

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة فرحات عباس – سطيف – 1 –  
Université Ferhat Abbas – Sétif – 1 –

Faculté de Technologie  
Département d'Électronique

## THÈSE

*Présentée pour l'obtention du diplôme de*

### DOCTORAT EN SCIENCES

En ÉLECTRONIQUE

*Par*

**M. Kamal BOUDJELABA**

*Thème*

---

***CONTRIBUTION À LA CONCEPTION DES FILTRES BIDIMENSIONNELS NON  
RÉCURSIFS EN UTILISANT LES TECHNIQUES DE L'INTELLIGENCE  
ARTIFICIELLE : APPLICATION AU TRAITEMENT D'IMAGES***

---

Date de soutenance : 11 juin 2014

Devant le jury composé de :

Président	: M. R.E. BEKKA	Prof.	Université Ferhat Abbas – Sétif – 1 – Algérie
Rapporteur	: M. D. CHIKOUCHE	Prof.	Université de M'sila – Algérie
Co-rapporteur	: M. F. ROS	H.D.R.	Université d'Orléans – France
Examineurs	: M. K. ROUABAH	M.C.A.	Université de Bordj Bou Arreridj – Algérie
	M. C. LEGER	Prof.	Université d'Orléans – France
	M. T. FOURNEL	Prof.	Université de Saint-Etienne – France

# Remerciements

Les travaux présentés dans cette thèse ont été réalisés dans le cadre de cotutelle internationale de thèse entre l'université d'Orléans (France) et l'université de Sétif – 1 (Algérie).

Je souhaite remercier Christine Rousselle, pour la confiance qu'elle m'a témoignée en m'accueillant dans le laboratoire PRISME alors qu'elle en était la directrice, ainsi que Azeddine Kourta, son successeur au sein de la direction du laboratoire PRISME, Christophe Léger, responsable de l'ancienne équipe ISS et actuelle directeur de Polytech'Orléans, de m'avoir permis de préparer cette thèse au sein du laboratoire PRISME, et Khaled R. de m'avoir mis en contact avec les responsables du laboratoire.

Je tiens à remercier et à témoigner ma sincère gratitude à Djamel Chikouche, Professeur à l'université de M'sila, qui a dirigé les travaux de cette thèse pour son encadrement de qualité, sa disponibilité, son aide et son encouragement et à Frédéric Ros, HDR, chercheur associé à l'université d'Orléans, qui a dirigé les travaux de cette thèse, qui m'a apporté une aide précieuse, à la fois par ses conseils sur le plan scientifique et par ses témoignages d'amitié et qui a considérablement contribué à bonifier mon travail, notamment lors de la rédaction des articles.

Je remercie Thierry Fournel, Professeur à l'université de Saint-Étienne, et Khaled Rouabah, Maître de Conférences à l'université de Bordj Bou-Arreridj, qui ont accepté d'être les rapporteurs de ce mémoire, pour la pertinence de leurs remarques et leur participation au jury de thèse.

Je remercie vivement Raïs El'hadi Bekka, Professeur à l'université de Sétif ; Christophe Léger, Professeur à l'université d'Orléans, pour l'honneur qu'ils m'ont témoigné en participant à ce jury, ainsi pour avoir consacré du temps à la lecture de ce travail.

Je remercie également Sylvie Treuillet, Maître de conférences à l'université d'Orléans et responsable de l'équipe "Image et Vision", et Philippe Ravier, Maître de Conférences à l'université d'Orléans et responsable de l'équipe "Signal" pour leur disponibilité et conseils précieux.

J'adresse mes remerciements à tous les amis et les collègues des équipes "Image et Vision" et "Signal", les membres du laboratoire PRISME, ainsi qu'au personnel de Polytech. En particulier, à Wael E., Raphaël D, Raphaël C., Meryem J., Guy L., Denis L., Roger L., Jean Yves C., Mohamed D., Clarisse N., Sylvie P., Rachid H., Ahmed D., Ahmad A., Arnaud M., Julien R., Olivier Bu., Karim A., Sonia A., Abdelouahab B., Abdelbassit B., Sandra S., Rachid J., Olivier Bo., Dalal A., Rémy L., Rodolphe W., Aladine C., Thien L., Smaïl A.

Je remercie également quelques amis et collègues de l'université de Béjaïa : Ouahib G., Loucif G., Hocine H., Samia M., Athmane B., Smaïl B.

Enfin j'exprime ici ma reconnaissance à ma famille et mes proches pour leurs encouragements.

# Résumé

La conception des filtres à réponse impulsionnelle finie (RIF) peut être formulée comme un problème d'optimisation non linéaire réputée pour être difficile à résoudre par les approches conventionnelles. Les contraintes sont élevées et un grand nombre de paramètres doit être estimé, surtout lorsqu'il s'agit de filtres RIF 2D. Afin d'améliorer les performances et d'optimiser la conception des filtres RIF, nous explorons plusieurs méthodes évolutionnaires capables de traiter de grands espaces de recherche.

Nous proposons un nouvel algorithme génétique dans lequel certains concepts innovants sont introduits pour améliorer la convergence et rendre son utilisation plus facile pour les praticiens. Le point clé de notre approche découle de la capacité de l'algorithme génétique (AG) à adapter les opérateurs génétiques au cours de la recherche génétique et en fonction de l'évolution de la population, tout en restant simple et facile à mettre en œuvre. La plupart des paramètres et les opérateurs sont modifiés par l'AG lui-même. Ensuite, la famille des méthodes d'optimisation par essaim de particules (PSO) est étudiée et une approche PSO est proposée pour la conception des filtres RIF. L'étude souligne l'importance de l'introduction de l'intelligence dans la métaheuristique pour les rendre plus efficaces en insérant des stratégies d'auto-réglage. Finalement, un algorithme génétique hybride (HGA) est proposé pour la conception de filtres numériques. L'algorithme est composé d'un processus génétique pur et d'une approche locale dédiée. Le processus génétique pur est consacré à la partie exploratoire du HGA. L'approche locale concerne la convergence de l'algorithme et est hautement optimisée de manière à être flexible. La contribution majeure consiste en l'élaboration d'une stratégie pour le guidage de l'algorithme de recherche des solutions, dans lequel les paramètres s'auto-adaptent par eux-mêmes en fonction de l'évolution de la recherche des solutions.

Nous avons démontré sur divers cas de conception de filtres RIF une amélioration significative des performances. Notre contribution concerne principalement l'utilisation de l'AG pour la conception des filtres RIF. Elle vise à relever le défi actuel de démocratisation de l'utilisation des AG's pour les problèmes d'optimisation du monde réel. Les expériences réalisées avec différents types de filtres mettent en évidence la contribution récurrente de l'AG et de l'hybridation dans l'amélioration des performances. Les expériences montrent également les avantages de notre AG par rapport à d'autres approches classiques de conception de filtres et des AG's de référence appliqués dans ce domaine.

**Mots clés :** Filtrés RIF 1D et 2D, Algorithme Génétique, Optimisation par Essaim de Particules, Hybridation, Recherche Tabou, Recuit Simulé, Ascension de Collines, Traitement d'Image.

# Abstract

The design of finite impulse response (FIR) filters can be formulated as a non-linear optimization problem reputed to be difficult to solve by conventional approaches. The constraints are high and a large number of parameters have to be estimated, especially when dealing with 2D FIR filters. In order to improve the performance of conventional approaches, we explore several evolutionary algorithms capable of handling large spaces.

We propose a new genetic algorithm in which some innovative concepts are introduced to improve the convergence and make its use easier for practitioners. The key point of our approach stems from the capacity of the genetic algorithm (GA) to adapt the genetic operators during the genetic life while remaining simple and easy to implement. Most of parameters and operators are changed by the GA itself. Then, the Particle Swarm Optimization (PSO) is proposed for FIR filter design. The study underlines the importance of introducing intelligence in metaheuristics to make them more efficient by inserting self-tuning strategies. Finally, a hybrid genetic algorithm (HGA) is proposed for the design of digital filters. The algorithm is composed of a pure genetic process and a dedicated local approach. The pure genetic process is devoted to the exploratory part of the HGA. The local approach concerns convergence of the algorithm and is highly optimized so as to be tractable. The main contribution is the development of a strategy for guiding the algorithm, in which the parameters auto-adapt by themselves according to the evolution of research solutions.

We demonstrate on various cases of filter design a significant improvement in performance. Our contribution mainly concerns the usability of GAs for the specific area of FIR filter design. It seeks to address the current challenge of democratizing the use of GAs for real world problems. Experiments performed with various types of filters highlight the recurrent contribution of hybridization in improving performance. The experiments also reveal the advantages of our proposal compared to more conventional filter design approaches and some reference GAs in this field of application.

**Keywords:** 1D and 2D FIR filters, Genetic Algorithm, Particles Swarm Optimization, Hybridization, Tabu Search, Simulated Annealing, Hill Climbing, Image Processing.

# Table des matières

Remerciements .....	
Résumé.....	ii
Abstract .....	iii
Table des matières .....	iv
Table des figures .....	viii
Liste des tableaux.....	x
Introduction générale .....	1
Chapitre I.....	9
1. Filtres à réponse impulsionnelle finie .....	9
1.1. Introduction .....	10
1.2. Théorie sur les filtres RIF unidimensionnels (1D) .....	11
1.2.1. Fonction de transfert et réponse fréquentielle .....	12
1.2.2. Méthodes traditionnelles de synthèse des filtres RIF 1D .....	13
1.2.2.1. Méthode des fenêtres.....	15
1.2.2.2. Méthode de l'échantillonnage fréquentiel .....	17
1.2.2.3. Méthodes itératives et d'optimisation .....	17
1.2.2.4. Méthode de Remez .....	18
1.2.2.5. Méthode de Parks-McClellan .....	21
1.2.3. Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 1D.....	21
1.3. Théorie sur les filtres RIF bidimensionnels (2D).....	23
1.3.1. Méthodes de synthèse des filtres RIF 2D.....	24
1.3.2. Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 2D .....	25
1.4. Limitation des méthodes conventionnelles de synthèse des filtres RIF .....	26
1.5. Conclusion.....	27
Chapitre II.....	29
2. Algorithmes évolutionnaires .....	29
2.1. Introduction .....	30
2.2. Recherche tabou (Tabu Search) .....	30
2.2.1. Concepts de base de la recherche tabou .....	31

2.3.	Recuit simulé (Simulated Annealing) .....	31
2.4.	Algorithme d'ascension de collines (Hill Climbing) .....	32
2.5.	Optimisation par essaim particulaire (de particules) (Particle swarm optimization : PSO) .....	33
2.6.	Algorithme génétique .....	36
2.6.1.	Chromosome (individu).....	37
2.6.2.	Evaluation.....	37
2.6.3.	Sélection.....	37
2.6.4.	Croisement (Crossover).....	37
2.6.5.	Mutation .....	38
2.6.6.	Amélioration de l'AG standard.....	39
2.7.	Hybridation .....	40
2.8.	Etat de l'art sur les méthodes évolutionnaires appliquées pour la synthèse des filtres RIF 1D .....	40
2.9.	Etat de l'art sur les méthodes évolutionnaires appliquées à la synthèse des filtres RIF 2D .....	42
2.10.	Conclusion .....	43
Chapitre III.....		45
3.	Elaboration d'approches évolutionnaires dédiées à la synthèse des filtres RIF .....	45
3.1.	Introduction .....	46
3.2.	AG adaptatif dédié à la synthèse des filtres RIF 1D et 2D.....	46
3.2.1.	Représentation des chromosomes .....	47
3.2.2.	Fonction fitness.....	47
3.2.3.	Population initiale .....	49
3.2.4.	Stratégie génétique.....	49
3.2.5.	Opérateurs génétiques et mécanismes dédiés à améliorer la convergence et les performances.....	52
3.2.5.1.	Opérateur de croisement.....	52
3.2.5.2.	Opérateur de mutation .....	52
3.2.5.3.	Procédure de sélection perfectionnée.....	54
3.2.5.4.	Processus de rafraîchissement pour éviter la convergence prématurée .....	56
3.3.	Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF .....	58

3.4.	Hybridation entre l'AG adaptatif et l'Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF .....	59
3.5.	L'algorithme hybride .....	60
3.5.1.	Détails de l'algorithme génétique hybride (HGA) .....	61
3.5.2.	Recherche locale .....	62
3.5.3.	Activation de l'approche locale et sélection des chromosomes.....	63
3.5.4.	Durée de la recherche locale .....	70
3.6.	Résumé de l'approche AGA .....	70
3.7.	Conclusion.....	71
Chapitre IV.....		73
4.	Résultats expérimentaux .....	73
4.1.	Application des algorithmes évolutionnaires pour la synthèse des filtres RIF 2D. 74	
4.1.1.	Synthèse d'un filtre RIF symétrique passe-bas .....	75
4.1.2.	Résultats avec d'autres filtres RIF .....	82
4.1.3.	Comparaison spécifique avec deux méthodes génétiques populaires.....	84
4.1.4.	D'autres résultats (tests) avec des filtres de dimensions plus élevées .....	86
4.1.5.	Discussion, étude statistique et de convivialité .....	86
4.2.	PSO pour la synthèse des filtres RIF 1D .....	89
4.2.1.	Réglage des paramètres.....	90
4.2.2.	Exemples de synthèse .....	92
4.2.3.	Etude statistique .....	98
4.2.4.	Discussion et perspectives .....	100
4.2.5.	Hybridation préliminaire de l'AGA avec la NPSO .....	101
4.3.	Algorithme hybride pour la synthèse des filtres RIF 1D et 2D .....	103
4.3.1.	Performance de notre système évolutionnaire .....	104
4.3.2.	Réglage des paramètres.....	104
4.3.2.1.	Paramètres génétiques de base.....	104
4.3.2.2.	Paramètres de domaine.....	105
4.3.2.3.	Paramètres de l'approche locale .....	105
4.3.3.	Résultats préliminaires relatifs aux composantes de l'algorithme .....	106
4.3.3.1.	Effet de l'initialisation .....	106
4.3.3.2.	Coût de la recherche locale.....	108
4.3.4.	Alternatives d'hybridation .....	109

4.3.5. Comparaisons avec d'autres approches .....	113
4.3.6. Application de l'algorithme hybride pour la synthèse d'un filtre RIF 2D.....	118
4.4. Résumé des performances pour la synthèse d'un filtre RIF 2D (Exemple 1, section 4.1 et 4.3.6) .....	122
4.5. Quantification .....	123
4.6. Contributions de la recherche.....	124
4.7. Application des filtres conçus pour le traitement d'images .....	125
4.7.1. Exemple de filtrage passe-bas et passe-haut.....	125
4.7.2. Ajout d'un bruit à une image et élimination de ce bruit par filtrage passe-bas 129	
4.8. Conclusion.....	132
Conclusion générale.....	133
Bibliographie .....	137
Bibliographie de l'auteur.....	145
.....	146

# Table des figures

Figure 1.1 – Réponse impulsionnelle symétrique .....	12
Figure 1.2 – Réponse impulsionnelle antisymétrique.....	13
Figure 1.3 – Gabarit du filtre (Linéaire).....	14
Figure 1.4 – Gabarit du filtre (en dB).....	15
Figure 1.5 – Réponses d’amplitude pour différents types de filtres. ....	15
Figure 2.1 – Croisement en 1 point .....	38
Figure 2.2 – Croisement directionnel (direction-based crossover).....	38
Figure 2.3 – Mutation.....	39
Figure 2.4 – Mutation arithmétique.....	39
Figure 3.1 – Processus génétiques.....	50
Figure 3.2 – Organigramme de l’AGA.....	51
Figure 3.3 – Organigramme du HGA. ....	61
Figure 3.4 – Gènes concernés par les similarités chromosomiques pour un filtre 1D (a) : RIF non symétrique, (b) : RIF avec symétrie impaire, (c) : RIF avec symétrie paire.....	65
Figure 3.5 – Gènes concernés par les similarités chromosomiques pour un filtre 2D (a) : RIF avec symétrie paire, (b) : RIF avec symétrie impaire.....	66
Figure 4.1 – Réponses fréquentielles (a) Adaptatif AG1 (b) Adaptatif AG2 (c) Adaptatif AG3 (d) Standard AG4.....	77
Figure 4.2 – Réponses fréquentielles : (a) Simulated Annealing (b) Tabu search (c) Hill-Climbing (d) Méthode de Remez (transformation fréquentielle). ....	78
Figure 4.3 – Réponses fréquentielles : (a) Méthodes des moindres carrés (transformation fréquentielle) (b) Méthode de l’échantillonnage en fréquence (c) Méthodes des fenêtres. ....	79
Figure 4.4 – Effet des processus de diversification et de rafraîchissement : (a) Processus de diversification (b) processus de rafraîchissement.....	81
Figure 4.5 – Réponses fréquentielles : (a) Passe-haut 7x7 (b) Coupe-bande 7x7 (c) Passe-bande 9x9 (d) Passe-bas 9x9 .....	83
Figure 4.6 – Réponses fréquentielles : (a) Passe-bas 10x10 (b) Passe-bas 11x11. ....	84
Figure 4.7 – Résultats avec différentes valeurs de la fréquence de coupure pour 10 exécutions. ....	88
Figure 4.8 – Erreurs moyennes et maximales pour 100 exécutions.....	89
Figure 4.9 – Réponses en amplitude (dB) obtenues pour un filtre RIF passe-bas d’ordre 40 ( $\omega_c = 0.5$ ) par les techniques AG, AGA, PSO, NPSO et CPSO.....	93
Figure 4.10 – Profile de convergence pour les techniques AG, AGA, PSO, NPSO et CPSO dans le cas d’un filtre RIF passe-bas d’ordre 40 ( $\omega_c = 0.5$ ).....	93
Figure 4.11 – Réponses d’amplitude (dB) d’un filtre RIF passe-bas d’ordre 20 ( $\omega_c = 0.5$ ) conçu par les algorithmes AG, AGA, PSO, NPSO et CPSO.....	95
Figure 4.12 – Profile de convergence pour les techniques AG, AGA, PSO, NPSO et CPSO dans le cas d’un filtre RIF passe-bas d’ordre 20 ( $\omega_c = 0.5$ ).....	96

---

Figure 4.13 – Réponse en amplitude (dB) d'un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.4$ ) conçu par les techniques AG, AGA, PSO, NPSO et CPSO.....	97
Figure 4.14 – Réponse en amplitude (dB) d'un filtre RIF passe-bas d'ordre 30 ( $\omega_c = 0.6$ ) conçu par les techniques AG, AGA, PSO, NPSO et CPSO.....	98
Figure 4.15 – Résultats obtenus pour 10 exécutions (variation et évolution des performances pour 10 exécutions).....	99
Figure 4.16 – Résultats obtenus pour 100 exécutions (variation et évolution des performances pour 100 exécutions).....	99
Figure 4.17 – Evolution de la fonction fitness dans l'algorithme hybride dans le cas d'un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.5$ ). .....	103
Figure 4.18 – Performance pour différentes valeurs du maximum de la fonction fitness de la population initiale .....	107
Figure 4.19 – Performance de l'approche locale pour différentes valeurs du maximum de la fonction fitness.....	108
Figure 4.20 – Performances en fonction de la taille du voisinage.....	109
Figure 4.21 – Performances en fonction du nombre de chromosomes.....	109
Figure 4.22 – Réponse fréquentielle (amplitude et phase) obtenue par HGA1 (problème 1).....	110
Figure 4.23 – Réponses d'amplitude : (a) AGA5, (b) Algorithme de CEN, (c) Méthode de Parks-McClellan, (d) Méthode des moindres carrés, (e) Méthode d'échantillonnage en fréquence, (f) Méthode des fenêtres (problème 1).....	111
Figure 4.24 – Evolution de la fonction fitness .....	112
Figure 4.25 – Réponse en fréquence (amplitude et phase) obtenue par HGA1 (problème 2).....	115
Figure 4.26 – Evolution de la fonction fitness.....	115
Figure 4.27 – Réponse fréquentielle (amplitude et phase) obtenue par HGA1 (problème 3).....	116
Figure 4.28 – Réponse fréquentielle (amplitude et phase) obtenue par HGA1 (problème 4).....	118
Figure 4.29 – Evolution de la fitness maximale dans l'algorithme HGA1 (Exemple 1, section 4.1).....	120
Figure 4.30 – Zoom sur les parties où l'approche locale est active dans l'algorithme HGA1 (Exemple 1, section 4.1).....	121
Figure 4.31 – Comparatif de performance entre les différentes approches.....	123
Figure 4.32 – Affichage des différentes versions de l'image de Lena. ....	126
Figure 4.33 – Histogrammes des images de la figure 4.32.....	127
Figure 4.34 – Représentation 3D du spectre des images de la figure 4.32.....	128
Figure 4.35 – Représentation du Logarithme du spectre des images de la figure 4.32.....	128
Figure 4.36 – Affichage des différentes versions de l'image de Lena. ....	129
Figure 4.37 – Histogrammes des images de la figure 4.36.....	130
Figure 4.38 – Représentation 3D du spectre des images de la figure 4.36.....	131
Figure 4.39 – Représentation du Logarithme du spectre des images de la figure 4.36.....	131

# Liste des tableaux

Table 2.1 – Algorithme de la recherche tabou (TS).....	31
Table 2.2 – Algorithme du recuit simulé (SA).....	32
Table 2.3 – Algorithme de la Hill Climbing (HC).....	33
Table 2.4 – Algorithme de l’optimisation par essaim de particules (PSO).....	35
Table 2.5 – Algorithme génétique (AG).....	37
Table 3.1 – Description et paramètres de l’AG proposé.....	57
Table 3.2 – Différentes étapes de l’algorithme génétique hybride (HGA).....	62
Table 3.3 – Algorithme de sélection pour la recherche locale.....	69
Table 3.4 – Comparatif entre les opérateurs de l’AG et de l’AGA et synthèse des mécanismes de l’AGA.....	71
Table 4.1 – Description et paramètres des différentes versions de l’AG.....	74
Table 4.2 – Paramètres de chaque algorithme évolutionnaire.....	75
Table 4.3 – Réglage des paramètres de l’AGA.....	76
Table 4.4 – Paramètres de performance obtenus pour un filtre passe-bas.....	80
Table 4.5 – Paramètres de performance obtenus pour un filtre passe-haut.....	82
Table 4.6 – Paramètres de performance obtenus par notre AG adaptatif et les AG’s utilisés par Mastorakis et Tzeng pour le premier exemple.....	85
Table 4.7 – Paramètres de performance obtenus par notre AG adaptatif et les AG’s utilisés par Mastorakis et Tzeng pour le second exemple.....	85
Table 4.8 – Paramètres de performance obtenus pour un filtre passe-bas (23x23).....	86
Table 4.9 – Paramètres de performance obtenus pour $\omega_{1,2} = 0.2$ .....	87
Table 4.10 – Paramètres de performance obtenus pour $\omega_{1,2} = 0.3$ .....	88
Table 4.11 – Paramètres de performance obtenus pour $\omega_{1,2} = 0.4$ .....	88
Table 4.12 – Paramètres de chaque algorithme.....	90
Table 4.13 – Réglage des paramètres de l’AGA.....	91
Table 4.14 – Paramètres de l’AGA.....	92
Table 4.15 – Performance de chaque algorithme ( $2*N+1=41$ ).....	94
Table 4.16 – Performance de chaque algorithme ( $2*N+1=21$ ).....	95
Table 4.17 – Performance de chaque algorithme ( $2*N+1=41$ ).....	97
Table 4.18 – Performance de chaque algorithme ( $2*N+1=31$ ).....	98
Table 4.19 – Moyenne et écart type des erreurs obtenues pour 100 exécutions.....	100
Table 4.20 – Paramètres de l’algorithme hybride HGA.....	105
Table 4.21 – Paramètres de chaque approche.....	106
Table 4.22 – Performances avec différentes valeurs de la fonction fitness moyenne de la population initiale.....	107
Table 4.23 – Performances avec différentes valeurs de la fonction fitness maximale de la population initiale.....	107
Table 4.24 – Performances de la recherche locale pour différentes valeurs du maximum de la fonction fitness initiale.....	107

---

Table 4.25 – Paramètres de performance obtenus pour un filtre passe-bas (problème 1).....	113
Table 4.26 – Paramètres de performance obtenus pour un filtre passe-haut (problème 2).....	114
Table 4.27 – Paramètres de performance obtenus pour un filtre passe-bas (problème 3).....	116
Table 4.28 – Paramètres de performance obtenus pour un filtre passe-bas (problème 4).....	117
Table 4.29 – Paramètres de performance obtenus pour un filtre passe-bas (Exemple 1, section 4.1). .....	120
Table 4.30 – Erreurs en coefficients produites dans le cas d'une quantification en représentation en virgule fixe. ....	123
Table 4.31 – Erreurs en réponses fréquentielles produites dans le cas d'une quantification en représentation en virgule fixe. ....	124

# Introduction générale

## *Contexte*

Au cours des dernières années et avec l'amélioration rapide de la technologie informatique, le traitement du signal numérique bidimensionnel (2D) est devenu plus important. En particulier, la conception de filtres numériques 2D a reçu beaucoup d'attention dans divers domaines. Ils ont été largement utilisés pour le traitement des photos aériennes et satellites, des données géophysiques et sismologiques, ou des images géologiques et médicales. Les filtres numériques peuvent être classés en deux familles : les filtres à réponse impulsionnelle finie (RIF) [Wang et al., 2008] et les filtres à réponse impulsionnelle infinie (RII). Etant donné que les filtres numériques RIF sont intrinsèquement stables et peuvent avoir une phase linéaire, ils sont souvent préférés aux filtres RII car ils possèdent plusieurs caractéristiques souhaitables [Cen, 2007], par exemple, la stabilité et la garantie de l'absence de distorsion de phase. Toutefois, le nombre de coefficients indépendants de filtre nécessaires pour concevoir de manière optimale un filtre RIF 2D est généralement élevé (filtre de grande taille) pour les applications pratiques. Cela rend complexe la conception efficace et met en évidence la nécessité de l'utilisation de techniques plus efficaces et plus évoluées pour la conception de filtres RIF 2D à phase linéaire [Lee et al., 2000].

Le problème de la conception de filtre consiste à trouver une réalisation du filtre qui satisfait à chacune des exigences à un degré suffisant pour le rendre utile. Les techniques de conception de filtres numériques RIF 2D ont été largement développées depuis plusieurs années [Kamp et al., 1975 ; Pei et al., 1999 ; Cen, 2007 ; Lu et al., 2009]. Les résultats de la plupart de ces techniques sont donnés sous la forme de la réponse impulsionnelle d'un filtre 2D, de sorte que le filtre est conçu pour une réalisation convenable de convolution directe. Différentes méthodes conventionnelles peuvent être utilisées pour trouver les coefficients à partir des spécifications en fréquence : la méthode des fenêtres, la méthode d'échantillonnage fréquentielle, la méthode des moindres carrés pondérés, ou la synthèse Equiripple [Kamp et al., 1975 ; Rajan et al., 1983]. L'inconvénient majeur des filtres conçus par ces techniques est qu'ils contiennent de nombreuses petites ondulations dans la bande passante et dans la bande atténuée. La plupart des méthodes ne permettent pas un contrôle indépendant des caractéristiques des bandes (passante, atténuée et de transition), en particulier la largeur de la bande de transition. Les méthodes classiques pour la conception des filtres RIF 2D utilisant les fonctions de la fenêtre et de l'échantillonnage en fréquence exigent un ordre de filtre élevé : un contrôle précis des fréquences de coupure entre la bande passante et la bande atténuée reste difficile. Dans la pratique, ces méthodes sont très limitées [Pei et al., 1996 ; Karaboğa et al., 2006].

De nouvelles approches de conception ont été élaborées et ont donné des résultats intéressants par rapport aux approches mentionnées ci-dessus. Par exemple, la méthode d'échange de Remez (Remez Exchange) est basée sur un algorithme de programmation

linéaire [Lee et al., 2000 ; Charalambous, 1985; Lu et al., 2000]. Bien que cette technique soit intéressante, les résultats sont limités, car la difficulté est de converger à l'une des solutions minimax. Malheureusement, on sait que pour certains cas, la méthode de Remez peut conduire à un comportement divergent, même si la première approximation de départ est assez bonne. L'algorithme des moindres carrés pondérés (WLS) fournit également quelques améliorations, et peut acquérir des solutions analytiques. Cette méthode souffre cependant de l'inconvénient de la précision de l'approximation qui se détériore considérablement au voisinage des bords des bandes. En outre, il faut calculer une matrice inverse qui peut être une problématique de calcul lorsque le degré des filtres est très élevé.

Récemment, plusieurs techniques d'optimisation usuelles [Lee et al., 2000] ont été développées. L'étude dans [Lee et al., 2000] traite le problème de conception minimax des filtres numériques RIF bidimensionnels (2D), à phase linéaire, à coefficients continus et de puissance de deux (POT). Dans cette étude, le problème de conception continu-minimax de coefficients est exprimé comme un problème de programmation linéaire avec des contraintes d'inégalité. Un certain nombre de méthodes efficaces et bien établies de conception numérique de filtres RIF 2D et ses associations complexes ont été proposées (par exemple, [Lee et al., 2000 ; Bhattacharya et al., 1999]).

La conception de filtres RIF 2D est extrêmement challengée par la communauté scientifique. Le challenge vise à obtenir des techniques plus générales et innovantes qui sont en mesure de relever de nouveaux et complexes problèmes d'ingénierie d'une grande pertinence aujourd'hui. Les algorithmes évolutionnaires et génétiques (AG) peuvent être une alternative intéressante pour la conception des filtres. Des scientifiques et des chercheurs dans une grande variété de domaines ont publié des articles et des études tutoriels informatiques documentant avec succès l'extension de la frontière des problèmes qui peuvent être traités efficacement par les algorithmes évolutionnaires. Les AG's ont montré qu'ils peuvent être plus efficaces et plus attrayants à trouver les solutions optimales ou quasi-optimales que les algorithmes classiques.

En conséquence, des approches, tels que les réseaux de neurones, les algorithmes génétiques, et d'autres outils liés à l'intelligence artificielle [Bhattacharya et al., 1999 ; Tzeng, 2007; Sriranganathan et al., 1995 ; Williams et al., 2001], ont été consacrées à la synthèse de méthodes capables de satisfaire les contraintes qui seraient inaccessibles si elles étaient traitées avec les techniques classiques de conception. Dans certains cas, ces initiatives ont montré de meilleurs indices de performance que les approches conventionnelles. Certaines méthodes évolutionnaires de conception de filtres ont été mises en œuvre [Pei et al., 1996 ; Yeh et al., 1996 ; Quinquis, 2000 ; Cen, 2007; Tzeng, 2007] avec des résultats prometteurs. Nous avons déjà étudié les algorithmes génétiques pour la conception de filtres RIF 1D et 2D [Boudjelaba et al., 2004 ; Boudjelaba et al., 2011 ; Boudjelaba et al., 2011] par le développement d'algorithmes génétiques dédiés à la synthèse de filtres RIF 1D et 2D. Nous avons démontré que l'ajout d'une certaine souplesse dans les versions classiques de l'AG, les ondulations peuvent être considérablement réduites par rapport aux approches traditionnelles et génétiques standards [Tzeng, 2007].

Cependant, il y a quelques points faibles associés aux méthodes évolutionnaires, comme l'augmentation du coût de calcul et la non-existence de la preuve théorique de convergence

vers un optimum global dans des conditions suffisamment générales [Karaboğa et al., 2006]. Bien que ces inconvénients ne ressortent pas au premier plan dans les travaux publiés, l'utilisation d'un AG pour un problème dédié reste difficile pour les praticiens qui sont souvent sceptiques concernant les AG's. Il n'y a pas de processus systématiques pour trouver un équilibre efficace entre l'exploration et l'exploitation des capacités de la recherche pendant l'exécution. Plus important encore, le paramétrage simultané et le réglage de plusieurs paramètres peuvent souvent conduire à des coûts de calcul extrêmement élevés sans assurer l'obtention de bons résultats. Pendant plus de 10 ans, de nombreuses études prometteuses [Boeringer et al., 2005] provenant de la communauté de la reconnaissance de formes et périphériques ont porté sur les paramètres d'adaptation (la mutation, les taux de croisement et la taille des populations) et sur les propositions de nouvelles variantes (par exemple un croisement intelligent, une recombinaison élitiste) pour lutter contre ces bien connues restrictions générales. Nous avons révélé un décalage entre les connaissances des AG's et leur implémentation pour les filtres. Les algorithmes génétiques utilisés pour traiter les problèmes de conception de filtre pourraient tirer davantage parti de ces progrès.

Généralement, les algorithmes qui utilisent des techniques d'intelligence artificielle, à savoir les Algorithmes Evolutionnaires (Evolutionary Algorithms : EA), la Recherche Tabou (Tabu Search : TS) [Glover, 1989, 1990 ; Wu et al., 2009], le Recuit Simulé (Simulated Annealing : SA) [Kirkpatrick et al., 1983 ; Bohachevsky et al., 1986 ; Bertsimas et al., 1993] et la méthode Hill-Climbing (HC) [Dunn, 1998; Akramifar et al., 2010 ; Johnson et al., 2002], ont reçu une attention croissante dans le domaine de l'optimisation numérique globale. Ils ont été appliqués avec succès dans une large variété de domaines. Cependant, ils ne peuvent pas être efficaces pour tous les types de problèmes d'optimisation. Pour résoudre un problème d'optimisation complexe, le moyen le plus efficace est de concevoir un algorithme en mesure de convenir aux besoins spécifiques. Par conséquent, il est nécessaire de chercher des méthodes plus innovantes, capables de surmonter ces faiblesses.

Ces dernières années, les techniques d'Optimisation par Essaim de Particules (Particle Swarm Optimization : PSO) [Kennedy et al., 1995] ont attiré une attention considérable parmi les diverses heuristiques d'optimisation modernes, en raison essentiellement de leur grande vitesse de convergence, de leur structure algorithmique simple et leur forte capacité d'optimisation globale. Ses avantages ont été largement soulignés ainsi que ses faiblesses, surtout du côté du réglage de leurs paramètres. Il est surprenant que de plus en plus de chercheurs affirment la supériorité des PSO sur les AG's et soulignent leur capacité à résoudre des problèmes complexes par une mise en œuvre plus facile. Il est vrai que par rapport aux AG's, la PSO n'a pas besoin d'opérateurs tels que le croisement et la mutation, et peut paraître moins complexe. De plus, le calcul est peu coûteux en termes de mémoire et de temps d'exécution surtout dans sa version primaire (initiale). Il convient de noter que, si l'inspiration de ces métaheuristiques (AG et PSO) n'est pas la même, les bases des algorithmes dérivés sont cependant assez similaires. Ils visent à explorer de manière optimale l'espace de recherche à travers des mécanismes de diversification tout en essayant d'exploiter les avantages de l'exploration plus localement. L'exploration et l'exploitation sont gérées par un maillage stochastique de processus et de tâches déterministes, les questions liées à la convergence prématurée ou à la stagnation sont identiques. Leurs différences sont davantage liées aux stratégies et aux mécanismes de leur mise en œuvre. Les

AG's sont plus axés sur des approches issues de l'élitisme alors que les PSO sont plus inspirées par un comportement de collaboration.

On trouve plusieurs centaines de documents pertinents représentant un large échantillon de l'utilisation des algorithmes génétiques dans la littérature au cours des 20 dernières années et couvrent un large éventail de sujets. Toutefois, les AG's dépendent grandement de la quantité de réglage nécessaire pour atteindre une performance donnée, et ce coût dépend de la configuration globale de l'AG qui dépend directement du temps et des efforts, pour permettre de régler l'AG. Tout au long de notre documentation pertinente, on peut voir que le coût du réglage des paramètres est largement ignoré par la communauté scientifique. L'adaptation et la flexibilité des AG's sont actuellement l'un des domaines les plus importants et les plus prometteurs de la recherche en calcul évolutif.

Notre travail est en ligne avec le défi actuel. Il présente des approches alternatives qui tirent parti de la puissance de la technique d'optimisation globale stochastique bien connue, appelée algorithme génétique et son application à la synthèse des filtres RIF 2D et 1D. L'objectif est de présenter une approche d'ingénierie conforme à la spécification formelle axée sur la conception optimale des filtres RIF. La tâche de conception de filtres non récursifs est formulée en tant que problème d'optimisation sous contraintes. Bien que l'algorithme est adaptatif, avec des adaptations mineures, pour son application à n'importe quel type de filtres RIF, nous avons choisi de mettre l'accent sur des filtres RIF à coefficients réels, en raison de leur importance dans la pratique de l'ingénierie et de leurs propriétés intrinsèques (phase linéaire, stabilité, flexibilité et mise en œuvre facile).

En utilisant l'algorithme génétique, des résultats prometteurs sont présentés ici pour la conception de filtres RIF. Le processus d'optimisation est principalement régi par les opérateurs génétiques qui effectuent le croisement et la mutation adaptative dont les principes initiaux sont fondés sur les meilleures avancées récentes en provenance de la communauté de la reconnaissance des formes. Pour une classe de problèmes et le savoir-faire associé, l'AG peut être robuste dans une gamme de paramètres qui sont pré-calibrés comme des points de départ pour travailler. La principale nouveauté est l'intégration de plusieurs mécanismes dédiés et complémentaires pour aider la convergence de l'AG pour une conception plus efficace et pour permettre une utilisation pratique par des non-experts avec des configurations différentes (passe-bas, passe-haut, passe-bande et coupe-bande). Ces mécanismes offrent une réelle flexibilité et une simplicité pour l'utilisateur, car ils permettent à l'AG d'être calibré et à plusieurs reprises de se réinitialiser et de s'étalonner. La pertinence de l'AG est réalisée grâce à la stratégie et non par des paramètres initiaux qui jouent un rôle secondaire. En outre, leurs coûts informatiques sont fortement optimisés. Ensuite, l'utilisateur a seulement besoin d'entrer les caractéristiques du filtre puis un ensemble de paramètres d'entrée à préciser et quelques paramètres supplémentaires, sans être confronté à toute la complexité du calcul évolutif. En combinant le savoir-faire dans la conception de filtres et les meilleurs progrès dans la conception génétique, nous avons obtenu un soi-disant "algorithme libre" qui optimise la convivialité de l'AG.

Nous pensons que les résultats récents mettant en évidence la supériorité des nouvelles versions de la PSO, aux algorithmes génétiques devraient être beaucoup plus modérés. Un examen attentif de la littérature montre que la plupart des comparaisons sont plutôt limitées à

des algorithmes spécifiques lancées dans des concepts spécifiques. Ils ne peuvent pas être généralisés car il est plus une question d'adaptabilité et de flexibilité que de performances pures. Le principal problème pour les concepteurs des algorithmes évolutionnaires est causé par le fait que les détails de conception, à savoir, les valeurs des paramètres influencent largement la performance de l'algorithme. Notre objectif est donc de montrer que les performances sont plus liées au potentiel des algorithmes d'être souples ou même auto-tuning (auto-réglables et auto-adaptatifs) que les mécanismes eux-mêmes. Trois versions d'algorithmes PSO sont comparés aux deux versions des AG's. En plus de la version de base, nous avons sélectionné deux nouvelles versions des algorithmes PSO qui se sont révélées être efficaces pour la conception de filtres. En ce qui concerne les AG's, deux versions sont sélectionnés. La première version est l'AG standard et la seconde est l'AG adaptatif dédié à la synthèse des filtres RIF. Cette version implique des paramètres pré-calibrés et adaptables. Nous avons spécialement développé ces méthodes pour la conception de filtres RIF 1D. L'objectif est de fournir des indications sur la façon dont les AG's et les PSO travaillent, et de proposer des mécanismes dont la performance pourrait être améliorée par l'intégration de caractéristiques d'un modèle à l'autre.

La dernière partie de nos travaux vise à proposer une approche génétique hybride basée sur les meilleurs mécanismes génétiques pour améliorer la mise en œuvre des AG's. Plusieurs mécanismes intelligents sont intégrés et combinés dans l'algorithme génétique, qui le différencie des algorithmes génétiques classiques. Pour rendre l'hybridation docile et efficace, nous proposons un algorithme qui consiste à sélectionner, pour l'approche locale, seul un nombre réduit de prototypes chromosomiques qui sont à la fois prometteurs et représentatifs de la population actuelle. Cette hybridation améliore non seulement l'efficacité, mais évite en particulier les coûts de temps de calcul excessifs qui ne sont pas acceptables pour les praticiens. Le cadre est un système auto-adaptatif basé sur un processus dans lequel les paramètres pertinents sont implicites, c'est à dire qu'ils sont choisis par le cycle évolutif lui-même.

Au meilleur de nos connaissances, il semble n'y avoir aucune approche génétique signalée qui va si loin dans cette direction pour réaliser la conception rapide et efficace des filtres FIR 1D et 2D.

### *Formulation de la problématique et contributions*

Un des principaux objectifs dans le domaine de conception des filtres RIF est l'optimisation des coefficients des filtres synthétisés et du temps nécessaire au calcul de ces coefficients. Une optimisation efficace des mécanismes et des opérateurs des métaheuristiques constitue une partie de la solution pour augmenter les performances de conception des filtres RIF. En effet, si les métaheuristiques permettent d'approcher précisément les spécifications du filtre, celles-ci ne sont pas capables de résoudre le problème rapidement et ne reproduisent pas les mêmes résultats à chaque exécution pour un même filtre, et les performances s'en trouvent aléatoires.

Dans ce contexte, la communauté scientifique du signal processing a adapté et a appliqué un ensemble d'algorithmes évolutionnaires pour résoudre le problème de synthèse des filtres RIF. Le premier objectif était d'utiliser la méthode la plus efficace avec le critère d'erreur le mieux adapté au problème traité (la communauté scientifique limite ses tests à 2 ou 3 exemples de filtres RIF). Par ailleurs, ces algorithmes ne peuvent pas être entièrement adaptés à un problème autre que celui traité par les auteurs. En effet, même si dans un premier temps les algorithmes étaient conçus, pour répondre à des besoins spécifiques, il est aujourd'hui nécessaire, pour des raisons de performance, de répétabilité et d'universalité, de définir et de mettre en œuvre des méthodes et des outils supportant la synthèse de tous types de filtres RIF. La problématique consistant à définir, développer et faire évoluer un algorithme d'optimisation pour être utilisé pour tout problème de synthèse de filtres RIF, n'est pas propre au domaine du signal processing, elle se rapporte au contraire à l'ensemble des secteurs d'optimisation et des algorithmes évolutionnaires. A cet effet, le second objectif est de généraliser cette technique pour traiter tous les cas (type de filtres, gabarit de filtres, ...). Le troisième objectif est de faire évoluer ces algorithmes pour qu'ils soient conviviaux et facilement exploitables par les utilisateurs et les praticiens.

Une éventuelle voie qui pourrait traiter efficacement de cette problématique, peut reposer sur des algorithmes ayant des mécanismes adaptatifs auto-réglables ou des algorithmes hybrides exploitant les avantages de chaque technique. Toutefois, l'utilisation dans le domaine de la synthèse des filtres RIF des opérateurs adaptatifs ou de l'hybridation reste une question ouverte. De plus, les travaux menés jusqu'à maintenant dans le domaine de la synthèse des filtres ne proposent pas des approches permettant de faire évoluer les performances en fonction de l'évolution de la recherche dans le domaine des métaheuristiques.

L'articulation du travail peut se résumer en quatre points :

- proposer de nouvelles approches de synthèse optimale des filtres RIF basées sur les techniques évolutionnaires
- intégrer dans ces techniques des mécanismes de diversification et des opérateurs adaptatifs (auto-réglables) permettant d'ajuster les différents paramètres en fonction du filtre traité et de l'état d'avancement de la recherche des solutions au problème
- implémenter un algorithme adaptatif hybride reposant sur une sélection intelligente des référents qui doivent être traités par l'approche locale
- élaborer un algorithme convivial, facilement exploitable par les praticiens.

Les performances des algorithmes élaborés ont été démontrées à travers plusieurs exemples de synthèse et surtout par les comparaisons réalisées avec les résultats et les performances des techniques élaborées par la communauté du traitement du signal pour la synthèse des filtres RIF. Les travaux présentés dans cette thèse sont validés par la publication de trois articles dans des revues sélectives et de haut rang, avec Impact Factor (IET Signal Processing ; Expert Systems With Applications et Circuits, Systems, and Signal Processing), et un autre article en cours de relecture dans une revue de même importance (Pattern Analysis and Applications). Deux papiers ont été présentés à "The 2011 IEEE Workshop on Statistical Signal Processing (SSP'11)" et à "The 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing".

## *Cadre de la thèse*

Le travail présenté dans cette thèse a été réalisé dans le cadre d'une cotutelle internationale de thèse entre l'université d'Orléans (France) et l'université de Sétif – 1 (Algérie). Cette thèse a pour but d'élaborer des techniques évolutionnaires auto-adaptatives pour optimiser les coefficients des filtres RIF tout en réduisant le temps de processing. Elle a pour objectif principal de répondre à la volonté de la communauté scientifique de s'appuyer sur des métaheuristiques intégrant des mécanismes auto-adaptatifs en fonction de l'évolution de la recherche des solutions et surtout en fonction de la diversité de l'échantillon d'individus manipulé par l'algorithme. Ces algorithmes ont été développés en utilisant le logiciel Matlab.

## *Organisation du document*

Le document est organisé en quatre parties. La première porte sur la problématique abordée dans la thèse ainsi que sur l'état de l'art des méthodes conventionnelles appliquées à la synthèse des filtres RIF.

La deuxième partie présente un résumé théorique de certaines techniques évolutionnaires et un état de l'art de leur application par la communauté scientifique pour la synthèse des filtres RIF. A la fin de ce chapitre, nous avons mentionné les limites de ces méthodes et les éventuelles voies à suivre afin d'améliorer leur performances.

La troisième partie traite en détail les différents algorithmes et mécanismes mis en œuvre pour optimiser la synthèse des filtres RIF. Elle intègre aussi une description détaillée de l'algorithme hybride élaboré et développé pour la synthèse des filtres RIF 1D et 2D.

La quatrième partie contient les résultats expérimentaux. Elle présente les résultats des différents exemples traités, la performance des algorithmes et des comparaisons avec les algorithmes élaborés par la communauté scientifique, ainsi qu'une section dédiée à l'application des filtres RIF 2D conçus pour le traitement d'images.

La conclusion propose un résumé des performances, un bilan du travail effectué durant la thèse et un ensemble de perspectives ouvertes par celui-ci.

Contenu détaillé des chapitres présentés :

- Le premier chapitre introduit la problématique de la synthèse des filtres RIF. Il aborde comment réaliser la synthèse au moyen des méthodes conventionnelles et souligne les limitations de ces méthodes. Le chapitre présente aussi les différents critères d'erreur (normes) utilisés pour l'optimisation des coefficients des filtres RIF.
- Le deuxième chapitre présente quelques méthodes et techniques évolutionnaires utilisées pour la synthèse des filtres RIF. En outre, ce chapitre expose les travaux qui se sont intéressés au problème d'optimisation des coefficients des filtres RIF.
- Les solutions proposées par cette thèse sont exposées dans le troisième chapitre. Il présente les métaheuristiques d'optimisation mises en œuvre en définissant les différents opérateurs ainsi que les améliorations proposées. Il expose aussi, les nouveautés et les stratégies propres à chaque algorithme. Il explique comment les opérateurs d'évolution et d'évaluation peuvent être mis à jour en fonction de l'avancement de la recherche des solutions et des besoins.

- Le quatrième chapitre évalue les solutions proposées en présentant les différents résultats obtenus par chaque technique et métaheuristique développée dans le cadre de la thèse. Il décrit l'amélioration des performances engendrées par l'introduction des nouveaux mécanismes dans les différents algorithmes. Pour chaque algorithme, une étude comparative avec d'autres méthodes de la communauté scientifique est réalisée afin de souligner les avantages et les inconvénients de chaque méthode. Cela permet de déterminer l'intérêt d'utiliser des opérateurs adaptatifs, de mettre en évidence l'influence des mécanismes sur les performances et l'utilité de fournir un support hybride adaptatif, et comment celui-ci permet de s'assurer que l'algorithme développé s'adapte aux attentes des utilisateurs. La dernière section de ce chapitre est consacrée au traitement d'images par les filtres RIF 2D.
- La conclusion propose une synthèse et un bilan du travail effectué durant la thèse et un ensemble de perspectives liées à la continuation du travail en exploitant les conclusions des différents tests.

# Chapitre I

---

## 1. Filtres à réponse impulsionnelle finie

### Sommaire

---

1.	Filtres à réponse impulsionnelle finie .....	9
1.1.	Introduction .....	10
1.2.	Théorie sur les filtres RIF unidimensionnels (1D) .....	11
1.2.1.	Fonction de transfert et réponse fréquentielle .....	12
1.2.2.	Méthodes traditionnelles de synthèse des filtres RIF 1D .....	13
1.2.2.1.	Méthode des fenêtres.....	15
1.2.2.2.	Méthode de l'échantillonnage fréquentiel .....	17
1.2.2.3.	Méthodes itératives et d'optimisation .....	17
1.2.2.4.	Méthode de Remez.....	18
1.2.2.5.	Méthode de Parks-McClellan.....	21
1.2.3.	Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 1D .....	21
1.3.	Théorie sur les filtres RIF bidimensionnels (2D) .....	23
1.3.1.	Méthodes de synthèse des filtres RIF 2D.....	24
1.3.2.	Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 2D .....	25
1.4.	Limitation des méthodes conventionnelles de synthèse des filtres RIF. 26	
1.5.	Conclusion.....	27

---

## 1.1. Introduction

Les filtres numériques ont une large gamme d'applications dans les domaines de communication, de traitement d'images, de reconnaissance de formes, ... Les filtres sont un élément clé dans la plupart des systèmes électroniques et plus généralement de tout dispositif de traitement du signal. Un filtre numérique fonctionne sur des signaux échantillonnés à temps discret, à la différence des filtres analogiques qui opèrent sur des signaux analogiques en temps continu. La plupart des systèmes électroniques contiennent un grand nombre de filtres qui répondent aux exigences variées.

Le filtrage est une forme de traitement du signal, obtenu par l'émission du signal utile par l'intermédiaire d'un ensemble de circuits électroniques, ce qui modifie de façon adaptative son spectre de fréquence, de phase, et donc sa forme. Le filtre est conçu pour extraire certaines informations relatives à ce signal, généralement pour retourner, à partir d'un signal incident, un signal qui est plus intelligible en termes de l'information qu'il contient et que l'on veut sélectionner. Le filtrage peut éliminer ou réduire les fréquences de bruit indésirables, soit isoler un signal complexe dans les bandes de fréquences utiles. Les avantages des filtres sont considérables : les modèles peuvent être personnalisés, précis et adaptables ; au-delà de leur fonction utile, ils peuvent être facilement intégrés dans des ensembles existants et peuvent générer des réductions de coûts dans la conception des composants.

Les filtres numériques peuvent être classés en plusieurs groupes, selon les critères de classification utilisés. Les deux principaux types de filtres numériques, classés selon la longueur de la réponse impulsionnelle, sont les filtres numériques à réponse impulsionnelle finie (filtres RIF) et les filtres numériques à réponse impulsionnelle infinie (filtres RII). Chaque type de filtres a des avantages et des inconvénients qui doivent être examinés avec soin lors de leur conception.

Les filtres numériques à réponse impulsionnelle finie sont également connus sous le nom de filtres numériques non récursifs car ils n'ont pas de réaction, bien que des algorithmes récursifs peuvent être utilisés pour concevoir les filtres RIF. Ces filtres ont de nombreux avantages tels que la stabilité, la possibilité d'obtenir une phase linéaire. La phase linéaire fait référence à l'état dans lequel la réponse en phase du filtre est une fonction linéaire de la fréquence. Cela se traduit par le même retard à travers le filtre pour toutes les fréquences. Par conséquent, le filtre ne provoque pas de distorsion de phase.

Pour certains signaux, la caractéristique de phase est une caractéristique essentielle. C'est le cas pour les différents capteurs industriels où il est nécessaire pour un filtre à avoir une caractéristique de phase linéaire pour éviter de perdre des informations importantes.

