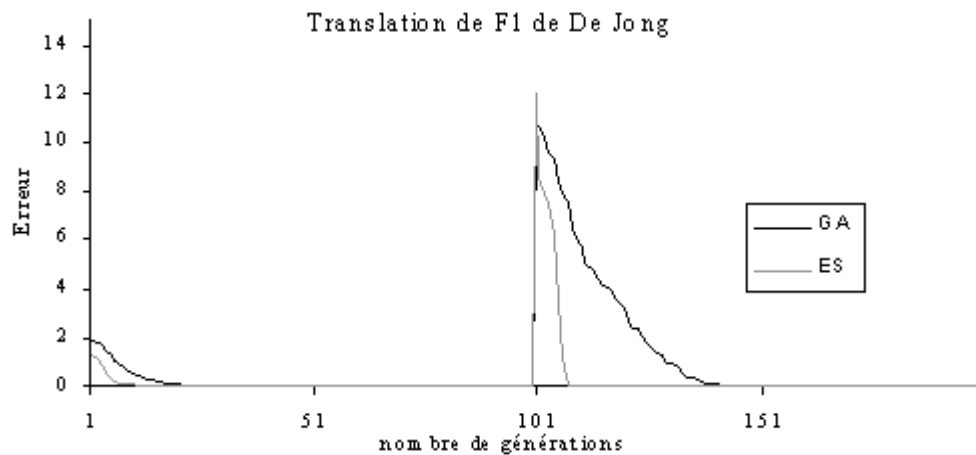


Figure 71 Evolution de l'erreur lors d'un changement d'environnement.

Afin de permettre aux ES de bien s'adapter, nous avons remis les écarts-types et les constantes d'apprentissage à des valeurs conformes à celles initiales. En effet, les paramètres de la mutation ES sont censés s'adapter aux besoins de la population en diminuant l'altération par mutation lorsque les individus sont proches de la solution optimale. Cependant, après convergence, les individus ont des écarts-types trop proches de 0 pour être capables de se ré-adapter à un changement brutal.

Les performances des ES avec ré-initialisation des paramètres de mutation sont meilleures que celles des GA de par leur capacité à se ré-adapter plus rapidement.

F. Série de changements d'environnement

Afin de tester la robustesse de nos populations à des changements multiples d'environnement, nous avons plongé les populations dans un environnement changeant toutes les 100 générations. Ainsi, l'expérience précédente est reproduite 3 fois en alternant l'environnement 1 (F1) et l'environnement 2 (F1 tradatée).

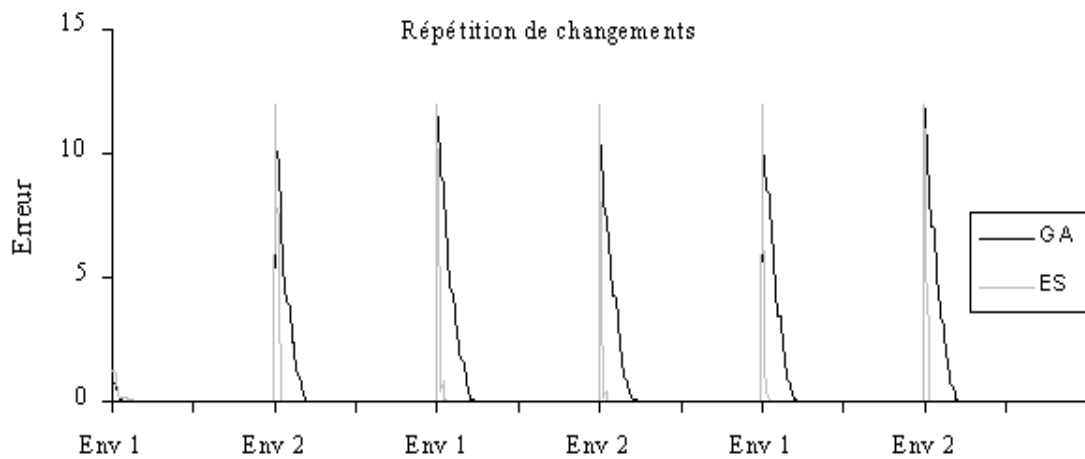


Figure 72 Evolution de l'erreur lors de plusieurs changements d'environnement.