Un filtre numérique est composé essentiellement par les éléments suivants :

- Un ou plusieurs organes de retard
- Plusieurs opérateurs arithmétiques (additionneurs et multiplicateurs)
- Plusieurs registres contenant les coefficients de pondération du filtre

La phase de conception d'un filtre numérique consiste à matérialiser l'algorithme de calcul, pour cela, il existe plusieurs possibilités de réalisation :

- Logique câblée (portes logiques, mémoires, réseaux systoliques,...)
- Logique programmée (processeur de traitement du signal "DSP : Digital Signal Processing", microprocesseur "ordinateur")

La synthèse et la conception complète des filtres RIF nécessite 4 étapes :

- La synthèse :
  1. Spécification des caractéristiques du filtre
  2. Calcul des coefficients du filtre
- La réalisation :
  3. Sélection (choix) de la structure de réalisation
  4. Implémentation

Notre travail consiste à élaborer les deux premières étapes (la synthèse des filtres RIF).

## 1.2. Théorie sur les filtres RIF unidimensionnels (1D)

La conception des filtres RIF est un processus itératif à plusieurs étapes. Les étapes clés sont la spécification du filtre, le calcul des coefficients et implémentation de la structure de réalisation.

La fonction de transfert en  $Z$  donne la relation entre l'entrée et la sortie dans le plan  $Z$ .

$$Y(Z) = H(Z) * X(Z) \text{ ou } H(Z) = \frac{Y(Z)}{X(Z)}$$

Tout filtre numérique linéaire peut être mathématiquement défini par une fonction polynomiale complexe. Cette fonction polynomiale est écrite comme le quotient de deux produits.

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\prod_{n=0}^{2N} (Z - Z_n)}{\prod_{m=0}^k (Z - p_m)} \quad (1.1)$$

Le numérateur spécifie les zéros du filtre et le dénominateur les pôles du filtre.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{n=0}^{2N} b_n * Z^{-n}}{1 + \sum_{m=1}^{2M} a_m * Z^{-m}} \quad (1.2)$$

Dans le cas d'un filtre RIF,  $a_1 = a_2 = \dots = a_{2M} = 0$

La fonction de transfert des filtres à réponse impulsionnelle finie (RIF) est de type polynomial, de ce fait, cette fonction de transfert n'a pas de pôles à l'extérieur du cercle unité. Cette absence de pôles permet d'obtenir des filtres inconditionnellement stables. De plus, ces filtres sont caractérisés par la possibilité d'avoir une phase linéaire grâce à la symétrie des coefficients de leur réponse impulsionnelle, ce qui leur confère un grand intérêt pour le filtrage.

### 1.2.1. Fonction de transfert et réponse fréquentielle

Ces filtres sont appelés aussi non-récurrents et leur fonction de transfert est exprimée comme un polynôme de variables complexes  $Z^{-1}$ .

$$H(z) = \sum_{n=0}^{2N} h(n) * Z^{-n} \quad (1.3)$$

$2N$  est l'ordre du filtre qui présente  $2N + 1$  coefficients et  $h(n)$  la réponse impulsionnelle du filtre.

La réponse fréquentielle est obtenue à partir de la fonction de transfert en remplaçant  $Z$  par  $e^{j\omega}$  ( $Z = e^{j\omega}$ ).

$$\hat{H}(\omega) = \sum_{n=0}^{2N} h(n) * e^{-jn\omega} \quad (1.4)$$

La propriété de causalité, qui traduit le fait que dans un filtre réel la sortie ne peut précéder l'entrée dans le temps, implique que la réponse impulsionnelle  $h(n)$  soit nulle pour  $n < 0$  [Bellanger, 2006].

Afin de rendre la caractéristique de phase d'un filtre RIF linéaire, la réponse impulsionnelle doit être symétrique ou antisymétrique.

$h(n) = h(N - n - 1)$  ; réponse impulsionnelle symétrique (autour de son élément central)

$h(n) = -h(N - n - 1)$  ; réponse impulsionnelle antisymétrique (autour de son élément central)

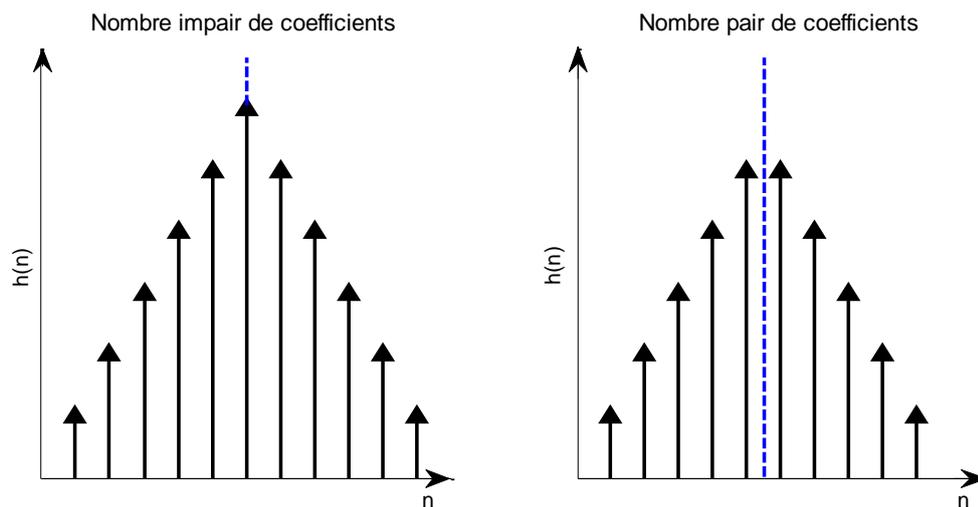


Figure 1.1 – Réponse impulsionnelle symétrique

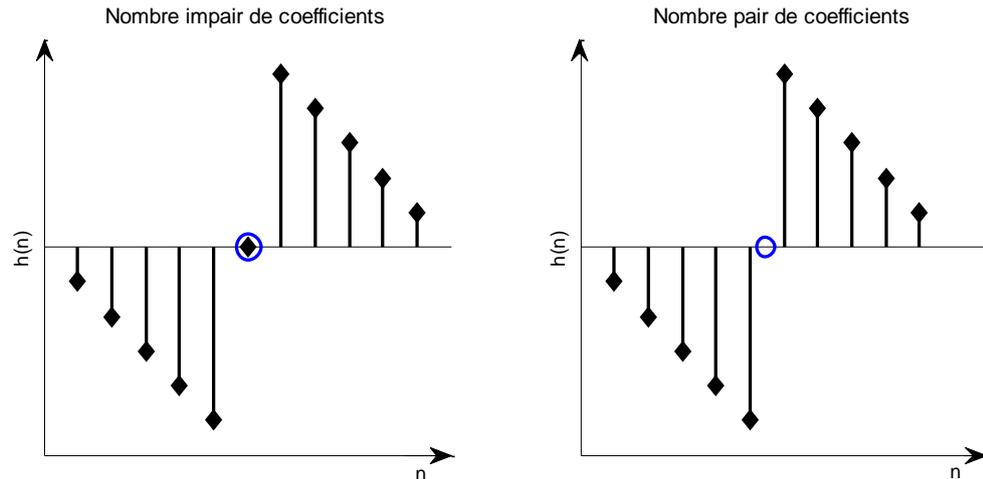


Figure 1.2 – Réponse impulsionnelle antisymétrique

La réponse fréquentielle d'un filtre RIF symétrique à phase linéaire avec une longueur  $2N + 1$  peut être caractérisée comme suit :

$$\hat{H}(\omega) = e^{-jN\omega} \sum_{n=0}^N b_n * \cos(n\omega) = e^{-jN\omega} * H(\omega) \quad (1.5)$$

$$H(\omega) = \sum_{n=0}^N b_n * \cos(n\omega) \quad (1.6)$$

Tel que :

$$b_n = \begin{cases} h(N) & n = 0 \\ 2h(N - n) & n = 1, 2, \dots, N \end{cases} \quad (1.7)$$

Et  $H(\omega)$  est la réponse d'amplitude utilisée pour approximer la réponse d'amplitude désirée  $D(\omega)$  au sens des moindres carrés.

Par conséquent, étant donné la réponse d'amplitude désirée  $D(\omega)$  et la longueur du filtre  $2N + 1$ , la synthèse du filtre RIF consiste à rechercher les coefficients optimaux du filtre  $b_n$  pour satisfaire les spécifications souhaitées du filtre. L'erreur à minimiser est l'erreur quadratique entre la réponse fréquentielle idéale et réelle du filtre.

$$E = \sum_{\omega} [D(\omega) - H(\omega)]^2 \quad (1.8)$$

### 1.2.2. Méthodes traditionnelles de synthèse des filtres RIF 1D

Traditionnellement, il existe de nombreux procédés bien connus pour la conception des filtres RIF, telles que la méthode de la fenêtre, la méthode de l'échantillonnage en fréquence, ... Ces méthodes sont rapides, pratiques, robustes mais généralement sous-optimales. Ces méthodes classiques ne peuvent pas optimiser les fonctions objectives (fitness) et ne peuvent pas converger vers la solution globale.

Les méthodes de synthèse classique peuvent être regroupées en trois classes :

- La méthode des fenêtres
- La méthode de l'échantillonnage en fréquence
- Les méthodes itératives et d'optimisation

La méthode d'optimisation la plus fréquemment utilisée pour la conception de filtres RIF à phase linéaire au sens de Chebyshev pondéré est celle basée sur l'algorithme de l'échange de Remez proposé par Parks et McClellan. La principale limitation de cette procédure est que les valeurs relatives de l'erreur d'amplitude dans les bandes de fréquences sont spécifiées à l'aide de la fonction de pondération, et non par les écarts eux-mêmes.

Ces méthodes de synthèse permettent de satisfaire les spécifications du filtre. Les spécifications qui définissent le gabarit du filtre peuvent être résumées comme suit :

- Définir la bande passante du filtre (la fréquence de coupure  $\omega_c$  ou les fréquences de coupures  $\omega_{1c}, \omega_{2c}$ )
- La largeur de la bande de transition ( $\Delta\omega$ )
- L'ondulation maximale (tolérée) en bande passante  $\delta_p$
- L'ondulation maximale (tolérée) en bande affaiblie (atténuée)  $\delta_a$
- L'atténuation maximale en bande affaiblie

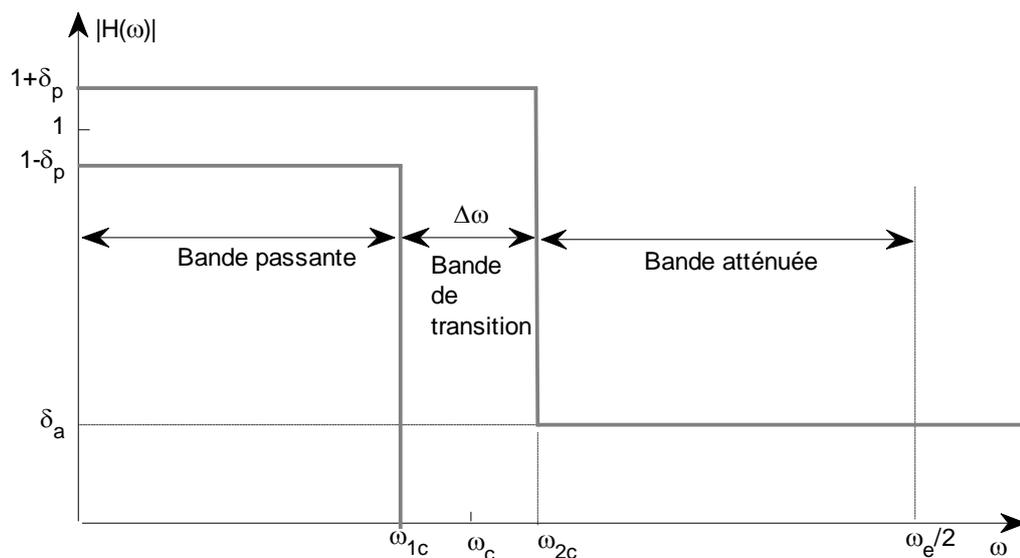


Figure 1.3 – Gabarit du filtre (Linéaire).

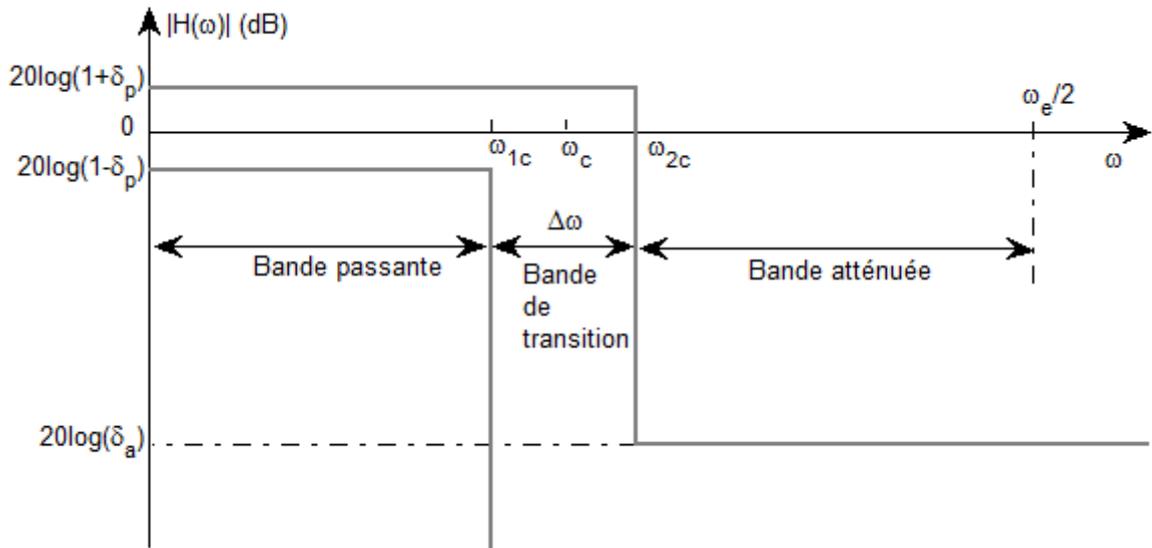


Figure 1.4 – Gabarit du filtre (en dB).

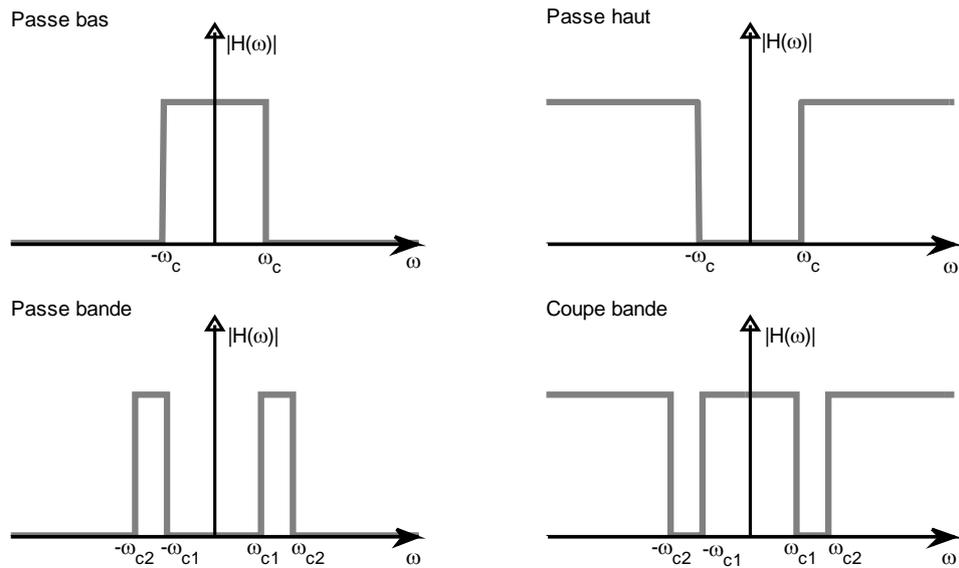


Figure 1.5 – Réponses d'amplitude pour différents types de filtres.

### 1.2.2.1. Méthode des fenêtres

La méthode des fenêtres permet de réaliser, à partir d'une réponse en fréquence idéale, la synthèse de filtres RIF [Blanchet et al., 1998].

Spécification au préalable de la réponse en fréquence idéale  $H_{\infty}(\omega)$  pour l'intervalle  $[-\pi, +\pi]$ :

$$H_{\infty}(\omega) = H_d(\omega) * e^{-j\frac{N-1}{2}\omega} \quad (1.9)$$

Détermination de la réponse impulsionnelle idéale (coefficients du filtre idéal) à l'aide de la transformée de Fourier inverse :

$$h_{\infty}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{\infty}(\omega) * e^{jn\omega} d\omega \quad (1.10)$$

A ce stade, on peut noter deux problèmes :

- Le nombre d'échantillons est infini ( $h_{\infty}(n)$  existe  $\forall n$ )
- $h_{\infty}(n)$  n'est pas causal ( $h_{\infty}(n) \neq 0$  pour  $n < 0$ )  $\rightarrow$  impossible de réaliser le filtre en temps réel.

Solution au premier problème : Comme la séquence  $h_{\infty}(n)$  n'est pas de support fini, pour obtenir un filtre RIF, on ne retient que  $N$  échantillons de celle-ci :

$$h(n) = \begin{cases} h_{\infty}(n) & n = -\frac{N-1}{2} : \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (1.11)$$

La réponse fréquentielle du filtre est obtenue par convolution de la réponse en fréquence souhaitée et du spectre de la fenêtre rectangulaire de longueur  $N$ .

$$H(\omega) = H_{\infty}(\omega) \otimes W_N(\omega) \quad (1.12)$$

Il existe plusieurs fenêtres de pondération : Hamming, Hanning, Blackman, Kaiser.

Le produit de convolution  $H_{\infty}(\omega) \otimes W_N(\omega)$  introduit :

- Le phénomène de Gibbs : élargissement de la bande de transition (la transition entre la bande passante et la bande atténuée est moins abrupte que pour  $H_{\infty}(\omega)$ ).
- L'apparition d'ondulations dans la bande passante et la bande atténuée.

La fenêtre rectangulaire donne une bande de transition étroite mais produit des ondulations élevées en bandes passante et atténuée. Par contre, la fenêtre de Hamming donne une large bande de transition mais produit des ondulations moins élevées.

La pondération temporelle introduit des ondulations et limite la raideur de coupure du filtre. Un compromis est à faire entre la raideur et l'amplitude des ondulations.

Solution au deuxième problème : on translate les échantillons de  $h(n)$  pour rendre le filtre causal.

$$h(n) = \begin{cases} h_{\infty}(n) & n = 0 : N - 1 \\ 0 & \text{ailleurs} \end{cases} \quad (1.13)$$

L'intérêt de la méthode des fenêtres réside dans la simplicité de calcul des coefficients et implique un nombre relativement faible de calcul. L'utilisation de cette méthode nécessite néanmoins, la connaissance de l'expression analytique de la réponse en fréquence

idéale  $H_\infty(\omega)$ . L'inconvénient majeur de cette méthode est la présence d'ondulations parasites dans sa réponse en fréquence (solution non optimale). On peut citer aussi, un filtre d'ordre supérieur est nécessaire pour satisfaire les spécifications du filtre.

### 1.2.2.2. Méthode de l'échantillonnage fréquentiel

Quand on ne connaît pas l'expression de la réponse fréquentielle, la méthode des fenêtres n'est pas applicable pour la synthèse des filtres RIF.

On échantillonne la réponse désirée dans le domaine fréquentiel ( $N$  échantillons uniformément espacés dans l'intervalle  $[0, 2\pi]$ ).

$$H(k) = H_\infty(\omega)_{(\omega=k\frac{2\pi}{N})} \text{ tel que } k = 0, \dots, N - 1$$

La réponse impulsionnelle est calculée par l'utilisation de la transformée de Fourier inverse de la séquence  $H(k)$ .

### 1.2.2.3. Méthodes itératives et d'optimisation

Parmi ces méthodes, on peut citer la méthode des moindres carrés, l'approximation pondérée au sens de Chebyshev, ...

Ce sont des méthodes qui réalisent l'optimisation au sens d'un certain critère par rapport au gabarit initial.

Dans ces méthodes, on utilise différentes normes du critère d'erreur.

$$E(\omega) = D(\omega) - H(\omega) \quad (1.14)$$

Erreur norme  $L_2$  :

$$E_2 = \left[ \sum_{\omega} |E(\omega)|^2 \right]^{1/2} \quad (1.15)$$

Erreur norme  $L_p$  :

$$E_p = \left[ \sum_{\omega} |E(\omega)|^p \right]^{1/p} \quad (1.16)$$

Erreur norme  $L_\infty$  : (Norme de Chebyshev)

$$E_\infty = \max |E(\omega)| \quad (1.17)$$

*Méthode des moindres carrés :*

La méthode des moindres carrés est basée sur le même principe que la méthode des fenêtres, la seule différence réside dans l'introduction et la spécification de quelques échantillons,

dans la bande de transition qui relie la bande passante et la bande atténuée. Cette technique permet d'obtenir des filtres de réponse impulsionnelle plus courte que par la méthode des fenêtres pour satisfaire le même gabarit et les mêmes contraintes.

Le critère d'erreur (d'optimalité) utilisé est la différence entre la réponse fréquentielle désirée et actuelle :

$$E(\omega) = W(\omega) * [D(\omega) - H(\omega)] \quad (1.18)$$

$W(\omega)$  est une fonction de pondération.

L'erreur pondérée est donnée par :

$$\varepsilon_2 = E_2^2 = \sum_{\omega} |E(\omega)|^2 \quad (1.19)$$

Connaissant le type de filtre et l'ordre du filtre, il ne reste qu'à déterminer les coefficients du filtre qui minimise  $\varepsilon_2$ .

*Equiripple design (oscillations contrôlées) :*

Cette méthode utilise la même erreur  $E(\omega)$  que la précédente technique mais la différence réside dans le critère à minimiser.

$$\varepsilon = \max_{\omega} |E(\omega)|$$

Le procédé est optimal dans le sens où on minimise le maximum des amplitudes des ondulations dans la bande passante et atténuée.

$$\min(\max_{\omega} |E(\omega)|)$$

On peut voir que ce critère mène à un filtre Equiripple ; un filtre dont la réponse en amplitude oscille de manière uniforme dans les limites de tolérance de chaque bande.

#### **1.2.2.4. Méthode de Remez**

La méthode de Remez permet d'introduire des critères d'optimisation tel qu'un réglage séparé des ondulations en bandes passantes et atténuées. C'est la méthode classique qui est considérée comme optimale pour déterminer les différents coefficients d'un filtre RIF. Cette méthode est basée sur une distribution uniforme de l'ondulation sur l'ensemble de la bande passante et sur une distribution uniforme de l'affaiblissement sur l'ensemble de la bande de rejection. Les filtres RIF résultants possèdent une réponse en phase linéaire. Ils sont nettement plus performants que les filtres RIF obtenus par la méthode des fenêtres. On les appelle aussi filtres minimax [Salman et al., 1982 ; Bellanger, 1987].

La méthode recherche itérativement les coefficients afin qu'avec un ordre minimal, le gabarit soit respecté au mieux. Par rapport à un filtre RIF obtenu avec la méthode des fenêtres, on a les avantages suivants [Salman et al., 1982]:

- Pour un gabarit identique, l'ordre d'un filtre RIF synthétisé par la méthode de Remez est nettement inférieur.
- L'ondulation dans la bande passante et l'affaiblissement minimal dans la bande de réjection (atténuée) ( $\delta_p, \delta_a$ ) sont configurables séparément.

*Détail de la méthode :*

Cette méthode réalise la synthèse comme étant un problème d'optimisation pondérée au sens de Chebyshev.

La fonction de pondération est donnée par :

$$W(e^{j\omega}) = \begin{cases} \delta_a/\delta_p & \text{dans la bande passante} \\ 1 & \text{dans la bande atténuée} \end{cases} \quad (1.20)$$

On peut exprimer l'erreur d'approximation sous la forme :

$$E(e^{j\omega}) = W(e^{j\omega}) * [H_d(e^{j\omega}) - Q(e^{j\omega}) * P(e^{j\omega})] \quad (1.21)$$

$H_d(e^{j\omega})$  est la réponse en fréquence désirée.

Comme  $Q(e^{j\omega})$  est fixé, on a :

$$E(e^{j\omega}) = W(e^{j\omega}) * Q(e^{j\omega}) * \left[ \frac{H_d(e^{j\omega})}{Q(e^{j\omega})} - P(e^{j\omega}) \right] \quad (1.22)$$

Cette formule n'est valable que dans le cas où  $Q(e^{j\omega})$  est différent de zéro.

$$E(e^{j\omega}) = W'(e^{j\omega}) * [H'_d(e^{j\omega}) - P(e^{j\omega})] \quad (1.23)$$

Tels que :  $H'_d(e^{j\omega}) = \frac{H_d(e^{j\omega})}{Q(e^{j\omega})}$  et  $W'(e^{j\omega}) = W(e^{j\omega}) * Q(e^{j\omega})$

La minimisation au sens de Chebyshev est caractérisée par ce critère :

$$\min_{coef\ de\ P} \left( \max_{\omega \in [0, \pi]} |E(e^{j\omega})| \right) \quad (1.24)$$

La résolution s'effectue à l'aide du théorème d'alternance :

Soit  $P$  une combinaison linéaire de  $M$  fonctions sinusoïdales :

$$P(e^{j\omega}) = \sum_{m=0}^{M-1} p_m \cos(m\omega) \quad (1.25)$$

Condition nécessaire et suffisante :  $P(e^{j\omega})$  est l'unique et meilleure approximation au sens de Chebyshev de  $H'_d(e^{j\omega})$  si l'erreur  $E(e^{j\omega})$  possède au moins  $M + 1$  extrema sur l'intervalle  $[0, \pi[$ .

Ou : il existe  $M + 1$  pulsations telles que :

$$E(e^{j\omega_i}) = -E(e^{j\omega_{i+1}}) \quad \omega_0 < \omega_1 < \dots < \omega_M$$

$$E(e^{j\omega_i}) = \max |E(e^{j\omega})| \text{ pour } \omega \in [0, \pi[ \quad (1.26)$$

L'algorithme de Remez consiste à chercher les pulsations  $\omega_i$  de la meilleure approximation.

On se donne un ensemble de  $M + 1$  pulsations.

Soit  $\delta$  l'erreur (ondulation) tolérée. On pose :

$$W'(e^{j\omega}) * [H'_d(e^{j\omega}) - P(e^{j\omega})] = (-1)^i \delta, \quad i = 0, 1, \dots, M$$

Avec

$$P(e^{j\omega}) = \sum_{m=0}^{M-1} p_m \cos(m\omega) \quad (1.27)$$

La solution est obtenue par inversion matricielle.

$$[A][p] = [H'_d] \quad (1.28)$$

$$[A] = \begin{pmatrix} 1 & \cos\omega_0 & \cos 2\omega_0 & \dots & \cos(M-1)\omega_0 & \frac{1}{W'(e^{j\omega_0})} \\ 1 & \cos\omega_1 & \cos 2\omega_1 & \dots & \cos(M-1)\omega_1 & \frac{-1}{W'(e^{j\omega_1})} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cos\omega_M & \cos 2\omega_M & \dots & \cos(M-1)\omega_M & \frac{(-1)^M}{W'(e^{j\omega_M})} \end{pmatrix} \quad (1.29)$$

$$[p] = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{M-1} \\ \delta \end{pmatrix} \quad (1.30)$$

$$[H'_d] = \begin{pmatrix} H'_d(e^{j\omega_0}) \\ H'_d(e^{j\omega_1}) \\ \vdots \\ H'_d(e^{j\omega_M}) \end{pmatrix} \quad (1.31)$$

Algorithme de la méthode de Remez :

1. Initialisation d'un ensemble de pulsations "extrema"
2. Résoudre le système d'équations (inversion de matrice) pour obtenir  $\delta$  et  $p_0, \dots, p_M$
3. Calculer  $P(\omega)$  et le module de l'erreur :  $|E(e^{j\omega})| = |W'(e^{j\omega}) * [H'_d(e^{j\omega}) - P(e^{j\omega})]|$

4. Localisation des nouvelles pulsations où  $|E(e^{j\omega})|$  est maximale et notamment où  $|E(e^{j\omega})| \geq \delta$

5. Calculer le paramètre de convergence  $Q = \frac{\max|E(e^{j\omega_i})| - \min|E(e^{j\omega_i})|}{\max|E(e^{j\omega_i})|}$

6. Rejeter les "extrema" superflus

7. Si  $Q > \varepsilon$  où  $\varepsilon$  est un paramètre de tolérance, itérer à partir de 2, sinon continuer en 8.

8. Calculer  $P(\omega)$  en utilisant le dernier ensemble de pulsations et en déduire  $h(n)$  la réponse impulsionnelle du filtre désiré

Malgré ces avantages, la méthode de Remez nécessite une grande quantité de calcul; ce qui la rend inappropriée pour les applications où la conception doit être faite en temps réel.

#### **1.2.2.5. Méthode de Parks-McClellan**

Cette méthode est basée sur une reformulation du problème de conception de filtre en un problème d'approximation polynomiale. Cet algorithme, n'est qu'une variante de l'algorithme de Remez appliqué à la synthèse des filtres RIF [Parks et McClellan, 1972].

C'est la méthode optimale selon le critère de Chebyshev pour déterminer les coefficients d'un filtre RIF [McClellan et Parks, 1973].

#### **1.2.3. Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 1D**

Les méthodes de synthèse traditionnelles ont l'avantage d'être simple à mettre en œuvre mais souffrent d'un inconvénient majeur qui est lié à la non-optimalité des solutions obtenues par ces méthodes. A cet effet, la synthèse des filtres RIF 1D et 2D a reçu un intérêt considérable de la part de la communauté scientifique. Dès les années 70', certains scientifiques ont commencé à proposer des méthodes d'optimisation afin d'éviter les inconvénients engendrés par l'utilisation des méthodes traditionnelles.

Parmi les précurseurs, on retrouve Parks et McClellan qui ont présenté dans [Parks et McClellan, 1972] une méthode classique efficace, qui porte leurs noms, pour la conception des filtres à réponse impulsionnelle finie à phase linéaire. Leur méthode vise à obtenir une approximation optimale au sens de Tchebychev à intervalles distincts correspondant à la bande passante et atténuée. Cette approche permet la spécification exacte des fréquences limites des bandes contrairement aux autres algorithmes classiques qui ne pouvaient pas contrôler directement les limites de la bande passante et atténuée. L'année suivante dans [McClellan et Parks, 1973], ils ont pu mettre en œuvre un programme polyvalent qui est capable de concevoir (au sens minimax) une grande classe de filtres numériques à phase linéaire. Le programme dispose d'options pour la conception de filtres standards comme le passe-bas, passe-haut, passe-bande et le coupe-bande. Le programme peut également être utilisé pour concevoir des filtres dont les spécifications arbitraires en fréquence approximatives peuvent être spécifiées par l'utilisateur. Et au milieu des années 90, Karam et

McClellan [Karam et McClellan, 1995] ont appliqué le théorème d'alternance au cas de filtres à coefficients complexes. Le problème de conception de filtre RIF complexe est reformulé de sorte qu'il satisfait clairement à la condition de Haar de l'approximation de Chebyshev. Un algorithme d'échange efficace est dérivé pour la synthèse des filtres RIF complexes au sens Chebyshev. En transformant la fonction d'erreur complexe, l'algorithme d'échange de Remez peut être utilisé pour calculer l'approximation complexe optimale au sens de Chebyshev. Ce nouvel algorithme est une généralisation de l'algorithme de Parks - McClellan, de manière que l'amplitude arbitraire et les réponses de phase peuvent être estimées.

Parmi les méthodes d'optimisation, on peut citer celle proposée par Potchinkov dans [Potchinkov, 1997]. Cette approche est basée sur une technique de programmation semi-infinie et permet la synthèse des filtres RIF avec ou sans contraintes, aussi bien au sens minmax qu'au sens des moindres carrés. Des solutions très précises pour des filtres pouvant atteindre 2000 coefficients ont été obtenues d'après l'auteur.

En 1999, McClellan et son collègue Karam proposent une nouvelle fois dans [Karam et McClellan, 1999], un nouvel algorithme d'échanges multiples pour la conception des filtres RIF de Chebyshev, avec des spécifications d'amplitude et de phase arbitraires. Comparé aux techniques de conception de filtres de Chebyshev existantes, ce nouvel algorithme présente une convergence plus rapide tout en maintenant une haute précision (convergence vers la solution optimale).

Une nouvelle version de l'algorithme des moindres carrés a été mise en œuvre par Kao dans [Kao et al., 2000]. Cet algorithme est différent de l'algorithme de recherche standard qui essaye directement toutes les combinaisons possibles. Cet algorithme utilise les propriétés géométriques d'un hyperespace pour mettre en évidence les solutions potentielles d'une manière beaucoup plus restrictive. La performance notable de cette technique est que cet algorithme produit des solutions aux moindres carrés avec un temps de recherche bien plus réduit que l'approche de recherche standard (méthode des moindres carrés).

[Shyu et al., 2008] ont réalisé la synthèse des filtres RIF 1D à coefficients complexes et à retard de groupe variable en utilisant la méthode des moindres carrés. Une fonction de pondération a été incluse dans le critère d'erreur afin d'optimiser la réponse en amplitude des filtres conçus.

Récemment, la norme  $L_\infty$  a été utilisée comme critère à optimiser par Apostolov dans [Apostolov, 2011], en introduisant une nouvelle fonction de modulation qui comprime les oscillations du cosinus. Le polynôme d'approximation est calculé en utilisant l'algorithme d'échange de Remez. Un polynôme optimal avec le degré le plus bas possible, qui se rapproche de la réponse du filtre idéal avec une grande précision, est proposé.

Pour le moment, Chan est le dernier auteur qui a proposé une méthode d'optimisation conventionnelle (non évolutionnaire). Dans [Chan et al., 2012], il propose une technique qui applique l'inférence bayésienne à la synthèse des filtres RIF avec "coefficients signés puissances de deux" (spot). Étant donné une réponse d'amplitude désirée spécifiée par les bornes supérieures et inférieures en décibels, l'estimation bayésienne des paramètres et la sélection de modèles sont adaptés pour produire une distribution de conceptions possibles.

En dépit du succès de certaines techniques proposées dans la littérature, ces méthodes basées sur des algorithmes non évolutionnaires souffrent de plusieurs limites inhérentes surtout en ce qui concerne les erreurs produites en bande passante et atténuée, sans oublier de citer la non applicabilité de ces algorithmes à toutes les catégories des filtres RIF. Ces approches restent donc dédiées à des exemples restreints et posent des problèmes dès qu'on s'éloigne des exemples proposés par les auteurs.

### 1.3. Théorie sur les filtres RIF bidimensionnels (2D)

La fonction de transfert d'un filtre RIF bidimensionnel (2D) de dimension  $N_1 * N_2$  est donnée par :

$$H(Z_1, Z_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} h(n_1, n_2) * Z_1^{-n_1} * Z_2^{-n_2} \quad (1.32)$$

Tel que  $h(n_1, n_2)$  est sa réponse impulsionnelle.

La réponse fréquentielle est donnée par :

$$H(\omega_1, \omega_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} h(n_1, n_2) * e^{-j.(n_1\omega_1+n_2\omega_2)} = M(\omega_1, \omega_2) * e^{j.\theta(\omega_1, \omega_2)} \quad (1.33)$$

Avec  $M(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|$  et  $\theta(\omega_1, \omega_2) = \arg(H(\omega_1, \omega_2))$

$M(\omega_1, \omega_2)$ , la réponse d'amplitude de  $H(\omega_1, \omega_2)$ , est une fonction à valeurs réelles.

Dans le cas d'un filtre RIF symétrique, la réponse fréquentielle peut s'écrire sous la forme suivante.

$$H(\omega_1, \omega_2) = M(\omega_1, \omega_2) * e^{-j\left[\frac{(N_1-1)}{2}\omega_1 + \frac{(N_2-1)}{2}\omega_2\right]} \quad (1.34)$$

$$M(\omega_1, \omega_2) = |F(\omega_1, \omega_2)| \quad (1.35)$$

$$F(\omega_1, \omega_2) = \sum_{k_1=0}^{P_1} \sum_{k_2=0}^{P_2} a(k_1, k_2) * \cos(k_1\omega_1) * \cos(k_2\omega_2) \quad (1.36)$$

$$P_1 = \frac{(N_1-1)}{2} \text{ et } P_2 = \frac{(N_2-1)}{2}$$

$N_1$  et  $N_2$  sont des entiers impairs.

L'idée principale est d'obtenir une approximation au sens des moindres carrés de la réponse d'amplitude, ce qui conduit à un rapprochement optimal de la solution. L'erreur quadratique utilisée à minimiser peut être exprimée sous la forme suivante :

$$E = \sum_{j=1}^{m_1} \sum_{k=1}^{m_2} [D(\omega_{1j}, \omega_{2k}) - F(\omega_{1j}, \omega_{2k})]^2 \quad (1.37)$$

Où  $D$  est la réponse d'amplitude désirée et  $F$  l'actuelle réponse d'amplitude.

$$E_i = \sum_{j=1}^{m_1} \sum_{k=1}^{m_2} [D(\omega_{1j}, \omega_{2k}) - C_1(\omega_{1j}) * A_i * C_2(\omega_{2k})^T]^2 \quad (1.38)$$

Tels que  $A_i$  et  $E_i$  sont la solution calculée et l'actuelle erreur quadratique obtenue avec cette solution, respectivement.

$$C_1(\omega_{1j}) = \begin{bmatrix} 1 & \cos(\omega_{11}) & \cdots & \cos(P_1 \cdot \omega_{11}) \\ 1 & \cos(\omega_{12}) & \cdots & \cos(P_1 \cdot \omega_{12}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos(\omega_{1m_1}) & \cdots & \cos(P_1 \cdot \omega_{1m_1}) \end{bmatrix} \quad (1.39)$$

$$C_2(\omega_{2k}) = \begin{bmatrix} 1 & \cos(\omega_{21}) & \cdots & \cos(P_2 \cdot \omega_{21}) \\ 1 & \cos(\omega_{22}) & \cdots & \cos(P_2 \cdot \omega_{22}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos(\omega_{2m_2}) & \cdots & \cos(P_2 \cdot \omega_{2m_2}) \end{bmatrix} \quad (1.40)$$

La matrice  $A_i$  formée par les coefficients du filtre de la solution calculée est donnée par :

$$A_i = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0P_2} \\ a_{10} & a_{11} & \cdots & a_{1P_2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{P_1 0} & a_{P_1 1} & \cdots & a_{P_1 P_2} \end{bmatrix} \quad (1.41)$$

$a_{k_1 k_2}$  est  $(k_1, k_2)^{\text{ème}}$  coefficient du filtre,  $k_1 = 0, 1, \dots, P_1$  et  $k_2 = 0, 1, \dots, P_2$ .

### 1.3.1. Méthodes de synthèse des filtres RIF 2D

*Méthode des fenêtres* : extension à 2D de la méthode utilisée dans le cas d'un filtre 1D.

*Méthode de l'échantillonnage en fréquence* : extension à 2D de la méthode utilisée dans le cas d'un filtre 1D.

*Méthodes itératives et d'optimisation* :

Dans ces méthodes, on utilise différentes normes du critère d'erreur.

$$E(\omega_1, \omega_2) = D(\omega_1, \omega_2) - H(\omega_1, \omega_2) \quad (1.42)$$

- Erreur norme  $L_2$  :

$$E_2 = \left[ \sum_{\omega_1} \sum_{\omega_2} |E(\omega_1, \omega_2)|^2 \right]^{1/2} \quad (1.43)$$

- Le critère des moindres carrés est donné par :

$$E_2^2 = \left[ \sum_{\omega_1} \sum_{\omega_2} |E(\omega_1, \omega_2)|^2 \right] \quad (1.44)$$

- Erreur norme  $L_p$  :

$$E_p = \left[ \sum_{\omega_1} \sum_{\omega_2} |E(\omega_1, \omega_2)|^p \right]^{1/p} \quad (1.45)$$

- Erreur norme  $L_\infty$  : (approximation au sens de Chebyshev)

$$E_\infty = \max |E(\omega_1, \omega_2)| \quad (1.46)$$

- Approximation minmax :

$$E = \min(\max |E(\omega_1, \omega_2)|) \quad (1.47)$$

- Algorithme de Parcks-McClellan :

Globalement, l'approximation (la synthèse) dans le domaine fréquentiel est réalisée en se basant sur deux mesures d'erreur différentes. La première mesure d'erreur est la norme minimisant le maximum de l'erreur (synthèse minmax) correspondant à une approximation uniforme dans tous les points fréquentiels. La deuxième mesure d'erreur est l'erreur utilisant la norme Euclidienne (ou l'erreur quadratique); les filtres conçus par cette mesure utilisent le procédé de conception par les moindres carrés.

### 1.3.2. Etat de l'art sur les méthodes classiques et d'optimisation pour la synthèse des filtres RIF 2D

La méthode des moindres carrés est l'une des méthodes les plus utilisées pour la synthèse des filtres RIF 2D. [Ahmad et al., 1989] ont utilisé la méthode des moindres carrés pour la synthèse des filtres RIF 2D symétriques et antisymétriques. Une étude approfondie des matrices impliquées dans le développement de la technique de conception révèle un certain nombre de propriétés utiles. Ils ont montré que ces propriétés conduisent à une solution analytique de bonne qualité qui rend inutile l'utilisation des méthodes d'optimisation, inversion de matrice ou itératives, selon leurs conclusions. Et en 1996, Pei a présenté deux travaux [Pei et al., 1996]<sup>1,2</sup> similaires à celui présenté dans [Ahmad et al., 1989] sur la synthèse des filtres RIF 2D. La méthode des moindres carrés a été généralisée et étendue pour la synthèse de 16 types de filtres RIF bidimensionnels à réponse fréquentielle symétrique ou antisymétrique. Etant données les spécifications de conception, les coefficients du filtre sont déterminés directement à partir d'une matrice de transformation fermée, ce qui fait que des procédures prenant du temps telles que l'optimisation, l'inversion de matrices, l'itération peuvent être évitées.

Le concept de paramètre de forme est présenté et appliqué par [Yeh et al., 1996] à la conception de filtres RIF bidimensionnels. En combinant ce concept avec la technique d'association de points, le filtre numérique RIF bidimensionnel dont le contour passe-bande correspond à la forme souhaitée est obtenu au moyen d'une transformation de McClellan. [Hsieh et al., 1997] ont introduit une fonction de pondération dans le critère d'erreur utilisé dans la méthode des moindres carrés, ce qui a permis d'améliorer les résultats obtenus par la méthode des moindres carrés.

[Lu et al., 1997] ont proposé une méthode pour déterminer les coefficients optimaux d'une transformation de McClellan sans mise à l'échelle et la détermination de la fréquence de coupure d'un filtre prototype 1D à l'aide de l'algorithme des moindres carrés récursifs. D'après les auteurs, cette méthode peut être utilisée pour concevoir de nombreux types de filtres 2D. D'ailleurs, [Pei et al., 1999] ont élaboré une méthode de recherche des coefficients de la transformation de McClellan pour la conception de filtres numériques RIF 2D. Cette méthode permet la conception de contours de transformation de forme arbitraire afin d'étendre des prototypes 1D à des filtres 2D de manière très efficace. Certaines conditions sont composées de telle sorte que le problème d'échelle de la transformation peut être évité. De plus, une nouvelle transformation de McClellan généralisée est présentée, celle-ci pouvant être appliquée à la conception de filtres 2D à coefficients complexes.

[Lee et al., 2000] ont réalisé la conception, par la méthode minimax, de filtres numériques RIF à phase linéaire bidimensionnels (2D) ayant des coefficients continus et de type puissance de deux (Powers-Of-Two : POT). Tout d'abord, le problème de conception minimax pour des coefficients continus est formulé comme un problème de programmation linéaire avec des contraintes d'inégalité. Leur projet présente une méthode basée sur une variante de l'algorithme de Karmarkar pour résoudre le problème de conception résultant. A chaque itération, le coût de calcul principal pour les coefficients du filtre est dû à la résolution d'un ensemble d'équations linéaires.

#### **1.4. Limitation des méthodes conventionnelles de synthèse des filtres RIF**

Il existe différentes méthodes conventionnelles pour trouver les coefficients du filtre à partir des spécifications de fréquence : la méthode de conception par les fenêtres, la méthode d'échantillonnage en fréquence, la conception par la méthode des moindres carrés pondérés, la conception equiripple ... L'inconvénient des filtres conçus par ces techniques est qu'ils contiennent de nombreuses petites ondulations dans la bande passante et la bande d'arrêt. Dans les méthodes des fenêtres et d'échantillonnage en fréquence utilisées pour concevoir les filtres RIF, le contrôle avec précision des fréquences de coupure entre la bande passante et atténuée reste difficile. La complexité de la conception est conforme aux limites de la conception classique qui ne peut gérer toutes les contraintes de performances du filtre. Alors que la science de la conception de filtres numériques est très bien établie et documentée, il n'existe pas de procédures de conception classiques qui conduisent à des plans optimaux pour la synthèse des filtres numériques.

De nouvelles approches classiques de conception ont été élaborées et ont obtenu des résultats intéressants par rapport aux approches mentionnées ci-dessus. Par exemple, le procédé de

Remez est basé sur un algorithme de programmation linéaire. Bien qu'intéressants, les résultats sont limités, car la difficulté devient alors de converger vers la solution minimax. Malheureusement, on sait que pour certaines fonctions, la méthode Remez peut conduire à un comportement divergent, même si la première approximation de départ est assez bonne. La méthode des moindres carrés pondérée fournit également quelques améliorations, mais elle souffre cependant d'un inconvénient majeur : la précision de l'approximation se détériore considérablement au voisinage des bords des bandes passantes et atténuées. De plus, il faut calculer une matrice inverse qui peut être problématique surtout lorsque l'ordre du filtre est très élevé.

En résumé, on peut dire que :

- Les conceptions par les méthodes conventionnelles ne sont pas optimales.
- Un filtre d'ordre supérieur est nécessaire pour satisfaire les spécifications requises.
- Un filtre d'ordre supérieur signifie plus de calculs par échantillon; ce qui implique que ces filtres sont plus lents et moins efficaces dans des applications en temps réel.

### 1.5. Conclusion

Au cours des dernières années, avec l'amélioration rapide de la technologie informatique, le traitement du signal numérique est devenu plus important. Par conséquent, le problème de conception de filtres numériques a reçu beaucoup d'attention. Ces filtres RIF ont de nombreuses applications importantes, par exemple, dans les systèmes radar, sonar, de traitement du signal et du traitement d'images. La conception de filtre est le procédé comprenant la synthèse et la mise en œuvre de telle sorte que le filtre résultant satisfait aux contraintes données (réponse en amplitude, en phase, ...). D'un point de vue mathématique, le problème de conception de filtre peut être considéré comme un problème de minimisation sous contrainte consistant à trouver une réalisation du filtre qui satisfait à chacune des exigences à un degré suffisant pour le rendre utile.

Récemment, des méthodes évolutionnaires d'optimisation ont été mises en œuvre pour la conception optimale des filtres numériques avec un meilleur contrôle et une meilleure optimisation des paramètres. Les heuristiques et les méthodes d'optimisation stochastiques ont montré leur capacité à approcher les solutions optimales au problème de synthèse de filtres RIF, comme l'algorithme génétique, le recuit simulé, la recherche tabou, l'optimisation par essaim de particules ...

Le chapitre suivant présente les principes de base et théoriques des heuristiques évolutionnaires, et les travaux menés, dans le contexte de problématiques similaires : développement de techniques évolutionnaires pour la synthèse des filtres RIF 1D et 2D.



---

# Chapitre II

---

## 2. Algorithmes évolutionnaires

### Sommaire

---

2. Algorithmes évolutionnaires .....	29
2.1. Introduction .....	30
2.2. Recherche tabou (Tabu Search) .....	30
2.2.1. Concepts de base de la recherche tabou .....	31
2.3. Recuit simulé (Simulated Annealing) .....	31
2.4. Algorithme d'ascension de collines (Hill Climbing) .....	32
2.5. Optimisation par essaim particulaire (de particules) (Particle swarm optimization : PSO) .....	33
2.6. Algorithme génétique .....	36
2.6.1. Chromosome (individu).....	37
2.6.2. Evaluation.....	37
2.6.3. Sélection.....	37
2.6.4. Croisement (Crossover).....	37
2.6.5. Mutation .....	38
2.6.6. Amélioration de l'AG standard.....	39
2.7. Hybridation .....	40
2.8. Etat de l'art sur les méthodes évolutionnaires appliquées pour la synthèse des filtres RIF 1D .....	40
2.9. Etat de l'art sur les méthodes évolutionnaires appliquées à la synthèse des filtres RIF 2D .....	42
2.10. Conclusion .....	43

---

## 2.1. Introduction

La conception de filtres à réponse impulsionnelle finie (RIF) peut être formulée comme un problème d'optimisation non linéaire réputée pour être difficile pour les approches conventionnelles. Les contraintes sont élevées et un grand nombre de paramètres doivent être estimés, surtout lorsqu'il s'agit de filtres RIF 2D.

Le chercheur ou l'ingénieur qui veut réaliser la synthèse des filtres RIF doit certainement s'appuyer sur des approches intelligentes et plus évoluées pour espérer optimiser les coefficients des filtres RIF. Mais la difficulté à laquelle sera confronté le chercheur est le choix d'une technique efficace, capable de trouver une solution optimale en un temps de calcul raisonnable. En plus, le réglage optimal des paramètres d'une approche, tel que préconisé par la théorie, peut s'avérer inefficace en pratique car il induit un coût de calcul élevé sans assurer une convergence aux solutions optimales. A cet effet, le choix de la bonne méthode et le réglage optimal des paramètres de celle-ci, doit faire appel au savoir-faire et à l'expérience du chercheur, plutôt qu'à l'application à la lettre des règles établies par la théorie de cette méthode.

Plusieurs articles traitent les limites des approches classiques et mettent en œuvre également plusieurs outils heuristiques modernes qui ont évolué au cours des deux dernières décennies et visant à résoudre le problème d'optimisation des coefficients des filtres RIF 1D et 2D, tels que le calcul évolutif, le recuit simulé (SA : Simulated Annealing), la recherche tabou (TS : Tabu Search), La méthode d'ascension de collines (HC : Hill Climbing), l'optimisation par essaim de particules (PSO : Particle Swarm Optimization), etc. Ces outils facilitent la résolution des problèmes d'optimisation qui étaient auparavant difficiles ou impossibles à résoudre. La majorité des métaheuristiques utilisent des processus aléatoires comme moyen d'exploitation pour récolter de l'information et peuvent être alors considérées comme des algorithmes stochastiques itératifs. Les algorithmes génétiques (AG's) sont probablement les plus populaires et les plus développés, et sont l'une des métaheuristiques ayant le plus grand nombre de versions. Pour rappel, ils ont été proposés par Holland [Holland, 1973] comme des algorithmes de recherche probabiliste, d'apprentissage et d'optimisation, et sont basés en partie sur le mécanisme de l'évolution biologique et la théorie de l'évolution de Darwin.

## 2.2. Recherche tabou (Tabu Search)

La recherche tabou (Tabu Search : TS) est une méthode d'optimisation mathématique, appartenant à la classe des techniques de recherche locale [Glover, 1989], [Glover, 1990], [Tai-Hsi Wu, 2009]. La recherche tabou améliore les performances d'une méthode de recherche locale en utilisant des structures adaptées de mémoire : une fois une solution potentielle a été déterminée, elle est marquée comme taboue afin que l'algorithme ne visite pas cette possibilité à plusieurs reprises. Pour explorer les régions de l'espace de recherche qui seraient laissées inexplorées par la procédure de recherche locale, la recherche tabou modifie la structure du voisinage de chaque solution au fur et à mesure que la recherche progresse. Les solutions admises sont déterminées par l'utilisation de structure de mémoire. Dans sa forme la plus simple, une liste taboue est une mémoire à court terme qui contient les solutions qui ont été visitées dans le passé récent. C'est le type le plus important de la structure de la mémoire utilisée pour déterminer les solutions admises pour la liste taboue.

### 2.2.1. Concepts de base de la recherche tabou

- Liste taboue : utilisation d'une structure de mémoires flexibles pour mémoriser les configurations ou régions visitées et inclure des mécanismes permettant d'interdire à l'algorithme certains mouvements pour ne pas retourner trop rapidement (temporairement) vers les régions visitées.
- Critère d'aspiration (assouplissement du mécanisme de la liste taboue) : autorisation et acceptation de certains mouvements tabous.

L'alternance entre ces deux premiers concepts dans le processus est réalisée à l'aide d'un mécanisme de contrôle.

- Stratégie d'intensification : exploitation plus approfondie des régions prometteuses trouvées (visitées) récemment et renforcement de la recherche dans ces régions.
- Stratégie de diversification : exploration des régions qui n'ont pas encore été visitées.

Tableau	Algorithme de base de la recherche tabou
Etape 1	Initialisation : générer une configuration initiale $x_0$ dans $S$ . Poser $x = x_0$ et $f = f(x_0)$ .
Etape 2	$k = 0$ , Liste taboue $T = \emptyset$ .
Etape 3	Répéter <ul style="list-style-type: none"> <li>- Incrémenter <math>k = k + 1</math> et générer un sous-ensemble <math>V</math> de solutions de telle sorte que la condition taboue ne soit pas violée</li> <li>- Choisir (en tenant compte des restrictions tabou et du critère d'aspiration) <math>x</math> le meilleur élément dans <math>V</math> (<math>best(V(x))</math>, par rapport à la fonction <math>f</math>)</li> <li>- <math>x_{k+1} = best(V(x))</math></li> <li>- Si <math>(f(x_{k+1}) &lt; f(x))</math> alors <math>x = x_{k+1}</math></li> <li>- Mettre à jour la liste taboue <math>T</math></li> <li>- Stop, si la condition d'arrêt est atteinte (éventuellement appliquer les stratégies d'intensification et de diversification). Sinon, aller à l'étape 3.</li> </ul>
Etape 4	Retourner $x$ , la meilleure solution trouvée

Table 2.1 – Algorithme de la recherche tabou (TS).

### 2.3. Recuit simulé (Simulated Annealing)

Dans le recuit simulé (Simulated Annealing : SA), un système est initialisé à une température  $T$  avec une configuration dont l'énergie est évaluée comme  $E$  [Kirkpatrick, 1983], [Bohachevsky, 1986], [Dimitris Bertsimas, 1993]. Une nouvelle configuration est réalisée en appliquant une variation aléatoire, et la variation de l'énergie  $dE$  est calculée. La nouvelle configuration est inconditionnellement acceptée si elle diminue l'énergie du système. Si l'énergie du système est augmentée par le changement, la nouvelle configuration est acceptée avec une certaine probabilité aléatoire. Dans le schéma original de Metropolis, la

probabilité est donnée par le facteur de Boltzmann  $\exp(-dE/kT)$ . Ce processus est répété plusieurs fois à la température actuelle afin de parcourir efficacement l'espace de recherche, puis la température est abaissée. Le procédé est répété à des températures successivement plus basses jusqu'à ce que l'état congelé soit atteint. Cette procédure permet au système de se déplacer à la baisse des états d'énergie, tout en sautant hors des minima locaux (en particulier à des températures plus élevées) en raison de l'acceptation probabiliste de quelques mouvements à la hausse.

Tableau	Algorithme de base du recuit simulé
Etape 1	<i>Définir une température initiale <math>T</math></i>
Etape 2	<i>Générer la solution initiale aléatoirement <math>x = x_0</math></i>
Etape 3	<i>Appliquer une Perturbation (aléatoire) à la solution actuelle pour obtenir une nouvelle solution</i>
Etape 4	<i><math>\Delta E</math> est la différence de la fonction fitness entre la nouvelle et l'actuelle solution</i>
Etape 5	<i>Si <math>\Delta E &lt; 0</math> (la nouvelle fitness est inférieure), accepter la nouvelle solution comme solution actuelle</i>
Etape 6	<i>Si <math>\Delta E \geq 0</math>, accepter la nouvelle solution avec une probabilité <math>Pr(\text{acceptation}) = \exp(-\Delta E / k_B \cdot T)</math></i>
Etape 7	<i>Lorsque l'équilibre est atteint, la température <math>T</math> peut être abaissée et le processus peut être répété</i>

Table 2.2 – Algorithme du recuit simulé (SA).

#### 2.4. Algorithme d'ascension de collines (Hill Climbing)

Hill Climbing (HC) tire son nom du fait qu'elle suppose que l'espace de fonction est "en forme de colline" [S.A. Dunn, 1998], [S.A. Akramifar, 2010], [Alan W. Johnson, 2002]. Si vous commencez au sommet de la colline (l'optimum local), et si vous vous déplacez dans n'importe quelle direction, vous vous déplacez vers le bas. Partout ailleurs, vous pouvez choisir n'importe quel axe et aussi longtemps que vous vous déplacez, vous vous rapprochez de l'optimum.

Hill Climbing ne parvient pas à trouver l'optimum lorsque l'espace de fonction contient des "maxima locaux". Un maximum (optimum) local est une petite colline laissée sur la surface dont le sommet n'est pas aussi élevé que le pic (optimum) principal. Si vous trouvez un maximum local de Hill Climbing, vous êtes coincé là-bas. Tout petit déplacement dans n'importe quelle direction rend les choses (temporairement) pires - même pour se déplacer vers l'optimum global.

Tableau	Algorithme de base de la Hill Climbing
Etape 1	Choisir une solution initiale $x_0$ dans $S$ . Poser $x = x_0$ et $f = f(x_0)$
Etape 2	$k = 0$
Etape 3	Répéter; <ul style="list-style-type: none"> <li>- <math>k = k + 1</math></li> <li>- Générer un sous-ensemble <math>V</math> de solutions <math>V(x)</math> <ul style="list-style-type: none"> <li>- Choisir <math>x</math> le meilleur élément dans <math>V</math> (<math>best V(x)</math>, par rapport à la fonction <math>f</math>)</li> <li>- <math>x_{k+1} = best(V(x))</math></li> <li>- Si <math>(f(x_{k+1}) &lt; f(x))</math> alors <math>x = x_{k+1}</math></li> <li>- Sinon <math>x = x</math></li> <li>- Stop, si la condition d'arrêt est satisfaite. Sinon, aller à l'étape 3</li> </ul> </li> </ul>
Etape 4	Retourner $x$ , la meilleure solution trouvée

Table 2.3 – Algorithme de la Hill Climbing (HC).

## 2.5. Optimisation par essaim particulaire (de particules) (Particle swarm optimization : PSO)

La PSO est une méthode de la grande famille de méthodes d'intelligence Swarm [Lingling Zhao, 2007] et est basée sur la reproduction d'un comportement social et la collaboration entre des individus souvent appelées particules. Ce comportement social s'appuie sur l'analyse de l'environnement et du voisinage et constitue alors une méthode de recherche d'optimum par l'observation des tendances des particules voisines. Chaque particule cherche à optimiser ses chances en suivant la tendance de son voisinage qu'il modère par ses propres vécus. L'algorithme est généralement initialisée de façon aléatoire et les particules (solutions candidates) sont placées au hasard dans l'espace de recherche de la fonction objectif, la PSO conduit avec succès à un optimum global. Le concept principal de la PSO est que les solutions potentielles sont accélérées vers les meilleures solutions. Les particules évaluent de manière itérative l'aptitude des solutions candidates et se souviennent de l'endroit où elles ont eu leur meilleure valeur de fitness. Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules de son entourage. A chaque itération, les particules se déplacent en tenant compte de leur meilleure position, mais aussi de la meilleure position de ses voisins. L'objectif est de modifier leur trajectoire pour qu'elles se rapprochent le plus possible de l'optimum. Cet optimum est obtenu par une procédure itérative sur la base des processus de mouvement et de l'intelligence dans un système évolutif. La PSO met l'accent sur la coopération plutôt que sur la compétition et il n'y a pas de sélection (au moins dans la version de base), l'idée étant qu'une particule encore aujourd'hui pauvre mérite d'être préservée, peut-être justement parce que c'est elle qui saura trouver la meilleure solution, précisément parce qu'elle agit hors de l'espace prometteur a priori. Selon certains résultats, cette technique ne souffre pas de problèmes rencontrés par d'autres méthodes évolutionnaires [R. Eberhart, 2000].

Un essaim de particules est défini par :

- Le nombre de particules constituant l'essaim
- La vitesse maximale d'une particule
- L'inertie d'une particule
- Les coefficients de pondération

A l'itération  $k$  (à l'instant  $k$ ), une particule  $i$  de l'essaim est caractérisée par :

Sa position dans l'espace de recherche :  $X_i^k$

Sa vitesse :  $V_i^k$

La position de la meilleure solution par laquelle elle est passée :  $pbest_i^k$

La position de la meilleure solution connue de son voisinage :  $gbest^k$

La fitness de sa meilleure solution :  $fitpbest_i^k$

La fitness de la meilleure solution connue de son voisinage :  $fitgbest^k$

Les vecteurs vitesse et position de la particule  $i$  sont modifiés comme suit :

$$V_i^{k+1} = \gamma_i * V_i^k + C_1 * rand_1 * (pbest_i^k - X_i^k) + C_2 * rand_2 * (gbest^k - X_i^k) \quad (2.1)$$

$$X_i^{k+1} = X_i^k + V_i^k \quad (2.2)$$

où  $X_i^k$  et  $V_i^k$  sont respectivement la position et la vitesse (inertie) de la  $i^{\text{ème}}$  particule à la  $k^{\text{ème}}$  itération,  $pbest_i^k$  est la meilleure position trouvée par la particule et  $gbest^k$  est la meilleure position trouvée par toutes les particules.

$\gamma_i$  est une fonction de pondération,  $C_1$  et  $C_2$  sont des facteurs positifs de pondérations (facteurs de poids positifs),  $rand_1$  et  $rand_2$  sont des nombres aléatoires entre 0 et 1.

[Clerc et al., 2003] a résumé l'évolution d'une particule comme étant une combinaison de trois types de comportements : égoïste (suivre sa voie en fonction de sa vitesse actuelle), conservateur (revenir en arrière en prenant en compte sa meilleure performance) et panurgien (suivre aveuglement la meilleure particule en considérant sa performance). La PSO exploite bien l'intelligence collective : coordination du groupe, instinct individuel et interaction locale entre les particules (compromis entre confiance en soi et influence des relations sociales).

La procédure de base d'implémentation de l'algorithme PSO est donnée dans ce tableau :

Tableau	Algorithme de la PSO (étapes de la PSO)
Etape 1	Initialiser chaque vecteur de particule de l'essaim en affectant une vitesse et une position aléatoires dans l'espace de recherche.
Etape 2	Evaluer les valeurs de la fonction fitness de chaque particule et comparer la fitness de chaque particule avec celle de sa meilleure valeur personnelle ( $fitpbest_i^{k-1}$ ). Si la valeur actuelle est meilleure que la valeur ( $fitpbest_i^{k-1}$ ), mettre à jour la valeur de ( $fitpbest_i^k$ ) et sa meilleure position ( $pbest_i^k$ ).
Etape 3	Identifier la particule qui a la meilleure valeur de la fonction fitness. La valeur de sa fonction fitness est identifiée comme ( $fitgbest^k$ ) et sa position comme ( $gbest^k$ ).
Etape 4	Mettre à jour les vitesses ( $V_i^{k+1}$ ) en utilisant l'équation 2.1 et les positions ( $X_i^{k+1}$ ) en utilisant l'équation 2.2 pour toutes les particules.
Etape 5	Remplacer les vecteurs de particules mis à jour comme vecteurs de particules initiaux pour l'étape 2.
Etape 6	Répéter les étapes 2-5 jusqu'à satisfaire le critère d'arrêt (nombre maximal d'itérations ou convergence vers la bonne valeur de la fitness).

Table 2.4 – Algorithme de l'optimisation par essaim de particules (PSO).

En général, pour les méthodes stochastiques, la convergence vers la solution optimale n'est pas garantie dans tous les cas même si les expériences ont montré la grande performance de l'approche d'optimisation par essaim de particules. A cet effet, il est nécessaire de définir un critère d'arrêt. En pratique, on préfère définir trois critères d'arrêt :

- Le nombre d'itérations maximales est atteint
- La variation de la vitesse est presque nulle
- Le seuil de la fitness maximale est atteint

Arrêter l'exécution de l'algorithme si l'un de ces trois critères d'arrêt est satisfait.

L'optimisation par essaim de particules présente un fort potentiel d'application pratique, mais le choix des paramètres reste l'un des problèmes de l'optimisation par particules d'essaim et de toutes les métaheuristiques, c'est très difficile de trouver les bons paramètres adaptés à la structure du problème.

Les résultats obtenus par la PSO sont très satisfaisants et confirment bien les performances de la méthode; sa simplicité d'implémentation lui donne un avantage conséquent. Néanmoins par l'expérimentation, on a remarqué que la qualité des solutions ne peut être garantie même en augmentant le nombre d'itérations. En résumé, la convergence de la PSO vers l'optimum global n'est pas systématique.

## 2.6. Algorithme génétique

L'algorithme génétique (AG) [Holland, 1973], [Holland, 1975], [Goldberg, 1989] a été proposé par Holland comme un algorithme de recherche probabiliste, d'apprentissage et d'optimisation, et est basé en partie sur les mécanismes de l'évolution biologique et la théorie de l'évolution de Darwin. Cet algorithme est un puissant outil de recherche, en particulier lorsqu'il est appliqué à des problèmes d'optimisation combinatoire. Toutefois, la mise en œuvre d'un AG efficace est souvent confronté à deux problèmes majeurs, d'un côté, la convergence prématurée vers les optima locaux et de l'autre côté, les exigences relatives à la recherche de longues périodes afin de parvenir à un optimum ou à une bonne solution sous-optimale.

D'autre part, l'AG fait évoluer sur plusieurs générations, une grande population en appliquant trois opérateurs, qui se composent de sélection, de croisement et de mutation. Ceci est essentiel pour produire plus de descendants qui tendent à homogénéiser la population et à améliorer le résultat moyen au fur et à mesure que l'algorithme progresse (évolue). L'algorithme commence avec une population initiale représentant des solutions candidates aléatoires. A chaque individu de la population, est attribué un score basé sur ses performances. Les individus ayant les meilleurs scores sont les plus susceptibles d'être sélectionnés pour produire une nouvelle génération.

Les chromosomes sélectionnés sont utilisées pour produire une nouvelle population sur la base de deux principaux opérateurs génétiques, croisement et mutation. Dans le croisement, deux individus sont utilisés pour produire deux nouveaux individus par l'échange de gènes entre les deux individus sélectionnés. La mutation aléatoire est également appliquée pour ajouter un peu de diversité à la population. Les descendants ayant les meilleures performances sont susceptibles d'être les parents de la prochaine génération. Le processus est répété jusqu'à atteindre un critère d'arrêt. Il est intéressant de souligner que dans la mise en œuvre d'un AG, les paramètres à optimiser (gènes) doivent être codés dans une chaîne de sorte que le croisement et la mutation peuvent être appliqués. Différents types de codage peuvent être utilisés, y compris avec des nombres réels. Le choix des individus est effectué à l'étape de sélection et peut être réalisé en utilisant diverses procédures. Deux procédures couramment utilisées sont la roulette et la sélection par tournoi. Dans la roulette, chaque individu se voit attribuer une taille de segment proportionnelle à son aptitude (fitness). La roue est ensuite lancée (tournée) et l'individu en face du marqueur devient l'un des parents. Dans la sélection par tournoi, plusieurs personnes sont choisies au hasard et le plus fort devient l'un des parents.

Chaque cycle de l'AG est gouverné par le processus d'évolution et à chaque itération, appelée génération, est créée une nouvelle population avec le même nombre d'individus à partir de la population précédente (parents). Cette nouvelle génération consiste généralement en des individus mieux adaptés à l'environnement tel qu'il est représenté par la fonction d'adéquation (fitness). Au fur et à mesure des générations, les individus vont tendre, en général, vers l'optimum de la fonction d'adéquation.

L'algorithme de l'AG peut être résumé comme suit :

Tableau	Algorithme génétique
Etape 1	Initialisation: Définir la taille de la population TP, le taux de mutation et de croisement, le type de sélection, choisir le critère d'arrêt et le codage à utiliser, générer aléatoirement la population initiale dans l'espace de recherche.
Etape 2	Evaluer la fonction fitness de chaque chromosome.
Etape 3	Appliquer l'opérateur de croisement.
Etape 4	Appliquer l'opérateur de mutation.
Etape 5	Sélectionner les chromosomes.
Etape 6	Mettre à jour la nouvelle population.
Etape 7	Répéter les étapes 2-6 jusqu'à atteindre le critère d'erreur et retourner la meilleure solution.

Table 2.5 – Algorithme génétique (AG).

Les AG's codés-réels, offrent généralement l'avantage d'être mieux adaptés aux problèmes d'optimisation numérique continus, d'accélérer la recherche et sont mieux adaptés au développement de techniques hybrides avec les méthodes évolutionnaires. Cependant, ces AG's nécessitent le développement d'opérateurs spécifiques au codage réel et à l'application.

### 2.6.1. Chromosome (individu)

Les chaînes des systèmes génétiques artificiels sont analogues aux chromosomes des systèmes biologiques. Ils portent les informations génétiques d'un individu. Ainsi, un individu se compose d'un ensemble d'éléments appelés caractéristiques ou gènes.

### 2.6.2. Evaluation

L'application d'un AG nécessite, en premier lieu, le calcul d'une fonction d'évaluation qui doit être minimisée ; ou d'une fonction fitness qui doit être maximisée. La fonction d'évaluation permet d'évaluer les individus et d'effectuer le processus de sélection.

### 2.6.3. Sélection

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais.

### 2.6.4. Croisement (Crossover)

Les croisements sont envisagés avec deux parents et génèrent deux enfants. Une fois la génération intermédiaire à moitié remplie, les individus sont aléatoirement répartis en

couples hermaphrodites. Les chromosomes des parents sont alors copiés et recombines de façon à former deux descendants possédant des caractéristiques issues des deux parents. L'opérateur de croisement assure donc le brassage du matériel génétique. Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes et il échange certains gènes d'un parent avec ceux de l'autre.

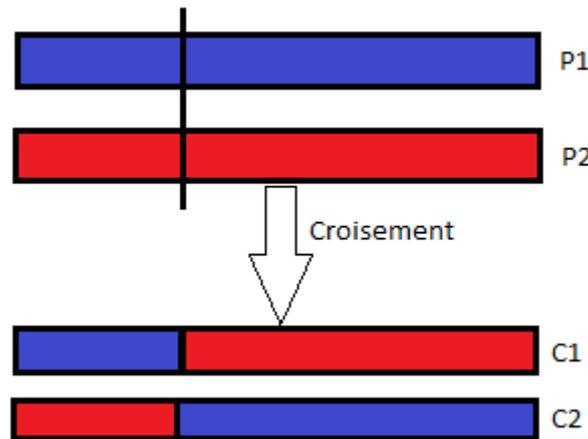


Figure 2.1 – Croisement en 1 point

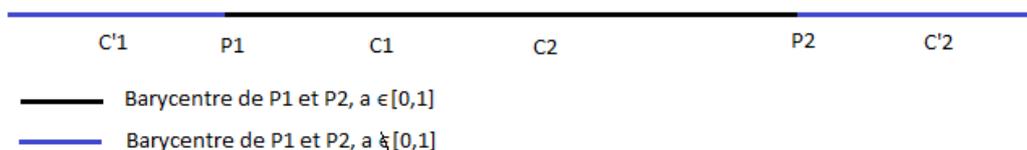


Figure 2.2 – Croisement directionnel (direction-based crossover).

### 2.6.5. Mutation

Les mutations sont des modifications aléatoires du génome. Bien sûr, il ne faut pas muter tous les gènes d'un individu, sinon il serait complètement déterminé aléatoirement. Il faut au contraire en modifier une petite partie, afin d'apporter quelque chose de nouveau à l'individu.

Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer et permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre de gènes mutés (taux de mutation). Les propriétés de convergence des algorithmes génétiques sont fortement dépendantes de cet opérateur sur le plan théorique. L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace de recherche. Cet opérateur nous garantit que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de la résolution du problème.

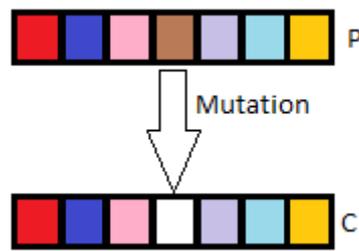


Figure 2.3 – Mutation.

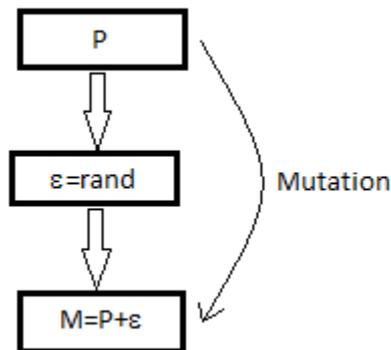


Figure 2.4 – Mutation arithmétique.

#### 2.6.6. Amélioration de l'AG standard

1. Les mécanismes de sélection standard (proportionnelle à la fonction fitness) ont des avantages mais aussi des inconvénients ; en particulier si un individu est très supérieur à la moyenne, il constituera presque exclusivement la population dans les générations suivantes et l'algorithme perdra toute diversité (convergence prématurée). Une solution possible est de procéder à un changement d'échelle de la fonction fitness (changement d'échelle statique ou dynamique, linéaire ou exponentiel). Une autre solution possible consistera à veiller à la diversification de la population en introduisant des mécanismes de rafraîchissement. Cette diversification vise à accroître la quantité des solutions potentielles en explorant de nouvelles régions de l'espace de recherche.
2. Dans certains problèmes, on veut garder intact le meilleur individu de la population lors du passage d'une génération à la suivante ; la stratégie d'élitisme concrétise cette idée en replaçant le meilleur individu dans la population suivante. Cet individu reste toutefois candidat pour la phase de reproduction (croisement et mutation) habituelle.
3. L'hybridation avec d'autres méthodes évolutionnaires ou classiques afin de tirer profit (exploiter, profiter) des avantages des deux méthodes.
4. Certaines applications ont pour but de trouver non pas l'optimum global, mais l'ensemble des optima. A cet effet, l'introduction d'un mécanisme de partage est nécessaire qui est connu sous le nom de niche.

Les métaheuristiques apportent plusieurs avantages et améliorations aux problèmes d'optimisation mais partagent deux inconvénients majeurs : les difficultés de réglage des paramètres propres à chaque méthodes et leur temps de calcul élevé. De plus, une analyse systématique des métaheuristiques du point de vue de la convergence, la complexité, la robustesse et la répétabilité des performances peut révéler plusieurs failles et inconvénients

de ces métaheuristiques. L'hybridation par combinaison ou coopération entre ces métaheuristiques pour tirer profit des avantages spécifiques d'approches différentes peut donner lieu à des approches plus efficaces et plus optimales.

## 2.7. Hybridation

Les AG's sont performants dans la recherche des régions prometteuses de l'espace de recherche, mais sont inefficaces dans la détermination du minimum local en termes de vitesse de convergence et de qualité de la solution. Le champ d'application des AG's a clairement progressé, principalement en raison de l'innovation au cours des dix dernières années, et a intégré des stratégies plus élaborées visant à remédier aux faiblesses des AG's. La solution est de proposer une approche génétique hybride basée sur les meilleurs mécanismes génétiques et d'améliorer la mise en œuvre des AG's.

La combinaison d'un algorithme génétique et d'une recherche locale permet d'accélérer la recherche pour localiser l'optimum global. Le lien entre la partie purement génétique et l'approche locale, et la profondeur de la recherche de l'approche locale sont, cependant, encore des questions qui ne peuvent pas être généralisées.

## 2.8. Etat de l'art sur les méthodes évolutionnaires appliquées pour la synthèse des filtres RIF 1D

[Lu et al., 2000] sont parmi les premiers auteurs à proposer un algorithme génétique pour la synthèse des filtres RIF. Cet algorithme génétique permet d'obtenir une approximation aux sens des moindres carrés du logarithme de la réponse en amplitude pondérée. À chaque itération, la fonction de pondération est mise à jour en utilisant le résultat de l'itération précédente, de telle façon que l'erreur pondérée se rapproche de l'erreur logarithmique réelle ou complexe. La même année, ils ont étendu leur algorithme aux cas complexes pour la conception de filtres numériques à réponse impulsionnelle finie complexe.

D'autres algorithmes évolutionnaires ont été élaborés pour la synthèse des filtres RIF, tels que les réseaux de neurones [Zeng et al., 2006], [Wang et al., 2008] et l'algorithme d'évolution différentielle (Differential Evolution) [Karaboğa et al., 2006]. Ces derniers auteurs ont réalisé une étude comparative entre leur méthode et l'algorithme génétique. En 2007, [Wang et al., 2007] a mis en œuvre quatre réseaux de neurones pour la synthèse de quatre types de filtres RIF. Les solutions ont été présentées comme des algorithmes parallèles pour approcher la réponse fréquentielle souhaitée et évitent ainsi l'inversion de matrice, et permettent de faire un calcul rapide des coefficients du filtre. [Wang et al., 2008] a proposé une technique d'optimisation en deux étapes avec diverses réponses en fréquence de masquage. Un filtre initial RFM est généré à la première étape en optimisant alternativement les sous-filtres à l'aide d'un algorithme standard simple. Lors de la deuxième étape, cette conception initiale est encore améliorée par l'optimisation des coefficients de sous-filtres globaux en utilisant simultanément un réseau neuronal. Comme les coefficients de l'ensemble des sous-filtres sont optimisés simultanément, par conséquent, l'optimisation conjointe mène à améliorer les performances de conception. Ce réseau de neurones de rétro-propagation utilise un taux d'apprentissage variable.

En 2007, [Cen, 2007] a proposé un algorithme génétique hybride pour résoudre le problème de convergence prématurée pour le cas de la synthèse des filtres RIF. Le système hybride est formé en intégrant les principales caractéristiques d'un algorithme génétique adaptatif (AGA), le recuit simulé (SA) et la recherche tabou (TS). L'utilisation du recuit simulé aide l'AGA à échapper aux optima locaux et empêcher la convergence prématurée. La notion de tabou est introduite pour augmenter la vitesse de convergence en réduisant l'espace de recherche en fonction des propriétés des coefficients du filtre. Il a montré par des exemples que les ondulations peuvent être largement réduites par cet algorithme hybride. Un autre algorithme hybride a été développé par [Hime et al., 2009] en utilisant une optimisation stochastique basée sur un recuit simulé combiné avec la logique floue pour la synthèse des filtres RIF. Ceci a été rendu possible par la grande applicabilité et l'efficacité de la mise en œuvre du recuit simulé adaptatif flou, qui exigeait quelques conditions théoriques pour assurer sa probabilité de convergence vers le minimum global.

Dès 2008, la PSO a reçu un intérêt particulier de la part de la communauté du traitement du signal en général et des concepteurs de filtres RIF 1D en particulier. [Ababneh et al., 2008] ont proposé un algorithme d'optimisation par essaim de particules (PSO) pour la synthèse des filtres RIF 1D. Une comparaison avec l'AG a été faite. Pour le problème traité, ils ont constaté que la PSO surpasse l'AG dans certains cas de conception traités. Puis, en 2011, [Kar et al., 2011] a introduit la notion des facteurs de constriction dans la PSO pour réaliser la synthèse des filtres RIF 1D passe-bas. Les résultats des simulations ont été comparés avec l'algorithme génétique et démontrent que l'approche proposée surpasse l'AG, non seulement dans la précision du filtre conçu, mais aussi dans la vitesse de convergence. En 2012, [Kar et al., 2012] a appliqué les résultats de la conception réalisée par la PSO pour sur-échantillonner un signal audio avec le filtre RIF passe-bas en forme personnalisée.

Récemment, des versions améliorées de la méthode PSO (IPSO, I: improved) ont été développées pour la synthèse des filtres RIF. Ces versions proposent une nouvelle définition du vecteur de vitesse et de mise à jour des essaims. [Mondal et al., 2011] ont élaboré deux variantes de la PSO (PSO standard et PSO améliorée "IPSO") et un AG pour la synthèse des filtres RIF 1D passe-haut. Une comparaison avec la méthode de Parks-McClellan, l'AG et la PSO conventionnelle a été réalisée. Les résultats des simulations justifient que l'algorithme IPSO proposé surpasse la méthode de Parks-McClellan, AG et PSO classique dans la précision de la réponse en amplitude du filtre ainsi que dans la vitesse de convergence. En 2012, [Mondal et al., 2012] ont appliqué l'IPSO pour la synthèse des filtres RIF 1D passe-bas. Dans cette version, la principale caractéristique du mécanisme de vitesse proposé est de surveiller et d'optimiser le poids des particules, qui a tendance à diminuer de façon linéaire dans les applications générales.

Parmi les algorithmes les plus performants basés sur la PSO, on trouve celui élaboré par [Kar et al., 2012] pour la synthèse des filtres RIF. Cet algorithme est appelé optimisation par essaim de particules en folie (Craziness PSO). Un opérateur de folie est introduit dans cette technique afin de s'assurer que les particules aient une certaine probabilité de folie (ne pas suivre les règles imposées) pour maintenir la diversité des particules. Cela lui a permis d'éviter les solutions sous-optimales et d'améliorer considérablement le temps de calcul et de diminuer le nombre d'itérations.

## 2.9. Etat de l'art sur les méthodes évolutionnaires appliquées à la synthèse des filtres RIF 2D

La majorité des travaux de recherche effectués dans le domaine de la synthèse des filtres RIF souligne la nécessité d'utiliser les techniques intelligentes et évolutionnaires. L'objectif est de tirer profit efficacement de leur aspect itératif et stochastique pour optimiser les coefficients des filtres et surtout rendre ces algorithmes plus conviviaux, facilement exploitables par les utilisateurs.

[Franzen et al., 1998] ont élaboré la synthèse des filtres RIF 2D par des stratégies évolutives et ils ont comparé les résultats obtenus par leur stratégie avec ceux obtenus en utilisant le recuit simulé et les stratégies de type gradient. Dans toutes ces méthodes, le même critère d'erreur a été utilisé, à savoir, l'erreur quadratique pondérée.

Les réseaux de neurones ont été appliqués pour la synthèse des filtres RIF 2D. Parmi les auteurs qui ont développé ces techniques, on peut citer [Zhao et al., 1997] et [Bhattacharya et al., 1999]. Ces auteurs utilisent un réseau de neurones dynamique de type Hopfield ayant une fonction de l'énergie de Lyapunov qui coïncide avec le critère d'erreur (erreur quadratique moyenne). Dans le cas de [Bhattacharya et al., 1999], le réseau a été simulé sur Hspice afin de permettre son utilisation en temps réel.

D'autres techniques, tels que les algorithmes génétiques, ont reçues un intérêt considérable de la part des concepteurs des filtres RIF 2D. Parmi les auteurs, on peut citer les travaux de [Lu et al., 2000], [Tzeng, 2004] et [Tzeng, 2007]. Dans tous ces articles, le critère d'erreur quadratique moyenne a été utilisé. Dans [Tzeng, 2007], l'algorithme est élaboré, en premier temps, pour la synthèse des filtres RIF 1D puis il a été étendu pour le cas 2D. Récemment, [Pham et al., 2010] a appliqué l'algorithme des abeilles (Bees Algorithm) pour la synthèse des filtres RIF 2D. Dans ces conclusions, il dit que cet algorithme apporte une nette amélioration des résultats par rapport à l'algorithme génétique.

Le problème de calcul des coefficients (synthèse) des filtres RIF 1D et 2D ne peut être résolu d'une manière optimale par les méthodes déterministes (conventionnelles), et ceci malgré la disponibilité de calculateurs (ordinateurs, DSP) de plus en plus performants. Afin de faire face à cette situation, il est nécessaire de développer et d'élaborer des méthodes, dites métaheuristiques, inspirées du comportement sociale (éthologie), de la génétique (de la biologie de l'évolution) et de l'intelligence artificielle pour espérer approcher les solutions optimales.

Des approches, telles que les réseaux de neurones, les algorithmes génétiques et d'autres outils liés à l'intelligence informatique et artificielle, ont été consacrés à la synthèse des filtres. Ces approches sont capables de satisfaire les contraintes qui seraient inaccessibles si elles étaient traitées avec les méthodes classiques de conception. Dans certains cas, ces initiatives ont réussi à se rapprocher de leur but et ont montré de meilleurs indices de performances que les approches conventionnelles.

Cependant, il y a quelques points faibles associés aux méthodes évolutionnaires, comme l'augmentation du coût de calcul et la non-existence de la preuve théorique de convergence

vers un optimum global dans des conditions suffisamment générales. Bien que ces inconvénients ne soient pas mis au premier plan dans les travaux publiés, l'utilisation d'un algorithme évolutionnaire pour un problème dédié reste difficile pour les praticiens qui sont souvent sceptiques concernant les algorithmes évolutionnaires en général et l'algorithme génétique en particulier. Il n'y a pas de processus systématiques pour trouver un équilibre efficace entre les capacités d'exploration et d'exploitation des capacités durant la phase de recherche des solutions. Plus important encore, le coût de paramétrage simultané et le réglage de plusieurs paramètres peuvent souvent conduire à des coûts de calcul extrêmement élevés sans assurer l'obtention de bons résultats.

## 2.10. Conclusion

Les algorithmes évolutionnaires et génétiques peuvent être une alternative intéressante pour la conception des filtres RIF. Des chercheurs ont publié des articles et des études traitant les algorithmes évolutionnaires dans une grande variété de domaines et participent ainsi à l'extension de la frontière des problèmes qui peuvent être traités efficacement par ces techniques. Les AG's sont réputés à être plus efficaces et plus attrayants à trouver les solutions optimales ou quasi-optimales que les algorithmes classiques.

De nombreuses études prometteuses provenant de la communauté de la reconnaissance de formes sont axées sur l'adaptation des paramètres (les taux de mutation et de croisement, la taille des populations ...) et sur la proposition de nouvelles variantes d'opérateurs génétiques (par exemple le croisement intelligent, la recombinaison élitiste) pour lutter contre ces bien connues restrictions générales. Les AG's utilisés pour traiter les problèmes de conception de filtres RIF pourraient tirer davantage parti de ces progrès.

Nous pensons qu'il y a des potentiels pour améliorer les versions de base des algorithmes génétiques et des optimisations par essaim de particules (PSO), en intégrant des mécanismes plus intelligents dans ces approches. L'hybridation entre les méthodes évolutionnaires est une direction intéressante, mais elle doit être bien gérée.

Les chapitres suivants traitent l'élaboration, la mise en œuvre et l'exploitation efficace des processus évolutionnaires des techniques citées dans ce chapitre. Le chapitre 3 présente en détail, des mécanismes adaptatifs intégrés dans des approches évolutionnaires pour améliorer leur performance et faciliter leur utilisation par les praticiens. Le chapitre 4 a pour but de valider les approches mises en œuvre à partir de plusieurs exemples de synthèse et expose les résultats et les performances de chaque technique élaborée.



## Chapitre III

---

# 3. Elaboration d'approches évolutionnaires dédiées à la synthèse des filtres RIF

### Sommaire

---

3.	Elaboration d'approches évolutionnaires dédiées à la synthèse des filtres RIF	45
3.1.	Introduction .....	46
3.2.	AG adaptatif dédié à la synthèse des filtres RIF 1D et 2D.....	46
3.2.1.	Représentation des chromosomes .....	47
3.2.2.	Fonction fitness.....	47
3.2.3.	Population initiale .....	49
3.2.4.	Stratégie génétique.....	49
3.2.5.	Opérateurs génétiques et mécanismes dédiés à améliorer la convergence et les performances.....	52
3.2.5.1.	Opérateur de croisement.....	52
3.2.5.2.	Opérateur de mutation .....	52
3.2.5.3.	Procédure de sélection perfectionnée.....	54
3.2.5.4.	Processus de rafraîchissement pour éviter la convergence prématurée	56
3.3.	Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF .....	58
3.4.	Hybridation entre l'AG adaptatif et l'Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF .....	59
3.5.	L'algorithme hybride .....	60
3.5.1.	Détails de l'algorithme génétique hybride (HGA) .....	61
3.5.2.	Recherche locale .....	62
3.5.3.	Activation de l'approche locale et sélection des chromosomes.....	63
3.5.4.	Durée de la recherche locale .....	70
3.6.	Résumé de l'approche AGA .....	70
3.7.	Conclusion.....	71

---

### **3.1. Introduction**

La complexité de la conception des filtres RIF se heurte aux limitations de la conception par les méthodes classiques, qui ne peuvent pas traiter toutes les contraintes de performances des filtres.

Les méthodes évolutionnaires, en particulier les algorithmes génétiques (AG's) ont le potentiel de fournir des réponses aux limitations des méthodes conventionnelles. Ils sont capables d'effectuer l'optimisation multicritère dans des chemins qui mènent automatiquement à des compromis de performances entre les spécifications de conception.

La performance de l'AG dépend de nombreux paramètres tels que la taille de la population, la probabilité de croisement, la probabilité de mutation. Les algorithmes génétiques sont inspirés du processus d'évolution, il est naturel de s'attendre à ce que l'adaptation n'est pas seulement utilisée pour trouver des solutions à un problème donné, mais aussi pour accorder ces algorithmes génétiques à un problème particulier. La question du contrôle des valeurs des différents paramètres d'un algorithme génétique est l'un des domaines les plus importants et les plus prometteurs de la recherche en calcul évolutif : il a le potentiel d'adaptation de l'algorithme au problème tout en résolvant le problème.

La mise en œuvre d'un algorithme évolutionnaire pour la conception des filtres RIF, est cependant confrontée à un certain nombre de défis. L'espace des filtres possibles est très large, et les paramètres individuels sont étroitement couplés, ce qui rend plus probable la convergence de l'algorithme vers des sous-optima locaux insatisfaisants. Un autre point est que, les AG's sont réputés pour être performants dans la recherche des régions prometteuses de l'espace de recherche, mais inefficaces dans la détermination du minimum local en termes de vitesse de convergence et de la qualité de la solution.

Le champ d'applications des AG's a clairement progressé, principalement en raison d'un certain nombre d'innovations durant les dix dernières années. Il a intégré des stratégies plus élaborées visant à remédier aux faiblesses de l'AG. Toutefois, une analyse approfondie des études relatives à l'application d'algorithmes génétiques pour la conception des filtres RIF montre que ce domaine spécifique ne bénéficie pas clairement de tous les progrès qui ont été faits dans la meilleure application d'algorithmes génétiques à des problèmes réels et généraux. Le défi dans les AG's d'aujourd'hui n'est pas dans la recherche de nouvelles alternatives conceptuelles, mais plutôt dans leur utilisation par des non-experts.

### **3.2. AG adaptatif dédié à la synthèse des filtres RIF 1D et 2D**

Il y a littéralement un certain nombre de travaux pertinents représentant un large échantillon de l'utilisation des algorithmes génétiques dans la littérature au cours des 20 dernières années et couvrent un large éventail de sujets. Toutefois, le succès des AG's est lié à la configuration globale de l'AG, au réglage et à l'adaptation de ses paramètres pour atteindre une performance donnée. Tout au long de la documentation pertinente, on peut voir que le coût et l'apport du réglage des paramètres sont largement ignorés par la communauté scientifique. L'adaptation et la flexibilité des AG's sont actuellement les domaines les plus importants et les plus prometteurs de la recherche en calcul évolutif.

Le premier algorithme proposé dans ce travail est un algorithme génétique adaptatif dédié à la synthèse des filtres RIF. Plusieurs mécanismes ont été étudiés pour rendre le processus plus rapide et évolutif afin d'accélérer la convergence. Notre algorithme est une version modifiée et améliorée de la version standard de l'AG. Il est codé en réel avec des fonctions supplémentaires telles que l'initialisation biaisée, des paramètres de mutation dynamiques, la diversification et l'élitisme. Dans cet algorithme, plusieurs paramètres ont été spécialement développés pour la conception de filtres FIR 1D et 2D.

L'utilisation d'une mutation adaptative (mutation non uniforme et un taux de mutation variable) à chaque itération pour permettre à l'AG d'échapper aux optimums locaux et pour empêcher la convergence prématurée. La génération de la population initiale dans un concept pseudo-aléatoire est introduite pour augmenter la vitesse de convergence en réduisant l'espace de recherche en fonction des propriétés des coefficients des filtres. D'autre part, cet algorithme utilise l'opérateur de croisement basé-direction (direction-based crossover), dans ce cas la connaissance du problème spécifique est introduite dans l'opération génétique afin de produire de meilleurs descendants. Le croisement basé-direction fait bon usage des valeurs de la fonction objective (fitness) pour déterminer l'orientation de la recherche génétique. En outre, afin d'effectuer une évaluation plus précise de la valeur de la fonction fitness des chromosomes, un autre paramètre est introduit dans la fonction coût, ce paramètre est appelé fonction de pondération. Un autre facteur important dans notre algorithme génétique est la sélection qui opère sur un espace élargi. Quand la sélection est réalisée dans un espace élargi, les parents et leurs descendants ont la même chance de rivaliser pour la survie. Un avantage évident de cette approche est que nous pouvons améliorer les performances de l'AG en augmentant les taux de croisement et de mutation.

### 3.2.1. Représentation des chromosomes

L'algorithme génétique est codé en réel car cette représentation est plus appropriée (précise) que le codage binaire pour la conception des filtres RIF. Le chromosome est une matrice de dimensions  $P_1 \times P_2$  représentant directement les coefficients du filtre RIF 2D à synthétiser. Cela signifie que pour un filtre de dimensions  $P_1 \times P_2$ , il est nécessaire de déterminer  $P_1 \times P_2$  coefficients, le but étant de trouver les coefficients qui minimisent l'erreur quadratique  $E$  sur  $m_1 \times m_2$  points fréquentiels. Les chromosomes ayant de faibles erreurs auront une plus grande probabilité d'être sélectionnés en tant que parents dans la génération suivante.

Dans le cas d'un filtre RIF 1D, le chromosome est un vecteur de longueur  $N+1$ . L'erreur quadratique est minimisée sur  $m_1$  points fréquentiels.

### 3.2.2. Fonction fitness

La qualité d'une solution représentée par un chromosome est donnée par la fonction de fitness (adéquation, adaptation, objective, coût ...) représentant son potentiel. Elle est utilisée par l'AG pour différencier les bonnes des moins bonnes solutions, de sorte qu'il est possible de sélectionner les meilleures en conséquence. Cette fonction de fitness est définie comme étant l'inverse de l'erreur  $E$  qui doit être maximisée.

Pour un gabarit donné de filtre, les ondulations sont généralement plus présentes dans certaines régions que dans d'autres, ce qui rend leur atténuation (minimisation) plus difficile.

Pour faire face à cette situation, nous suggérons la pondération de l'erreur quadratique, ce qui contribuera à compenser ce déséquilibre récurrent. Comme les poids de pondérations sont difficiles à régler théoriquement en raison de l'absence de progrès dans l'AG lui-même, ils ont été fixés expérimentalement pour les différents types de filtres. Le choix des valeurs de la fonction de pondération est donné dans le tableau 3.1, qui résume les différents paramètres de réglage de notre algorithme génétique adaptatif.

Le critère d'erreur utilisé est donné par l'équation 3.1.

$$E_i = \sum_{j=1}^{m_1} \sum_{k=1}^{m_2} W(\omega_{1j}, \omega_{2k}) * [D(\omega_{1j}, \omega_{2k}) - C_1(\omega_{1j}) * A_i * C_2(\omega_{2k})^T]^2 \quad (3.1)$$

Où  $W$  est la matrice de pondération de dimensions  $m_1 \times m_2$

$D$  est la matrice de dimensions  $m_1 \times m_2$  qui donne la réponse d'amplitude désirée

$C_1$  est une matrice cosinus de dimensions  $m_1 \times P_1$

$C_2$  est une matrice cosinus de dimensions  $m_2 \times P_2$

La matrice  $A_i$  formée par les coefficients du filtre du  $i^{\text{ème}}$  chromosome de dimensions  $P_1 \times P_2$

La fonction fitness est calculée pour chaque chromosome dans les différentes étapes de l'algorithme génétique adaptatif. A cet effet, il est nécessaire de multiplier la matrice cosinus  $C_1$  par la matrice coefficients  $A_i$  ; le résultat est multiplié par la transposée de la matrice cosinus  $C_2$  ( $C_2^T$ ). Ensuite, on effectue la soustraction entre la matrice contenant la réponse désirée et le résultat du produit matriciel. Finalement, la valeur de l'inverse de la fonction fitness est obtenue en multipliant le résultat de la soustraction par la matrice de pondération  $W$ . Ces opérations (évaluation de la fonction fitness) sont effectuées à trois étapes de notre algorithme, c'est-à-dire à l'étape de croisement, de mutation et de sélection.

Pour un filtre RIF unidimensionnel (1D), l'erreur est donnée par la formule suivante :

$$E_i = \sum_{k=1}^{m_1} W(\omega_k) * [D(\omega_k) - C(\omega_k)^T * B_i]^2 \quad (3.2)$$

Où  $W$  est le vecteur de pondération de longueur  $m_1$ .

$$W(\omega_k) = \begin{cases} W_p & \omega_k \in \text{passband} \\ W_s & \omega_k \in \text{stopband} \end{cases} \quad (3.3)$$

$H(\omega_k) = C(\omega_k)^T * B_i$  est la réponse en amplitude.

Le vecteur  $B_i$  de longueur  $N+1$  est formé par les coefficients du  $i^{\text{ème}}$  chromosome.

$$B_i = [b_i(0) \ b_i(1) \ b_i(2) \ \dots \ b_i(N)]^T \quad (3.4)$$

$C$  est le vecteur cosinus de longueur  $N+1$ .

$$C(\omega) = [1 \cos(\omega) \cos(2\omega) \dots \cos(N\omega)]^T \quad (3.5)$$

$D$  est un vecteur de longueur  $m_1$  donnant la réponse d'amplitude désirée.

La fonction fitness représente l'inverse de l'erreur pour les deux cas (1D et 2D) :

$$f_i = 1/(1 + E_i)$$

### 3.2.3. Population initiale

Les performances de l'AG dépendent fortement de la population initiale à partir de laquelle l'algorithme commence à évoluer.

Le chromosome est généré de la manière suivante :

$$POP_0 = \left(\frac{1}{a}\right) * randn(P_1, P_2) \quad (3.6)$$

Où  $P_1 \times P_2$  est la dimension du filtre,  $a$  est un entier  $\in [5, 20]$

Dans le cas unidimensionnel, le chromosome est généré par la formule suivante :

$$POP_0 = \frac{randn(1, N + 1)}{a} \quad (3.7)$$

En Matlab, cette instruction génère des nombres pseudo-aléatoires d'une distribution normale de moyenne 0 et d'écart type  $(1/a)$ .

Cela permet à la population initiale d'avoir des valeurs appropriées par rapport à notre problème d'optimisation (des expériences ont montré que l'écart type entre les gènes des chromosomes est inférieur à 0.2). La façon de générer la population initiale n'affecte pas la qualité des résultats, mais peut augmenter ou diminuer légèrement le temps nécessaire à la convergence. Le cas échéant,  $POP_0$  peut être partiellement représentée par des solutions mutées obtenues avec les approches standards. Bien que n'étant pas systématique, ce processus accélère généralement la convergence génétique (le temps CPU et le nombre d'itérations pour obtenir des solutions satisfaisantes).

### 3.2.4. Stratégie génétique

Le schéma choisi est classiquement entraîné par les différents opérateurs de croisement, de mutation, de sélection et de reproduction comme illustré sur la figure 3.1 et 3.2. Il comprend toutefois quelques différences majeures qui seront décrites plus tard dans la façon dont les opérations successives sont appliquées.

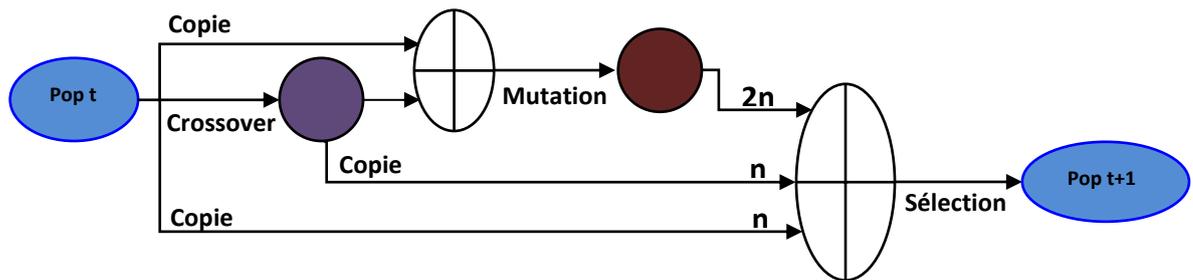


Figure 3.1 – Processus génétiques.

Les  $n$  chromosomes de la génération  $t$  produisent  $2 * n$  chromosomes à partir de l'opérateur de croisement (crossover). Dans ce cas, il n'est pas nécessaire de fixer un taux de croisement (prendre  $C_R = 100\%$ ). Une copie de la population mère et de descendants est sauvegardée. Les parents et les enfants sont regroupés et soumis à l'opérateur de mutation. Chaque composant du chromosome a une probabilité  $M_P$  d'être muté. La valeur  $M_P$  est calculée individuellement pour chaque chromosome ainsi que le niveau (l'amplitude) de mutation et dépendent de l'avancement de l'algorithme génétique. C'est ce qu'on appelle un paramètre auto-adaptatif. Une population élargie (de taille  $4 * n$ ) chromosomes est ensuite soumise à l'opérateur de sélection qui élimine les mauvais chromosomes et les chromosomes redondants pour obtenir une population de taille  $n$ . L'opérateur de sélection prend en compte à la fois l'élitisme et la diversité. La décision de choisir un chromosome donné dans la nouvelle population est basée sur deux critères : le premier est le score de fitness. S'il existe un chromosome dans la population de candidats (parmi les  $4 * n$ ) avec un score plus élevé que les autres candidats, il est susceptible d'être sélectionné. C'est le côté élitiste du processus. Toutefois, si un candidat répond à ce critère élitiste mais présente une structure comparable à un chromosome qui a déjà été sélectionné, il est susceptible d'être rejeté. C'est le rôle du processus de diversification. Notre programme inclut implicitement l'éviction bien connue et le concept de niches. Il est très rapide et facile à mettre en œuvre car il nécessite seulement la définition de la similarité entre deux chromosomes. Bien que l'évaluation est réalisée sur une population de  $4 * n$  individus et prend plus de temps, mais le nombre d'itérations est réduit.

Lorsque ce processus génère moins de  $n$  chromosomes, la population est complétée par de nouveaux chromosomes générés à partir de la dernière population après être soumise à une grande mutation.

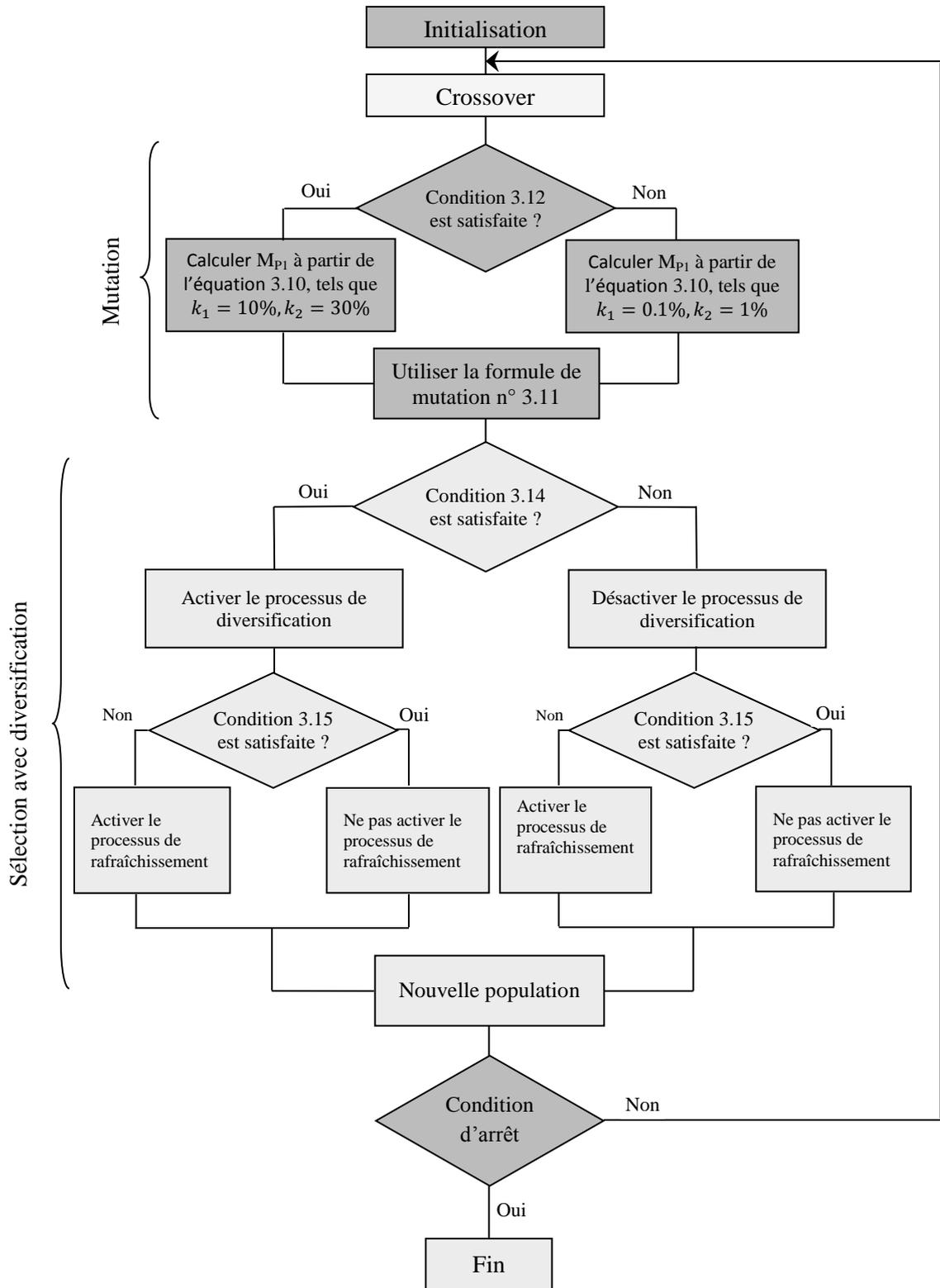


Figure 3.2 – Organigramme de l'AGA.