Nous pouvons constater la stabilité dans le temps des performances des deux méthodes. En effet, l'adaptation des individus se fait de manière similaire lors des 5 changements brutaux d'environnement. Les individus ainsi adaptés sont capables de se ré-adapter successivement à plusieurs changements d'environnement sans perdre de performances entre deux convergences vers une même solution présentée à deux instants différents.

Par cette série d'expériences, nous pouvons garantir la validité des méthodes évolutionnaires GA et ES que nous avons implémentées. De plus, nous avons montré qu'elles sont capables de s'adapter rapidement à un ou plusieurs changements d'environnement et que le comportement des populations reste stable dans le temps. Les ES semblent cependant être plus performantes en temps. Lorsque les simulations sont de grandes durées (supérieures au quart d'heure par exemple) et que le moindre gain du nombre de générations nécessaires peut être significatif, les ES semblent être la méthode la plus efficace.

Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance de la parole.

Résumé : Les systèmes de reconnaissance automatique de la parole sont de plus en plus répandus et utilisés dans des conditions acoustiques très variées, par des locuteurs très différents. De ce fait, ces systèmes, généralement conçus en laboratoire, doivent être robustes afin de garder des performances optimales en situation réelle. Les travaux que nous avons effectués sont partis de l'idée que si les systèmes de reconnaissance étaient capables de s'auto-modifier dans le temps, afin de s'adapter aux changements de leurs environnements acoustiques, ils pourraient être beaucoup plus robustes. En effet, les organismes vivants ont montré à la fois des capacités à sélectionner les informations utiles et à modifier leur traitement en vue de survivre dans leur environnement.

Dans un premier temps, nous avons cherché à adapter le système de reconnaissance lui-même aux différents environnements. Nous avons étudié les capacités du système à s'adapter aux changements de conditions acoustiques, à l'aide d'une approche locale (par rétro-propagation du gradient) et d'une approche globale (par algorithmes évolutionnaires), en vue de trouver un système optimal.

Dans un second temps, nous nous sommes placée dans le cadre du traitement des données en entrée du système. Partant d'une base de projection issue d'une analyse en composantes principales, nous avons cherché à trouver à l'aide des algorithmes évolutionnaires une base de projection adaptée à chaque environnement rencontré et permettant de retrouver les conditions acoustiques connues du système.

Nous avons mis en place une plate-forme de simulation permettant de faire évoluer des populations de systèmes de reconnaissance. Les résultats obtenus montrent qu'en moyenne l'hybridation des algorithmes évolutionnaires et des techniques de reconnaissance classiques améliore sensiblement, et de manière stable, les performances du système de reconnaissance, et ceci dans les deux types d'hybridation que nous avons testés.

Mots-clés : algorithmes évolutionnaires, reconnaissance de la parole, robustesse.

Evolutionary Algorithms to study the robustness of speech recognition systems.

Summary : Automatic speech recognition systems are becoming ever more common and are increasingly deployed in more variable acoustic conditions, by very different speakers. So these systems, generally conceived in a laboratory, must be robust in order to provide optimal performance in real situations. This ph-D explores the possibility of gaining robustness by designing speech recognition systems able to auto-modify in real time, in order to adapt to the changes of acoustic environment. As a starting point, the adaptive capacities of living organisms were considered in relation to their environment. Analogues of these mechanisms were then applied to automatic speech recognition systems. It appeared to be interesting to imagine a system adapting to the changing acoustic conditions in order to remain effective regardless of its conditions of use.

Initially, the speech recognition system itself was adapted to various environments. Its capacity to adapt to the changes of acoustic conditions was studied, using a local approach (by retro-propagation of the gradient) and a global solution (by evolutionary algorithms), in order to find an optimal system.

Secondly, the specific aspects of the system's input data processing were examined. A projection base adapted to each environment was sought, based on a principal component analysis of acoustic data, using evolutionary algorithms to set the system's knowledge of acoustic conditions.

A simulation platform was set up, to allow the evolution of populations of recognition systems. Results obtained show that on average the hybridization of the evolutionary algorithms and traditional techniques of recognition improves the performance of the speech recognition system.

Key words : evolutionary algorithms, speech recognition, robustness.