### 3.2.5. Opérateurs génétiques et mécanismes dédiés à améliorer la convergence et les performances

Certaines études ont déployé des efforts considérables en vue de trouver les valeurs de paramètres qui donnent des performances raisonnables pour un large éventail de problèmes. Des problèmes spécifiques nécessitent cependant des paramètres dédiés soulignant la nécessité de mécanismes supplémentaires qui permettent un bon réglage des paramètres. Le bon jeu de paramètres à un niveau d'avancement de la vie génétique n'est pas le même à un autre moment. Notre système de croisement est basé sur une approche standard améliorée, tandis que les régimes de mutation et de sélection sont plus sophistiqués. Ils comprennent des mécanismes qui contribuent à la souplesse de l'AG et encouragent un minimum de diversité au sein de la population.

#### 3.2.5.1. Opérateur de croisement

Le croisement est l'opérateur principal dans l'AG et la clé de leur succès. Le croisement basé-direction utilise les valeurs de la fonction objective (fitness) pour déterminer l'orientation de la recherche génétique. Dans ce cas, nous n'avons pas à nous soucier d'un taux de croisement élevé (égal à 1) parce que la nouvelle population est choisie parmi les meilleurs entre parents et enfants.

L'opérateur génère deux descendants  $Ch_C^{child}$  à partir de deux parents  $i$  et  $j$  selon la formule suivante :

$$\begin{cases} Ch_C^{child1} = (Ch^i - Ch^j) * \lambda_{c1} + Ch^i \\ Ch_C^{child2} = (Ch^j - Ch^i) * \lambda_{c2} + Ch^j \end{cases} \quad (3.8)$$

Où  $Ch^i$  ( $Ch^j$ ) représente l'individu  $i$  ( $j$ ), et  $\lambda_{c1}$  et  $\lambda_{c2}$  sont des nombres aléatoires entre 0 et 1.

$$\begin{cases} Si (fit(i) > fit(j)) & \lambda_{c1} > \lambda_{c2} \\ Sinon & \lambda_{c1} < \lambda_{c2} \end{cases} \quad (3.9)$$

Le croisement permet la génération de nouveaux chromosomes sans introduire de nouvelles caractéristiques génétiques dans la population au niveau des gènes.

La probabilité de croisement est  $C_R = 100\%$ .

#### 3.2.5.2. Opérateur de mutation

L'opérateur de mutation modifie arbitrairement la valeur des gènes selon une probabilité prédéterminée. L'opérateur de mutation apporte une contribution significative à l'efficacité de la recherche. Il introduit la diversité et reflète les caractéristiques qui ne sont pas présentes dans la population actuelle, et peut donc protéger contre une telle perte nuisible et empêcher la convergence prématurée. La stratégie de mutation est reconnue comme l'une des composantes de l'AG qui a le plus d'influence sur les performances. Classiquement, l'opérateur de mutation est appliqué pour les individus choisis de manière aléatoire. Par rapport à l'opérateur de mutation traditionnelle, nous proposons une mutation non-uniforme

directement applicable à chaque chromosome de la population, la non-uniformité est liée à la fitness des chromosomes. Afin de contribuer efficacement à une meilleure diversité, on propose également une version adaptative qui est capable de renforcer l'action de la mutation, si nécessaire.

L'idée d'une mutation non-uniforme est relativement bien connue dans la communauté génétique. Le taux de mutation adaptatif varie en fonction de l'amélioration moyenne des meilleures valeurs de la fonction fitness pendant un nombre spécifié de générations. Le taux de mutation est limité pour les meilleurs chromosomes et encouragé (conséquent) pour les mauvais. Pour un chromosome donné  $i$ ,  $M_{P_i}$  est appliquée à chaque gène.

$$M_{P_i} = \begin{cases} \frac{k_1 * (f_{max} - f_i)}{(f_{max} - f_{avg})} & f_i \geq f_{avg} \\ k_2 & f_i < f_{avg} \end{cases} \quad (3.10)$$

Où  $f_{max}$  et  $f_{avg}$  sont, respectivement, les valeurs maximales et moyennes de la fitness dans la population actuelle.  $f_i$  est la valeur de la fitness du chromosome  $i$  sélectionné à la mutation.  $k_1$  et  $k_2$  sont des paramètres de pondération avec  $k_1$  et  $k_2 \leq 1$ . Ils sont respectivement la borne inférieure et la borne supérieure des paramètres de pondération du taux de mutation.

Comme expliqué plus haut, cette adaptation concerne la différence entre les valeurs de fitness des chromosomes. Cette dynamique est d'une certaine manière dissociée du progrès génétique qui peut laisser les praticiens sans aucune aide pour une situation donnée.

Les intervalles de confiance pour les bonnes valeurs pour les paramètres de pondération  $k_1$  et  $k_2$  ont été déterminées de manière à être disponibles avant l'exécution de l'algorithme. Les valeurs ont été évaluées que sur une échelle limitée et la configuration parfaite n'existe pas. Le paysage change la performance des paramètres au fil du temps car il est dépendant des progrès génétiques. Le réglage de ces paramètres est inutile et n'est donc pas nécessaire. La nouveauté de notre approche réside dans les conditions d'utilisation en distinguant l'état où la population évolue d'une manière normale et l'état où la population stagne.

L'idée consiste à relancer un processus dévolution qui stagne en rafraîchissant plusieurs chromosomes. Le taux de mutation  $M_{P_i}$  est augmenté une fois par les valeurs  $k_1$  et  $k_2$ . Il est effectivement augmenté chaque fois que la population a tendance à se coincer dans un optimum local et diminué lorsque la population est dispersée dans l'espace de recherche. La probabilité de mutation  $M_{P_i}$  est appliquée à chaque gène, comme expliqué précédemment. Si la mutation est activée, le niveau de mutation prend également les informations à partir de l'état actuel de la recherche, et modifie en conséquence les chromosomes.

Différentes approches peuvent être étudiés. Nous suggérons l'adoption d'une version à deux cas décrite comme suit:

$$Ch^{new} = \begin{cases} Ch^{old} * (1 + \lambda_{m1}) & \text{si minimum local} \\ Ch^{old} * (1 + \lambda_{m2}) & \text{sinon} \end{cases} \quad (3.11)$$

L'opérateur génère un nouveau chromosome  $Ch^{new}$  à partir de l'ancien chromosome  $Ch^{old}$ . Où  $\lambda_{m1}$ ,  $\lambda_{m2}$  sont des paramètres dépendant de l'avancement de la recherche ( $\lambda_{m1} \gg \lambda_{m2}$ ) et  $Ch$  définit chaque composante du chromosome.

La présence d'un minimum local n'est validée que si trois critères sont simultanément activés :

- Une avance suffisante de vie génétique caractérisée par le niveau moyen de la fonction fitness des solutions candidates qui est comparé à un seuil  $th_1$  ( $f_{avg} \leq th_1$ ).
- Absence d'une variation de la population entre deux itérations génétiques caractérisée par la valeur de la dérivée de la fonction fitness moyenne ( $\Delta_{fitness(avg)} \leq th_2$  où  $th_2$  est un seuil).
- Absence d'un écart entre la meilleure solution génétique et la moyenne des solutions ( $\frac{f_{max} - f_{avg}}{f_{avg}} \leq th_3$ , où  $f_{max}$  et  $f_{avg}$  sont respectivement la meilleure fitness et la fitness moyenne des solutions et  $th_3$  est un seuil).

Lorsque ce mécanisme est actif dans notre algorithme, nous disons qu'une mutation non-uniforme adaptative est appliquée.

Le procédé de vérification de la présence d'un minimum local peut être résumé dans cette équation :

$$\begin{cases} \text{If } (f_{avg} \leq th_1 \text{ et } \Delta_{fitness(avg)} \leq th_2 \text{ et } \frac{f_{max} - f_{avg}}{f_{avg}} \leq th_3) \text{ Presence de minimum local} \\ \text{Sinon pas de présence de minimum local} \end{cases} \quad (3.12)$$

La détection d'un minimum local est uniquement basée sur les valeurs de fitness. Elle présente un très faible coût de calcul car elle est basée sur des calculs simples. Une procédure d'étalonnage élémentaire permet de trouver des intervalles de confiance.

### 3.2.5.3. Procédure de sélection perfectionnée

La stratégie de sélection des parents a un grand impact sur les performances de l'AG. Plusieurs recherches pensent que la manipulation des paramètres concernant les mécanismes de sélection est plus intéressante que celles qui se rapportent au croisement et à la mutation. Le dispositif mis en œuvre est un système auto-adaptatif basé sur un processus dans lequel les paramètres sont implicites, c'est à dire qu'ils sont sélectionnés par le cycle évolutif lui-même. Ils ne nécessitent aucune intervention de l'utilisateur ni tout autre réglage durant l'exécution. L'équilibre entre la pression de sélection et la diversification est obtenu grâce à la stratégie et non par des paramètres qui doivent être modifiés au cours du processus évolutif. La stratégie a été largement validée pour le problème de conception de filtres.

Nous avons utilisé la sélection basée sur un espace élargi; les parents et leur progéniture ont la même chance de rivaliser pour la survie.

Notre schéma de sélection vise à améliorer la qualité de la population actuelle en sélectionnant des chromosomes de qualité supérieure et ayant une plus grande probabilité d'être insérés dans la population suivante, sans réduire la diversité de la population. Ensuite,

le schéma de sélection permet à l'AG de traiter avec des fonctions multimodales et peut explorer plusieurs collines dans le paysage de la fonction fitness en même temps.

$$P_i = f_i / \sum_{j=1}^n f_j \quad (3.13)$$

Où  $f_j$  est la valeur de la fitness du  $j^{\text{ème}}$  chromosome.

La procédure n'est pas un processus de classement pur mais agit avec le même objectif, à savoir empêcher les individus très en forme (ayant une fitness très grande) de gagner la domination précoce au détriment de ceux moins adaptés.

Le classement est basé uniquement sur les différences de probabilité, mais l'algorithme de sélection inclut le matériel génétique.

L'algorithme commence avec le meilleur chromosome appelé *courant*. Ce chromosome est automatiquement sélectionné et le premier par rapport au deuxième appelé *inspecté* par l'intermédiaire de la fonction de fitness. S'il n'y a pas de correspondance (similitude) entre les deux, le second devient *courant* et la procédure se poursuit. Sinon, la comparaison est effectuée via un gène métrique afin d'éviter les rejets de deux chromosomes ayant des valeurs similaires de fitness tout en étant génétiquement différents. *Inspecté* est rejeté s'il n'y a pas de différence entre eux dans cet espace. Dans ce cas, le premier chromosome est toujours d'actualité. La comparaison se fait ensuite avec le troisième chromosome qui devient *inspecté*. La procédure se poursuit jusqu'à ce que toute la population ait été inspectée.

Ce mécanisme contribue à une meilleure diversification et est donc appelé "diversification".

$$\begin{cases} \text{si } \Delta_{fitness}(ch_{courant}, ch_{inspecté}) \geq f_{\epsilon} \text{ continuer sinon} \\ \text{si } \Delta_{gene}(ch_{courant}, ch_{inspecté}) \geq g_{\epsilon} \text{ continuer sinon} \\ \text{rejeter } ch_{inspecté} \end{cases} \quad (3.14)$$

$ch_{courant}$  ( $ch_{inspecté}$ ) représente le chromosome *courant* (*inspecté*) et  $\Delta_{fitness}$  (resp.  $\Delta_{gene}$ ) représente la distance dans l'espace des fitness (la distance dans l'espace des gènes), et  $f_{\epsilon}$  (resp.  $g_{\epsilon}$ ) sont des seuils prédéfinis.

Le coût du programme de sélection est lié au classement en fonction de la fitness et en fonction de l'espace génétique;  $n-1$  distances doivent être calculées dans chaque espace. En ce qui concerne l'espace des gènes, des distances différentes peuvent être sélectionnées : distances euclidienne entre tous les gènes des deux chromosomes et distances euclidiennes directes dans des sous-espaces de coefficients sélectionnés en tant que différence de coefficients de filtre. Le second est préférable car le coût est lourd pour le premier. Dans le cas bidimensionnel, nous commençons par comparer les gènes de positions (1,1), (1,  $P_2$ ), ( $P_1$ , 1) et ( $P_1$ ,  $P_2$ ) de l'individu préalablement sélectionné et de l'individu inspecté. Puis, nous tirons au hasard d'autres positions pour les comparer. En général, les quatre premières

comparaisons sont suffisantes pour confirmer ou infirmer une similitude entre les gènes des deux chromosomes.

L'élément gauche de la position est l'indice de la ligne du gène de la matrice  $A_i$  et l'élément de droite de la position est l'indice de la colonne du gène de la matrice  $A_i$ .

Dans le cas unidimensionnel, nous commençons par comparer les gènes en positions 1 et  $N + 1$  de l'individu sélectionné précédemment et l'individu inspecté. Nous pouvons ensuite tirer au hasard d'autres positions et les comparer, si nécessaire. En général, les deux premières comparaisons sont suffisantes pour confirmer ou infirmer une similitude entre les gènes des deux chromosomes.

La position est l'indice ligne du gène donné par le vecteur  $B_i$ .

Il convient de noter que le second calcul est traité uniquement lorsque le chromosome *courant* et le chromosome *inspecté* ont des fonctions fitness semblables. Les nouveaux chromosomes prometteurs dans la population sont insérés en remplaçant les plus semblables. Ce mécanisme est appelé sélection adaptative avec diversification car il induit la modification automatique de la probabilité de sélection. Les individus avec une meilleure fitness sont sélectionnés en tenant compte du maintien de la diversité de la population.

### **3.2.5.4. Processus de rafraîchissement pour éviter la convergence prématurée**

L'équilibre entre la pression de sélection et la diversité de la population dépend du problème à résoudre. Malheureusement, il n'existe pas de manière universelle pour gérer simultanément la pression de sélection et la diversification. Comme déjà mentionné, notre stratégie consiste à définir les paramètres qui donnent une performance raisonnable pour un large éventail de problèmes, tout en offrant à notre algorithme une certaine souplesse et adaptabilité. Pour ces raisons, la flexibilité intégrée dans notre opérateur de mutation est une caractéristique intéressante, mais elle ne peut pas prévenir et gérer tous les cas de convergence prématurée. L'évolution génétique même guidée n'est pas toujours contrôlable. La même chose vaut pour le schéma de sélection.

Nous introduisons donc la notion de rafraîchissement (rupture) de la population qui implique un changement profond dans la population actuelle.

Avant d'apporter une solution pratique, la convergence prématurée doit être détectée. La détection consiste à identifier un processus dévolution qui stagne, matérialisé par la présence de nombreux chromosomes semblables au sein de la population. Selon une métrique donnée, le procédé vise à calculer ou estimer le pourcentage de chromosomes semblables au sein de la population et le réensemencement de la population en conséquence. Deux chromosomes sont dits similaires si leur distance est inférieure à un seuil prédéfini en fonction de la métrique active. Bien qu'efficace, cette approche peut être fastidieuse. Pour  $n$  chromosomes, ce procédé nécessite le calcul de  $n * (n - 1)/2$  distances sur l'espace des gènes.

Une façon plus rapide consiste à travailler directement sur l'espace de la fonction fitness et en appliquant la règle suivante : la population actuelle doit être réensemencée lorsque la différence entre la fonction fitness du meilleur chromosome et la fonction fitness moyenne

de la population est inférieure à un seuil donné. Dans les deux cas, le réensemencement est soumis au niveau du progrès génétique si cette information est disponible. C'est généralement le cas.

$$\begin{cases} \text{si } \Delta f_{fitness}(f(ch_{best}), f(ch_{ave})) \leq \Delta f_{min} & \text{et} \\ f(ch_{best}) \leq f_{end} & \text{processus de rafraîchissement actif} \\ \text{sinon} & \text{processus de rafraîchissement inactif} \end{cases} \quad (3.15)$$

Où  $\Delta f_{min}$  et  $f_{end}$  sont les différents seuils et  $f(x)$  la fitness du chromosome  $x$ .

Nous proposons un mécanisme de rafraîchissement (ou processus de rupture) capable de réensemencer la population active sans perdre son avancement actuel (garder la fitness moyenne supérieure ou égale à celle calculée avant le processus de rafraîchissement).

Le rafraîchissement est tout simplement une mutation forte, caractérisé par une probabilité plus élevée et une amplitude plus forte. Ce mécanisme est hautement optimisé et agit en complémentarité avec les autres mécanismes à savoir la mutation adaptative et la procédure de diversification. Dans ce cas, la probabilité de mutation est fixée à une valeur élevée telle que  $M_p = 40\%$ .

		Paramètres	
Approches	<b>Adaptatif GA<sub>1</sub></b>	<i>Mutation adaptative non-uniforme avec la procédure de diversification</i> <i>Présence de minimum local</i> <i>Processus de rafraîchissement actif</i> <i>Direction-based crossover</i> <i>Sélection adaptative avec la procédure de diversification (Processus de rafraîchissement et de rupture)</i>	$k_1 = 0.1\%, k_2 = 1\%$ $k_1 = 10\%, k_2 = 30\%$ $M_p = 40\%$ $R_c = 100\%$
	<b>Adaptatif GA<sub>2</sub></b>	<i>Mutation non uniforme</i> <i>Processus de rafraîchissement actif</i> <i>Direction-based crossover</i> <i>Sélection adaptative avec la procédure de diversification (Processus de rafraîchissement et de rupture)</i>	$M_p = 0.5\%$ $M_p = 40\%$ $R_c = 100\%$
	<b>Adaptatif GA<sub>3</sub></b>	<i>Mutation adaptative non-uniforme avec la procédure de diversification</i> <i>Présence de minimum local</i> <i>Direction-based crossover</i> <i>Sélection basée sur un espace d'échantillons large</i>	$k_1 = 0.1\%, k_2 = 1\%$ $k_1 = 10\%, k_2 = 30\%$ $R_c = 100\%$
	<b>Standard GA<sub>4</sub></b>	<i>Mutation non-uniforme</i> <i>Direction-based crossover</i> <i>Sélection basée sur un espace d'échantillons large</i>	$M_p = 0.5\%$ $R_c = 100\%$

Table 3.1 – Description et paramètres de l'AG proposé.

### 3.3. Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF

Pour rechercher efficacement les optima dans un espace de solutions aussi large, exige de répondre à deux autres questions : comment les particules peuvent mettre à jour leurs meilleures positions connues afin de suivre les optima et comment peut-on organiser une population pour concilier la nécessité de suivre les optima existants contre la nécessité de distribuer les particules afin de rechercher dans l'espace restant pour trouver de nouveaux Optima ? Ces questions d'équilibre entre l'exploration et l'exploitation sont essentielles pour tous les algorithmes évolutionnaires. Les versions de base de la PSO offrent une nouvelle stratégie. Toutefois, elles ne peuvent pas gérer les problèmes liés à fixer les paramètres de contrôle de l'algorithme, la convergence prématurée, la stagnation et le fait de revoir la même solution maintes fois. Ainsi, plusieurs algorithmes dérivés ont été proposés au cours des dernières années pour améliorer l'efficacité de la PSO. Ils ont en commun l'objectif d'ajouter plus de flexibilité et permettre une contribution efficace afin d'éviter les problèmes cités ci-dessus. Nous avons choisi deux versions de PSO (et aussi la version standard) récemment développées pour la conception de filtres qui ont prouvé leur efficacité.

L'inconvénient de la PSO classique utilisée pour la génération de coefficients optimaux pour la conception des filtres RIF est que les résultats sont sous-optimaux. En général, les solutions initiales sont généralement loin de l'optimum global et donc un poids d'inertie  $w$  plus grand peut être bénéfique pour la recherche des optima globaux. De grands poids d'inertie permettent à la PSO d'explorer globalement l'espace des solutions et de petits poids d'inertie permettent à la PSO d'explorer localement l'espace. Ces poids d'inertie  $w$  (qu'on note  $\gamma$  dans les versions sophistiquées de la PSO) jouent un rôle important dans l'équilibre entre les capacités d'exploration globales et locales. La valeur de  $w$  pour toutes les particules va diminuer en même temps que le nombre d'itérations augmente, et est calculée en utilisant l'expression suivante :

$$\gamma_{qi}^k = \gamma_{max} - (\gamma_{max} - \gamma_{min}) * \frac{Z_{iter,qi}^k}{Z}, \text{ if } V_{qi}^k(x_{i,gbest}^k - x_{qi}^k) > 0 \quad (3.16)$$

$$\gamma_{qi}^k = \gamma_{qi}^{k-1}, \text{ if } V_{qi}^k(x_{i,gbest}^k - x_{qi}^k) < 0 \quad (3.17)$$

Où  $\gamma_{max}$  et  $\gamma_{min}$  sont respectivement le poids initial et final.

$q = 1, 2, \dots, n_p; i = 1, 2, \dots, N$ .  $\gamma_{qi}^k$  est le poids d'inertie  $i$  de la particule  $q$  à l'itération  $k$ .

$Z$  est un paramètre contrôlant la diminution dynamique et linéaire des paramètres dans l'intervalle décroissant  $\gamma_{max}$  à  $\gamma_{min}$ .

La vitesse de la particule est mise à jour comme suit :

$$V_i^{k+1} = \gamma * V_i^k + C_1 * rand_1 * (pbest_i^k - X_i^k) + C_2 * rand_2 * (gbest^k - X_i^k) \quad (3.18)$$

La formule suivante montre la mise à jour de la position des particules :

$$X_i^{k+1} = X_i^k + V_i^k \quad (3.19)$$

NPSO est une version améliorée de la PSO qui propose une nouvelle définition du vecteur de vitesse et de mise à jour des essais.

Un choix approprié de  $Z$  assure un équilibre entre les explorations globales et locales, nécessitant ainsi moins d'itérations en moyenne pour trouver une solution suffisamment optimale. La principale caractéristique intéressante du mécanisme des poids d'inertie décrit ci-dessus est de contrôler le poids d'une particule, diminués linéairement dans des applications générales, pour éviter de mémoriser un trop grand nombre de particules similaires, à la fin du processus d'optimisation. L'importance du contrôle des poids d'inertie  $w$  ( $\gamma$ ) dans l'algorithme PSO est également retenue pour augmenter la possibilité d'échapper à des solutions optimales locales.

Les étapes de la NPSO élaborée pour la conception des filtres RIF sont :

Etape 1 : Initialisation de la taille de la population de vecteur de particules (taille de l'essaim), du nombre maximal d'itérations, du nombre de coefficients du filtre, du paramètre  $Z$ , des vitesses de particules ; choix des valeurs de  $C_1$ ,  $C_2$  et le nombre d'échantillons de fréquence.

Etape 2 : Générer aléatoirement les vecteurs de particules initiaux (coefficients du filtre) et calculer les fitness initiales de toutes les particules.

Etape 3 : Calculer les fitness et les meilleures positions de chaque particule. Comparer la fitness actuelle de chaque particule avec  $fpbest$ . Si la fitness actuelle est meilleure que  $fpbest$ , remplacer  $fpbest$  par la fitness actuelle.

Etape 4 : Comparer la fitness actuelle de chaque particule avec  $fgbest$ . Mettre à jour  $fgbest$ , la meilleure fitness des particules.

Etape 5 : Mettre à jours toutes les vitesses et les positions des particules.

Etape 6 : Répéter les étapes 3–5 jusqu'à atteindre le nombre maximum d'itérations ou jusqu'à atteindre le seuil minimum de l'erreur. Finalement,  $gbest$  est le vecteur de particule représentant les coefficients du filtre.

### **3.4. Hybridation entre l'AG adaptatif et l'Optimisation avancée par essaim de particules (NPSO) pour la synthèse des filtres RIF**

Nous pensons qu'il y a un potentiel à améliorer les versions de PSO en intégrant des mécanismes plus intelligents que cela a été fait pour l'AGA. L'hybridation entre les systèmes est un domaine prometteur qui a démontré sa capacité à améliorer les résultats. Il doit cependant être bien géré. L'hybridation ne se comporte pas toujours selon les attentes (manque de résultat, répétabilité, flexibilité ...). La mauvaise utilisation des mécanismes peut gaspiller les ressources de l'algorithme et même conduire à des performances plus mauvaises. La mise en œuvre de régimes hybrides est encore un problème. Aujourd'hui, il faut dire que cela met en évidence une limitation essentielle de l'utilisation de l'hybridation mais encourage les chercheurs travaillant sur ce champ libre pour développer des modèles

hybrides plus efficaces. En outre, il semble possible de contester les chercheurs d'unifier les concepts derrière les PSO et les AG : amélioration de la coopération entre les chromosomes dans l'AG, une meilleure sélectivité et la capacité de diversification de PSO et à la fois, une meilleure analyse de l'état de l'évolution et d'adapter intelligemment les mécanismes intégrés.

La NPSO est plus rapide que l'AGA, l'AGA est plus optimal que la NPSO. Le but est de combiner ces deux techniques afin de profiter simultanément des avantages des deux méthodes.

L'idée est d'initialiser la recherche des coefficients du filtre par l'optimisation avancée par essaim de particules (NPSO) et dès que cette technique plafonne (stagne), l'algorithme génétique adaptatif (AGA) prend le relais afin d'améliorer davantage les résultats obtenus par la NPSO. Le but étant d'obtenir des résultats optimaux tout en minimisant le temps de calcul.

### **3.5. L'algorithme hybride**

L'algorithme se compose de deux phases qui sont initialement séquentielles mais peut alterner régulièrement les progrès génétiques. Dans la première, seule la partie pure de l'AG est active et l'objectif est d'explorer l'espace des solutions pour détecter les zones d'intérêt où des solutions peuvent être réglées. La deuxième phase est l'endroit où l'AG est hybride. Certaines régions prometteuses de l'espace de recherche ont été atteintes et donc peuvent être exploitées par la recherche locale (Fig. 3.3). L'équilibre entre l'exploration et l'exploitation est optimisé.

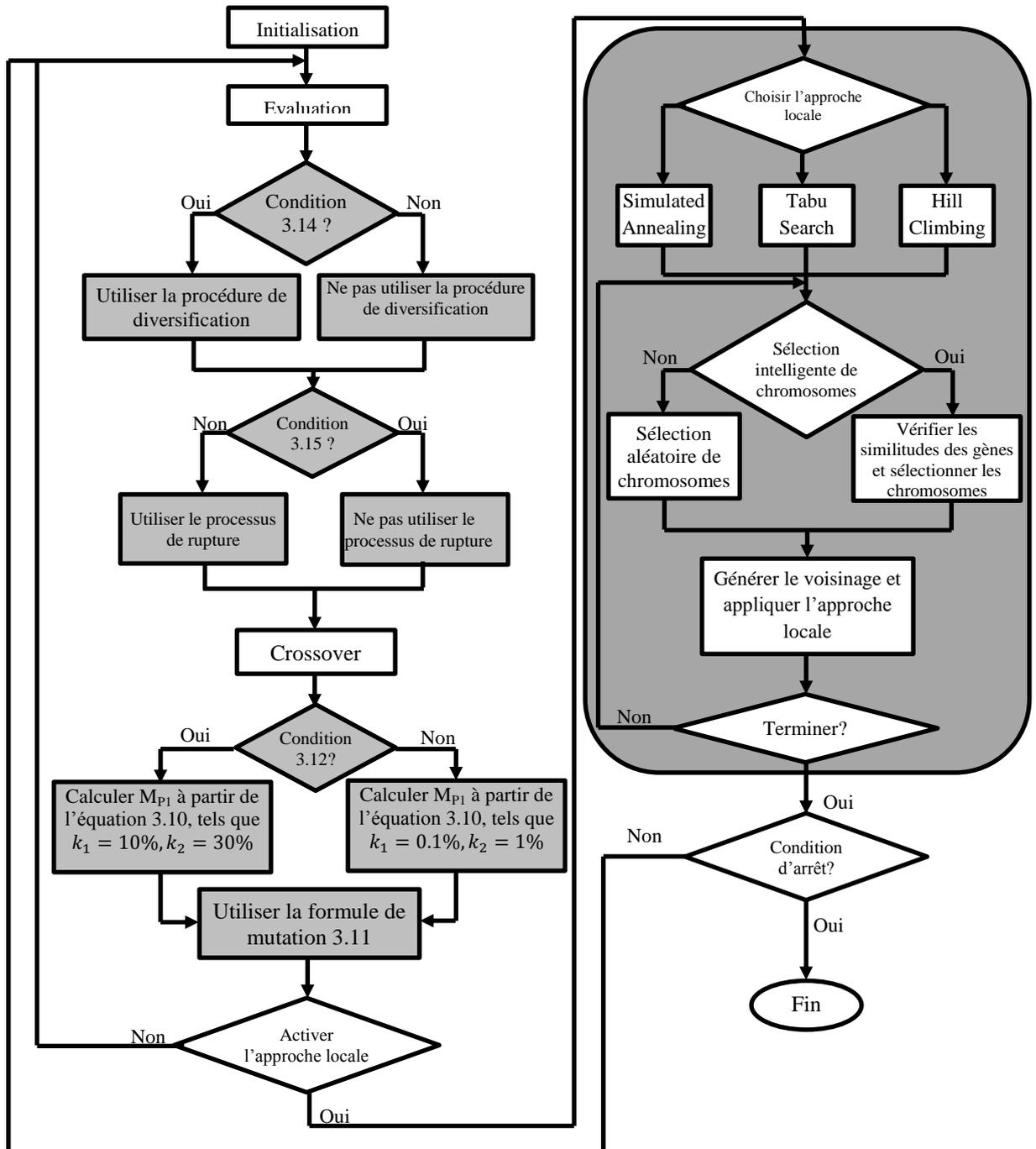


Figure 3.3 – Organigramme du HGA.

### 3.5.1. Détails de l'algorithme génétique hybride (HGA)

La mise en place de la structure d'un AG est un point critique, c'est la ligne directrice qui conduit à une bonne architecture. L'obtention des meilleurs résultats possibles est nécessaire, la répétabilité de l'optimisation et de la couverture de l'espace de recherche sont également très importants. La figure 3.3 présente l'organigramme de l'algorithme hybride.

---

**Algorithme 1** Différentes étapes du HGA

---

- Etape 1: (Initialisation) Définir la taille de la population  $Size_{pop}=TP$  et indiquer le nombre d'itérations maximales; utiliser le codage réel; générer aléatoirement  $Size_{pop}=TP$  individus comme population initiale
- Etape 2: Evaluer la population
- Etape 3: Sélectionner la population
- Etape 4: Réensemencer ou ne pas réensemencer la population par le processus de rafraîchissement
- Etape 5: Appliquer l'opérateur de croisement
- Etape 6: Appliquer l'opérateur de mutation avec ou sans le mécanisme de diversification
- Etape 7: Effectuer la recherche locale avec les paramètres appropriés sur les chromosomes sélectionnés
- Etape 8: Stop ou aller à l'étape 2
- 

Table 3.2 – Différentes étapes de l'algorithme génétique hybride (HGA)

### 3.5.2. Recherche locale

Le concept de base de l'application d'un schéma de recherche locale adaptative à l'AG est d'examiner si l'AG converge vers une solution optimale globale ou non. De toute évidence, lorsque la solution est proche, elle est continuellement améliorée par l'application d'une recherche locale qui peut être considérée comme une méthode de descente multi-départs sophistiquée. Comme les évaluations de la fonction fitness sont la partie la plus fastidieuse de l'algorithme, la préoccupation n'est pas seulement la performance, mais aussi l'optimisation du mécanisme qui équilibre l'AG pur et la recherche locale pour parvenir à une solution de la qualité souhaitée. La façon dont l'information acquise grâce à une recherche locale est utilisée dans un algorithme génétique hybride a un grand impact sur la performance du processus de recherche.

Les points clés de notre système hybride sont les suivants :

- **Equilibre génétique** : L'AG pur est chargé de l'exploration de l'espace, il est probable de trouver un ensemble de solutions acceptables, et les procédures locales visent à améliorer ces solutions par une recherche appropriée dans leur quartier (voisinage). La recherche locale doit être utilisée en particulier pour améliorer les chances à une solution de propager ses caractéristiques dans les générations à venir et non le contraire.
- **Interaction** : L'approche interagit directement avec la structure génétique en propageant le résultat de la recherche locale. L'état de la vie génétique est lié à l'information de la fitness. L'information de la fitness dans la population est ensuite utilisée pour polariser la recherche locale vers les chromosomes qui ont une meilleure fonction fitness.
- **Sélectivité** : Nous supposons que les chromosomes avec une meilleure fitness sont plus susceptibles d'être dans les bassins d'attraction des régions les plus prometteuses. Nous proposons d'adapter la recherche locale au niveau de la fitness des chromosomes qui réduit le nombre de candidats. Nous proposons également d'éviter l'utilisation de la recherche locale autant que possible en sélectionnant un ensemble réduit de chromosomes de référence parmi l'ensemble des candidats. Ainsi que quelques chromosomes de référence seront soumis à la recherche locale.

Notre approche peut fonctionner avec plusieurs types d'approches locales évolutives. Plusieurs méthodes peuvent jouer le rôle, nous recommandons la recherche tabou (TS), recuit simulé (SA) et Hill Climbing (HC). Pour chaque approche, les paramètres de fonctionnement doivent être spécifiés.

### 3.5.3. Activation de l'approche locale et sélection des chromosomes

Dans les algorithmes génétiques hybrides traditionnels, une recherche locale est appliquée à chaque individu de la population. Comme déjà mentionné précédemment, l'application d'une recherche locale pour chaque individu de la population est coûteuse et peut gaspiller des ressources sans fournir d'informations plus utiles. La solution efficace est donc de considérer qu'une partie (fraction) des chromosomes de la population définie comme la probabilité de la recherche locale. De notre expérience et de la littérature, il semble difficile de définir la fraction idéale pour un problème donné afin d'obtenir les meilleures performances pour l'algorithme hybride. Néanmoins, deux principes généraux peuvent être indiqués :

- La probabilité de la recherche locale  $P_{local}$  doit être faible dans les phases initiales et incrémentée dans les générations suivantes. Dans les premières étapes, la diversité de la population assure un bon échantillon de l'espace de recherche et l'algorithme ne nécessite pas beaucoup d'aide de la part de la recherche locale.
- Une utilisation plus sélective de la recherche locale pourrait améliorer l'efficacité des algorithmes hybrides.

Ces principes généraux ont été pris en considération et adaptés à notre algorithme.

- L'approche locale est activée lorsque la valeur de la fitness maximale dépasse un certain seuil défini par notre savoir-faire dans le domaine. Cela limite le rôle de l'approche locale à la partie convergence, au moins dans le cas où le meilleur chromosome est prometteur.
- Elle peut également être activée lorsque l'algorithme stagne (pas d'évolution de la fitness maximale) après un nombre donné d'itérations  $I_s$ , la fourchette raisonnable étant d'environ 50 itérations pour notre type de problème. Nous pouvons seulement dire que si  $I_s$  est trop petit ( $<10$ ) peut fausser le processus d'évolution de l'AG, alors un trop grand  $I_s$  ( $> 100$ ) réduit son profit. En tout cas, cela donne à l'approche locale un rôle de diversification.

Afin de surmonter la difficulté d'estimer la fraction optimale de la population qui devrait effectuer une recherche locale, nous suggérons d'utiliser une sélection de chromosomes plus intelligente. Dans l'hypothèse que les différents bassins d'attraction peuvent être identifiés, même grossièrement, nous suggérons de choisir automatiquement un ou plusieurs individus représentatifs dans le même bassin afin de subir l'approche locale. Tout en améliorant la performance, ce processus favorise la diversité et aide à lutter contre le problème de la convergence prématurée. Grâce à cette approche sélective, une valeur très faible de  $P_{local}$  peut être fixée à n'importe quel moment de l'état génétique. La performance dépend de la disponibilité d'avoir un algorithme de "sélection" qui n'est pas coûteux en calcul. Les ingrédients de l'algorithme de sélection sont dépendants du problème, et deux éléments

doivent être précisés : la métrique de similarité qui permet la comparaison entre les chromosomes et la définition du bassin d'attraction.

Cette mesure de similarité peut être basée sur la distance Euclidienne pour les génotypes codés par des réels, ou la distance de Hamming pour les génotypes avec une représentation binaire. Plus petite est la distance entre deux individus, plus ils sont similaires. Comme de nombreux calculs de distance sont nécessaires, le choix doit prendre en compte la pertinence, mais aussi le temps consommé pour effectuer les calculs. Pour la conception des filtres, les chromosomes sont représentés par des patterns de grandes dimensions. Cela limite fortement les comparaisons génétiques. Pour accélérer les calculs sans perte de performance, nous suggérons de travailler à partir de deux éléments complémentaires : les gènes et les valeurs de fitness des chromosomes.

Les chromosomes avec des gènes différents peuvent avoir des valeurs de fitness similaires. Mais deux chromosomes avec des gènes similaires sont proches les uns des autres et donc donnent toujours des valeurs de fitness semblables. Ainsi, pour les chromosomes avec des valeurs similaires de fitness, nous avons sélectionné (via un AG dédié) les gènes dont les chromosomes sont les plus proches. En d'autres termes, lorsque deux chromosomes ont des valeurs de fitness similaires et sont représentés par des patterns similaires dans l'espace de dimension réduite, ils sont certains d'être proches les uns des autres.

Lorsque les deux gènes avec les positions 1 et  $N + 1$  du premier et du deuxième chromosome sont similaires, nous pouvons conclure que les deux chromosomes sont identiques. Cette approche a montré son efficacité et est suffisante pour notre problème.

Dans le cas d'un filtre symétrique 2D (à coefficients symétriques), il suffit de vérifier les quatre valeurs extérieures ((1,1), (1,  $P_2+1$ ), ( $P_1+1$ , 1) et ( $P_1+1$ ,  $P_2+1$ )) afin de comparer les gènes des deux chromosomes et de détecter les similitudes entre les chromosomes (voir figures 3.4 et 3.5).

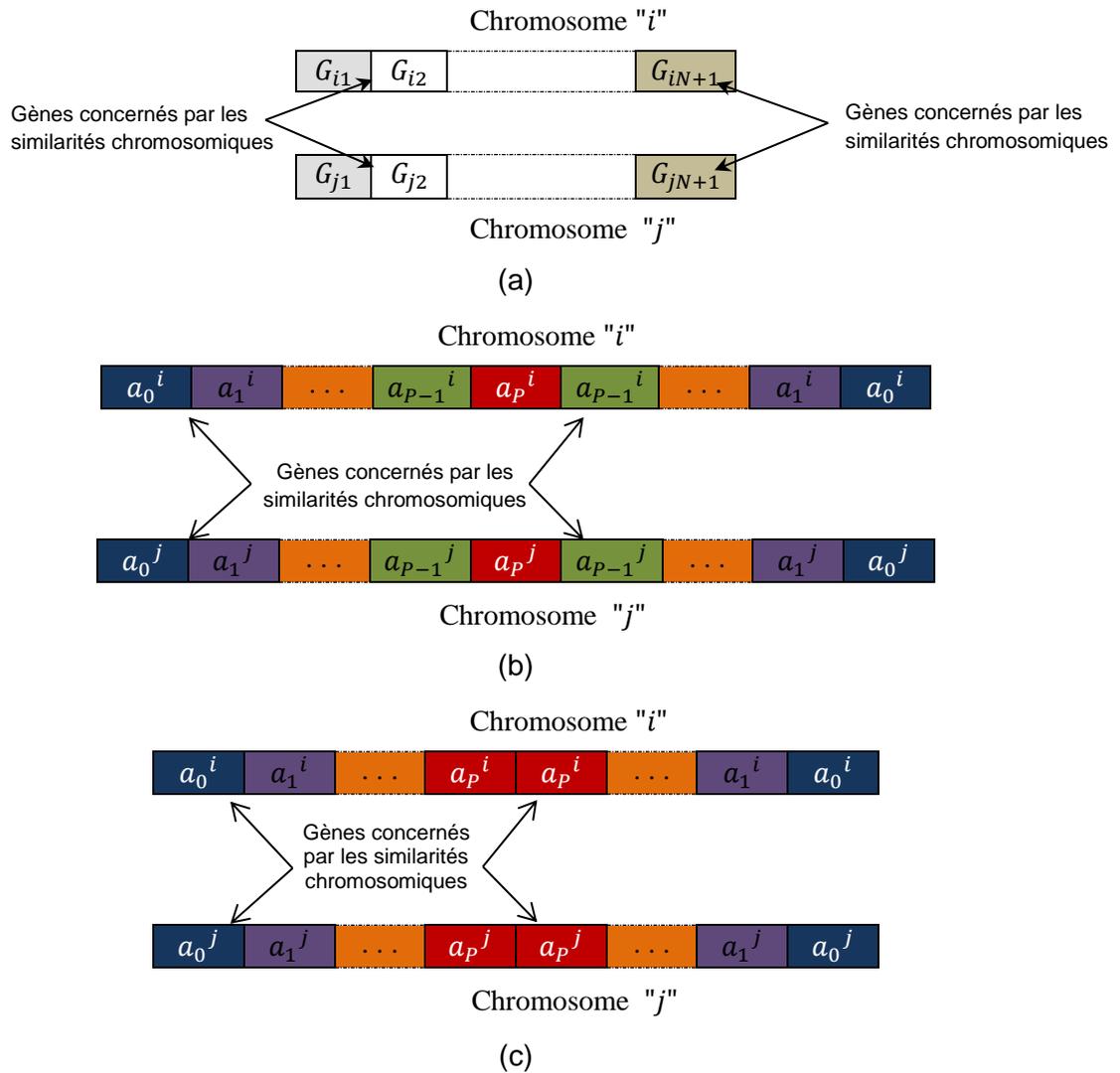


Figure 3.4 – Gènes concernés par les similarités chromosomiques pour un filtre 1D  
 (a) : RIF non symétrique, (b) : RIF avec symétrie impaire, (c) : RIF avec symétrie  
 paire

$a_{00}^i$	$a_{01}^i$	$\dots$	$a_{0P_2}^i$	$a_{0P_2}^i$	$\dots$	$a_{01}^i$	$a_{00}^i$
$a_{10}^i$	$a_{11}^i$	$\dots$	$a_{1P_2}^i$	$a_{1P_2}^i$	$\dots$	$a_{11}^i$	$a_{10}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{P_10}^i$	$a_{P_11}^i$	$\dots$	$a_{P_1P_2}^i$	$a_{P_1P_2}^i$	$\dots$	$a_{P_11}^i$	$a_{P_10}^i$
$a_{P_10}^i$	$a_{P_11}^i$	$\dots$	$a_{P_1P_2}^i$	$a_{P_1P_2}^i$	$\dots$	$a_{P_11}^i$	$a_{P_10}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{10}^i$	$a_{11}^i$	$\dots$	$a_{1P_2}^i$	$a_{1P_2}^i$	$\dots$	$a_{11}^i$	$a_{10}^i$
$a_{00}^i$	$a_{01}^i$	$\dots$	$a_{0P_2}^i$	$a_{0P_2}^i$	$\dots$	$a_{01}^i$	$a_{00}^i$

$a_{00}^j$	$a_{01}^j$	$\dots$	$a_{0P_2}^j$	$a_{0P_2}^j$	$\dots$	$a_{01}^j$	$a_{00}^j$
$a_{10}^j$	$a_{11}^j$	$\dots$	$a_{1P_2}^j$	$a_{1P_2}^j$	$\dots$	$a_{11}^j$	$a_{10}^j$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{P_10}^j$	$a_{P_11}^j$	$\dots$	$a_{P_1P_2}^j$	$a_{P_1P_2}^j$	$\dots$	$a_{P_11}^j$	$a_{P_10}^j$
$a_{P_10}^j$	$a_{P_11}^j$	$\dots$	$a_{P_1P_2}^j$	$a_{P_1P_2}^j$	$\dots$	$a_{P_11}^j$	$a_{P_10}^j$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{10}^j$	$a_{11}^j$	$\dots$	$a_{1P_2}^j$	$a_{1P_2}^j$	$\dots$	$a_{11}^j$	$a_{10}^j$
$a_{00}^j$	$a_{01}^j$	$\dots$	$a_{0P_2}^j$	$a_{0P_2}^j$	$\dots$	$a_{01}^j$	$a_{00}^j$

(a)

$a_{00}^i$	$a_{01}^i$	$\dots$	$a_{0P_2}^i$	$\dots$	$a_{01}^i$	$a_{00}^i$
$a_{10}^i$	$a_{11}^i$	$\dots$	$a_{1P_2}^i$	$\dots$	$a_{11}^i$	$a_{10}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{P_10}^i$	$a_{P_11}^i$	$\dots$	$a_{P_1P_2}^i$	$\dots$	$a_{P_11}^i$	$a_{P_10}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{10}^i$	$a_{11}^i$	$\dots$	$a_{1P_2}^i$	$\dots$	$a_{11}^i$	$a_{10}^i$
$a_{00}^i$	$a_{01}^i$	$\dots$	$a_{0P_2}^i$	$\dots$	$a_{01}^i$	$a_{00}^i$

$a_{00}^j$	$a_{01}^j$	$\dots$	$a_{0P_2}^j$	$\dots$	$a_{01}^j$	$a_{00}^j$
$a_{10}^j$	$a_{11}^j$	$\dots$	$a_{1P_2}^j$	$\dots$	$a_{11}^j$	$a_{10}^j$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{P_10}^j$	$a_{P_11}^j$	$\dots$	$a_{P_1P_2}^j$	$\dots$	$a_{P_11}^j$	$a_{P_10}^j$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{10}^j$	$a_{11}^j$	$\dots$	$a_{1P_2}^j$	$\dots$	$a_{11}^j$	$a_{10}^j$
$a_{00}^j$	$a_{01}^j$	$\dots$	$a_{0P_2}^j$	$\dots$	$a_{01}^j$	$a_{00}^j$

(b)

Figure 3.5 – Gènes concernés par les similarités chromosomiques pour un filtre 2D  
 (a) : RIF avec symétrie paire, (b) : RIF avec symétrie impaire

Pour la conception des filtres RIF, deux individus sont dits "identiques" s'ils ont la même valeur de la fonction fitness et les valeurs des gènes sélectionnés sont identiques, c'est à dire

leur différence est inférieure à un certain seuil exprimé en pourcentage de la moyenne des deux gènes.

Soient  $a^i, a^j$  les gènes des chromosomes  $i$  et  $j$ ,  $fit(i)$  et  $fit(j)$  leur fitness,  $\varepsilon_f$ ,  $\varepsilon_{g1}$  et  $\varepsilon_{gN+1}$  les seuils pour les comparaisons des fitness et des gènes.

$$\Delta fit \leq \varepsilon_f$$

$$\Delta G(1) \leq \varepsilon_{g1}$$

$$\Delta G(P + 1) \leq \varepsilon_{gN+1}$$

$\Delta$  représente l'opérateur de différence absolue, et  $\varepsilon_f$  est fixé à un taux de 0.5% de la fitness moyenne des deux chromosomes  $i$  et  $j$ . De même,  $\varepsilon_{g1}$  et  $\varepsilon_{gN+1}$  sont fixés à 1% de la valeur moyenne des gènes.

Le bassin d'attraction peut être considéré comme une niche ou une zone présentant des caractéristiques communes. Il est extrêmement difficile d'estimer exactement le nombre idéal de niches et de leurs bassins associés. Si les niches sont trop petites, le nombre de niches sera initialement égal approximativement au nombre de chromosomes. L'algorithme va alors prendre beaucoup plus de temps à converger si toutes les niches sont considérées, par ailleurs certaines d'entre elles doivent être rejetées (écartées). Si les niches sont trop grandes; donc autant d'espace de recherche à englober (recouvrir), alors qu'elles ne présentent aucun intérêt. Ces deux situations sont à éviter, et par conséquent nous suggérons une approche différente qui n'est pas limitée par ces paramètres. Les entrées disponibles sont notre savoir-faire dans le domaine et le nombre maximum de chromosomes qui subissent l'approche locale via la probabilité de l'approche locale. Notre savoir-faire permet de décider si deux chromosomes sont similaires, et s'ils appartiennent donc à la même niche. La distance  $d_c$  reflète cette proximité entre deux chromosomes. Une entrée supplémentaire est liée à la pertinence minimum d'une niche : une niche pour être pertinente, elle doit être une représentativité minimale en termes de densité de chromosome, elle doit couvrir au moins  $k_r$  proches chromosomes voisins. Les valeurs aberrantes doivent être évitées.

L'algorithme est basé sur un principe de double distance et le concept de voisinage. Il vise à trouver des chromosomes de manière itérative pour couvrir l'espace de recherche en respectant certaines règles. Chaque chromosome  $C_i$  sélectionné est fixé à un ensemble (lot) de chromosomes qui sont plus proches. Le voisin  $C_j$  parmi les chromosomes attachés est définitivement rattaché en tant que voisin si  $dist(C_j, C_i) < d_c$  où  $dist(C_j, C_i) = \|C_j, C_i\|$  est la norme Euclidienne.

Le sous-ensemble de chromosomes  $C_j$  définit la région d'influence de  $C_i$ . Chaque nouveau chromosome candidat pour la sélection doit être à la fois loin de ceux de la référence existante, et en dehors de leurs régions d'influence qui inhibent la sélection des chromosomes supplémentaires. A chaque étape, le nouveau chromosome est le pattern le plus éloigné des références existantes. L'idée de l'algorithme consiste à déterminer automatiquement un sous-ensemble de chromosomes qui est plus grand que le nombre nécessaire pour l'approche locale et inférieure à une limite supérieure  $It_{max}$ . Les chromosomes sélectionnés sont alors déterminées par l'examen de ce sous-ensemble.

Supposons qu'une niche doit couvrir au moins  $k_r$  proches voisins. On peut en déduire le nombre maximal correspondant de niches sur l'hypothèse d'une distribution uniforme par :

$$\text{Max}_n = \text{Size}_{\text{pop}} / k_r \quad (3.20)$$

Le nombre d'itérations  $\text{It}_{\text{max}}$  est conditionné par le nombre de chromosomes désirés et la valeur de  $k_r$ . Par approximation à une distribution symétrique, on en déduit la taille moyenne et le nombre maximum d'itérations  $\text{It}_{\text{max}}$ .

$$\text{It}_{\text{max}} = 0.5 * (\text{Max}_n + \text{Min}_n) \quad (3.21)$$

$\text{It}_{\text{max}}$  est la limite supérieure des itérations de l'algorithme, mais il peut être arrêté avant automatiquement lorsque le sous-ensemble candidat est vide. Par exemple, en présence d'un superbe chromosome,  $d_c$  est plus actif : il peut inactiver de nombreux chromosomes, car ils sont similaires à la référence. Le nombre final des étapes est alors dépendant de la distribution des chromosomes eux-mêmes.

L'algorithme est très rapide car il consiste seulement à calculer à chaque étape  $i$ , les distances entre les chromosomes candidats et la dernière distance est choisie. Il convient de souligner qu'on n'a pas besoin de calculer les  $k_r$  voisins les plus proches à chaque étape.

Cette procédure peut être comparée aux algorithmes de type clustering issus de la "méthode de colline" [Pérez et al., 2012], mais elle n'implique pas l'ajustement de tous les paramètres influents et est moins coûteuse.

A l'exécution, l'algorithme a besoin de  $k_r$ ,  $d_c$  et  $P_{\text{local}}$ . Le nombre de chromosomes est généré automatiquement.

Soit  $T = [C_1, \dots, C_n]$  un ensemble de  $n$  chromosomes représentés par un pattern dans un espace de dimension  $p$ ,  $C_b$  le chromosome ayant la meilleure fitness,  $C_{cu}$  le dernier chromosome sélectionné,  $ST_d(A)$  l'ensemble des éléments  $j$  tel que  $\text{dist}(A, C_j) < d_c$  et  $SI$  l'union de ces éléments pour les chromosomes de référence.

L'objectif est de trouver  $S = \{C_{\sigma_1}, \dots, C_{\sigma_m}\} \in \mathbb{R}^p$  l'ensemble des chromosomes de référence.

L'algorithme peut être résumé comme suit :

**Algorithme de sélection** [Input ( $T, P_{local}, k_r, d_c$ ), Ouput: ensemble de chromosomes  $S$ ]

---

Etape 0: Premier chromosome

- Calculer  $It_{max}$
- $S = \{C_{\sigma 1}\}$  où  $C_{\sigma 1} = C_{b_i}$
- $C_{Cu} = C_b$
- $SI = \emptyset$
- $N_{ref} = 1$

Etape 1: Déterminer les chromosomes influents de chaque référence de chromosomes

Pour chaque  $C_k \in \{T - S - SI\}$

- (a) calculer  $d(C_{cu}, C_k)$
- (b) Trouver son plus proche voisin  $y_k(1nn)$  dans  $S$  :  $y_k(1nn) = \min d(C_k, C_{\sigma l})$  pour  $l \in [1, N_{ref}]$
- (c) Attribuer  $C_k$  à  $y_k(1nn)$  et sauvegarder la distance liée
- (d)  $SI = SI + ST_d(C_{Cu})$
- Mettre à jour le sous-ensemble de patterns  $T_i$  de  $T$  attaché à chaque chromosome de référence  $i$ .

Etape 2: Déterminer le nouveau chromosome de référence

Pour chaque  $C_{\sigma k} \in S$ ,

- Trouver son plus loin chromosome influent  $Ok_f / d(C_{\sigma k}, Ok_f) = \max d(C_{\sigma k}, C_i)$  pour  $C_i \in T_k$
- Sauvegarder  $d_k(C_{\sigma k})$  la distance entre  $C_{\sigma k}$  et  $Ok_f$

Etape 3: - Déterminer le chromosome gagnant  $O_{wf} / d_w(C_{\sigma w}) = \max (d_k(C_{\sigma k})) k \in [1, N_{ref}]$

- Déduire le chromosome gagnant  $x_w = O_{wf}$
- $C_{Cu} = O_{wf}$
- $S = S + x_w$
- $N_{ref} = N_{ref} + 1$

Etape 4: Si  $N_{ref}$  égale  $It_{max}$  ou  $T - S - SI = \emptyset$  Fin sinon aller à l'étape 1.

---

**Table 3.3 – Algorithme de sélection pour la recherche locale.**

L'algorithme détermine automatiquement  $N_{ref}$  chromosomes en  $It_{max}$  itérations ou moins. Chaque chromosome est sélectionné lors d'une itération donnée  $i$  et est attaché à un ensemble de chromosomes voisins de taille  $w(i)$ .  $It(i)$  définit le numéro de l'itération où il a été détecté. La probabilité de la recherche locale  $P_{local}$  définit le nombre maximal  $N_{max}$  de chromosomes utilisés par la recherche locale. Si  $N_{ref} < N_{max}$ , tous les chromosomes de  $N_{ref}$  sont soumis à la recherche locale. Dans le cas contraire, les chromosomes sont triés (classés) sur la base d'une fonction d'agrégation définie ci-dessous. Cette fonction  $S$  est composée de trois termes. Le premier concerne la qualité de la solution via la fonction de fitness. Le second concerne la diversité et donne plus d'importance aux chromosomes sélectionnés au cours des premières itérations car ils sont plus éloignés. Le troisième est lié à la densité et encourage les voisinages denses.

$$S(i) = \alpha * (fit(i) / fit_{max}) + \beta * (It_{max} - It(i)) / It_{max} + \gamma * (w(i) / w_{max}) \quad (3.22)$$

Où  $\alpha$ ,  $\beta$  et  $\gamma$  sont les coefficients de pondération associés. Dans notre problème, comme il n'y a pas de règle qui donne plus d'importance à un critère particulier, nous les avons fixés au même niveau.

### 3.5.4. Durée de la recherche locale

Quelques études ont été menées afin de déterminer la durée optimale de la recherche locale. Différentes durées se sont avérées produire des performances similaires et les résultats obtenus ont une applicabilité limitée. Même si quelques chromosomes sont concernés par l'approche locale via l'utilisation de notre approche de sélection, limiter le coût de calcul est très important. La limitation est motivée par les progrès de la génétique et s'ajuste dynamiquement lors de l'avance du processus local lui-même.

La durée de l'approche locale est limitée par le nombre d'itérations pour la Hill Climbing et la recherche tabou, et par la valeur de la température initiale et finale pour le recuit simulé. Cette durée est directement reliée au potentiel de chaque chromosome concerné par l'approche locale, par exemple, évaluation de sa fitness. En dessous d'un seuil de fitness bas, la durée est égale à zéro, au-dessus d'un seuil haut, la durée est limitée, et entre celles-ci la durée suit une règle linéaire simple.

En outre, le procédé est interrompu si, pendant l'activation de l'approche locale, la fitness maximale de la population ne change pas après un nombre donné d'itérations. Le procédé est également arrêté si la valeur de la fitness souhaitée est atteinte. Seul un petit nombre de chromosomes subissent l'approche locale avec notre processus de sélection. Cela rend plus facile la durée problématique. La durée de notre processus est conditionnée uniquement par des critères liés aux progrès de la génétique et non par un moyen artificiel pour réduire le temps de calcul sans contrôler l'effet des procédures locales.

## 3.6. Résumé de l'approche AGA

Dans l'objectif de résumer les procédés nouveaux mis en œuvre dans l'AGA élaboré dans ce travail, une synthèse mentionnant les différents mécanismes, une définition des tâches propres à chaque opérateur, ainsi qu'une connaissance des différences avec l'AG standard sont nécessaires. Il est en outre indispensable de consulter la section 3.2 pour prendre connaissance des formules permettant d'activer certains processus et de calculer les différents paramètres.

Pour chaque mode opérationnel (chaque étape et processus de l'AG), l'algorithme exploite les informations de la configuration de la population et de son évolution pour adapter et optimiser les différents mécanismes de l'algorithme. Cette adaptation est alors optimisée pour le problème de recherche des solutions, par une étroite collaboration entre l'opérateur de mutation et de sélection.

Les informations dont l'algorithme exploite pour adapter ses paramètres et ses mécanismes sont (les détails sont données dans la section 3.2) :

- Définition et détection des signes de similarité entre les chromosomes
- Détection de la présence de minimum local
- Détection de la convergence prématurée

En regardant attentivement les deux derniers points, il est difficile, à première vue, de faire la différence entre la détection d'une convergence prématurée et la présence d'un minimum local. Cette distinction devient plus claire en consultant les équations 3.12 (vérification de la présence de minimum local) et 3.15 (vérification de la convergence prématurée).

Afin de développer une synthèse exploitable des mécanismes, l'accent doit particulièrement être mis sur les nouveautés de l'AGA par rapport à l'AG standard (tableau 3.4).

AG standard	AG adaptatif (AGA)
- Aléatoire	<p><b>Initialisation</b></p> <p>- Pseudo-aléatoire (biaisée) : optimisation de la moyenne et de l'écart type de la population initiale.</p>
- Standard	<p><b>Croisement</b></p> <p>- Directionnel : les descendants générés par le croisement dépendent de la fitness des parents sélectionnés à l'opérateur de croisement.</p>
- Standard	<p><b>Mutation</b></p> <p>- Non-uniforme : le taux de mutation dépend de la fitness de l'individu sélectionné à la mutation, de la présence de minimum local et dépend aussi de la diversification de la population.</p> <p>- Adaptative : l'amplitude de la mutation dépend de la diversification de la population et de la présence de minimum local.</p>
- Standard	<p><b>Sélection</b></p> <p>- Diversification : les individus similaires à ceux déjà sélectionnés sont écartés même s'ils ont des valeurs de fitness élevées.</p> <p>- Rupture : rafraîchissement de la population dès détection de convergence prématurée.</p>

Table 3.4 – Comparatif entre les opérateurs de l'AG et de l'AGA et synthèse des mécanismes de l'AGA.

Toutes les définitions citées jusqu'à maintenant ne concernent que la collecte d'informations sur l'évolution (des données d'évolution), mais l'algorithme contient aussi des mécanismes permettant de traiter ces données pour activer ou désactiver les procédures de diversification et de rupture.

### 3.7. Conclusion

Dans ce chapitre, nous avons présenté les différents algorithmes des techniques utilisées pour une synthèse optimale des filtres RIF 1D et 2D. Notre première approche (algorithme génétique adaptatif dédié à la synthèse des filtres RIF) vise à réduire simultanément le critère d'erreur et le temps CPU nécessaire à la convergence. Plus intéressant encore, nos mécanismes offrent une grande souplesse à l'AG, contribuant ainsi à sa facilité d'utilisation qui est le vrai défi aujourd'hui. Ceci permet à la majorité des paramètres d'être pré-calibrés, laissant seulement un petit ensemble de paramètres à régler par l'utilisateur. L'utilisateur peut créer son propre filtre avec confiance, car il n'est pas confronté à toute la complexité du calcul évolutif.

La seconde technique élaborée est l'optimisation par essaim de particules, en incluant deux nouvelles versions améliorées de cette technique. Un avantage attrayant de la PSO est la facilité de sa mise en œuvre à la fois dans le contexte de codage et de sélection des paramètres. L'algorithme est beaucoup plus simple et intuitif à mettre en œuvre que les opérateurs de sélection et de mutation dans un algorithme génétique. Nous pensons qu'il y a un potentiel à améliorer les versions de PSO en intégrant des mécanismes plus intelligents et en l'hybridant avec notre algorithme génétique adaptatif. Cette hybridation a pour but d'améliorer l'analyse de l'état de l'évolution de la recherche des coefficients optimaux du filtre à concevoir.

L'algorithme hybride proposé est essentiellement une combinaison de notre algorithme génétique adaptatif et des approches locales de sorte que nous pouvons utiliser efficacement la capacité de recherche de l'AG, ainsi que les capacités de recherche localisée d'une approche locale. En introduisant des mécanismes spécifiques et en optimisant le rôle de la recherche locale, notre processus d'évolution peut être accéléré sans perte de précision. En particulier, la capacité de l'algorithme pour déterminer le moment approprié pour appliquer les mécanismes de l'approche locale et de diversification locale est très efficace dans l'optimisation des coefficients de filtre.

---

# Chapitre IV

---

## 4. Résultats expérimentaux

---

4.	Résultats expérimentaux .....	73
4.1.	Application des algorithmes évolutionnaires pour la synthèse des filtres RIF 2D 74	
4.1.1.	Synthèse d'un filtre RIF symétrique passe-bas .....	75
4.1.2.	Résultats avec d'autres filtres RIF .....	82
4.1.3.	Comparaison spécifique avec deux méthodes génétiques populaires 84	
4.1.4.	D'autres résultats (tests) avec des filtres de dimensions plus élevées 86	
4.1.5.	Discussion, étude statistique et de convivialité .....	86
4.2.	PSO pour la synthèse des filtres RIF 1D .....	89
4.2.1.	Réglage des paramètres .....	90
4.2.2.	Exemples de synthèse .....	92
4.2.3.	Etude statistique .....	98
4.2.4.	Discussion et perspectives .....	100
4.2.5.	Hybridation préliminaire de l'AGA avec la NPSO .....	101
4.3.	Algorithme hybride pour la synthèse des filtres RIF 1D et 2D .....	103
4.3.1.	Performance de notre système évolutionnaire .....	104
4.3.2.	Réglage des paramètres .....	104
4.3.2.1.	Paramètres génétiques de base .....	104
4.3.2.2.	Paramètres de domaine .....	105
4.3.2.3.	Paramètres de l'approche locale .....	105
4.3.3.	Résultats préliminaires relatifs aux composantes de l'algorithme 106	
4.3.3.1.	Effet de l'initialisation .....	106
4.3.3.2.	Coût de la recherche locale .....	108
4.3.4.	Alternatives d'hybridation .....	109
4.3.5.	Comparaisons avec d'autres approches .....	113
4.3.6.	Application de l'algorithme hybride pour la synthèse d'un filtre RIF 2D 118	
4.4.	Résumé des performances pour la synthèse d'un filtre RIF 2D (Exemple 1, section 4.1 et 4.3.6) .....	122
4.5.	Quantification .....	123
4.6.	Contributions de la recherche .....	124
4.7.	Application des filtres conçus pour le traitement d'images .....	125
4.7.1.	Exemple de filtrage passe-bas et passe-haut .....	125
4.7.2.	Ajout d'un bruit à une image et élimination de ce bruit par filtrage passe-bas 129	
4.8.	Conclusion .....	132

---

La fonction de transfert d'un filtre RIF s'approche de l'idéal quand l'ordre du filtre augmente, ce qui accroît la complexité et la quantité du temps nécessaire pour sa synthèse. Pour résoudre un problème d'optimisation complexe, le moyen le plus efficace est de concevoir un algorithme en mesure de convenir aux besoins spécifiques. Par conséquent, il est nécessaire de chercher des méthodes plus innovantes, capables de surmonter ces faiblesses.

#### 4.1. Application des algorithmes évolutionnaires pour la synthèse des filtres RIF 2D

Dans cette section, nous cherchons à démontrer expérimentalement l'efficacité de notre approche lorsqu'elle est appliquée sur divers cas. Différentes versions de notre approche AGA (Algorithme Génétique Adaptatif) sont évaluées pour souligner le rôle des différents mécanismes mis en place. AG<sub>1</sub> comprend tous les nouveaux mécanismes intégrés dans notre version initiale AG<sub>4</sub> (AG standard). Les composantes de chaque version sont détaillées dans le tableau 4.1.

Des comparaisons sont faites avec d'autres algorithmes. Ces algorithmes sont bien connus : les approches conventionnelles, ainsi que les approches évolutionnaire. Parmi les approches évolutionnaires, quatre approches non-génétiques sont utilisées dans leurs versions initiales (standards), leurs paramètres sont répertoriés dans le tableau 4.2. Parmi les approches conventionnelles, deux d'entre elles basées sur l'algorithme de Remez et des moindres carrés ont été testées. Pour ces deux méthodes, le filtre 1D a été synthétisé et puis nous avons utilisé la transformation de fréquence pour concevoir le filtre 2D. Une attention particulière a été accordée aux deux approches génétiques populaires souvent citées dans des documents récents et développées spécifiquement pour la conception de filtres.

Approches	Paramètres	
<b>AG Adaptatif</b> AG <sub>1</sub>	<ul style="list-style-type: none"> <li>• Mutation adaptative Non-uniforme avec la procédure de diversification</li> </ul>	$k_1 = 0.1\%, k_2 = 1\%$
	<ul style="list-style-type: none"> <li>• Présence de minimum local</li> </ul>	$k_1 = 10\%, k_2 = 30\%$
	<ul style="list-style-type: none"> <li>• Processus de rafraîchissement actif</li> </ul>	$M_p = 40\%$
	<ul style="list-style-type: none"> <li>• Croisement basé direction</li> </ul>	$C_R = 100\%$
<b>AG Adaptatif</b> AG <sub>2</sub>	<ul style="list-style-type: none"> <li>• Sélection adaptative avec le processus de rafraîchissement (procédure de diversification et de rupture)</li> </ul>	
	<ul style="list-style-type: none"> <li>• Mutation non-uniforme</li> </ul>	$M_p = 0.5\%$
	<ul style="list-style-type: none"> <li>• Processus de rafraîchissement actif</li> </ul>	$M_p = 40\%$
	<ul style="list-style-type: none"> <li>• Croisement basé direction</li> </ul>	$C_R = 100\%$
<b>AG Adaptatif</b> AG <sub>3</sub>	<ul style="list-style-type: none"> <li>• Sélection adaptative avec le processus de rafraîchissement (procédure de diversification et de rupture)</li> </ul>	
	<ul style="list-style-type: none"> <li>• Mutation adaptative Non-uniforme avec la procédure de diversification</li> </ul>	$k_1 = 0.1\%, k_2 = 1\%$
	<ul style="list-style-type: none"> <li>• Présence de minimum local</li> </ul>	$k_1 = 10\%, k_2 = 30\%$
	<ul style="list-style-type: none"> <li>• Croisement basé direction</li> </ul>	$C_R = 100\%$
<b>AG Standard</b> AG <sub>4</sub>	<ul style="list-style-type: none"> <li>• Sélection basée sur un espace d'échantillons élargi</li> </ul>	
	<ul style="list-style-type: none"> <li>• Mutation non-uniforme</li> </ul>	$M_p = 0.5\%$
	<ul style="list-style-type: none"> <li>• Croisement basé direction</li> </ul>	$C_R = 100\%$

Table 4.1 – Description et paramètres des différentes versions de l'AG.

Approches		Paramètres		
<b>Simulated Annealing</b>	Choix aléatoire de la solution initiale $x_0$ dans $S$ .	Températures initiale et finale: $T_0 = 2000 T_F = 10^{-5}$	$k_B = 1$	$Pr(\text{accepté}) = \exp(-\Delta E / k_B \cdot T)$
<b>Tabu search</b>	Choix aléatoire de la solution initiale $x_0$ dans $S$ .	Liste tabou initiale : $T = \emptyset$	Taille de la liste tabou $T : STL = 80$	Taille du sous ensemble : $SS = 100$
<b>Hill-climbing</b>	Choix aléatoire de la solution initiale $x_0$ dans $S$ .	$k = n_1 * n_2$		Taille du sous ensemble : $SS = 100$

Table 4.2 – Paramètres de chaque algorithme évolutionnaire.

Finalement, l'ensemble des algorithmes est :

1. AG<sub>1</sub>
2. AG<sub>2</sub>
3. AG<sub>3</sub>
4. AG<sub>4</sub>
5. Simulated Annealing
6. Tabu Search
7. Hill Climbing
8. Méthodes de Remez et des moindres carrés (Transformation Fréquentielle)
9. Echantillonnage Fréquentiel
10. Méthode des fenêtres
11. AG de Mastorakis [Mastorakis et al., 2003] réel (codage réel)
12. AG de Mastorakis [Mastorakis et al., 2003] binaire (codage binaire, version identique à leur étude initiale)
13. AG de Tzeng [Tzeng, 2007] (approche Shian-Tzan Tzeng)

Pour illustrer la répétabilité et l'efficacité de l'algorithme génétique adaptatif proposé, dix exécutions indépendantes ont été réalisées pour chaque exemple. Les simulations sont réalisées sous Matlab 7.8.0 (R2009a) en utilisant un processeur Intel Core i7.

#### 4.1.1. Synthèse d'un filtre RIF symétrique passe-bas

La réponse en amplitude du filtre passe-bas est donnée par l'équation suivante :

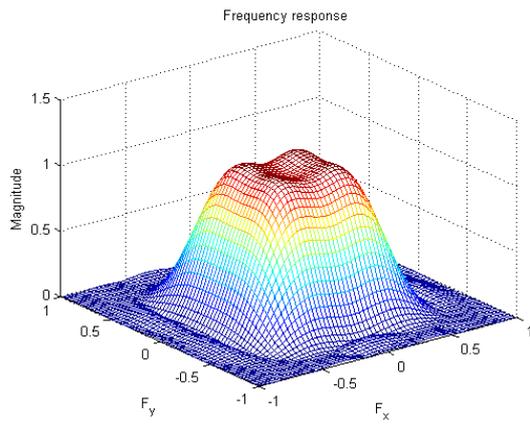
$$D(\omega_1, \omega_2) = \begin{cases} 1 & |\omega_{1,2}| < 0.5 \\ 0 & |\omega_{1,2}| \geq 0.5 \end{cases}$$

Le filtre est symétrique et de dimensions 7\*7.

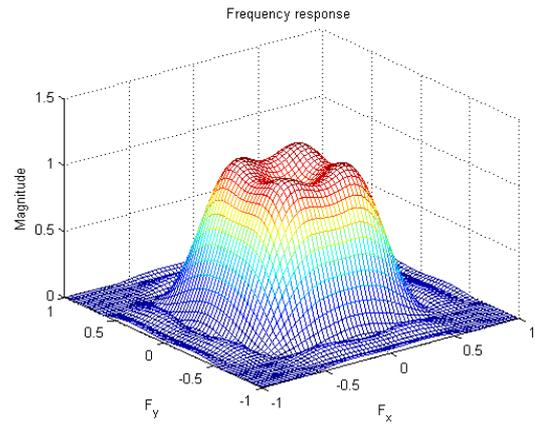
Dans cette simulation, nous avons testé AG<sub>1</sub> en variant l'ensemble des paramètres d'entrée afin de démontrer la flexibilité de l'AGA.

	Paramètres	Erreur moyenne	Erreur maximale	Temps CPU	Ite
AG Adaptatif AG <sub>1</sub>	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0092	0.0803	49.74s	298
	$C_R = 80\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0120	0.1019	56.73s	391
	$C_R = 40\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0122	0.1112	70.62s	628
	$C_R = 100\%$ , $M_P = 1\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0114	0.1013	73.39s	457
	$C_R = 100\%$ , $M_P = 80\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0099	0.0977	49.94s	306
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.01\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0108	0.0955	67.81s	425
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.9\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0099	0.0885	63.66s	398
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 10\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0109	0.1397	60.99s	383
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 0.2\%$ $W = 1$ in PB, $W = 5/8$ in SB	0.0104	0.0867	68.69s	426
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 7/8$ in SB	0.0121	0.0911	54.13s	331
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 6/8$ in SB	0.0147	0.1023	62.24s	375
	$C_R = 100\%$ , $M_P = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $W = 1$ in PB, $W = 4/8$ in SB	0.0132	0.1436	56.34s	340

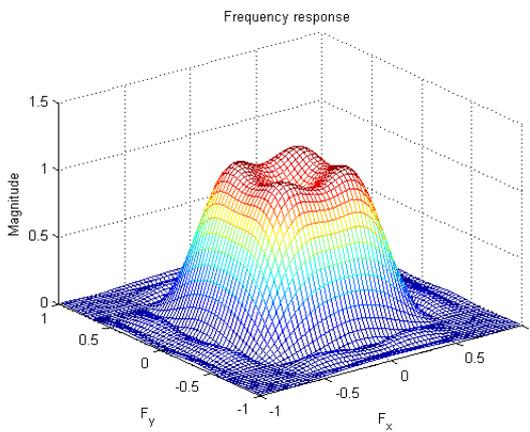
Table 4.3 – Réglage des paramètres de l'AGA.



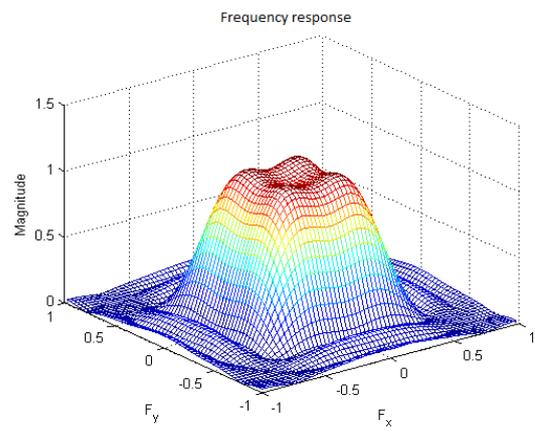
(a)



(b)



(c)



(d)

Figure 4.1 – Réponses fréquentielles (a) Adaptatif AG<sub>1</sub> (b) Adaptatif AG<sub>2</sub> (c) Adaptatif AG<sub>3</sub> (d) Standard AG<sub>4</sub>.

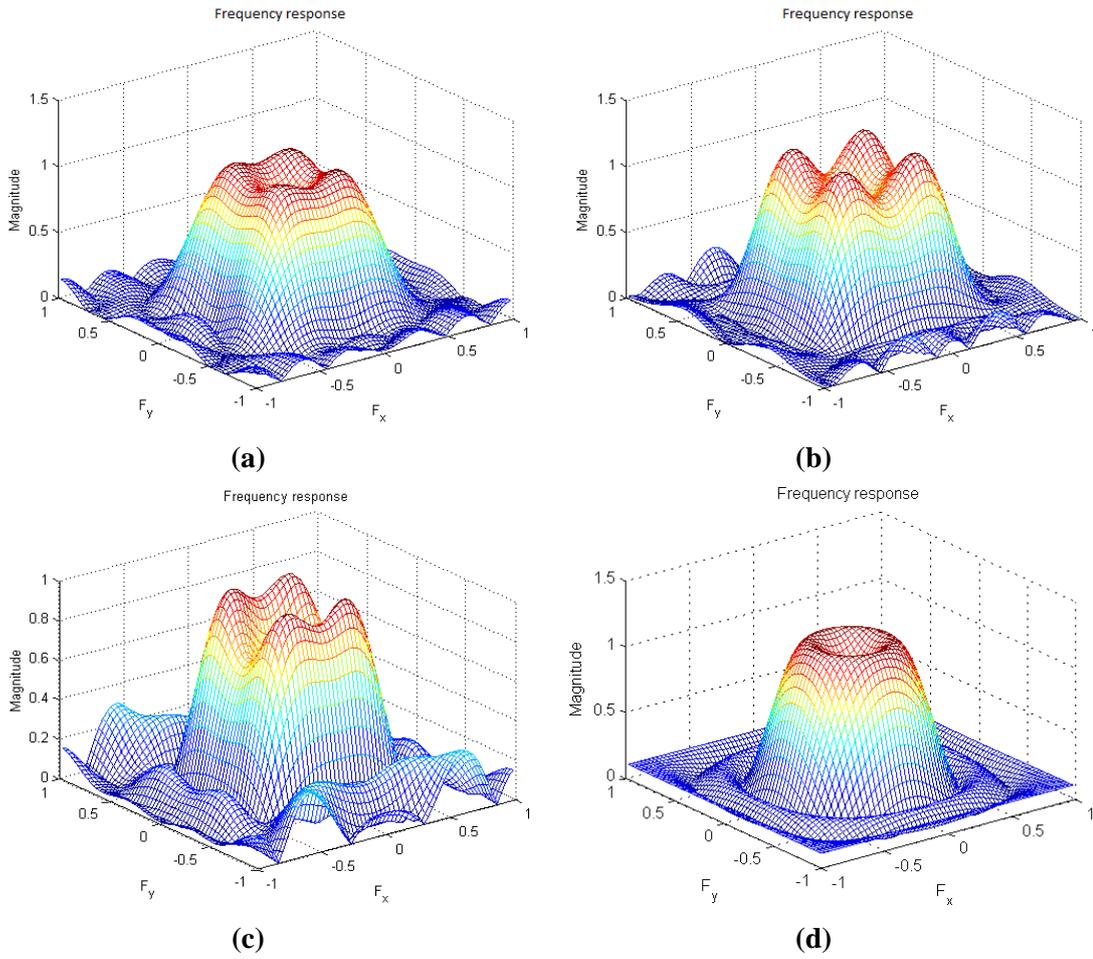


Figure 4.2 – Réponses fréquentielles : (a) Simulated Annealing (b) Tabu search (c) Hill-Climbing (d) Méthode de Remez (transformation fréquentielle).

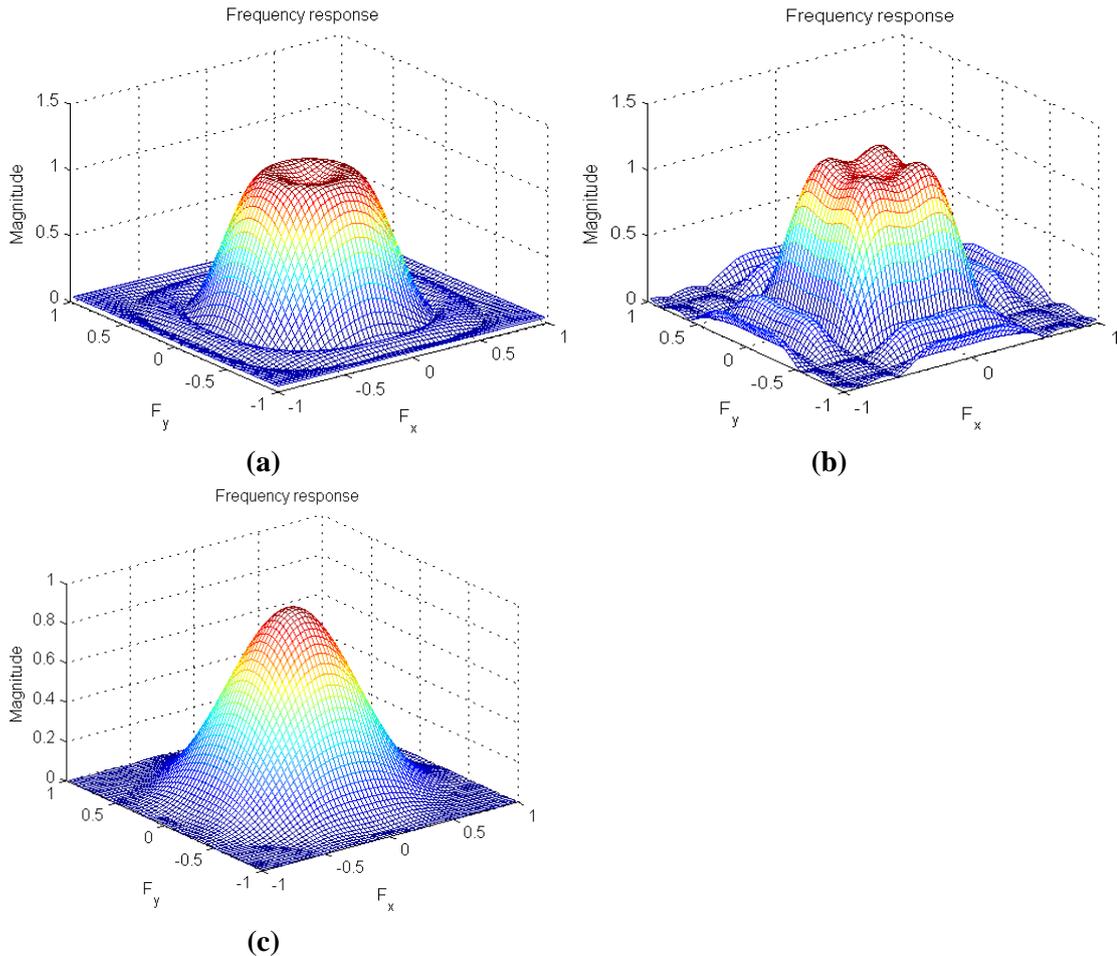


Figure 4.3 – Réponses fréquentielles : (a) Méthodes des moindres carrés (transformation fréquentielle) (b) Méthode de l'échantillonnage en fréquence (c) Méthodes des fenêtres.

Les principales caractéristiques de notre AG (de AG<sub>4</sub> à AG<sub>1</sub>) sont représentées dans le tableau 4.1. La première ligne résume les paramètres de notre AGA dans sa version complète (AG<sub>1</sub>) pour la synthèse de filtres RIF. C<sub>R</sub> et M<sub>p</sub> représentent respectivement le taux de croisement et la probabilité de mutation.

Les paramètres  $f_{\epsilon}$ ,  $g_{\epsilon}$ , et  $f_{end}$  ont été fixés respectivement à 0.00001, 0.0001 et 0.008.  $\Delta f_{min} = 0.07 * fit_{ave}$ ,  $fit_{ave}$  étant la fitness moyenne de l'ensemble de la population. Les paramètres  $k_1$  et  $k_2$  ont été fixés respectivement à 0.001 et 0.01 dans le cas d'une évolution normale de notre algorithme et à 0.1 et 0.3 lorsque le signe d'une convergence prématurée est détectée.

Les paramètres  $th_1$ ,  $th_2$  et  $th_3$  ont été respectivement fixées pour tous les essais à 0.01, 0.07 et 0.00001.

Les paramètres  $\lambda_{m1}$  et  $\lambda_{m2}$  ont été fixés à  $0.5 * randn$  et  $0.05 * randn$ , où  $randn$  représente un nombre aléatoire compris entre 0 et 1. Si le processus de rafraîchissement est actif, la probabilité de mutation a été fixée à  $M_p = 40\%$ .

Pour une comparaison équitable, nous ne considérons que les chromosomes initiaux qui ont été générés au hasard, le temps CPU est généralement réduit de 0% à 30% lorsque les

chromosomes initiaux sont constitués par d'autres solutions issues des méthodes conventionnelles (mutées ou non).

Dans l'AG et l'AGA (toutes les versions de l'AGA), la taille de la population  $P_S$  a été fixée à 100, le nombre maximal de générations génétiques a été fixé à 1500 et la probabilité de croisement ( $C_R$ ) à 100 %.

La matrice de pondération  $W$  est composée de deux valeurs:  $C_0 = 1$  dans la bande passante et  $C_1 = 5/8$  à l'extérieur de cette bande (dans la bande de transition et atténuée).

Les figures 4.1, 4.2 et 4.3 montrent les réponses en fréquence obtenues pour un filtre passe-bas de dimensions  $7 \times 7$  pour les différents algorithmes conventionnels et évolutionnaires introduits dans les sections précédentes.

La moyenne des résultats obtenus par chaque algorithme pour l'ensemble des données (critères) évaluées est indiquée dans le tableau 4.4, qui résume les performances caractérisées par cinq critères : l'erreur moyenne, l'erreur maximale, la largeur de bande de transition, le temps CPU et le nombre d'itérations. On peut constater que les résultats sont satisfaisants, mais plus important encore, que les variations entre les configurations sont relativement faibles.

Les erreurs des AGA proposées sont nettement inférieures à celles obtenues lorsque les autres techniques sont utilisées. Les erreurs maximales de notre approche  $AG_1$ , AG standard et SA sont respectivement 0.0803, 0.1086 et 0.1320.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
$AG_1$	0.0092	0.0803	0.113	49.74s	298
$AG_2$	0.0200	0.1055	0.166	75.63s	458
$AG_3$	0.0193	0.1092	0.142	59.62s	359
$AG_4$	0.0212	0.1086	0.170	205.42s	1380
Simulated Annealing	0.0255	0.1320	0.174	108s	
Tabu Search	0.0570	0.2190	0.139	660s	
Hill Climbing	0.0810	0.3850	0.121	Aléatoire	
Remez (Transformation Fréquentielle)	0.0566	0.1668	0.163	< 2s	
Moindres carrés (Transformation Fréquentielle)	0.0424	0.1412	0.193	< 2s	
Echantillonnage fréquentiel	0.0551	0.2203	0.115	< 2s	
Méthode des fenêtres	0.0400	0.4920	0.322	< 2s	

Table 4.4 – Paramètres de performance obtenus pour un filtre passe-bas.

Nous pouvons observer précisément dans les figures 4.1 et 4.2 que l'AG adaptatif et l'AG standard ont les meilleures réponses en amplitude pour les régions de la bande passante et de la bande d'arrêt. Les AGA ont des bandes de transition étroites par rapport aux filtre conçus par la technique SA et par la méthode des fenêtres. Pour la région de la bande atténuée, l'algorithme HC et la méthode de l'échantillonnage en fréquence produisent un filtre dont la réponse est un peu moins bonne que les autres techniques.

Nous voyons clairement que l' $AG_1$  produit de meilleurs résultats que les autres versions pour presque tous les critères et que la technique HC est moins pertinente que tous les autres

algorithmes. Selon l'erreur directement liée à la fonction fitness, AG<sub>1</sub> est le plus efficace. Il convient de souligner le rôle particulier du couple "taux de mutation adaptatif" et "processus de rafraîchissement". De deux erreurs [0.0092 ; 0.0803] obtenues avec AG<sub>1</sub>, nous allons à [0.0193 ; 0.1092] pour l'AG<sub>3</sub>. Pour l'AG<sub>1</sub>, nous pouvons observer que les performances et le temps CPU sont améliorés. Seul un petit nombre d'itérations génétiques (298) est nécessaire pour atteindre des erreurs faibles (0.0092 pour l'erreur moyenne et 0.0803 pour l'erreur maximale).

La figure 4.4 montre l'effet du couple "processus de rafraîchissement" et "mutation adaptative". Nous pouvons observer une série de niveaux plats successifs. Chaque fois qu'un signe de convergence prématurée est détecté, la population est réensemencée, ce qui affecte la moyenne de fitness sans perdre la qualité du meilleur chromosome. Cela accélère les progrès de la génétique et améliore la convergence finale.

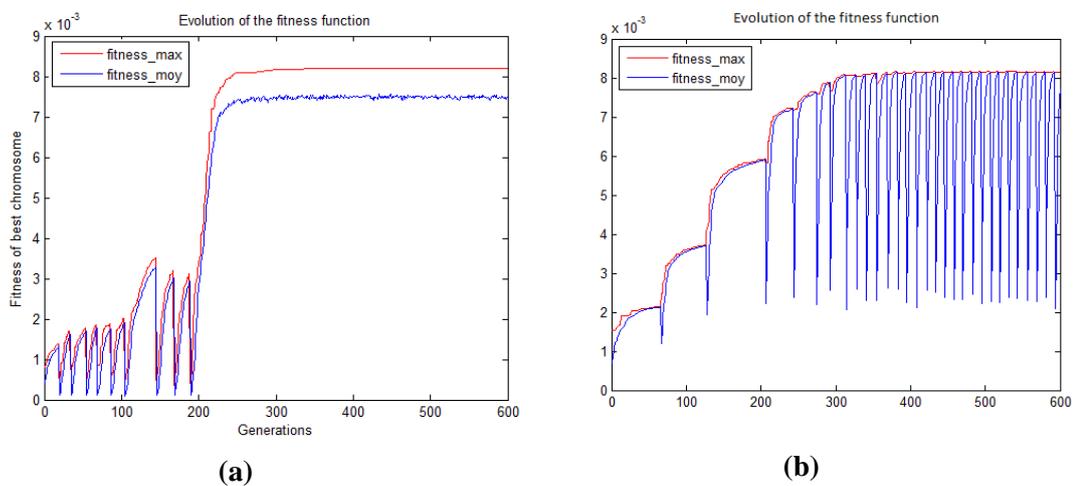


Figure 4.4 – Effet des processus de diversification et de rafraîchissement : (a) Processus de diversification (b) processus de rafraîchissement

On peut voir qu'il y a une différence entre la sélection avec le processus de rafraîchissement et la sélection avec le processus de diversification. Dans le premier cas, le processus de diversification se produit presque instantanément une fois que l'algorithme n'évolue plus, alors que dans le second, le processus de rafraîchissement (afin de réensemencer la population) se produit lorsque certaines conditions sont remplies. Dans le premier cas, le processus de diversification permet d'éviter la stagnation de l'algorithme (la population est diversifiée à tout moment). Notez que la sélection avec diversification n'empêche pas le processus de rupture (breaking) mais raccourcit le réensemencement de la population car la population est plus diversifiée (l'algorithme ne permet pas les similitudes flagrantes entre les individus). Le processus de rafraîchissement doit être suivi par le processus de rupture dans l'algorithme pour donner de meilleurs résultats. L'AG<sub>1</sub> est meilleur dans la plupart des critères, soulignant son rôle intéressant. Nous avons également vérifié le cas où la population initiale a été générée par l'ajout d'un bruit aléatoire aux solutions calculées par des méthodes conventionnelles. L'évolution dans l'AGA est différente, mais les résultats finaux sont semblables à ceux obtenus à partir d'une population initiale pseudo-aléatoire.

#### 4.1.2. Résultats avec d'autres filtres RIF

Le tableau 4.5 montre une étude comparative et de performance entre les diverses techniques mentionnées dans cette partie, appliquées à la synthèse d'un filtre RIF 2D passe-haut symétrique de dimensions 7\*7.

Comme dans le cas d'un filtre passe-bas, on peut voir que l'AG<sub>1</sub> (tableau 4.5 et figure 4.5.a) est meilleur que les autres méthodes pour presque tous les critères, sauf pour le temps CPU nécessaire à la convergence : la synthèse par les algorithmes classiques est caractérisée par un très faible temps CPU.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
AG <sub>1</sub>	0.0108	0.0879	0.108	90.27s	539
AG <sub>2</sub>	0.0199	0.1460	0.169	101.66s	623
AG <sub>3</sub>	0.0133	0.1096	0.121	143.58s	864
AG <sub>4</sub>	0.0182	0.1397	0.137	158.29s	982
Simulated Annealing	0.0343	0.1867	0.176	104.64s	
Tabu Search	0.0855	0.2045	0.135	352.13s	
Hill Climbing	0.2512	0.9803	0.130	Aléatoire	
Remez (Transformation fréquentielle)	0.0250	0.1026	0.166	< 2s	
Moindres carrés (Transformation fréquentielle)	0.0208	0.1002	0.191	< 2s	
Echantillonnage en fréquence	0.0352	0.2019	0.119	< 2s	
Méthode des fenêtres	0.0099	0.2374	0.318	< 2s	

Table 4.5 – Paramètres de performance obtenus pour un filtre passe-haut.

La performance de la méthode des fenêtres est très compétitive en ce qui concerne l'erreur moyenne (0.0099), mais beaucoup moins performante que l'AG<sub>1</sub> concernant le critère d'erreur maximale (0.2374 à 0.0879) et la largeur de bande de transition (0.318 à 0.108). En revanche, l'approche de Remez (transformation fréquentielle) semble être efficace pour l'erreur maximale (0.1026), mais satisfaisant seulement modérément pour le critère concernant la largeur de bande de transition (0.166), et médiocre pour l'erreur moyenne (0.0250). On constate que les méthodes des moindres carrés et de Remez utilisées dans la méthode de la transformation fréquentielle donnent des résultats inférieurs à la moyenne. La méthode des moindres carrés donne une légère diminution des ondulations (erreur maximale) par rapport à la méthode de Remez, mais elle se caractérise par une augmentation de la largeur de la bande de transition.

Compte tenu de l'erreur maximale, les résultats des AG<sub>2</sub> et AG<sub>4</sub> sont de moins bonne qualité que les autres algorithmes génétiques mis en œuvre. Ceci indique que le processus de rafraîchissement ne fonctionne bien que s'il est accompagné par le processus de sélection avec la diversification et la mutation adaptative (AG<sub>1</sub> et AG<sub>3</sub>). Les mécanismes mis en place sont complémentaires. Le processus de rafraîchissement est en mesure de réensemencer la population génétique qui conduit à des modifications substantielles. Les autres mécanismes agissent également à accroître la diversification, mais de façon modérée et qui contribue à la convergence.

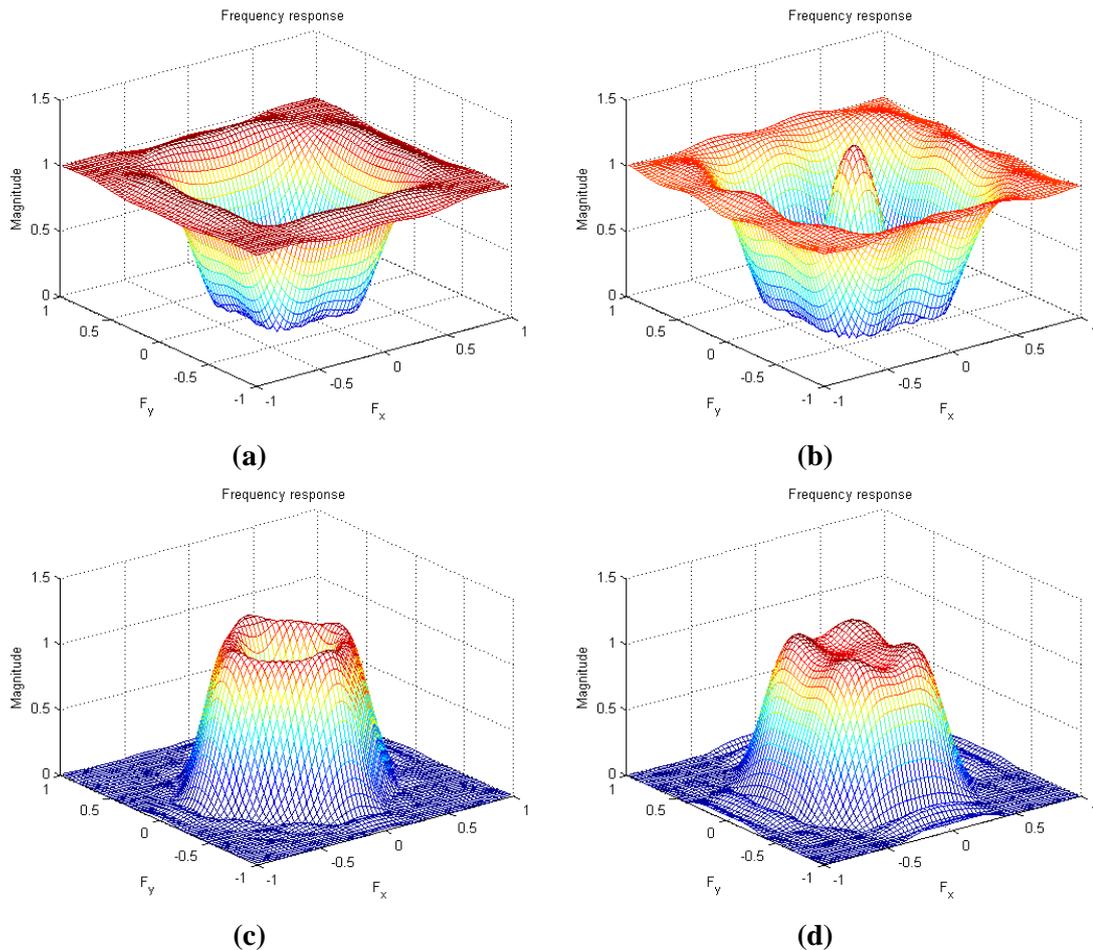


Figure 4.5 – Réponses fréquentielles : (a) Passe-haut 7x7 (b) Coupe-bande 7x7 (c) Passe-bande 9x9 (d) Passe-bas 9x9

Les algorithmes génétiques adaptatifs réussissent mieux que les autres techniques présentées dans cette partie. L'approche Hill Climbing donne les plus mauvais résultats, ce qui peut s'expliquer par le manque d'intelligence embarquée dans cet algorithme qui ne fonctionne que sur la base d'une "marche aléatoire". Nous pensons que plus de la démarche elle-même, les mauvais résultats proviennent principalement de la difficulté de réglage des paramètres d'entrée. Compte tenu de l'absence de progrès théoriques, la seule façon d'améliorer la performance consiste à intégrer une certaine intelligence via l'inclusion de mécanismes supplémentaires.

Les figures 4.5 et 4.6 montrent les résultats de l'application de l'approche AG adaptatif proposée ( $AG_1$ ) à plusieurs problèmes de conception.

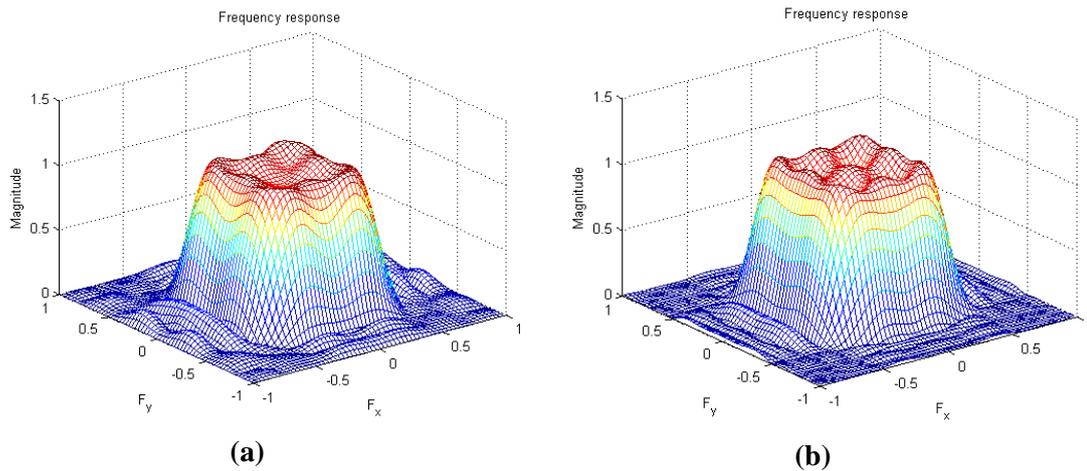


Figure 4.6 – Réponses fréquentielles : (a) Passe-bas 10x10 (b) Passe-bas 11x11.

#### 4.1.3. Comparaison spécifique avec deux méthodes génétiques populaires

Parmi les différentes études déjà publiées, celles de Mastorakis [Mastorakis et al., 2003] et de Tzeng [Tzeng, 2007] semblent être parmi les plus populaires et sont régulièrement citées comme références. Pour cette raison, nous proposons de les comparer à notre approche afin de montrer son efficacité.

Les algorithmes dans [Tzeng, 2007] et [Mastorakis et al., 2003] ont été utilisés pour concevoir des filtres RIF 2D avec deux supports. Le premier exemple correspond au cas déjà proposé dans la section précédente. Le deuxième exemple porte sur la conception d'un filtre ayant la réponse en amplitude désirée donnée par l'équation suivante. Nous avons choisi ce cas, car il est très similaire à celui traité dans [Mastorakis et al., 2003].

$$D(\omega_1, \omega_2) = \begin{cases} 1 & |\omega_{1,2}| \leq 0.1 \\ 0 & |\omega_{1,2}| > 0.1 \end{cases}$$

L'algorithme de Mastorakis est basé sur une représentation chromosomique binaire. Pour notre essai, nous avons développé la version en codage binaire pour avoir une reproduction exacte de leur exemple et une version en codage réel afin de mieux évaluer la méthodologie. Pour l'algorithme codé binaire, différentes longueurs de chromosomes ont été étudiées. Chaque gène est codé sur 10 bits, puis 20 et enfin 30 bits. Par conséquent, la taille du chromosome dépend de la taille du filtre et du nombre de bits des gènes. Pour les résultats mentionnés dans les tableaux 4.6 et 4.7, nous avons donné les résultats des gènes codés sur 20 bits afin de ne pas alourdir le temps de calcul. Les résultats de codage 20 bits et 30 bits sont essentiellement similaires, à l'exception du temps d'exécution (CPU time).

Les résultats sont présentés dans les tableaux 4.6 et 4.7, et seulement les performances avec AG<sub>1</sub> sont comparées. Ces résultats montrent l'amélioration apportée par l'approche proposée par rapport aux autres techniques efficaces.

Les résultats de la méthode utilisée par Mastorakis dans la version codée réelle sont acceptables pour l'exemple présenté dans leur article. Cependant, la précision de leur méthode est plus faible pour un support de filtre différent. La version binaire semble être inefficace pour la synthèse de filtres RIF 2D, car ces filtres nécessitent une grande précision et l'optimisation d'un grand nombre de paramètres.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
AG <sub>1</sub>	0.0092	0.0803	0.113	49.74s	298
Mastorakis réel	0.0289	0.1870	0.167	202.47s	977
Mastorakis binaire	0.0327	0.2102	0.196	471.23s	1259
Tzeng	0.0198	0.1156	0.156	140.36s	928

Table 4.6 – Paramètres de performance obtenus par notre AG adaptatif et les AG's utilisés par Mastorakis et Tzeng pour le premier exemple.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
AG <sub>1</sub>	0.0101	0.0924	0.085	50.31s	304
Mastorakis réel	0.0192	0.1103	0.124	188.02s	930
Mastorakis binaire	0.0292	0.1411	0.147	450.68s	1231
Tzeng	0.0194	0.1116	0.133	132.25s	910

Table 4.7 – Paramètres de performance obtenus par notre AG adaptatif et les AG's utilisés par Mastorakis et Tzeng pour le second exemple.

Les résultats de la méthode utilisée par Tzeng sont comparables à ceux obtenus par l'algorithme génétique standard (AG<sub>4</sub>) mentionnés précédemment. La seule différence est que plusieurs essais sont nécessaires à l'algorithme de Tzeng pour atteindre les résultats de l'algorithme génétique standard. Il n'est pas efficace à chaque exécution de l'algorithme. Les résultats obtenus ne sont pas surprenants puisque notre mécanisme a été conçu dans le but d'améliorer la performance des AG nommés standards.

Concernant l'algorithme de Mastorakis, nous étions sceptiques sur le codage binaire. Cette représentation s'est avérée très efficace pour une sélection de problèmes pour laquelle il est tout à fait adapté. Pour les autres cas, la question de l'irrégularité reste un problème : pour un problème donné, quelle longueur de chromosomes et quels paramètres d'entrée doivent être sélectionnés pour qu'il soit efficace ?

Dans tous les cas, notre algorithme AG<sub>1</sub> donne de meilleurs résultats que ceux utilisés par Mastorakis et Tzeng. Les tableaux 4.6 et 4.7 montrent que la méthode de conception proposée fournit les meilleurs résultats pour la synthèse de filtre RIF 2D.

Il est important de souligner qu'il était difficile de trouver des paramètres conduisant à des solutions acceptables sans un réglage extensif. C'est une critique récurrente adressée à des techniques évolutives.

#### 4.1.4. D'autres résultats (tests) avec des filtres de dimensions plus élevées

L'AG a été appliqué avec succès à divers filtres classiques, y compris, des filtres passe-bande, passe-bas, passe-haut et coupe-bande d'ordres différents avec le même jeu de paramètres. Pour illustrer l'efficacité de l'approche proposée pour les filtres ayant des dimensions plus élevées, nous considérons ici la conception d'un filtre RIF passe-bas 2D de dimensions 23\*23, dont la réponse d'amplitude est donnée par l'équation suivante :

$$D(\omega_1, \omega_2) = \begin{cases} 1 & |\omega_{1,2}| \leq 0.1 \\ 0 & |\omega_{1,2}| > 0.1 \end{cases}$$

Les indices de performance sont présentés dans le tableau 4.8, ainsi que ceux obtenus avec les autres approches discutées dans cette partie, à titre de comparaison, et indiquent la meilleure performance globale de l'approche AGA proposée. Encore une fois, comme on peut le voir dans le tableau 4.8, l'approche AGA (AG<sub>1</sub>) donne de meilleurs résultats que les autres approches pour l'ensemble des critères.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
AG <sub>1</sub>	0.0087	0.0872	0.079	191.04s	742
AG <sub>2</sub>	0.0189	0.1153	0.126	235.52s	961
AG <sub>3</sub>	0.0179	0.1133	0.117	300.44s	1325
AG <sub>4</sub>	0.0203	0.1185	0.138	331.18s	1608
Simulated Annealing	0.0216	0.1399	0.141	220.12s	
Tabu Search	0.0502	0.2208	0.114	515.86s	
Hill Climbing	0.0734	0.3896	0.108	Aléatoire	
Remez (Transformation Fréquentielle)	0.0487	0.1699	0.130	< 2s	
Moindres carrés (Transformation Fréquentielle)	0.0396	0.1431	0.172	< 2s	
Echantillonnage en fréquence	0.0512	0.2310	0.105	< 2s	
Méthode des fenêtres	0.0371	0.4987	0.233	< 2s	

Table 4.8 – Paramètres de performance obtenus pour un filtre passe-bas (23x23).

A l'exception de la vitesse de convergence (temps CPU et nombre d'itérations) qui est affectée par une dimension plus grande, la différence entre les approches concernant les autres critères sont similaires. Par exemple, alors que AG<sub>1</sub> donne une erreur moyenne de 0.0087, les méthodes de l'échantillonnage en fréquence et des fenêtres donnent respectivement 0.0512 et 0.071. L'AG<sub>1</sub> est performant pour découvrir et trouver des solutions pertinentes que les autres versions d'AGA pour tous les critères. Cela illustre la complémentarité des rôles des différents mécanismes mis en œuvre.

#### 4.1.5. Discussion, étude statistique et de convivialité

Cette section passe en revue l'ADN de notre AGA et démontre sa cohérence par des expérimentations supplémentaires.

L'ensemble des paramètres suivants sont calibrés pour la classe de filtres étudiés. Ils sont fixés pour l'utilisateur.

- Les paramètres liés à l'initialisation de la population
- Les paramètres de mutation  $k_1$  et  $k_2$ . Ils ont été calibrés pour être globalement efficaces, par exemple, un bon compromis entre la diversité et l'élitisme
- La probabilité de croisement  $C_R$
- Le niveau de la probabilité de mutation  $M_P$  quand un processus de rafraîchissement est nécessaire
- Les paramètres  $\lambda_{m1}$  et  $\lambda_{m2}$  utilisés dans la formule de mutation
- Les paramètres  $f_\varepsilon$ ,  $g_\varepsilon$ ,  $f_{end}$
- La matrice de pondération  $W$  pour jouer sur les ondulations : le meilleur compromis est d'environ (5/8). Il est toutefois possible d'améliorer légèrement les performances en sélectionnant des valeurs dans l'intervalle  $(5/8 \pm 5\%)$

Les paramètres secondaires sont peu nombreux et peuvent être considérés comme des paramètres de l'utilisateur :

- $\Delta f_{min}$
- Les paramètres  $th_1$ ,  $th_2$  et  $th_3$  utilisés pour détecter un minimum local

Pour ces paramètres, les meilleures valeurs sont fonction de la taille du filtre et le nombre de points de fréquence (voir calcul d'erreur). Différentes valeurs ont été calibrées pour correspondre avec des configurations différentes (différentes tailles de filtre et différents points de fréquence) et ils servent de références pour les nouvelles configurations. Leurs rôles sont parfaitement compréhensibles car ils n'ont pas d'interaction avec la complexité de l'évolution.

Afin d'évaluer l'utilisabilité (répétabilité, la performance et le temps de calcul), nous avons généré 100 types de filtres différents et avons changé la fréquence de coupure (de 0.1 à 0.8). Pour chacun deux, nous avons lancé 10 exécutions avec exactement les mêmes paramètres.

Les résultats pour  $\omega_{1,2} = 0.2, 0.3$  et  $0.4$  sont rapportées dans les tableaux suivants et affichés sur la figure suivante (Fig. 4.7).

$\omega_{1,2} = 0.2$	1	2	3	4	5	6	7	8	9	10
Erreur moy	0.0087	0.0087	0.0089	0.0085	0.0087	0.0090	0.0086	0.0086	0.0086	0.0087
Erreur max	0.0801	0.0784	0.0739	0.0796	0.0793	0.0828	0.0820	0.0811	0.0783	0.0801
Temps CPU	49.87	46.17	48.79	51.59	50.20	50.74	48.03	49.23	50.02	50.20
Ite	299	280	301	324	313	317	294	303	307	310

Table 4.9 – Paramètres de performance obtenus pour  $\omega_{1,2} = 0.2$ .

$\omega_{1,2} = 0.3$	1	2	3	4	5	6	7	8	9	10
Erreur moy	0.0092	0.0093	0.0091	0.0094	0.0090	0.0096	0.0099	0.0093	0.0095	0.0095
Erreur max	0.0905	0.0883	0.0830	0.0847	0.0914	0.0846	0.0896	0.0845	0.0838	0.0878
Temps CPU	50.32	49.93	48.96	50.73	51.74	47.30	52.08	50.29	49.90	53.19
Ite	297	296	291	307	315	276	322	303	300	322

Table 4.10 – Paramètres de performance obtenus pour  $\omega_{1,2} = 0.3$ .

$\omega_{1,2} = 0.4$	1	2	3	4	5	6	7	8	9	10
Erreur moy	0.0094	0.0096	0.0104	0.0098	0.0100	0.0093	0.0095	0.0099	0.0104	0.0099
Erreur max	0.0866	0.0826	0.0888	0.0828	0.0855	0.0832	0.0802	0.0805	0.0816	0.0899
Temps CPU	53.69	49.67	55.51	51.66	49.72	49.35	51.86	52.50	50.04	51.93
Ite	332	292	337	305	292	287	307	314	293	308

Table 4.11 – Paramètres de performance obtenus pour  $\omega_{1,2} = 0.4$ .

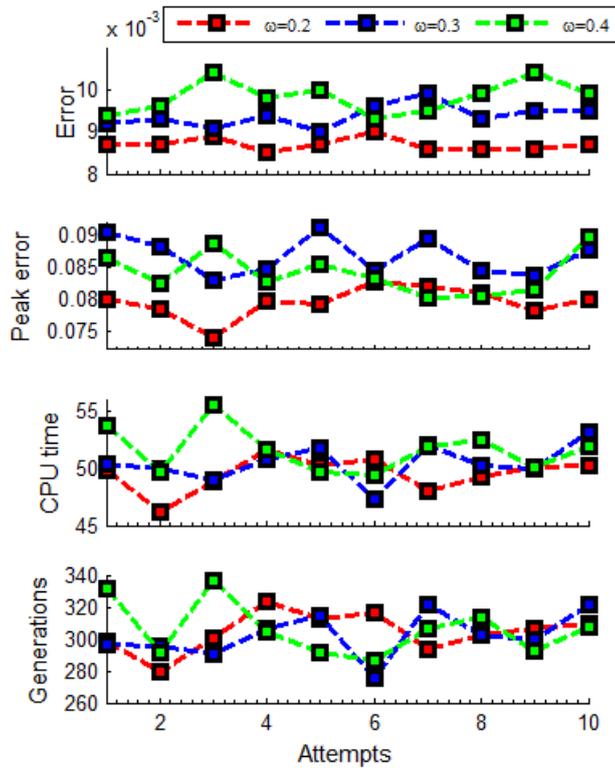


Figure 4.7 – Résultats avec différentes valeurs de la fréquence de coupure pour 10 exécutions.

Pour approfondir davantage la régularité et la répétabilité de notre algorithme, nous avons généré aléatoirement 100 filtres et lancé leurs exécutions. La figure suivante (Fig. 4.8) représente les erreurs moyenne et maximale obtenues.

Les variations d'erreur (moyenne et maximale) sur de grandes échelles de tests sont très petites, ce qui démontre la cohérence de notre algorithme. Certains cas sont plus faciles à distinguer lorsque l'erreur maximale est analysée, mais la variation de la quantité reste très faible. Et pour chaque opération, l'algorithme donne des résultats satisfaisants en termes de performance.

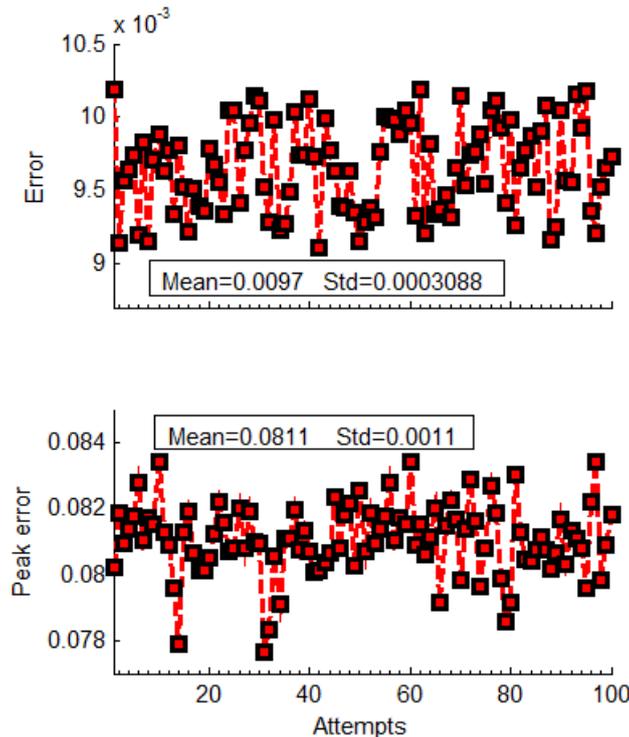


Figure 4.8 – Erreurs moyennes et maximales pour 100 exécutions.

## 4.2. PSO pour la synthèse des filtres RIF 1D

Les tests dans [Ababneh et al., 2008], [Kar et al., 2011] et [Monadal et al., 2012] traitent la synthèse des filtres RIF passe-bas d'ordre 20 à 30. À cet effet, les simulations ont été effectuées pour concevoir des filtres RIF passe-bas avec des longueurs de coefficients 21, 31 et 41. Le nombre d'échantillons de fréquence choisi est de 128.

Le tableau 4.12 résume les meilleurs paramètres choisis pour les algorithmes (GA, AGA, PSO et NPSO, CPSO) utilisés pour la conception des filtres RIF. Dans cette partie, il n'y a aucune comparaison avec les approches classiques de conception de filtres comme de nombreuses études traitant des approches évolutionnistes ont déjà traité ce sujet et mener à des conclusions similaires. Pour la même raison, l'AG le plus élémentaire avec des paramètres statiques et des opérateurs génétiques classiques (croisement, mutation et sélection) n'a pas été inclus dans la comparaison. Bien que les résultats obtenus avec ces algorithmes (AG's élémentaires) pour ce type de filtres soient intéressants, ils ne peuvent pas

être extrapolés à d'autres problèmes car ils dépendent de la capacité humaine à régler les paramètres.

Paramètres	GA	AGA	PSO	NPSO	CPSO
Nombre d'échantillons de fréquence	128	128	128	128	128
Taille de la population	120	120	120	120	120
Nombre max d'itérations	500	500	500	500	500
Croisement	direction-based crossover (1 descendant)	direction-based crossover (2 descendant)	-	-	-
Taux de croisement	100%	100%	-	-	-
Mutation	Mutation non-uniforme	Mutation adaptative non-uniforme avec la procédure de diversification	-	-	-
Probabilité de mutation	0.5%	Probabilité de mutation adaptative	-	-	-
Processus de rafraîchissement	-	Actif	-	-	-
$C_1$	-	-	2.05	2.05	2.05
$C_2$	-	-	2.05	2.05	2.05
$v_i^{min}$	-	-	0.01	0.01	0.01
$v_i^{max}$	-	-	1	1	1
$\gamma^{min}$	-	-	0.4	0.4	-
$\gamma^{max}$	-	-	1	1	-
$P_{cr}$	-	-	-	-	0.3
$v_{craziness}$	-	-	-	-	0.0001

Table 4.12 – Paramètres de chaque algorithme.

Le bon choix des paramètres joue un rôle important dans le profil de convergence des algorithmes cités précédemment. Ils ont été sélectionnés à partir des documents de référence pour les algorithmes PSO [Mondal et al., 2012] et [Kar et al., 2012]. Les résultats présentés pour chaque algorithme sont les meilleures (selon l'erreur moyenne) parmi les 5 tests (essais) successifs lancés avec les mêmes paramètres. La répétabilité des résultats est assez bonne, ce qui souligne la cohérence des algorithmes : le rapport entre la plus mauvaise et la meilleure erreur est inférieur à 5% pour tous les algorithmes sauf pour la PSO qui présente généralement un écart plus grand, mais moins de 10%.

Ces algorithmes sont également utilisés pour la conception de filtres RIF passe-haut et ont obtenu les mêmes profils de performance que les filtres RIF passe-bas.

#### 4.2.1. Réglage des paramètres

*Paramètres génétiques de base :*

Ces paramètres sont pré-calibrés et n'ont pas besoin d'être réglés par l'utilisateur.

- La taille de la population initiale  $Size_{pop}$  est égale à 120. Le cycle évolutionnaire est répété jusqu'à 500 générations.
- Les valeurs des paramètres tels que les probabilités de croisement et de mutation  $C_R$ ,  $P_M$  et les paramètres  $k_1$ ,  $k_2$ ,  $\lambda_{m1}$ ,  $\lambda_{m2}$  sont d'une grande importance pour que l'algorithme trouve une solution optimale avec rapidité, précision et diversité. Le tableau 4.13 montre les performances obtenues pour différentes valeurs des paramètres mentionnés ci-dessus.

	Paramètres	Ondulations max en bande atténuée	Ondulations max en bande passante	Erreur moyenne	Ite
AG Adaptif (AGA)	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0119	0.0831	0.0458	215
	$C_R = 80\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0142	0.0906	0.0492	308
	$C_R = 40\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0145	0.0918	0.0509	423
	$C_R = 100\%$ , $P_M = 1\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0128	0.0865	0.0471	377
	$C_R = 100\%$ , $P_M = 80\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0124	0.0852	0.0464	223
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.01\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0122	0.0847	0.0469	358
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.9\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0120	0.0835	0.0463	310
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 10\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0198	0.1012	0.0470	302
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 0.2\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0123	0.0834	0.0468	359
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 5 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0163	0.0985	0.0466	389
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 0.5 * rand$ , $\lambda_{m2} = 0.2 * rand$	0.0151	0.0936	0.0502	374
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 2 * rand$	0.0172	0.0989	0.0518	415
	$C_R = 100\%$ , $P_M = 40\%$ , $k_1 = 0.1\%$ , $k_2 = 1\%$ $\lambda_{m1} = 2 * rand$ , $\lambda_{m2} = 0.05 * rand$	0.0164	0.0974	0.0501	428

Table 4.13 – Réglage des paramètres de l'AGA.

Les paramètres de l'AGA ont été déterminés à partir de plusieurs tests. Les valeurs de ces paramètres permettent aux algorithmes d'avoir des performances optimales, c'est à dire minimiser les erreurs et réduire le temps de calcul nécessaire pour la convergence.

Dans ce qui suit (tableau 4.14), les paramètres sont répertoriés avec les valeurs utilisées dans les tests.

	Paramètres	
AG Adaptatif (AGA)	• Mutation adaptative non-uniforme avec la procédure de diversification	$k_1 = 0.1\%, k_2 = 1\%$
	• Présence de minimum local	$k_1 = 10\%, k_2 = 30\%$
	• Processus de rafraîchissement actif	$P_M = 40\%$
	• Direction-based crossover	$C_R = 100\%$
	• Sélection adaptative avec la procédure de rafraîchissement (Diversification et processus de rupture)	

Table 4.14 – Paramètres de l’AGA.

Paramètre de domaine :

- La valeur de  $W$  qui donne les meilleurs résultats est 1 aux fréquences incluses dans la bande passante et  $(5/8)^2$  aux fréquences incluses dans la bande atténuée. Ces valeurs ont été étalonnées après plusieurs essais pour différentes valeurs de  $W$  et différentes configurations de filtres.

#### 4.2.2. Exemples de synthèse

**Exemple 1 :**

$$D(\omega) = \begin{cases} 1 & 0 \leq \omega \leq 0.5 \\ 0 & \omega > 0.5 \end{cases}$$

Les figures 4.9 et 4.10 montrent respectivement, la réponse en amplitude pour un filtre RIF passe-bas d’ordre 40 obtenu en utilisant GA, AGA, PSO, NPSO et CPSO et les profils de convergence de chaque algorithme.

La figure 4.10 montre une comparaison complète de l’évolution de la fonction fitness donnée par les différents algorithmes. Cette courbe met en évidence les avantages apportés par la CPSO en terme du nombre d’itérations nécessaires à la convergence. En comparaison avec les autres algorithmes, il est clair que lorsqu’on utilise les versions sophistiquées de l’algorithme PSO, toutes les particules ont tendance à converger vers la meilleure solution rapidement dans la plupart des cas, alors que la version de base de la PSO produit les moins meilleurs résultats et présente de nombreuses oscillations au début du procédé.

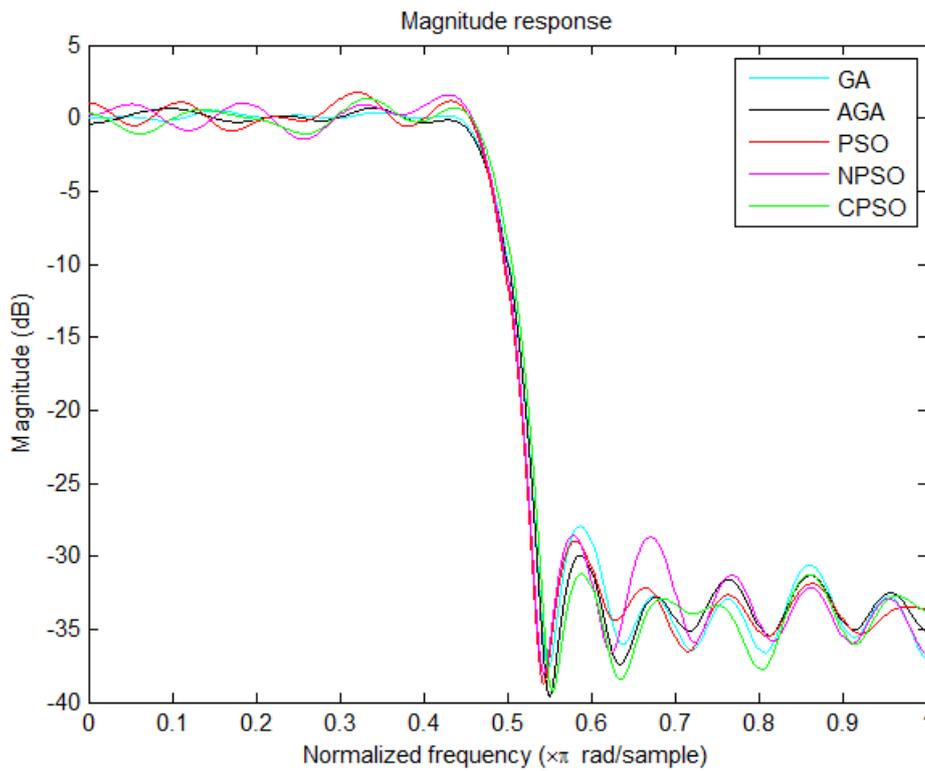


Figure 4.9 – Réponses en amplitude (dB) obtenues pour un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.5$ ) par les techniques GA, AGA, PSO, NPSO et CPSO.

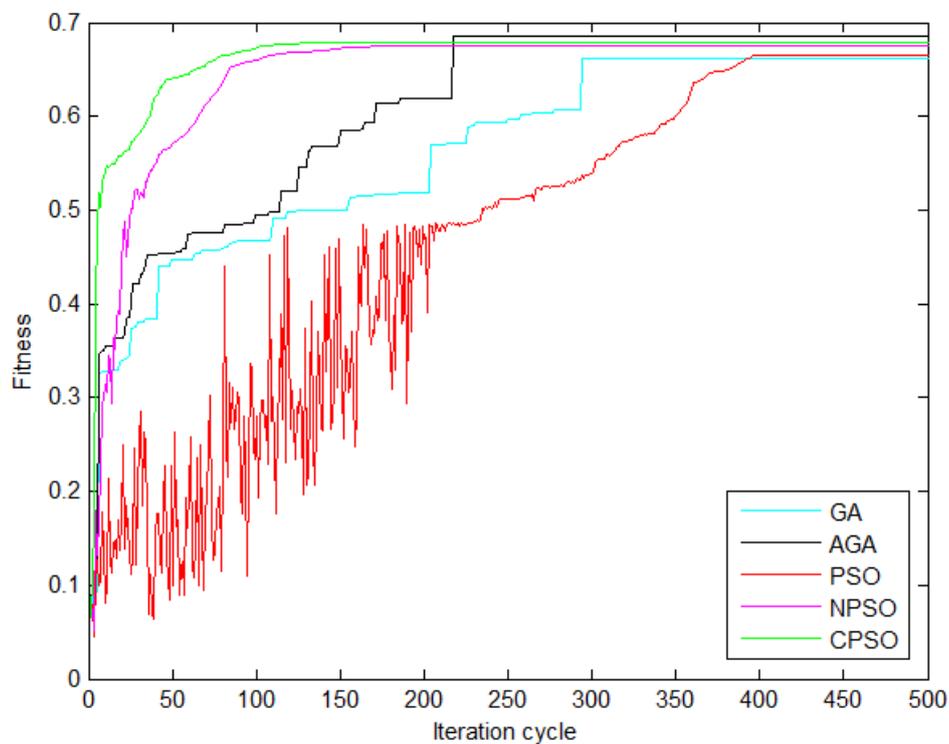


Figure 4.10 – Profile de convergence pour les techniques GA, AGA, PSO, NPSO et CPSO dans le cas d'un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.5$ ).

Le tableau 4.15 résume les résultats et un comparatif des différents paramètres de performance obtenus avec les algorithmes GA, AGA, PSO, NPSO et CPSO pour un filtre RIF passe-bas d'ordre 40.

Paramètres	GA	AGA	PSO	NPSO	CPSO
Ondulations max dans la bande d'arrêt	0.0201	0.0119	0.0155	0.0173	0.0087
Ondulations max dans la bande passante	0.0604	0.0831	0.2252	0.2015	0.1648
Erreur moyenne	0.0512	0.0458	0.0644	0.0641	0.0509
Nombre d'itérations	294	215	398	172	128

Table 4.15 – Performance de chaque algorithme ( $2*N+1=41$ ).

Les algorithmes GA et AGA donnent de faibles ondulations dans la bande passante, mais l'algorithme AGA donne la meilleure valeur en terme d'erreur moyenne. Le NPSO et CPSO convergent vers la valeur désirée après, respectivement, 172 et 128 itérations. Ces résultats indiquent que l'AGA et CPSO donnent de meilleurs résultats en termes d'erreur et de nombre d'itérations. Ces résultats de simulation démontrent clairement la supériorité de l'algorithme génétique adaptatif et du Craziness PSO sur les autres méthodes utilisées dans la présente partie (GA, PSO, NPSO).

**Exemple 2 :**

$$D(\omega) = \begin{cases} 1 & 0 \leq \omega \leq 0.5 \\ 0 & \omega > 0.5 \end{cases}$$

La figure 4.11 montre la réponse en amplitude obtenue pour un filtre RIF passe-bas d'ordre 20 à l'aide des GA, AGA, PSO, NPSO et CPSO. Le tableau 4.16 mentionne les ondulations maximales dans les bandes passantes et atténuées, l'erreur moyenne et le nombre d'itérations nécessaires à la convergence lorsque tous les algorithmes sont utilisés pour la conception d'un filtre RIF passe-bas d'ordre 20. Les algorithmes peuvent être comparés en termes de vitesse de convergence. La figure 4.12 montre le tracé des valeurs maximales de la fonction fitness en fonction du numéro d'itération. De la figure 4.12, on peut voir que le CPSO est nettement plus rapide que les autres algorithmes pour trouver le meilleur filtre. L'AGA converge vers une meilleure valeur de fitness dans un nombre raisonnable d'itérations. Le tableau 4.16 confirme notre affirmation selon laquelle l'AGA est plus efficace pour la conception des filtres RIF. Sauf pour la technique PSO, tous les algorithmes présentent une courbe similaire au début, mais le niveau de fitness est continuellement faible pour l'AG. Concernant la PSO, les oscillations sont toujours présentes dans la courbe de fitness, mais la fitness atteinte à la convergence est meilleure que celle de l'AG et la même que celle obtenue par la NPSO. Il est intéressant de comparer le GA et l'AGA : on peut observer que les courbes de fitness changent étape par étape. Ceci diffère des versions PSO qui présentent une progression plus régulière. Les différentes versions de l'AG ont en commun une mutation non-uniforme, la différence entre elles réside dans le niveau de l'escalier. Cela peut être interprété comme une meilleure capacité de l'AGA à trouver de meilleures régions de l'espace de recherche. Cette dynamique est d'une certaine manière dissociée du progrès génétique qui peut laisser les praticiens sans aucune aide pour une situation donnée.

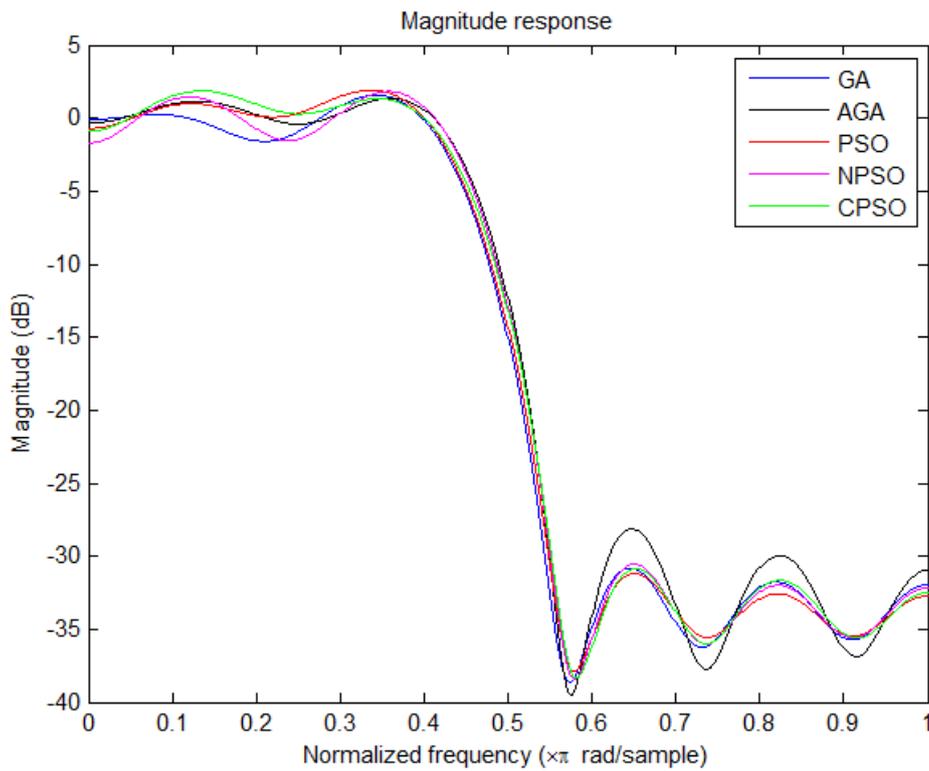


Figure 4.11 – Réponses d’amplitude (dB) d’un filtre RIF passe-bas d’ordre 20 ( $\omega_c = 0.5$ ) conçu par les algorithmes GA, AGA, PSO, NPSO et CPSO.

Paramètres	GA	AGA	PSO	NPSO	CPSO
Ondulations max dans la bande atténuée	0.0088	0.0191	0.0076	0.0097	0.0086
Ondulations max dans la bande passante	0.1989	0.1697	0.2392	0.2342	0.2468
Erreur moyenne	0.0625	0.0598	0.0755	0.0799	0.0802
Nombre d’itérations	302	177	419	99	118

Table 4.16 – Performance de chaque algorithme ( $2*N+1=21$ ).

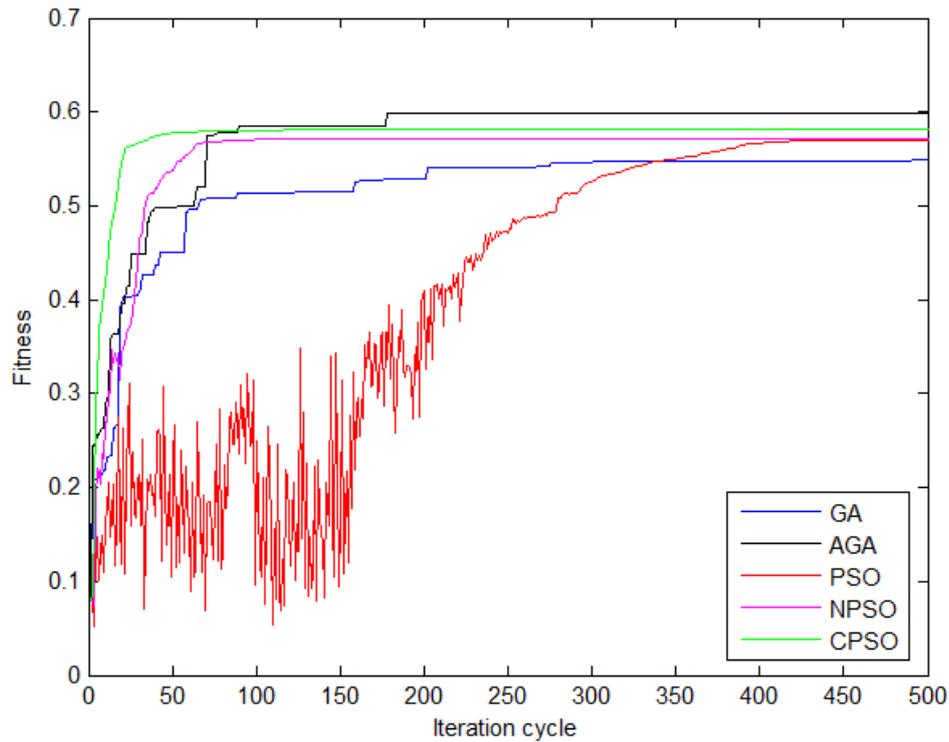


Figure 4.12 – Profile de convergence pour les techniques GA, AGA, PSO, NPSO et CPSO dans le cas d'un filtre RIF passe-bas d'ordre 20 ( $\omega_c = 0.5$ ).

**Exemple 3:**

$$D(\omega) = \begin{cases} 1 & 0 \leq \omega \leq 0.4 \\ 0 & \omega > 0.4 \end{cases}$$

Le tableau 4.17 présente la comparaison complète entre les algorithmes GA, AGA, PSO, NPSO et CPSO pour la conception d'un filtre RIF d'ordre 40. La figure 4.13 montre les réponses en amplitude obtenues en utilisant les GA, AGA, PSO, NPSO et CPSO.

Le tableau 4.17 montre que les plus faibles ondulations en bande passante sont obtenues à l'aide de l'AGA (0.0507). On constate dans le tableau 4.17 que la CPSO réalise le meilleur cycle d'itérations pour la convergence. Seuls 134 itérations sont nécessaires pour sa convergence alors que 487 itérations sont nécessaires à la PSO pour des performances plus faibles. Pour cet exemple, les algorithmes NPSO et CPSO présentent des performances similaires. L'AGA a une meilleure réponse en amplitude, de faibles ondulations en bande passante et une meilleure erreur moyenne que les autres algorithmes. 215 itérations sont toutefois nécessaires pour atteindre la convergence.

Paramètres	GA	AGA	PSO	NPSO	CPSO
Ondulations max dans la bande atténuée	0.0313	0.0361	0.0112	0.0108	0.0100
Ondulations max dans la bande passante	0.0939	0.0507	0.1971	0.1888	0.1320
Erreur moyenne	0.0763	0.0679	0.0794	0.0735	0.0717
Nombre d'itérations	365	215	487	168	134

Table 4.17 – Performance de chaque algorithme ( $2*N+1=41$ ).

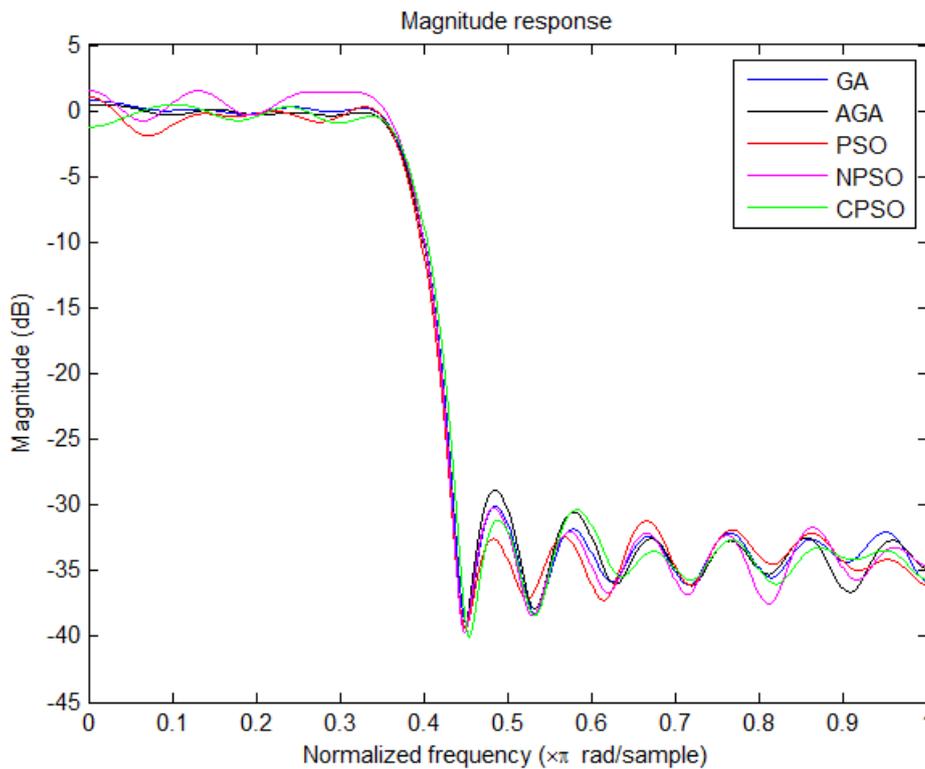


Figure 4.13 – Réponse en amplitude (dB) d'un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.4$ ) conçu par les techniques GA, AGA, PSO, NPSO et CPSO.

**Exemple 4 :**

$$D(\omega) = \begin{cases} 1 & 0 \leq \omega \leq 0.6 \\ 0 & \omega > 0.6 \end{cases}$$

Comme on le voit sur la figure 4.14, pour la région de la bande passante, l'AGA, NPSO et CPSO produisent une meilleure réponse que celle des GA et PSO. Pour la région de la bande d'arrêt, le filtre conçu par la PSO produit une meilleure réponse que les autres. Pour cet exemple, l'erreur moyenne obtenue avec l'AGA (0.0185) est beaucoup plus faible que celles obtenues par les autres algorithmes (de 0.0419 à 0.0674). Le tableau 4.18 résume les performances de chaque algorithme.

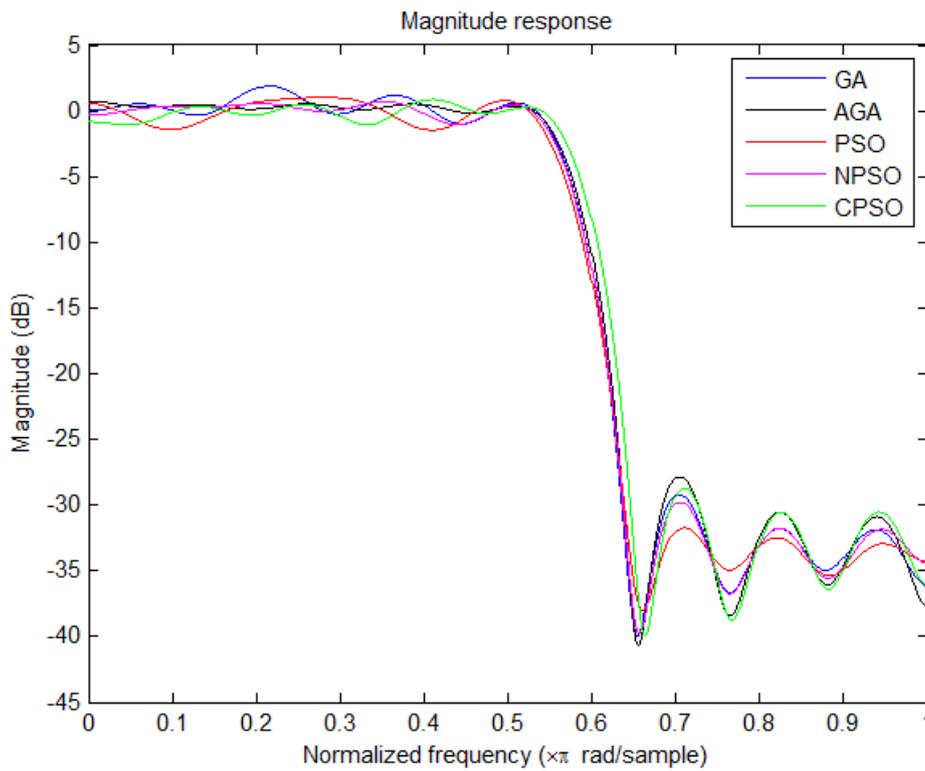


Figure 4.14 – Réponse en amplitude (dB) d'un filtre RIF passe-bas d'ordre 30 ( $\omega_c = 0.6$ ) conçu par les techniques GA, AGA, PSO, NPSO et CPSO.

Paramètres	GA	AGA	PSO	NPSO	CPSO
Ondulations max en bande atténuée	0.0145	0.0194	0.0072	0.0124	0.0165
Ondulations max en bande passante	0.2302	0.0191	0.1551	0.1129	0.1144
Erreur moyenne	0.0674	0.0185	0.0665	0.0419	0.0422
Nombre d'itérations	297	203	471	135	163

Table 4.18 – Performance de chaque algorithme ( $2*N+1=31$ ).

### 4.2.3. Etude statistique

Afin d'évaluer l'utilisabilité (répétabilité, performance et nombre d'itérations) de l'AGA, le filtre de l'exemple 1 a été synthétisé en exécutant l'algorithme AGA plusieurs fois successivement. Les résultats en termes d'erreur d'ondulation maximale dans la bande passante et atténuée, et d'erreur moyenne sont donnés dans les figures 4.15 et 4.16 pour 10 et 100 exécutions respectivement.

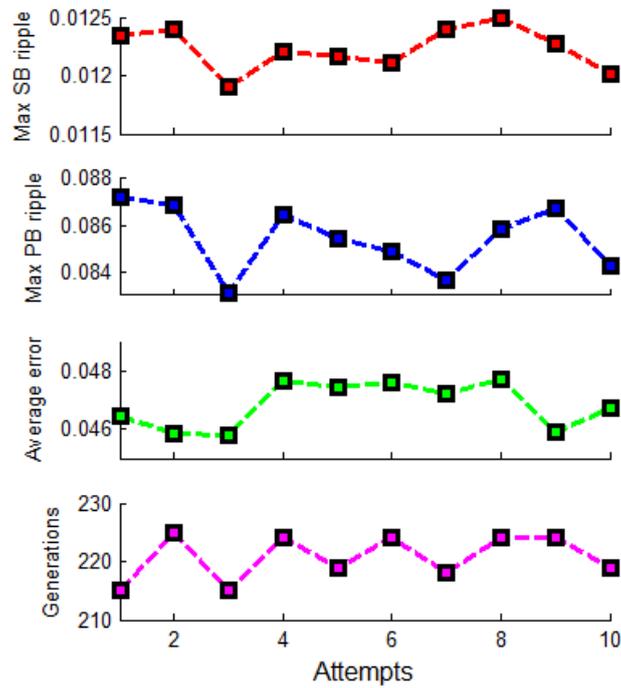


Figure 4.15 – Résultats obtenus pour 10 exécutions (variation et évolution des performances pour 10 exécutions)

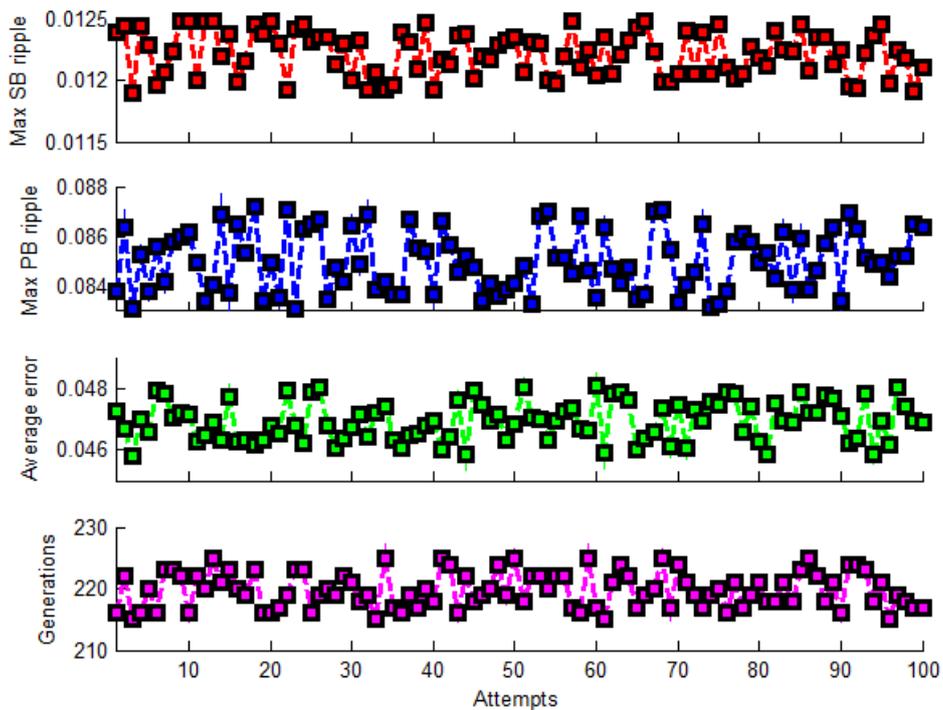


Figure 4.16 – Résultats obtenus pour 100 exécutions (variation et évolution des performances pour 100 exécutions)

Pour plus d’approfondissement, 100 exécutions du même exemple ont été réalisées. Le tableau 4.19 représente l’erreur moyenne et l’erreur maximale en bande passante et en bande atténuée. Les variations des erreurs (moyenne et maximale) pour un large échantillon de tests sont très faibles, ce qui démontre l’efficacité et la robustesse de notre algorithme génétique adaptatif. Le plus intéressant est que l’algorithme donne des résultats satisfaisants en terme de performance. La moyenne et l’écart type des erreurs démontrent la répétabilité de l’algorithme génétique adaptatif.

	<i>Paramètres</i>	<i>Moyenne</i>	<i>Ecart type</i>
	Erreur d’ondulations max en bande atténuée	0.0122	$1.7962 \times 10^{-4}$
AG Adaptatif (AGA)	Erreur d’ondulations max en bande passante	0.0851	$12.0036 \times 10^{-4}$
	Erreur moyenne	0.0469	$6.5761 \times 10^{-4}$

Table 4.19 – Moyenne et écart type des erreurs obtenues pour 100 exécutions.

#### 4.2.4. Discussion et perspectives

Les mérites des algorithmes PSO résident dans leur simplicité de mise en œuvre ainsi que du fait que leur convergence peut être contrôlée via quelques paramètres. Toutefois, la version de base, malgré son intérêt, donne des résultats non-optimaux et moins répétitif que les autres versions plus sophistiquées de ces algorithmes. Il y a des paramètres de contrôle impliqués dans l’algorithme PSO et le réglage approprié de ces paramètres reste un point clé de la réussite. Une certaine forme de réglage essai-erreur est nécessaire pour chaque cas particulier du problème de conception de filtre RIF. Bien que la technique PSO est probablement plus simple que l’AG, la situation des praticiens est similaire. Aucune conclusion ne peut être tirée au sujet de la supériorité d’une méthode sur l’autre en termes de performances. Il faut dire que les versions de base de l’AG sont moins conviviales et dépendent davantage sur les efforts et la capacité à gérer le réglage des paramètres que les algorithmes eux-mêmes.

Les algorithmes NPSO et particulièrement CPSO sont plus sophistiqués. Ils produisent clairement de meilleurs résultats que la version de base de la technique PSO. Bien que les paramètres de contrôle soient impliqués dans les algorithmes, ils intègrent certains mécanismes qui permettent d’optimiser la recherche et d’atteindre des résultats plutôt bons. Ils permettent de mettre à jour les connaissances de l’essaim de l’environnement ; l’équilibre entre l’exploration et l’exploitation est mieux géré. La NPSO et la CPSO reposent sur la même idée générale : au début ils facilitent la recherche globale, mais se concentrent davantage sur l’exploitation de l’espace à la fin. Le réglage des paramètres appropriés pour une classe de problèmes ne semble pas être difficile. La CPSO intègre une certaine intelligence en la rendant plus auto-réglable (adaptative), ce qui donne une différence significative dans les résultats.

Comme on peut le constater à partir de cette étude, la NPSO et la CPSO sont meilleurs en terme du nombre de cycles d'itérations nécessaires pour la convergence que les AGA, GA et PSO. Les AGA, NPSO et CPSO convergent vers leurs minimums d'ondulations respectifs en moins de 250 itérations. En outre, la NPSO et la CPSO donnent des valeurs plus élevées (sous-optimales) d'erreur mais l'AGA atteint une erreur presque optimale (faibles). Notons que 100 itérations pour la PSO correspondent à 3.44 s et 19.28 s pour l'AGA, amplifiant ainsi la différence dans le taux de convergence entre PSO et AGA. Nous considérons qu'il s'agit là d'un atout majeur de ces algorithmes. Compte tenu du fait ci-dessus, on peut enfin en déduire que les performances de la technique AGA est meilleure que celle des NPSO et CPSO, mais l'AGA a besoin de plus de temps pour explorer et exploiter l'espace de recherche. Une explication plausible de cette supériorité est que dans un système AG classique, il est difficile dans la pratique à un chromosome d'occuper tout l'espace de recherche et que de multiples itérations sont nécessaires. Puis, à l'approche de la convergence, la sélectivité nécessaire introduite dans l'algorithme pénalise la diversité. Dans notre processus, la diversité est mieux gérée par les différents mécanismes qui permettent aux autres zones de l'espace de recherche d'être explorées, même si la convergence est presque atteinte et qui augmentent la probabilité d'améliorer les performances générales. Ceci peut être facilement observé par l'analyse de la fitness maximale. Contrairement aux autres algorithmes qui présentent une forte continuité, la meilleure fonction fitness pour l'AGA s'améliore avec une continuité donnée, mais par l'intermédiaire d'étapes consécutives.

#### 4.2.5. Hybridation préliminaire de l'AGA avec la NPSO

Les AG's classiques sont réputés pour être efficaces à explorer la totalité de l'espace de recherche, mais pauvres à trouver un optimum local précis. Pour cette raison, ils sont souvent combinés avec des approches locales pour compenser cette faiblesse. Les algorithmes PSO explorent différemment l'espace de recherche. Nos simulations prouvent qu'au départ ils ont un taux de convergence plus rapide, mais deviennent rapidement inefficaces à trouver l'optimum local avec précision. Dans les PSO, une particule ne peut atteindre n'importe quel point de l'espace de recherche rapidement. En théorie, les particules survivent d'une itération à l'autre, toute particule peut finalement aller n'importe où. Cela pourrait être possible au début de la recherche, mais il semble plus difficile par la suite à l'approche de la convergence, car le processus est partiellement bloqué. Cette question peut être traitée par une forte coopération avec les mécanismes mis en place dans notre AGA. La stagnation doit être détectée et le processus est relancé via un processus de rafraîchissement. Ce processus est cependant important non seulement pour détecter la convergence prématurée, mais aussi pour l'empêcher. L'idée est de maintenir autant que possible une population présentant un minimum de diversité afin de renforcer la probabilité d'échapper à un minimum local. Les particules redondantes ne contribuent pas davantage à l'amélioration de la convergence. Ainsi, la prévention peut être réalisée en inspectant en permanence la présence de particules redondantes et le remplacement de certaines d'entre elles par de nouvelles particules via un opérateur spécialisé comme cela se fait dans notre AGA.

En outre, une coopération étroite entre les mécanismes des PSO pour l'exploration et l'AGA est susceptible d'améliorer les performances de l'AGA, en particulier le temps d'exécution

(temps CPU). L'exploration s'effectue différemment, en général rapidement et cette différence peut être exploitée avantageusement.

Après avoir souligné certaines complémentarités évidentes, il convient de mentionner plusieurs analogies entre les algorithmes.

L'adaptation de la mutation en AG qui a prouvé son efficacité est analogue à contrôler le poids de la vitesse (inertie) comme dans la NPSO et la CPSO. Les deux paramètres restent importants. Leur réglage doit être étalonné, ces paramètres varient en fonction de la durée de vie de l'évolution et sont consacrés à un chromosome ou à une particule spécifique. En outre, la limitation est similaire : comme la souligne Eberhart dans [Eberhart et al., 2000], quand la population est proche de la convergence et que la valeur de la fitness moyenne est élevée, la mutation sera assez grande dans un chromosome avec une faible fitness et qui ne survit pas au processus de sélection. Ainsi, même si un certain nombre de mutations porterait le chromosome dans une région de haute-fitness, le chromosome n'y arrive jamais parce qu'il ne survit pas à la sélection.

L'opération de croisement dans l'AG peut être partiellement récupérée dans les différentes équations gérant l'évolution des particules. L'analogie est ici moins évidente sauf que nous pouvons dire que l'échange entre les éléments reste statique et pas optimisé dans la version de base. Certains schémas de croisement sophistiqués intègrent plus d'intelligence et sont analogues à l'évolution dans les PSO surtout si l'opération de croisement est élargie sur plus de deux chromosomes.

Dans la PSO, la notion de sélectivité est indirectement présente, même si toutes les particules continuent en tant que membres de la population pendant toute la durée de la recherche. A notre avis, la faiblesse est plus due à l'absence de contrôle de la sélectivité qui est mieux gérée par l'AGA. Notre stratégie de sélection maintient le classement des meilleurs chromosomes mais peut éliminer les redondants pour faire place à une plus grande diversité et d'augmenter les chances de trouver une meilleure zone de l'espace. Sans surprise, on constate que la précision moyenne augmente clairement avec le nombre d'itérations, pour toutes les méthodes, et que l'ordre relatif des différentes méthodes ne dépend pas fortement du nombre d'échantillons. Bien qu'il soit impossible d'extrapoler les résultats à l'ensemble de la classe des problèmes, il n'est pas difficile d'imaginer que les approches PSO pourraient être encore améliorées en intégrant cette intelligence comme cela se fait pour les AGA. L'hybridation entre ces deux techniques est une direction intéressante, mais elle doit être bien gérée. Nous avons mis en place un système hybride préliminaire dans le cadre de cette étude. Cet exemple a été traité pour souligner la puissance et illustrer l'amélioration offerte par l'hybridation de la PSO avec l'AGA. L'idée est d'utiliser la capacité de la NPSO d'atteindre rapidement des zones intéressantes de l'espace de recherche et de les exploiter par l'AGA. L'algorithme commence la recherche des solutions optimales avec la NPSO pour accélérer la phase d'exploration. Ensuite, l'AGA est activé pour exploiter pleinement l'espace des solutions et ainsi éviter les minima locaux. Avec cette version d'hybridation de base (chromosomes AGA initialisés par NPSO), des résultats similaires avec l'AGA ont été obtenus tout en réduisant le temps d'exécution de l'AGA. Les résultats (figure 4.17) montrent que l'algorithme hybride nécessite moins d'itérations que l'algorithme AGA pour le problème d'optimisation tout en préservant des résultats (erreurs) comparables à ceux

obtenus par l'AGA qui offrent une meilleure base que ceux obtenus par d'autres méthodes mentionnées dans la présente partie du document. Le temps d'exécution de l'algorithme hybride a été considérablement réduit par rapport à l'AGA, car comme mentionné précédemment, 100 itérations dans PSO nécessitent seulement 3.44s. Dans les versions futures, nous envisageons d'utiliser NPSO à chaque fois que le processus de rafraîchissement est activé. Nous envisageons également une coopération renforcée pour la gestion de l'évolution des différentes sous-populations en exploitant plusieurs combinaisons d'opérateurs PSO et génétiques. L'idée derrière cette coopération est à l'origine d'optimiser la recherche via la combinaison de mécanismes. En outre, elle permet à une stratégie de niches à être introduite : si les sous-populations peuvent être mélangées à différents niveaux, elles explorent l'espace de recherche en parallèle, augmentant ainsi la probabilité d'atteindre l'optimum en trouvant des solutions multiples.

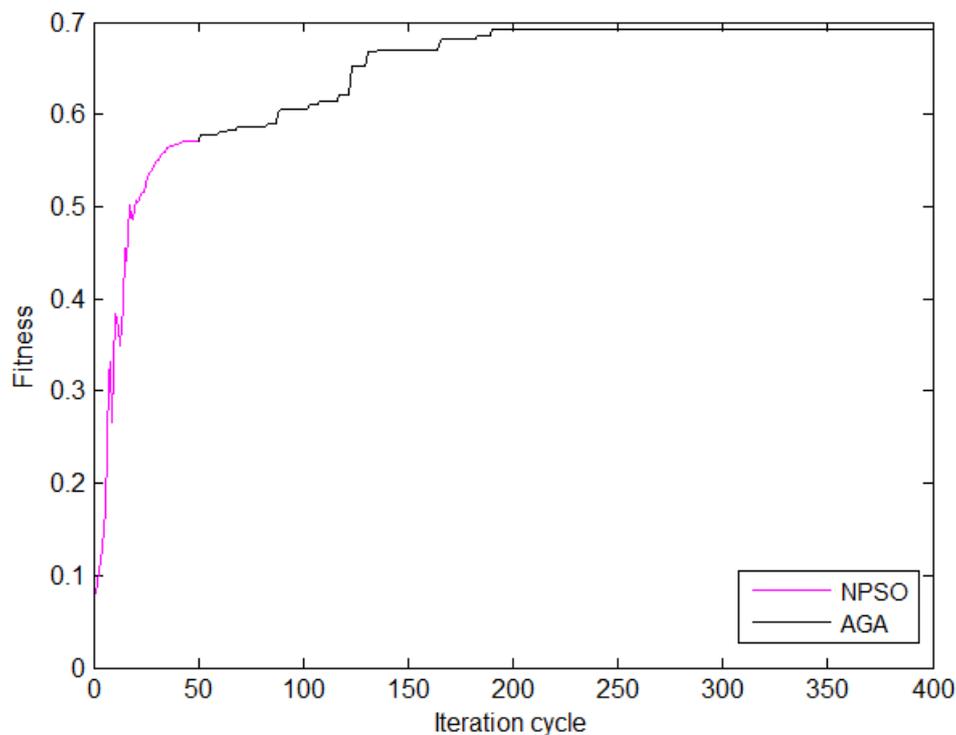


Figure 4.17 – Evolution de la fonction fitness dans l'algorithme hybride dans le cas d'un filtre RIF passe-bas d'ordre 40 ( $\omega_c = 0.5$ ).

### 4.3. Algorithme hybride pour la synthèse des filtres RIF 1D et 2D

Dans la synthèse des filtres RIF, il est nécessaire d'optimiser les coefficients du filtres afin d'approximer la réponse fréquentielle désirée en minimisant le critère d'erreur. Afin d'illustrer l'efficacité et l'applicabilité des approches proposées, cette section présente quatre problèmes de synthèse. Les deux premiers cas traitent la synthèse des filtres RIF passe-bas et passe-haut. Le troisième cas étudie l'exemple traité par Cen dans [Cen, 2007] pour comparer notre algorithme avec un autre algorithme hybride dédié à la synthèse des filtres RIF 1D. Le dernier problème est consacré à la synthèse d'un filtre RIF d'ordre 60 pour

montrer les performances de notre algorithme hybride quand il est appliqué à des filtres d'ordre élevé.

**Problème 1 :** Cet exemple traite la synthèse d'un filtre RIF ayant la réponse d'amplitude suivante

$$D1(\omega) = \begin{cases} 1 & |\omega| < 0.45 \\ 0.03 & |\omega| \geq 0.45 \end{cases}$$

Filtre RIF 1D passe-bas symétrique de longueur 41.

**Problème 2 :** Cet exemple traite la synthèse d'un filtre RIF ayant la réponse d'amplitude suivante

$$D2(\omega) = \begin{cases} 0.03 & |\omega| < 0.4 \\ 1 & |\omega| \geq 0.4 \end{cases}$$

Filtre RIF 1D passe-haut symétrique de longueur 31.

**Problème 3 :** Cet exemple traite la synthèse d'un filtre RIF passe-bas, de phase linéaire et d'ordre 22 avec une bande passante comprise dans l'intervalle  $[0, 0.3]$  et une bande atténuée comprise dans l'intervalle  $[0.5, 1]$ .

$$D3(\omega) = \begin{cases} 1 & |\omega| \leq 0.25 \\ 0.03 & |\omega| \geq 0.35 \end{cases}$$

**Problème 4 :** Cet exemple traite la synthèse d'un filtre RIF ayant la réponse d'amplitude suivante

$$D4(\omega) = \begin{cases} 1 & |\omega| < 0.5 \\ 0.03 & |\omega| \geq 0.5 \end{cases}$$

La longueur du filtre est 61.

#### 4.3.1. Performance de notre système évolutionnaire

Quatre versions de notre algorithme génétique hybride "HGA" (HGA<sub>1</sub> à HGA<sub>4</sub>), ont été testées selon le type d'approches locales utilisées. La procédure de sélectivité ( $P_{local}$ ) a été utilisée pour limiter les chromosomes qui subissent l'approche locale.

#### 4.3.2. Réglage des paramètres

##### 4.3.2.1. Paramètres génétiques de base

Ces paramètres sont pré-étalonnés et n'ont pas besoin d'être modifiés.

La taille initiale de la population  $Size_{pop}$  est fixée à 100 individus. Si la condition de convergence de premier niveau est atteinte de sorte qu'il n'y a pas d'amélioration sur le processus dévolution génétique pendant 50 générations, l'approche locale est appliquée sur les chromosomes générés par notre algorithme de sélection. Le cycle évolutif est répété

jusqu'à ce que la valeur de la fitness souhaitée soit obtenue ou la meilleure valeur de la fitness ne peut pas être améliorée au bout de 1000 générations.

Les valeurs des paramètres tels que la probabilité de croisement et de mutation déterminent, dans une large mesure, si un algorithme peut trouver une solution optimale avec rapidité, précision et diversité. Dans ce qui suit, nous énumérons les paramètres et leurs valeurs utilisées dans les essais.

#### 4.3.2.2. Paramètres de domaine

La valeur de la fonction de pondération  $W$  qui donne les meilleurs résultats est de 1 dans la bande passante et de  $(5/8)^2$  dans la bande d'arrêt. Ces valeurs ont été étalonnées après plusieurs essais pour différentes valeurs de  $W$  et pour différentes configurations de filtre.

#### 4.3.2.3. Paramètres de l'approche locale

Les paramètres des approches locales (SA, TS et HC) ont été déterminés à partir de plusieurs tests. Les valeurs de ces paramètres peuvent donner des algorithmes avec des performances optimales, c'est à dire réduire les erreurs et le temps de calcul nécessaires pour la convergence. L'approche locale est activée uniquement lorsque la valeur de fitness moyenne est supérieure à 0,5. La sélection des chromosomes pour la recherche locale a été menée dans le but d'assurer la diversité en sélectionnant les chromosomes de référence. La procédure est conduite par  $P_{local}$ ,  $k_r$  et  $d_c$  qui ont été calibrés à partir de plusieurs tests.

Le tableau 4.20 résume les paramètres de l'algorithme génétique hybride et le tableau 4.21 résume les paramètres et les processus de chaque variante de l'approche.

	Processus	Paramètres	Valeurs
<b>HGA</b>	AG Adaptatif (AGA)	<i>Probabilité de croisement</i>	$C_R = 100\%$
		<i>Procédure de rafraîchissement active</i>	$M_P = 20\%$
		<i>Mutation non-uniforme adaptative avec le processus de diversification</i>	$k_1 = 0.1\%$
			$k_2 = 1\%$
		<i>Présence de minimum local</i>	$k_1 = 5\%$
			$k_2 = 15\%$
	<i>Fonction de pondération</i>	$W = 1$ en PB	
		$W = (5/8)^2$ en SB	
	Simulated Annealing (SA)	<i>Température initiale</i>	$T_0 = 10000$
		<i>température finale</i>	$T_F = 10^{-4}$
		<i>Probabilité d'acceptation</i>	$k_B = 1$
			$Pr(\text{accepted}) = \exp(-\Delta E / k_B \cdot T)$
	Tabu search (TS)	<i>Liste tabou initiale</i>	$T = \emptyset$
		<i>Taille de la liste tabou <math>T</math></i>	$STL = 10$
		<i>Taille de sous-ensemble</i>	$SS = 10$
	Hill Climbing (HC)	<i>Taille de sous-ensemble</i>	$k = n_1 * n_2$
$SS = 10$			
Chromosomes locaux	$P_{local}$	4%	
	$k_r$	4	
			$\varepsilon_{g1,N} = 1\%$ $\varepsilon_f = 0.5\%$ $I_s = 50$ Ite

Table 4.20 – Paramètres de l'algorithme hybride HGA.

	Paramètres		
	Recherche locale	Chromosomes	AG Adaptatif (AGA)
HGA <sub>1</sub>	<input checked="" type="checkbox"/> Simulated Annealing (SA)	<input checked="" type="checkbox"/> Référence	<input checked="" type="checkbox"/>
HGA <sub>2</sub>	<input checked="" type="checkbox"/> Tabu Search (TS)	<input checked="" type="checkbox"/> Référence	<input checked="" type="checkbox"/>
HGA <sub>3</sub>	<input checked="" type="checkbox"/> Hill Climbing (HC)	<input checked="" type="checkbox"/> Référence	<input checked="" type="checkbox"/>
HGA <sub>4</sub>	<input checked="" type="checkbox"/> Simulated Annealing (SA)	<input checked="" type="checkbox"/> Sélection aléatoire	<input checked="" type="checkbox"/>
AGA <sub>5</sub>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SGA <sub>6</sub>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Table 4.21 – Paramètres de chaque approche.

### 4.3.3. Résultats préliminaires relatifs aux composantes de l’algorithme

Dans cette première partie, l’objectif est d’illustrer le rôle de plusieurs composantes de la méthode proposée. La première concerne l’effet de l’initialisation des chromosomes par les méthodes traditionnelles en minimisant le coût de calcul. La seconde concerne l’approche locale. Plusieurs aspects sont envisagés : d’abord, l’effet de l’hybridation de l’AG et le type de l’hybridation sur les résultats globaux obtenus; puis, nous soulignons l’importance de la sélectivité des chromosomes pour obtenir des résultats satisfaisants pour améliorer les performances et le coût de traitement.

#### 4.3.3.1. Effet de l’initialisation

Le tableau 4.22 donne les performances (le temps CPU et le nombre d’itérations nécessaires pour la convergence) en fonction de la fitness moyenne de la population initiale. Le tableau 4.23 donne les performances en fonction de la fitness maximale de la population initiale. La figure 4.18 résume ces résultats. De ces deux tableaux, on peut voir que le temps CPU et le nombre d’itérations dépendent à la fois de la moyenne de la fitness et du maximum de la fitness de la population initiale. Dès que les fitness initiales moyenne et maximale sont élevées, le temps CPU et le nombre d’itérations baissent. L’effet est reproductible (cette constatation est obtenue pour tous les tests effectués) mais il convient de souligner que le gain en coût de calcul demeure faible. Pour les différentes alternatives de sélection, le tableau 4.24 et la figure 4.19 résument l’évolution de la fitness maximale dans l’approche locale, basée sur le niveau de la fitness maximale au moment de l’activation de l’approche locale. On a constaté, en utilisant l’approche locale avec des chromosomes de référence, une évolution cohérente et une meilleure valeur de la fitness finale par rapport à la non utilisation des chromosomes de référence. La fonction fitness maximale évolue de 0.52 à 0.72 dans le cas d’une sélection aléatoire des chromosomes et de 0.52 à 0.77 dans le cas d’une sélection intelligente de chromosomes. Il convient de noter que cette tendance est moins importante si l’algorithme commence à partir de valeurs élevées de la fonction fitness, par exemple, lorsque l’algorithme hybride est proche de la convergence. La fonction fitness maximale évolue de 0.77 à 0.79 dans le cas d’une sélection aléatoire des chromosomes et de 0.77 à 0.81 lorsque les chromosomes de référence sont sélectionnés.

Fitness moyenne initiale	0.1	0.15	0.2	0.25
Nombre d'itérations	177	172	168	161
CPU time (sec)	144.03	139.87	136.65	130.54

Table 4.22 – Performances avec différentes valeurs de la fonction fitness moyenne de la population initiale.

Fitness maximale initiale	0.2	0.225	0.25	0.275
Nombre d'itérations	169	166	164	161
CPU time (sec)	136.86	134.49	132.92	130.54

Table 4.23 – Performances avec différentes valeurs de la fonction fitness maximale de la population initiale.

Niveau de la fonction fitness initiale		0.52	0.77
Fonction fitness finale	Recherche locale avec chromosomes de référence	0.77	0.81
	Recherche locale avec sélection aléatoire	0.72	0.79

Table 4.24 – Performances de la recherche locale pour différentes valeurs du maximum de la fonction fitness initiale.

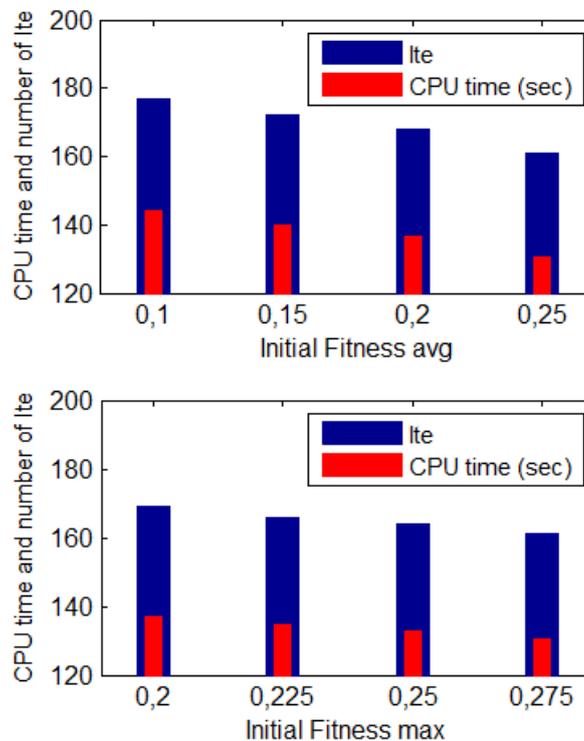


Figure 4.18 – Performance pour différentes valeurs du maximum de la fonction fitness de la population initiale

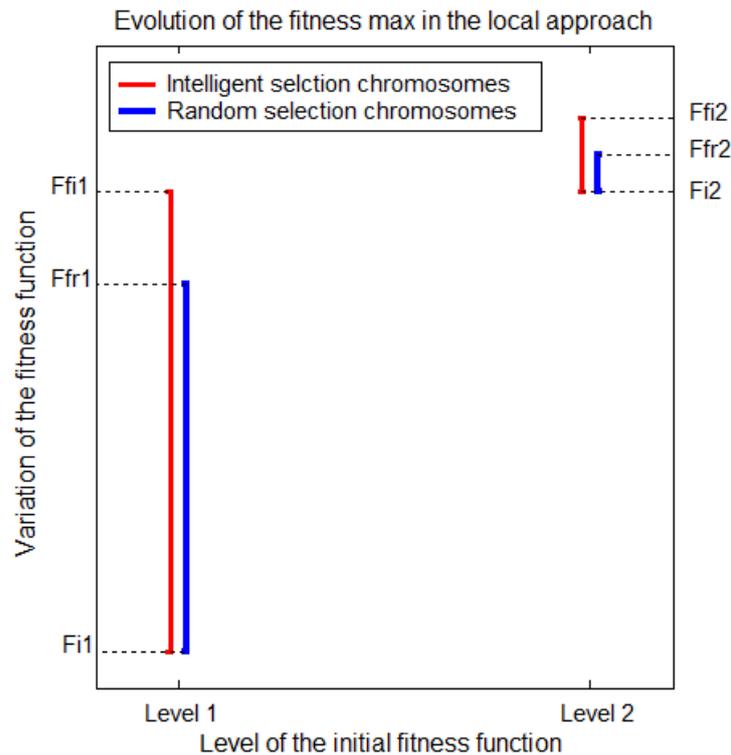


Figure 4.19 – Performance de l’approche locale pour différentes valeurs du maximum de la fonction fitness

#### 4.3.3.2. Coût de la recherche locale

Afin de souligner l’importance de notre processus de sélection, nous avons testé différentes versions de notre algorithme de sélection. La première consiste à sélectionner au hasard les chromosomes de l’approche locale. La seconde suit le même schéma que notre algorithme, mais fixe la taille du voisinage des chromosomes de référence : A chaque étape,  $k_r$  voisins les plus proches sont traitées pour constituer la zone d’influence.

Les résultats concernant la première version ne sont pas surprenants. Pour le même nombre de chromosomes, plus de générations génétiques sont nécessaires pour atteindre les mêmes résultats. Avec plus de chromosomes, le temps CPU est toujours plus élevé. Les résultats sont régulièrement comparables seulement quand une grande partie des chromosomes est considérée ce qui est très coûteux.

Concernant la deuxième version, les figures 4.20 et 4.21 montrent en outre les effets de la taille du voisinage et le nombre de chromosomes sur le comportement de l’algorithme HGA. Les résultats montrent que l’utilisation du nombre approprié de chromosomes et de la meilleure taille des voisins peut conduire à des améliorations significatives du temps d’exécution et de la qualité de la solution.

Il est à noter que le temps CPU augmente lorsque la taille du voisinage augmente et que l’influence de la taille du voisinage est minime, sur l’erreur moyenne et maximale.

Lorsque le nombre de chromosomes augmente, le temps CPU augmente tandis que l'erreur moyenne et l'erreur maximale varient légèrement. Ce résultat est un bon indicateur pour déterminer  $k_r$  et  $P_{local}$ .

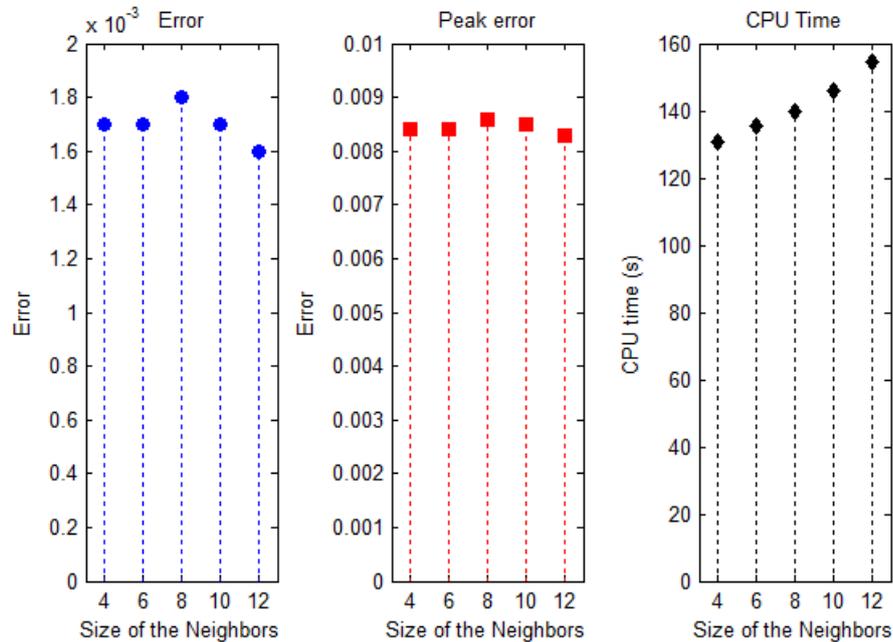


Figure 4.20 – Performances en fonction de la taille du voisinage.

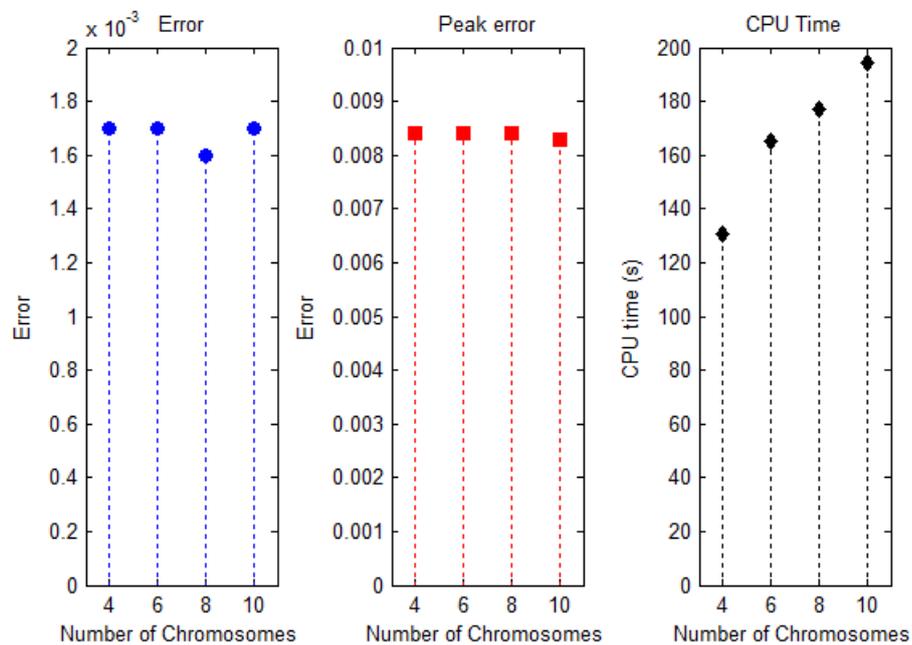


Figure 4.21 – Performances en fonction du nombre de chromosomes.

#### 4.3.4. Alternatives d'hybridation

L'utilisation des techniques d'optimisation hybrides impliquent les meilleures pratiques des deux algorithmes (algorithme principale et approche locale) et donc de réduire le temps de synthèse. Cela est vrai pour les quatre problèmes traités et quel que soit le type d'approche

locale utilisée. La fonction fitness est également considérablement améliorée parce que l'hybridation permet d'éviter aux chromosomes d'être piégés dans des minima locaux, donc de les guider vers la solution globale. Comme on le voit sur les figures 4.22 et 4.23, les filtres conçus par l'AG hybride (HGA<sub>1</sub>) et l'AG adaptatif (AGA<sub>5</sub>) ont des réponses de même allure dans toutes les régions de fréquence.

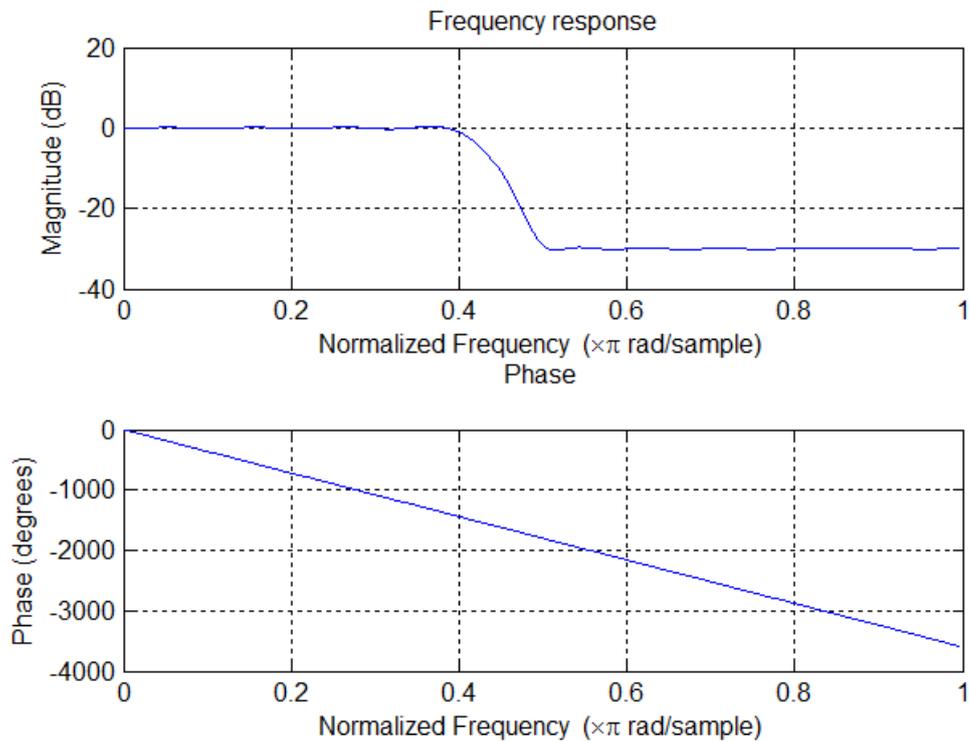


Figure 4.22 – Réponse fréquentielle (amplitude et phase) obtenue par HGA<sub>1</sub> (problème 1).

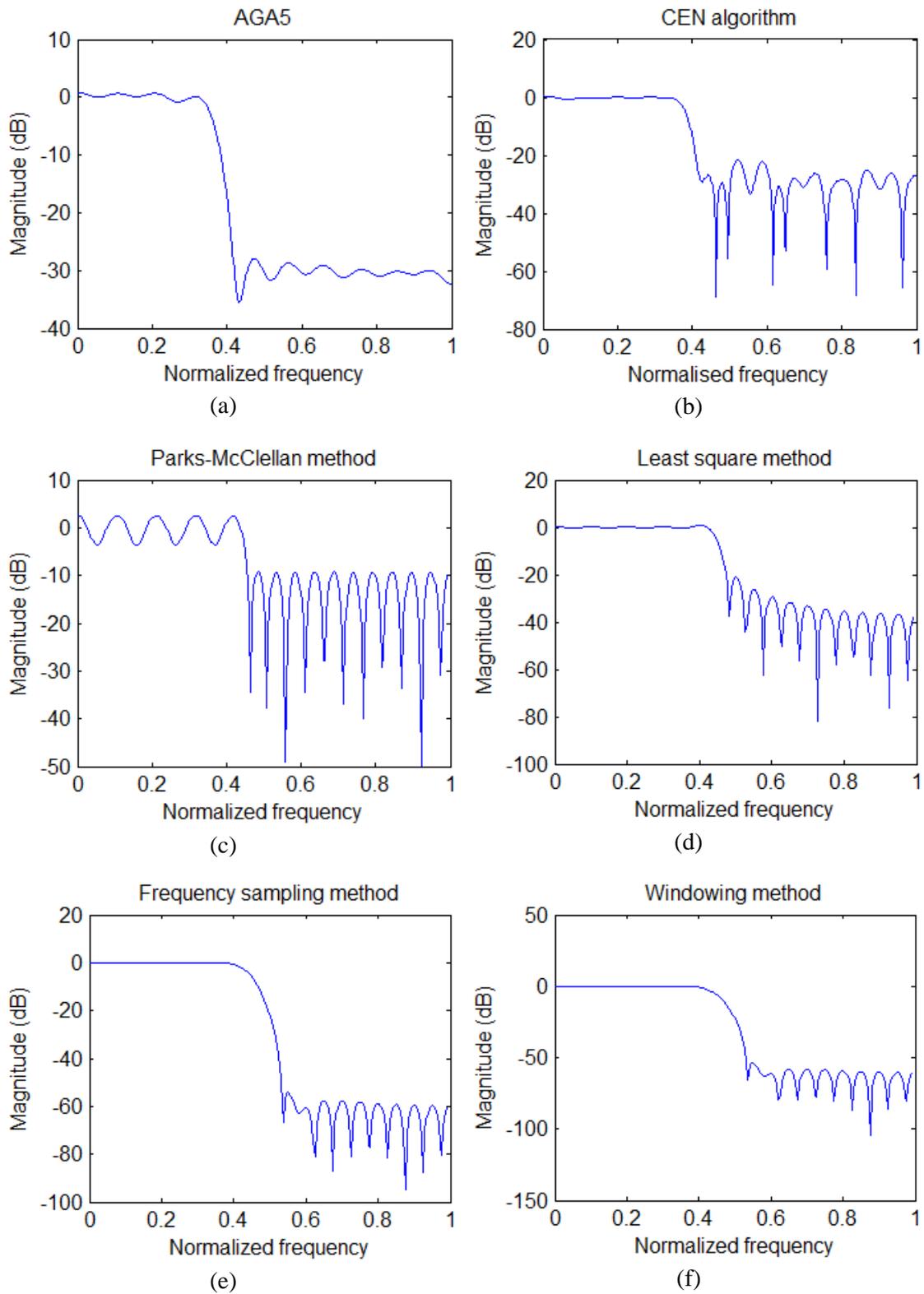


Figure 4.23 – Réponses d’amplitude : (a) AGA<sub>5</sub>, (b) Algorithme de CEN, (c) Méthode de Parks-McClellan, (d) Méthode des moindres carrés, (e) Méthode d’échantillonnage en fréquence, (f) Méthode des fenêtres (problème 1).

Les algorithmes ont été comparés en fonction de la vitesse de convergence et de l'évolution de la fonction fitness. La figure 4.24 montre l'évolution des meilleures solutions obtenues lorsque HGA<sub>1</sub> et SGA<sub>6</sub> sont employés. Il est clair que la fitness maximale dans l'algorithme HGA<sub>1</sub> passe de la valeur 0.52 à 0.77 lorsque l'approche locale est active (ce qui représente un bond de 30% de la valeur de la fitness finale). D'après les graphes de la figure, on peut voir que HGA<sub>1</sub> est nettement plus rapide que l'AGA<sub>5</sub> pour trouver les meilleurs coefficients du filtre. Par exemple, l'optimum est atteint après 161 itérations dans l'algorithme HGA<sub>1</sub>, l'AGA<sub>5</sub> non seulement a mis plus de temps (40 itérations de plus), mais aussi les erreurs produites sont supérieures à celles obtenues par l'algorithme HGA<sub>1</sub>. L'étude statistique réalisée montre que les solutions trouvées pour dix essais indépendants ne sont que légèrement différentes, ce qui nous confirme la régularité et l'optimalité de nos résultats. Parfois, pendant le processus d'évolution, les meilleures valeurs de fitness sont temporairement stabilisées à des optima médiocres ou que la population soit piégée dans des bassins isolés de l'espace de recherche (Fig. 4.24). Les différents mécanismes mis en œuvre permettent de briser ces états en se jetant dans un bassin de fitness supérieure. Dans ces cas, les meilleures valeurs de la fonction fitness sont rapidement augmentées, tandis que de nouveaux "chromosomes" sont introduits dans la population car ils envahissent ce nouveau bassin caractérisé par une meilleure fonction fitness. Par rapport à l'AGA<sub>5</sub> et SGA<sub>6</sub>, HGA<sub>1</sub> réalise non seulement une amélioration de la qualité de la solution, mais aussi une réduction de l'effort de calcul. L'intérêt de la recherche locale est d'aider l'AG Adaptatif (AGA<sub>5</sub>) à échapper aux optima locaux et empêcher la convergence prématurée. Notre algorithme de sélection pour l'approche locale peut accélérer la convergence en réduisant l'espace de recherche en fonction de la diversité de la population. Les erreurs moyenne et maximale sont respectivement d'environ 0.0025 et 0.0121 pour l'AGA<sub>5</sub> tandis qu'elles sont de 0.0017 et 0.0084 pour l'algorithme HGA<sub>1</sub>. D'après le tableau 4.25, les approches locales "recuit simulé" et "recherche tabou" donnent des résultats presque similaires, tandis que les résultats de l'approche "Hill Climbing" sont légèrement inférieurs en terme de performance que ceux obtenus par les deux autres approches locales.

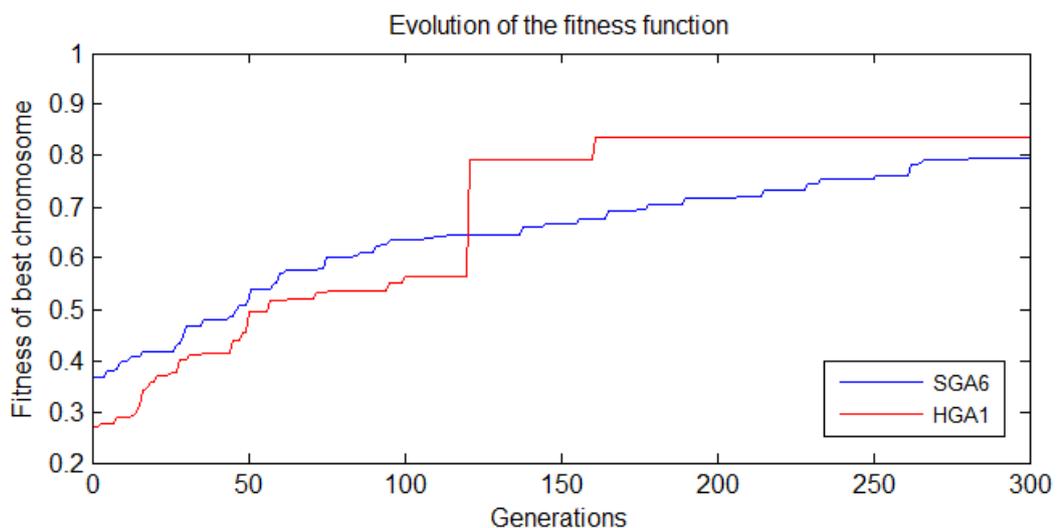


Figure 4.24 – Evolution de la fonction fitness

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
HGA <sub>1</sub>	0.0017	0.0084	0.050	130.54s	161
HGA <sub>2</sub>	0.0019	0.0089	0.050	135.02s	172
HGA <sub>3</sub>	0.0021	0.0093	0.051	146.17s	170
HGA <sub>4</sub>	0.0023	0.0103	0.051	140.98s	181
AGA <sub>5</sub>	0.0025	0.0121	0.058	153.29s	201
SGA <sub>6</sub>	0.0048	0.0295	0.064	175.61s	281
Algorithme de Cen	0.0068	0.0661	0.061	152.27s	197
Remez / Parks-McClellan	0.0499	0.0573	0.0908	< 2s	
Moindres carrés	0.0207	0.0827	0.0904	< 2s	
Echantillonnage en fréquence	0.0107	0.0954	0.1476	< 2s	
Méthode des fenêtres	0.0094	0.0779	0.1351	< 2s	

Table 4.25 – Paramètres de performance obtenus pour un filtres passe-bas (problème 1).

#### 4.3.5. Comparaisons avec d'autres approches

En commençant par Suckley [Suckly, 1991], plus d'une centaine de méthodes de conception de filtres RIF utilisant les AG's ont déjà été publiées, par exemple, Cemes et Ait - Boudaoud [Cemes et al., 1993], Lee [Lee et al., 1998] et Karaboğa et Çetinkaya [Karaboğa et al., 2003].

Nos résultats sont comparés avec ceux obtenus à partir des différentes approches

- Les approches évolutionnistes non classiques: Remez / Parks- McClellan, les moindres carrés, échantillonnage en fréquence et la méthode des fenêtres.
- L'approche hybride populaire de CEN [Cen, 2007]
- L'Algorithme Génétique Adaptatif (AGA) : L'AGA est le même que notre algorithme HGA mais sans l'aide de l'approche locale. Les paramètres de cet algorithme sont les mêmes que ceux de l'algorithme HGA, ici nous avons désactivé l'approche locale, à savoir l'algorithme est basé sur un AG Adaptatif.
- L'Algorithme Génétique Standard (SGA) : cet algorithme est appelé la norme car il ne comporte pas de mécanismes ou d'hybridation dédiés. Il utilise les mécanismes de base de l'évolution de l'algorithme génétique (croisement basé-direction, mutation standard, nouvelle population choisit parmi les meilleurs entre parents et enfants) avec des paramètres calibrés mais statiques. C'est une approche similaire à celle de Lu H-C et Tzeng S-T [Lu et al., 2000] appliquée à la conception de filtres RIF.

Afin de comparer l'algorithme HGA avec les méthodes proposées par CEN, nous avons choisi les mêmes paramètres de conception comme dans le papier de CEN [Cen, 2007] pour le quatrième exemple. Dix essais indépendants de chaque algorithme évolutionnaire ont été effectués. Les résultats moyens des dix exécutions sont donnés comme exemples dans ce document.

Le tableau 4.25 résume les résultats de l'analyse de performance pour le problème 1. Nous comparons nos résultats avec ceux obtenus par les méthodes citées précédemment. Le tableau 4.25 montre que les erreurs maximale et moyenne des filtres conçus par notre algorithme (0.0084 et 0.0017) sont plus faibles que celles du SGA6 (0.0295 et 0.0048) et de

l'algorithme de CEN (0.0661 et 0.0068) respectivement. Les réponses en fréquence sont présentées dans les figures 4.22 et 4.23. Les erreurs obtenues en utilisant l'algorithme de CEN sont comparables à celles obtenues par le SGA, tandis que le temps CPU et le nombre de générations sont similaires à ceux de l'algorithme génétique adaptatif. Les résultats obtenus en utilisant AGA<sub>5</sub> sont intéressants et révèlent l'importance des mécanismes mis en œuvre. Cependant, les versions des algorithmes HGA surpassent légèrement l'AGA<sub>5</sub> quelle que soit l'approche locale utilisée et pour les trois critères. Ils ont également besoin de moins de temps de calcul. Les approches non évolutives ne remplissent pas tous au même niveau les différents critères, mais leurs performances sont majoritairement loin des performances des algorithmes évolutionnistes. Alors que HGA<sub>1</sub> produit respectivement une erreur moyenne de 0.0017 et une erreur maximale de 0.0084, les erreurs correspondantes pour les approches Remez / Parks-McClellan et les moindres carrés sont respectivement (0.0499 et 0.0573) et (0.0207 et 0.0827).

Le tableau 4.26 compare la qualité de la solution pour les différentes approches et résume leurs performances pour le problème 2. La réponse en fréquence de HGA<sub>1</sub> est représentée sur la figure 4.25. Pour cet exemple HGA<sub>1</sub> donne également les meilleurs résultats. Grâce à notre processus de sélection, de meilleurs résultats sont obtenus et un temps de calcul réduit est atteint. De HGA<sub>1</sub> à HGA<sub>4</sub>, 18 itérations supplémentaires sont nécessaires et le temps CPU est augmenté de 11.37 secondes. La figure 4.25 montre que la réponse en fréquence satisfait aux spécifications de conception. La synthèse finale atteint une erreur d'ondulation (erreur maximale) de 0.0099.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
HGA <sub>1</sub>	0.0024	0.0099	0.073	129.37s	181
HGA <sub>2</sub>	0.0025	0.0099	0.071	136.11s	190
HGA <sub>3</sub>	0.0027	0.0101	0.078	145.23s	185
HGA <sub>4</sub>	0.0027	0.0107	0.080	140.74s	199
AGA <sub>5</sub>	0.0032	0.0118	0.082	152.01s	210
SGA <sub>6</sub>	0.0043	0.0261	0.085	173.25s	311
Algorithme de Cen	0.0058	0.0562	0.086	149.64s	205
Remez / Parks-McClellan	0.0688	0.0948	0.0859	< 2s	
Moindres carrés	0.0175	0.0601	0.0876	< 2s	
Echantillonnage en fréquence	0.0191	0.1878	0.1752	< 2s	
Méthode des fenêtres	0.0176	0.1501	0.1712	< 2s	

Table 4.26 – Paramètres de performance obtenus pour un filtre passe-haut (problème 2).

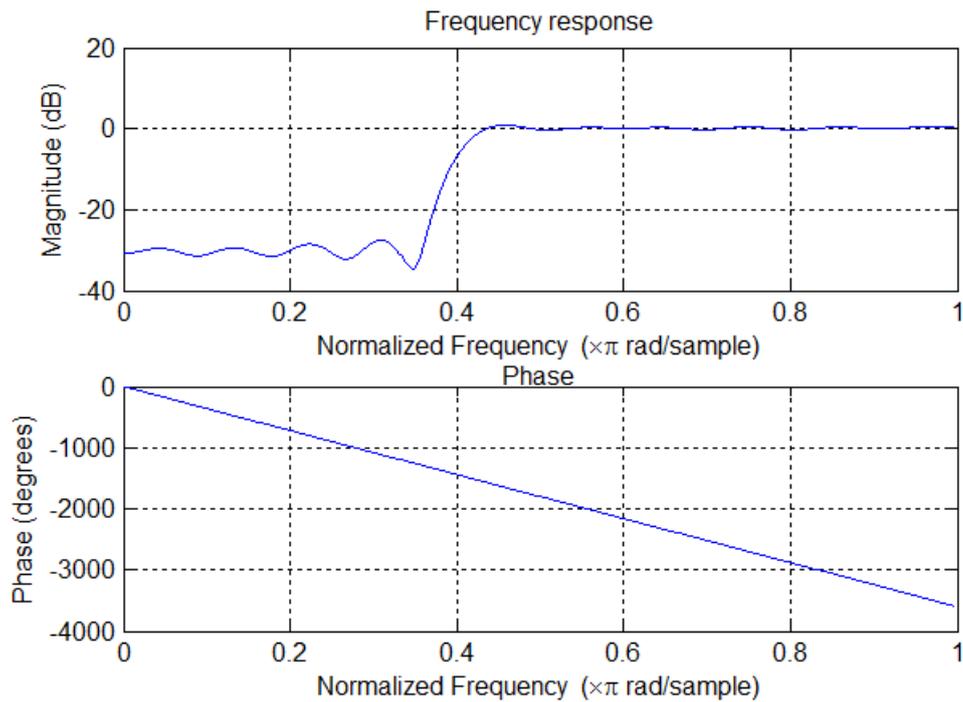


Figure 4.25 – Réponse en fréquence (amplitude et phase) obtenue par HGA1 (problème 2).

De la figure 4.26, on peut remarquer que la fitness maximale de l’algorithme HGA<sub>1</sub> atteint rapidement sa valeur finale car l’approche locale empêche la stagnation de l’algorithme. Dès que l’algorithme détecte une non-évolution de la fonction fitness, l’approche locale est activée pour permettre à l’algorithme d’évoluer.

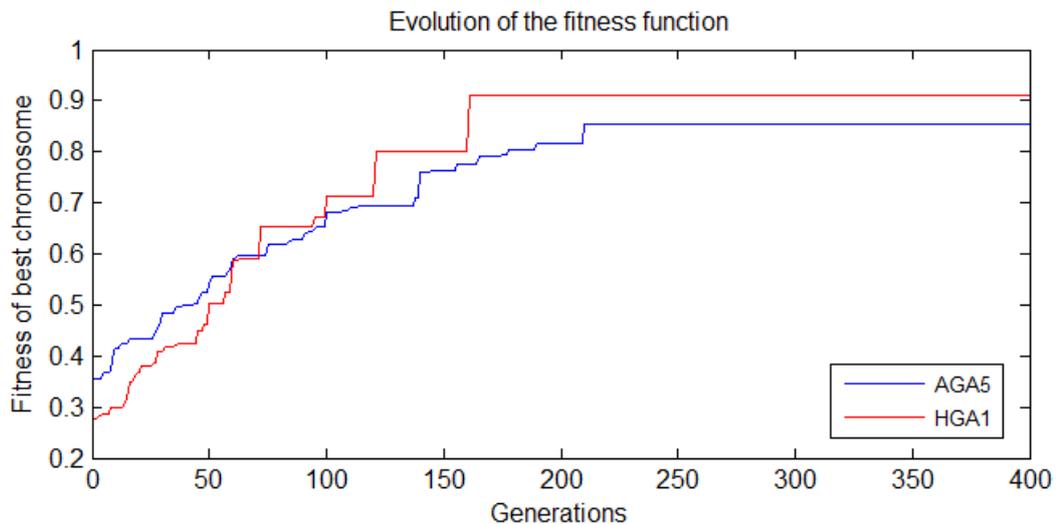


Figure 4.26 – Evolution de la fonction fitness.

Le tableau 4.27 résume les résultats de l’analyse de performance pour le problème 3 et la figure 4.27 montre la réponse fréquentielle obtenue par l’algorithme HGA<sub>1</sub>. Les résultats en terme de performance sont comparables à ceux obtenus pour les deux premiers problèmes.

Ce qui confirme l'avantage de l'introduction de la recherche locale et la stratégie de sélection par la procédure de rafraîchissement dans l'algorithme génétique, c'est-à-dire, l'évolution et la sélection des parents et des survivants.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
HGA <sub>1</sub>	0.0026	0.0108	0.080	110.32s	196
HGA <sub>2</sub>	0.0028	0.0109	0.080	115.13s	205
HGA <sub>3</sub>	0.0029	0.0110	0.083	126.33s	209
HGA <sub>4</sub>	0.0033	0.0113	0.083	121.65s	233
AGA <sub>5</sub>	0.0035	0.0132	0.092	131.45s	251
SGA <sub>6</sub>	0.0057	0.0309	0.099	151.28s	342
Algorithme de Cen	0.0074	0.0699	0.085	128.69s	241
Remez / Parks-McClellan	0.0524	0.0605	0.0908	< 2s	
Moindres carrés	0.0239	0.0887	0.0904	< 2s	
Echantillonnage en fréquence	0.0140	0.0991	0.1076	< 2s	
Méthode des fenêtres	0.0124	0.0814	0.1051	< 2s	

Table 4.27 – Paramètres de performance obtenus pour un filtre passe-bas (problème 3).

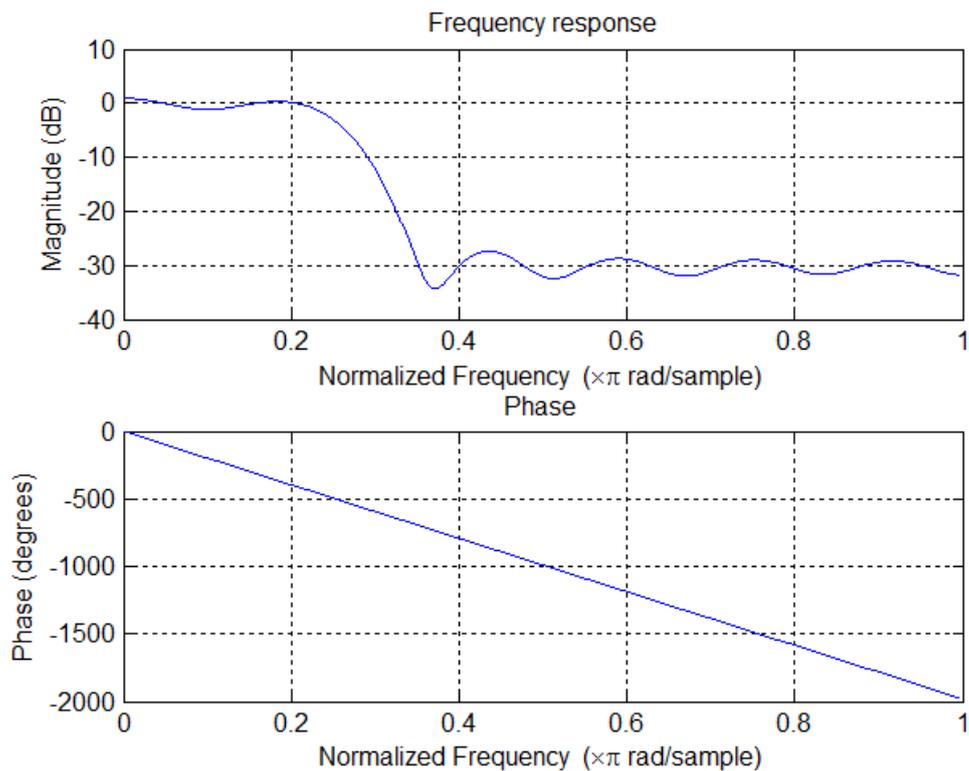


Figure 4.27 – Réponse fréquentielle (amplitude et phase) obtenue par HGA1 (problème 3).

Pour le problème 4, les performances de chaque algorithme sont résumées dans le tableau 4.28. Il aborde la conception de filtres d'ordre élevé. La réponse en fréquence est donnée dans la figure 4.28. Sans modifier les paramètres, les mécanismes de diversification et l'hybridation avec des approches locales ont amélioré les performances et ont permis

d'éviter de tomber sur des minima locaux. L'algorithme est capable de gérer les conceptions de filtres d'ordre élevé au même niveau que les autres filtres d'ordre moins élevé. Nous pouvons constater que les erreurs moyennes pour les versions HGA sont inférieures à 0.0031 et les erreurs maximales sont de l'ordre de 0.011. La différence notable avec les autres problèmes concerne le temps de convergence, qui augmente d'environ 20%. Il est intéressant de noter que la version sans les approches locales ou même sans le processus de sélection adaptative a plus de difficulté à converger. L'algorithme hybride HGA<sub>1</sub> donne des erreurs moyennes qui sont deux fois plus basses que celles données par le SGA et une erreur maximale qui est le tiers de celle du SGA. Pour tous les tests, l'algorithme hybride donne des résultats satisfaisants à chaque fois, même pour des filtres d'ordre élevé. En outre, nous avons obtenu une bande de transition très étroite pour ce type de filtre. Compte tenu du problème 4 qui traite un filtre d'ordre élevé (61), nous pouvons dire que notre algorithme hybride est également efficace pour les filtres d'ordre élevé.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
HGA <sub>1</sub>	0.0025	0.0111	0.032	151.01s	204
HGA <sub>2</sub>	0.0027	0.0117	0.040	155.24s	215
HGA <sub>3</sub>	0.0027	0.0116	0.041	166.00s	209
HGA <sub>4</sub>	0.0031	0.0119	0.046	161.33s	222
AGA <sub>5</sub>	0.0034	0.0143	0.052	172.72s	232
SGA <sub>6</sub>	0.0057	0.0325	0.060	191.14s	340
Algorithme de Cen	0.0070	0.0708	0.035	170.44s	221
Remez / Parks-McClellan	0.0511	0.0615	0.0604	< 2s	
Moindres carrés	0.0218	0.0892	0.0701	< 2s	
Echantillonnage en fréquence	0.0125	0.0998	0.0862	< 2s	
Méthode des fenêtres	0.0117	0.0822	0.0825	< 2s	

Table 4.28 – Paramètres de performance obtenus pour un filtre passe-bas (problème 4).

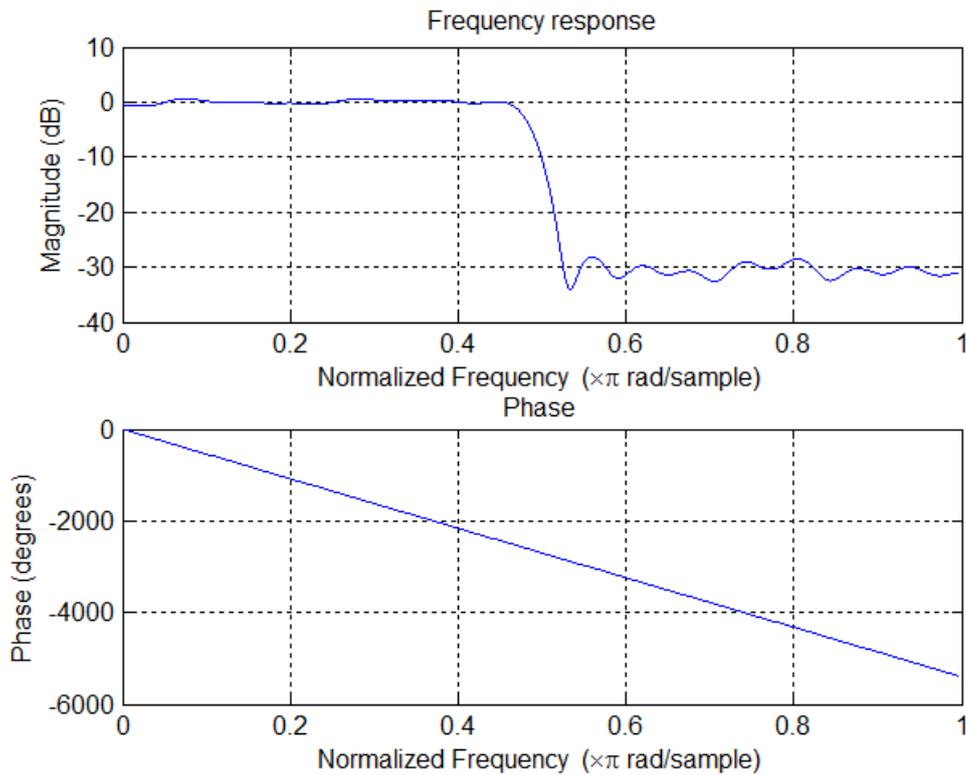


Figure 4.28 – Réponse fréquentielle (amplitude et phase) obtenue par HGA1 (problème 4).

#### 4.3.6. Application de l’algorithme hybride pour la synthèse d’un filtre RIF 2D

##### *Articulation entre les opérateurs génétiques et les mécanismes intelligents*

Le HGA présente des mécanismes intéressants et une certaine adaptativité à travers la mutation non-uniforme. Toutefois, comme pour la plupart des régimes existants, l’hybridation seule n’est pas suffisante tant que son rendement est largement influencé par les valeurs des paramètres et par l’articulation entre les paramètres et les opérateurs génétiques. Les performances de l’AG dépendent du réglage paramétrique pour atteindre une performance donnée. Afin d’éviter ces faiblesses, nous proposons une stratégie qui consiste à définir les paramètres qui donnent un rendement raisonnable pour un large éventail de problèmes, tout en offrant une certaine souplesse et adaptativité à notre algorithme. Pour atteindre cette exécution autonome, il faut distinguer les paramètres qui doivent être réglés et contrôlés. Pour rappel, le réglage des paramètres se réfère à des paramètres fixés dans la phase d’initialisation en tant que paramètres de contrôle qui subissent des changements pendant l’exécution de l’AG. Cela n’est pas suffisant. Les paramètres doivent être liés à l’ensemble de la stratégie et ils doivent être synchronisés avec les mécanismes impliqués. Notre objectif est de générer un ensemble de paramètres d’accordage qui tendent initialement à favoriser la sélectivité à l’encontre de la diversité et de promouvoir la diversité par l’intermédiaire des mécanismes.

Certes, la flexibilité de l’opérateur de mutation est une caractéristique intéressante néanmoins elle ne peut pas gérer toute seule les problèmes de convergence prématurée. Le

processus de sélection avec diversification qui élimine les chromosomes similaires, la procédure de croisement et la mutation non-uniforme sont tous des procédures auto-adaptatives. Tous ces paramètres sont appelés paramètres de contrôle. Les mécanismes interagissent les uns avec les autres et sont complémentaires : certains sont récurrents alors que d'autres ne sont activés que dans le cas de perte de contrôle, lorsque la population n'évolue pas de façon normale. Ils peuvent s'auto-régler et s'adapter aux progrès de la génétique, puis mettre l'AG à tout moment dans une configuration dans laquelle il peut évoluer.

Lorsque la sélectivité est favorisée, la population a tendance à converger rapidement, et elle peut alors converger à un optimum local et dans le cas l'extrême conduire à une convergence prématurée. Par conséquent, nous proposons comme nouveauté dans notre approche de distinguer l'état où la population évolue d'une manière normale de l'état où la population a un comportement spécial.

Concrètement, trois états sont considérés :

- Evolution de manière normale
- Présence d'un minimum local
- Présence (détection) de convergence prématurée

De toute évidence, la pertinence de l'AG est réalisée grâce à la stratégie et non par des paramètres initiaux qui jouent un rôle secondaire. La seule condition est que les paramètres de réglage  $k_1$  et  $k_2$  (quand la population évolue d'une manière normale) doivent permettre un niveau suffisant de pression pour éviter une recherche aléatoire alors que la stratégie a été créée pour maintenir un niveau donné de la diversité. La force de notre proposition est la manière dont les mécanismes sont adaptés à travers les paramètres alors que la recherche progresse. Ces mécanismes offrent une réelle flexibilité et simplicité pour l'utilisateur, car ils permettent à l'AG d'être calibré et à plusieurs reprises de s'auto-étalonner.

L'objectif de cette section est d'évaluer la contribution de l'hybridation et des différents mécanismes mis en place pour la synthèse des filtres RIF 2D.

L'exemple suivant montre les résultats de l'application de notre approche hybride pour la conception de filtre RIF 2D. Le filtre conçu a les mêmes spécifications que le filtre traité dans le premier exemple de la section 4.1.

La réponse en amplitude du filtre passe-bas est donnée par l'équation suivante :

$$D(\omega_1, \omega_2) = \begin{cases} 1 & |\omega_{1,2}| < 0.5 \\ 0 & |\omega_{1,2}| \geq 0.5 \end{cases}$$

Le filtre est symétrique et de dimensions 7\*7.

Les résultats de la synthèse en termes d'erreur moyenne et d'erreur maximale par rapport à la réponse en fréquence souhaitée, de largeur de bande de transition et de temps CPU sont représentés dans le tableau suivant.

Algorithmes	Erreur moyenne	Erreur maximale	Largeur de la bande de transition	Temps CPU	Ite
HGA <sub>1</sub>	0.0091	0.0798	0.109	38.14s	205
AGA	0.0092	0.0803	0.113	49.74s	298
Simulated Annealing	0.0255	0.1320	0.174	108s	
Tabu Search	0.0570	0.2190	0.139	660s	
Hill Climbing	0.0810	0.3850	0.121	Aléatoire	
Remez (Transformation Fréquentielle)	0.0566	0.1668	0.163	< 2s	
Moindres carrés (Transformation Fréquentielle)	0.0424	0.1412	0.193	< 2s	
Echantillonnage fréquentiel	0.0551	0.2203	0.115	< 2s	
Méthode des fenêtres	0.0400	0.4920	0.322	< 2s	

Table 4.29 – Paramètres de performance obtenus pour un filtre passe-bas (Exemple 1, section 4.1).

On constate que les performances de l’algorithme hybride sont significativement meilleures que celles de toutes les autres méthodes traitées dans cette section. Nous pouvons même noter une diminution d’environ un tiers du nombre d’itérations nécessaires à la convergence par rapport à l’algorithme génétique adaptatif dédié à la synthèse des filtres RIF (AGA).

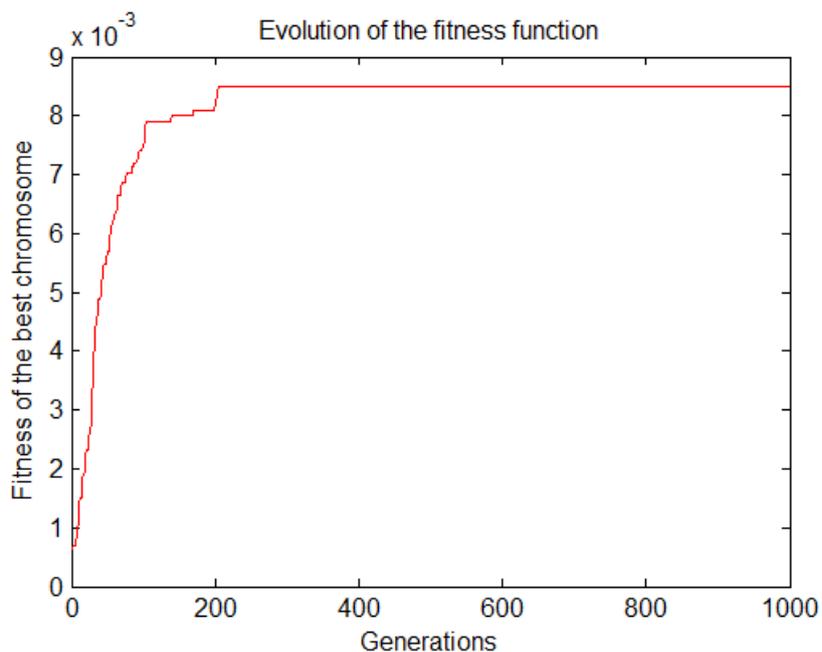


Figure 4.29 – Evolution de la fitness maximale dans l’algorithme HGA1 (Exemple 1, section 4.1).

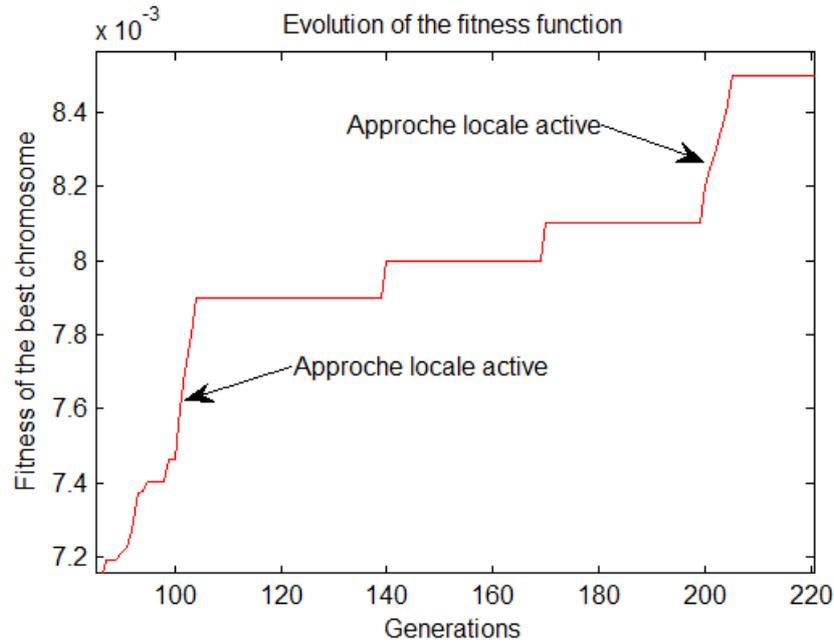


Figure 4.30 – Zoom sur les parties où l’approche locale est active dans l’algorithme HGA1 (Exemple 1, section 4.1).

Les figures 4.29 et 4.30 montrent l’évolution de la fitness du meilleur individu dans l’algorithme hybride HGA<sub>1</sub>. On note que l’approche locale est activée à deux reprises afin d’éviter la stagnation de l’algorithme. Les mécanismes de diversifications et de rupture sont activés après le premier et le deuxième pallier alors que l’approche locale est activée une première fois aux alentours de l’itération 100 et une seconde fois après le troisième pallier. Cette activation (de l’approche locale ou des mécanismes de rafraîchissement) est gérée par les conditions citées dans les sections précédentes.

Il est très clair de constater que l’apport en termes de variation de la fitness est plus significatif dans l’approche locale que dans les mécanismes de rafraîchissement.

Les résultats obtenus par les différentes approches et différents types de filtres montrent que les approches évolutionnaires sont plus efficaces que les approches conventionnelles pour la synthèse des filtres RIF. Les approches évolutionnaires sont meilleures en termes d’erreur moyenne, d’erreur maximale et de largeur de la bande de transition. Le seul critère où les méthodes conventionnelles sont plus efficaces est le temps CPU car ces méthodes ne sont pas des méthodes itératives.

De toute évidence, les AG’s sont entraînés sur la base de divers paramètres. Ces paramètres peuvent ne pas être optimaux pour toutes les configurations et ils ne peuvent pas avoir la même efficacité dans les différents états des progrès de la génétique. L’étalonnage des paramètres optimaux est fastidieux. L’étalonnage est toutefois suffisamment précis pour que l’AG soit toujours conduit correctement. L’espace des possibilités est réduit et la recherche génétique facilitée. Les mécanismes sont indispensables car ils permettent de placer l’AG à tout moment dans une configuration dans laquelle il peut évoluer. Nous pouvons dire que l’AG est contrôlé via certains mécanismes ou opérations à suivre (opérateurs de mutation, schéma de sélection). D’autres sont activés uniquement en cas de perte de contrôle (la forte

mutation, la procédure de diversification et de rupture). Cela permet à la majorité des paramètres à être pré-calibrée, ne laissant qu'une petite série dédiée aux utilisateurs. Nous pouvons considérer cela comme un système auto-adaptatif basé sur un processus dans lequel les paramètres sont implicites, c'est à dire qu'ils sont choisis par le cycle évolutif lui-même. Ils ne nécessitent aucune intervention de l'utilisateur ou tout réglage avant l'exécution. Plus intéressant encore, nos mécanismes offrent une grande souplesse à l'AG, contribuant ainsi à sa facilité d'utilisation qui est le vrai défi aujourd'hui. L'équilibre entre la pression de la sélection et la diversification est obtenu grâce à la stratégie et non par des paramètres qui doivent être modifiés au cours du processus évolutif. Cela permet d'éviter un certain nombre de problèmes pathologiques potentiels qui peuvent surprendre les utilisateurs imprudents et les décourager.

Plusieurs filtres RIF ont été testés et les résultats montrent que le HGA est plus performant que l'AG et les méthodes classiques. Le processus de diversification est en mesure d'atteindre une solution optimale en évitant les optima locaux, la procédure de rafraîchissement est en mesure de parvenir à une solution optimale sans convergence prématurée et l'algorithme montre une capacité d'atteindre des valeurs de fitness élevées avec moins de générations pour tous les filtres testés.

Nous avons observé que le HGA est plus stable, plus rapide, plus exact, plus robuste et converge vers les solutions optimales. Il y a toujours un compromis entre le temps de calcul et la qualité de la solution. Si la qualité de la solution est la principale préoccupation et le temps de calcul est encore négociable, l'algorithme génétique hybride est le meilleur choix.

En résumé, notre HGA surpasse clairement les approches classiques, des résultats très pertinents peuvent être obtenus avec un coût de calcul très réduit. L'approche hybride atteint un niveau de performance proche de l'optimalité, elle surpasse toutes les autres approches évolutionnaires utilisées dans le présent document. Plus qu'un algorithme, le cadre que nous proposons donne de bons résultats et reproductible, ce qui est loin d'être le cas avec de nombreuses solutions existantes qui ne sont pas suffisamment auto-adaptatives.

#### **4.4. Résumé des performances pour la synthèse d'un filtre RIF 2D (Exemple 1, section 4.1 et 4.3.6)**

Afin de pouvoir analyser efficacement les performances de chaque approche, il a été reporté sur la figure suivante (Fig. 4.31) une synthèse des performances permettant ainsi d'exploiter et de comparer plus aisément les différentes approches. Certes, d'un point de vue global, le HGA et l'AGA sont meilleurs pour tous les critères, néanmoins pour le temps CPU les méthodes classiques s'avèrent plus performantes, à cause de leur nature non itérative.

Le graphe suivant donne un récapitulatif des différents paramètres de performances obtenus par chaque approche utilisée pour la synthèse des filtres RIF 2D.

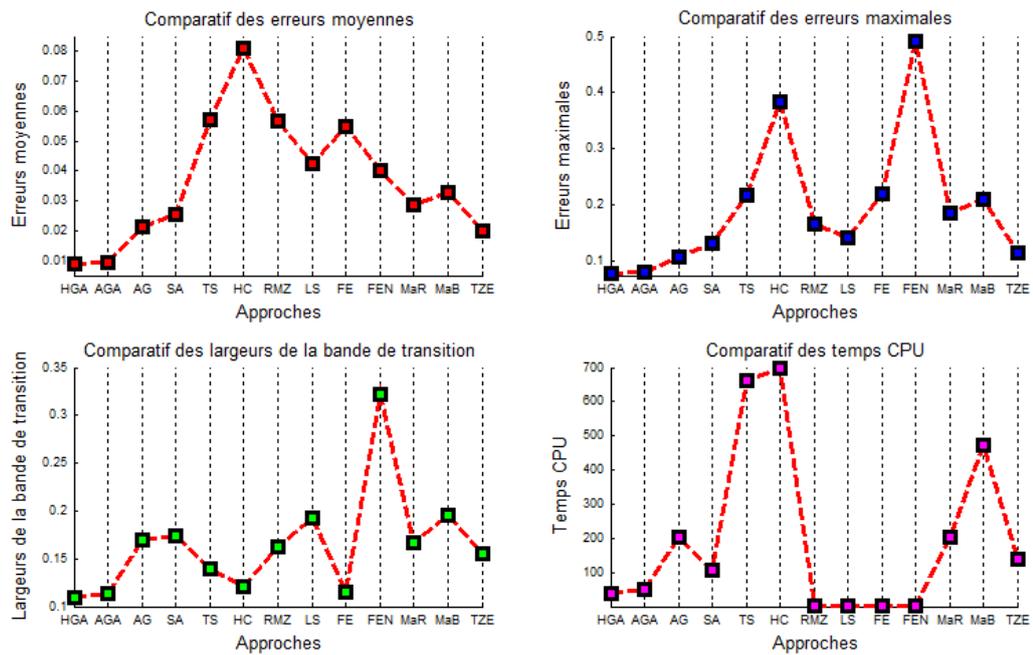


Figure 4.31 – Comparatif de performance entre les différentes approches

**HGA** : AG Hybride, **AGA** : AG Adaptatif, **AG** : Algorithme Génétique, **SA** : Recuit Simulé, **TS** : Recherche Tabou, **HC** : Hill Climbing, **RMZ** : Méthode de Remez, **LS** : Méthode des Moindres Carrés, **FE** : Echantillonnage Fréquentiel, **FEN** : Méthode des Fenêtres, **MaR** : Algorithme de Mastorakis Réel, **MaB** : Algorithme de Mastorakis Binaire, **TZE** : Algorithme de Tzeng.

## 4.5. Quantification

La conception complète des filtres RIF requiert 4 étapes essentielles : - Spécification du filtre - Calcul des coefficients - Choix de la structure - Mise en œuvre. La synthèse traite les deux premières étapes mais nous tenons à aborder les erreurs engendrées lors de la quantification des coefficients dans les deux dernières phases.

La conception pratique des filtres RIF peut être réalisée à l'aide de logiciels ou implémentée matériellement. Peu importe le procédé de conception, le problème le plus connu qui peut être engendré par ces procédés est l'effet des longueurs de mot (de quantification). L'un des objectifs, lors de la conception des filtres, est de réduire ces effets de mot de longueur finie, autant que possible, répondant ainsi aux exigences (spécifications du filtre). Il est possible d'utiliser soit des mots en virgule fixe ou en virgule flottante. Les deux représentations des nombres ont des avantages et des inconvénients. Dans cet exemple, la représentation en virgule fixe est utilisée pour enregistrer les coefficients et les échantillons en mémoire.

	4 bits	8 bits	16 bits
Erreur min de quantification	$4.3167 \times 10^{-4}$	$1.0350 \times 10^{-4}$	$3.6465 \times 10^{-7}$
Erreur moy de quantification	0.0109	$8.5058 \times 10^{-4}$	$3.7351 \times 10^{-6}$
Erreur max de quantification	0.0309	0.0018	$6.7033 \times 10^{-6}$
Ecart-type	0.0085	$5.5287 \times 10^{-4}$	$1.7410 \times 10^{-6}$
SNR (Signal-to-Noise Ratio)	138.5496	$3.2502 \times 10^{+4}$	$3.2777 \times 10^{+9}$
SNR (dB)	49.3123	103.8906	219.1040

Table 4.30 – Erreurs en coefficients produites dans le cas d'une quantification en représentation en virgule fixe.

On peut noter que la précision augmente à mesure que le nombre de bits est élevé.

Les filtres RIF conservent leur caractéristique de phase linéaire après quantification. Le filtre RIF à phase linéaire est symétrique, ce qui signifie que les paires correspondantes de coefficients seront quantifiées à la même valeur. Elle se traduit par la symétrie de la réponse impulsionnelle restant donc inchangée.

	4 bits	8 bits	16 bits
Erreur min de quantification	$7.6049 \times 10^{-4}$	$3.0350 \times 10^{-6}$	$2.1390 \times 10^{-7}$
Erreur moy de quantification	0.071353	0.004620	$1.9327 \times 10^{-5}$
Erreur max de quantification	0.187988	0.017565	$7.5584 \times 10^{-5}$
Ecart-type	0.051486	0.004578	$1.7458 \times 10^{-5}$
SNR (Signal-to-Noise Ratio)	161.3676	$2.0401 \times 10^{+4}$	$1.4034 \times 10^{+9}$
SNR (dB)	50.8368	99.2338	210.6220

Table 4.31 – Erreurs en réponses fréquentielles produites dans le cas d’une quantification en représentation en virgule fixe.

La position des zéros des filtres RIF change, alors que la position de ses pôles reste inchangée car ils sont situés en  $z = 0$ . La quantification est sans effet sur les pôles. La conclusion est que la quantification des coefficients des filtres RIF ne peut pas rendre les filtres RIF instables, comme c’est le cas avec les filtres RII.

#### 4.6. Contributions de la recherche

L’algorithme génétique hybride et l’algorithme génétique adaptatif sont des approches heuristiques avec deux avantages principaux : ils ont une convergence rapide, et nécessitent seulement quelques paramètres de contrôle. Mais la performance de ces approches dépend de ses paramètres et peut être influencée par la convergence prématurée et le problème de stagnation. Pour surmonter ces problèmes, l’AG et le HGA ont été modifiés et optimisés dans ce travail et sont utilisés pour la conception de filtres RIF. Plusieurs opérateurs nouveaux ont été introduits avec succès dans l’algorithme proposé pour améliorer ses performances. L’algorithme évalue l’interaction entre la pression de sélection et la procédure de diversification (la stratégie). Ces opérateurs, bien adaptés à notre problème, peuvent trouver des applications dans de nombreux autres domaines d’optimisation et de nature mathématique similaire.

Un algorithme génétique hybride est introduit pour trouver la solution optimale au problème de conception des filtres RIF 1D et 2D. Le HGA et l’AGA proposent une nouvelle définition pour les opérateurs de sélection, d’évolution et la qualité de la solution est améliorée. Une comparaison des résultats de simulation révèle l’efficacité de l’optimisation et la supériorité de l’algorithme sur les techniques d’optimisation en vigueur pour la résolution des problèmes de conception des filtres RIF. Les résultats des simulations indiquent clairement que le HGA possède les meilleures performances en termes de réponse en amplitude, de minimum d’ondulations dans la bande passante et d’arrêt, et d’atténuation maximale avec la plus étroite largeur de la bande de transition.

L’algorithme a la capacité de déterminer automatiquement les paramètres génétiques appropriés devenus plus précieux pour les plus complexes des problèmes de conception. Compte tenu de cette capacité unique, le HGA élaboré offre un grand potentiel pour la

conception améliorée des filtres numériques RIF. Avec la poursuite du développement, l'approche présentée ici est susceptible de devenir un outil clé pour la communauté des concepteurs de filtres numériques. Ainsi, le HGA peut être utilisé comme un bon optimiseur pour calculer les coefficients optimaux pour n'importe quel problème pratique de conception de filtres numériques pour les systèmes de traitement du signal. Par rapport aux techniques déjà publiées, la méthode a montré de bonnes performances et elle est adéquate pour une utilisation dans d'autres problèmes liés à la conception.

## 4.7. Application des filtres conçus pour le traitement d'images

Le traitement des images constitue l'une des grandes orientations du traitement de l'information. Il désigne un ensemble de méthodes dont l'objectif est soit de transformer des images et soit d'en extraire de l'information.

Le filtrage est considéré comme étant l'opération fondamentale en traitement d'images. Il permet de réduire le bruit, de compenser les défauts de capteur, d'améliorer la perception de certains détails ...

Le filtrage de l'image  $I$  par le filtre  $h$  est une image  $F$  dont les luminances sont données par :

$$F(x, y) = \sum_{a, b} h(a, b) \cdot I(x + a, y + b) \quad (4.1)$$

### 4.7.1. Exemple de filtrage passe-bas et passe-haut

Dans cette section et pour nos exemples de filtrage, nous utilisons l'image de Lena.

Afin de voir les effets des filtrages passe-bas et passe-haut sur les images, l'image de Lena est filtrée par un filtre RIF 2D passe-bas puis par un filtre RIF 2D passe-haut.

Lors d'un filtrage par un filtre RIF, les pixels de l'image de sortie (filtrée) sont uniquement dépendants de la valeur des pixels de l'image d'entrée (originale) (pas de récurrence). Lors d'un filtrage par un filtre RII, les pixels de l'image de sortie (filtrée) sont calculés à partir de l'image présentée à l'entrée mais aussi en fonction de ceux du voisinage du pixel de l'image résultante (récurrence entre les pixels de l'image de sortie).

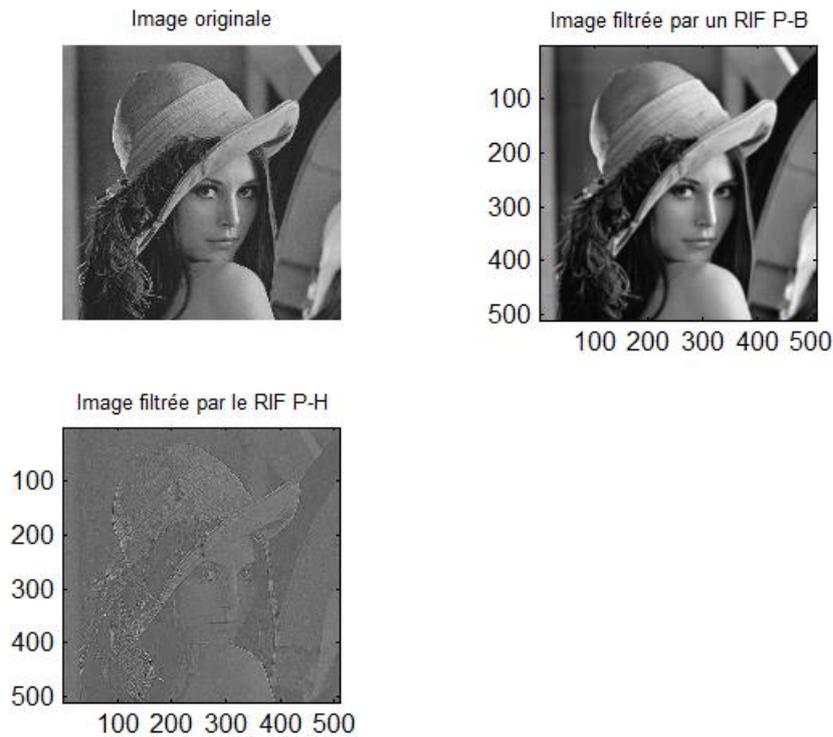


Figure 4.32 – Affichage des différentes versions de l’image de Lena.

A partir de la figure 4.32, on peut remarquer que le filtrage par un filtre RIF 2D passe-bas, de l’image de Lena, atténue les variations de luminance. L’effet du filtrage passe-bas réalisé ici est le lissage du contenu de l’image et la limitation des variations brusques d’intensité. Cet effet de lissage est très visible au niveau du chapeau et des cheveux (lissage de la partie représentant les cheveux, pas les cheveux au sens physiologique). En général, ces effets sont difficilement observables à l’œil nu et le filtrage passe-bas n’entraîne que peu de pertes en qualité visuelle.

Ce genre de filtre est utilisé typiquement pour réduire les effets du bruit, et pour supprimer le contenu haute fréquence (détails de l’image) avant un sous-échantillonnage (afin de limiter le risque d’aliasing ou d’apparition de fréquences parasites).

L’image filtrée par le filtre passe-haut met en évidence les variations de luminance qui caractérisent traditionnellement les contours des objets ou de la texture d’une image.

La figure 4.33 montre les histogrammes des différentes images de Lena. L’histogramme de l’image originale est constitué de pixels compris en majorité entre 0.1 et 0.7 car cette image possède peu de noir (pixels=0) et peu de blanc (pixels=1). L’image filtrée par le passe-bas possède des pixels compris majoritairement entre 0.2 et 0.75, ce qui confirme que le filtrage passe-bas réalisé ici apporte peu de modifications à l’image originale. Concernant l’image filtrée par un filtre passe-haut, on peut voir que la majorité du spectre est compris entre 0 et 0.1 car cette image possède beaucoup de pixels foncés (proches du noir) : image sombre.

Il faut noter qu’avec la fonction histogramme, on perd l’information spatiale contenue dans l’image (la forme ou les motifs de l’image).

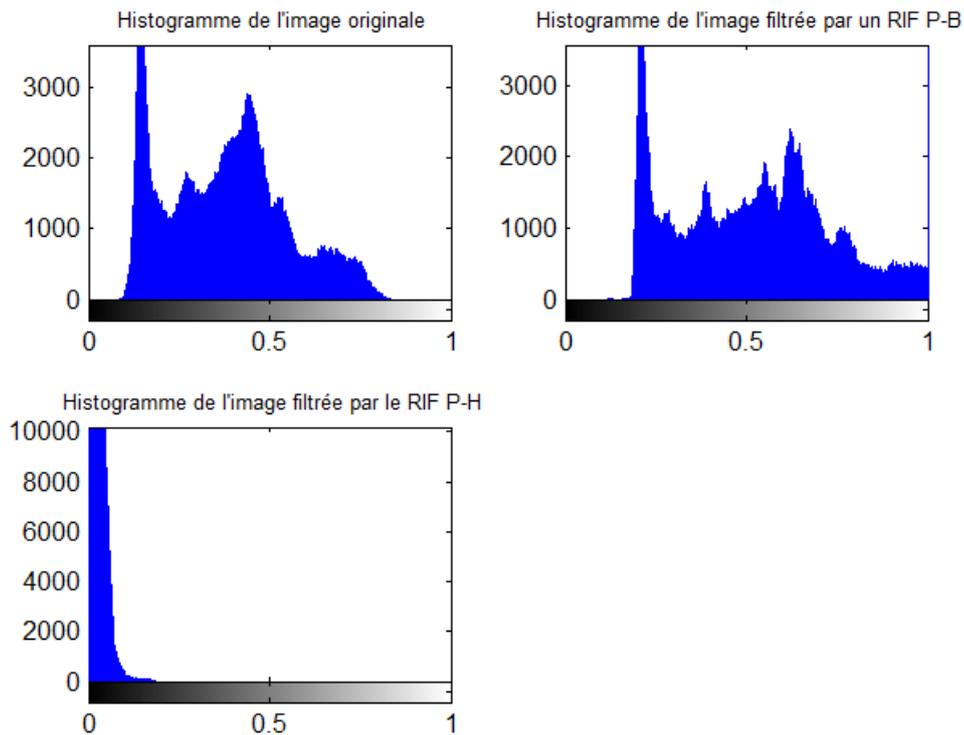


Figure 4.33 – Histogrammes des images de la figure 4.32

Comme pour la plupart des images naturelles, le spectre de l'image de Lena possède une forte composante continue, et très peu de composantes en moyennes et hautes fréquences spatiales. Compte tenu de la dynamique du module du spectre (valeurs importantes en basses fréquences), on a opté pour l'affichage en 3D du module du spectre et aussi pour l'affichage du logarithme décimal (base 10) de ce module. L'échelle logarithmique permet de visualiser avec précisions la valeur du module du spectre aux fréquences où ce module est très faible.

Du spectre de l'image avant filtrage on aperçoit une zone claire au centre du spectre, qui correspond à la composante continue de l'image ainsi qu'un nuage de points autour de ce centre qui correspond à quelques composantes moyennes et hautes fréquences de l'image (contours, détails, zones de transition).

Le spectre de l'image filtrée par un filtre passe-bas conserve la composante continue (le point central) et voit éliminer le nuage de points représentant les moyennes et hautes fréquences. Cependant, beaucoup de points apparaissent tout autour du centre engendrés par le module de la fonction de transfert du filtre. A partir du spectre de l'image filtrée par le filtre passe-haut, on observe un nuage de points plus dense aux fréquences moyennes et aux fréquences hautes. Cela est dû à l'atténuation des basses fréquences qui avaient un module très grand avant le filtrage.

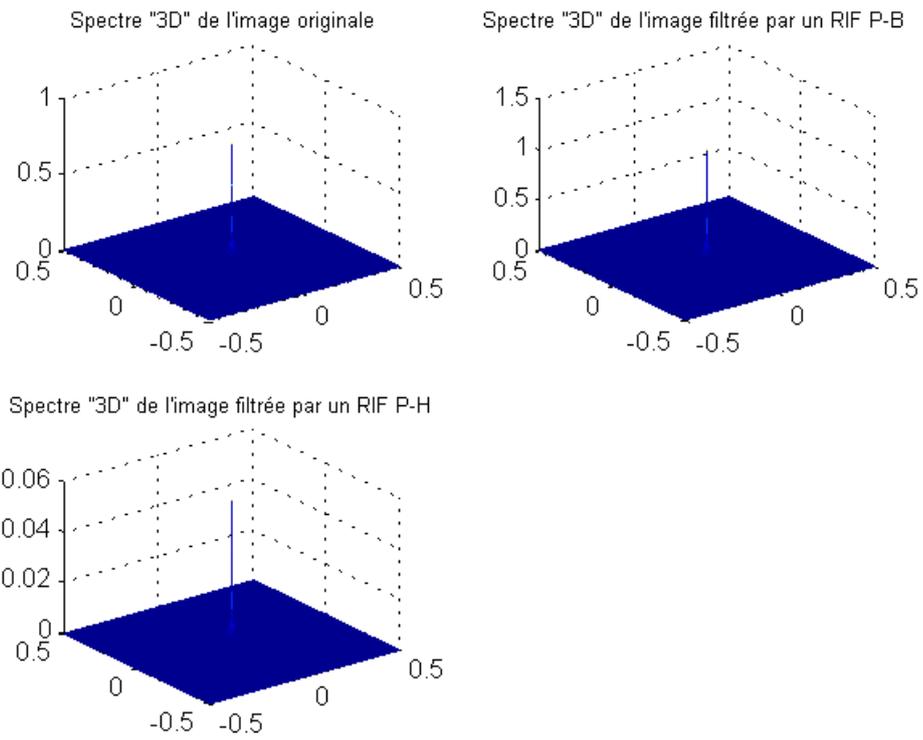


Figure 4.34 – Représentation 3D du spectre des images de la figure 4.32

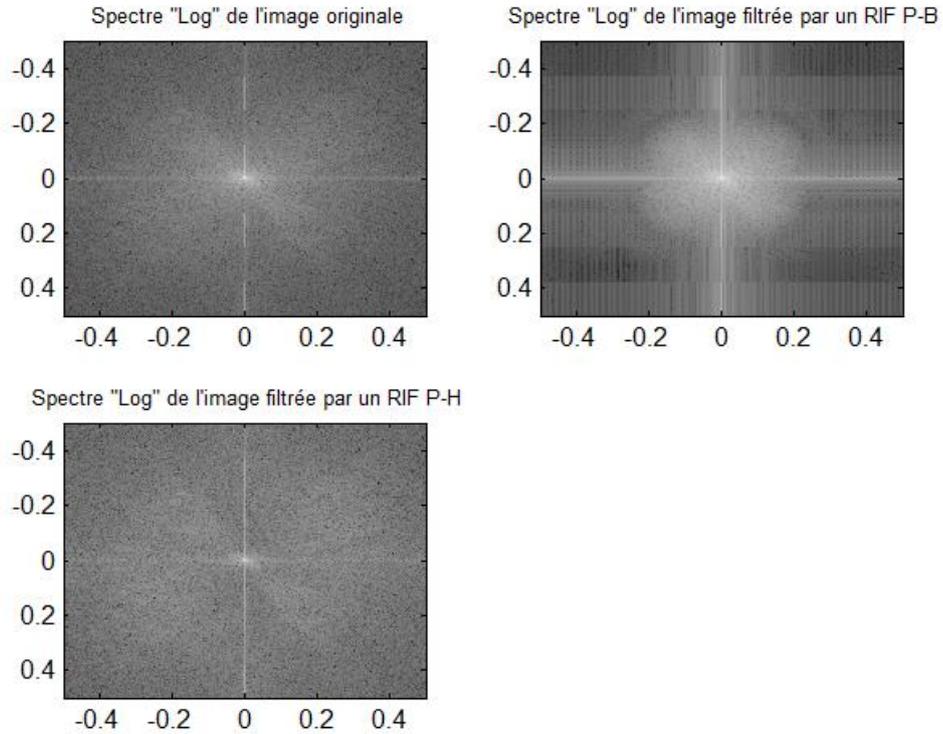


Figure 4.35 – Représentation du Logarithme du spectre des images de la figure 4.32

#### 4.7.2. Ajout d'un bruit à une image et élimination de ce bruit par filtrage passe-bas

On a ajouté un bruit haute fréquence à l'image de Lena, puis on a procédé à un filtrage passe-bas.

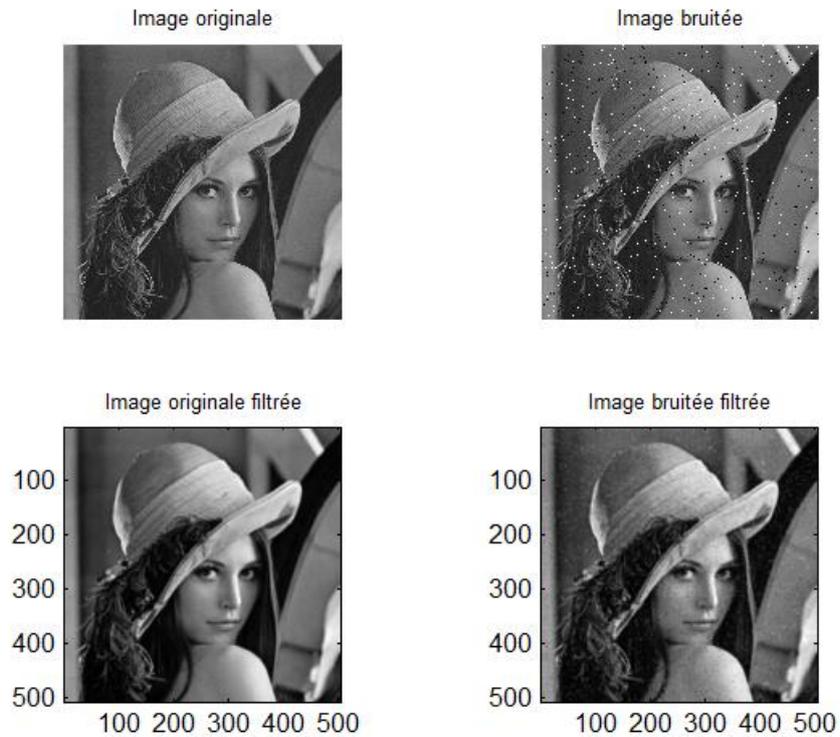


Figure 4.36 – Affichage des différentes versions de l'image de Lena.

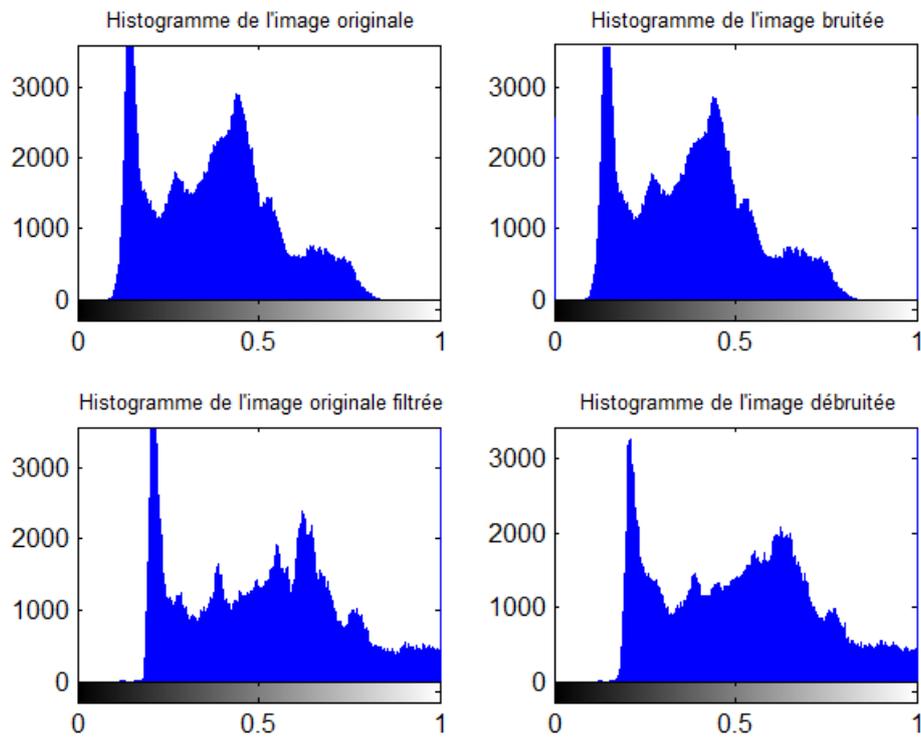


Figure 4.37 – Histogrammes des images de la figure 4.36

On constate que l'ajout d'un bruit "poivre et sel" (de haute fréquence) à l'image de Lena est très visible sur l'image elle-même mais ne produit presque aucun changement à l'histogramme par rapport à celui de l'image originale. Concernant les histogrammes après filtrage, on remarque aussi qu'il y a peu de différences entre ces deux histogrammes.

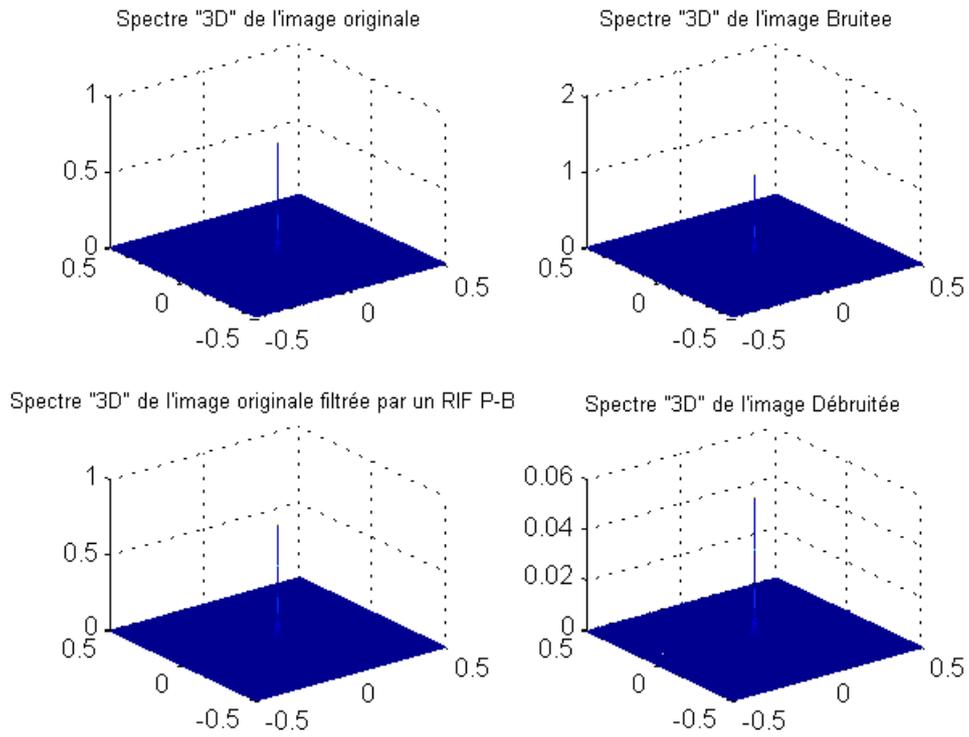


Figure 4.38 – Représentation 3D du spectre des images de la figure 4.36

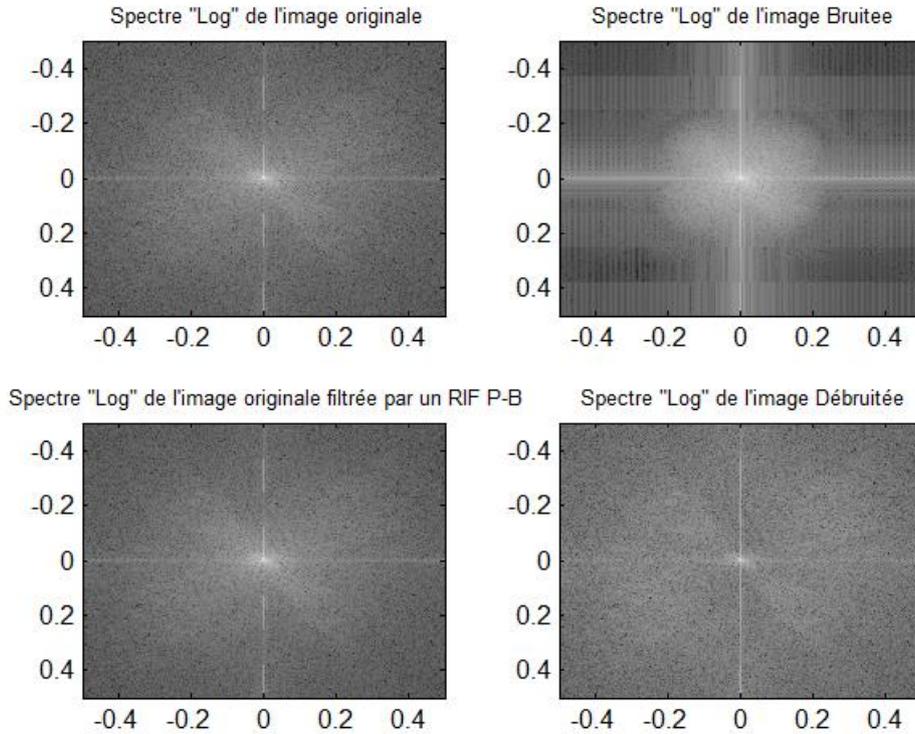


Figure 4.39 – Représentation du Logarithme du spectre des images de la figure 4.36

L'image bruitée possède un spectre ayant beaucoup de points aux moyennes fréquences et après filtrage (élimination du bruit par un filtre passe bas), on remarque que le nuage de points occupe toute la gamme de fréquence. La différence entre l'image originale filtrée et l'image bruitée filtrée réside dans la densité du nuage : cette densité est plus grande dans le cas de l'image bruitée filtrée.

#### 4.8. Conclusion

Notre algorithme génétique adaptatif dédié à la conception des filtres FIR 2D peut produire des filtres avec de bonnes réponses fréquentielles tout en réduisant considérablement les critères d'erreur et de temps CPU. Les expériences réalisées avec différents filtres 2D ont démontré l'efficacité et l'applicabilité de l'approche proposée.

Un nouvel algorithme génétique hybride dédié à la conception de filtres est proposé dans le présent travail. L'algorithme est essentiellement une combinaison de l'algorithme génétique adaptatif et des approches locales de telle sorte que nous pouvons utiliser efficacement la capacité stochastique de recherche par l'AG ainsi que les capacités de recherche locales d'une approche locale. Par la gestion des mécanismes spécifiques d'une manière innovante, efficace et l'optimisation du rôle de la recherche locale, notre processus d'évolution peut être accéléré sans perte de précision. En particulier, la capacité de l'algorithme pour déterminer le moment approprié pour appliquer les mécanismes de diversification et de l'approche locale est très efficace dans l'optimisation des coefficients des filtres et améliore les performances de conception. Ces améliorations contribuent à réduire l'espace de recherche en fonction de la diversification de la population et à aider l'AG à échapper à des optima locaux et empêcher aussi la convergence prématurée.

Plus précisément, notre proposition présente deux caractéristiques importantes nouvelles : d'abord, elle permet de générer seulement un petit nombre de chromosomes prometteurs à soumettre à la procédure locale. Cela évite d'appliquer la recherche locale à tous les chromosomes dans le cas extrême qui est pratiquement contraignant tout en préservant la qualité de la solution et le coût de calcul. La procédure de sélection est hautement auto-optimisée en termes de coûts de calcul (informatiques).

Les exemples confirment que notre HGA a une puissante capacité à trouver un optimum global. Les résultats obtenus par les différentes approches pour les différents types de filtres montrent que les approches évolutionnaires sont plus efficaces que les approches conventionnelles. Les approches évolutionnaires sont meilleures en termes d'erreur moyenne, d'erreur d'ondulation et de largeur de la bande transition. Le seul critère où les méthodes conventionnelles sont plus efficaces est le temps CPU, car ces méthodes ne sont pas des méthodes itératives. L'utilisation de l'hybridation avec des approches locales permet de réduire le temps de convergence (entre 5% et 20 %) par rapport à l'algorithme génétique adaptatif, même si plus de calculs sont nécessaires. Le niveau d'amélioration pourrait être encore meilleur, mais au détriment du coût de calcul qui reste la question cruciale.

# Conclusion générale

Cette thèse traite et présente les méthodes de synthèse optimale pour les filtres numériques RIF 1D et en particulier 2D sur la base de différentes versions d'algorithmes génétiques et d'algorithmes évolutionnaires. Notre approche de l'algorithme génétique adaptatif dédié à la conception des filtres RIF peut produire des filtres avec de bonnes caractéristiques de réponse tout en réduisant considérablement les critères d'erreur et les temps CPU nécessaires à la convergence. Les expériences réalisées avec différents filtres 2D ont démontré l'efficacité et l'applicabilité de l'approche proposée. Plus intéressant encore, nos mécanismes offrent une grande souplesse à l'AG, contribuant ainsi à sa facilité d'utilisation qui est le vrai défi aujourd'hui. Ceci permet à la majorité des paramètres d'être pré-calibrés, laissant seulement un petit ensemble dédié à l'utilisateur. Plus qu'un algorithme, le "système" que nous avons proposé donne de bons résultats et reproductibles, ce qui est loin d'être le cas avec de nombreuses solutions existantes qui ne sont pas suffisamment auto-adaptatives. L'utilisateur peut créer son propre filtre avec confiance, car il n'est pas confronté à toute la complexité du calcul évolutif.

Un avantage attrayant de l'algorithme PSO est sa mise en œuvre facile, à la fois dans le contexte de codage et de sélection des paramètres. L'algorithme est beaucoup plus simple et intuitif à mettre en œuvre que les opérateurs, de sélection et de probabilité de mutation complexes requises pour les algorithmes évolutionnaires telles que l'AG. L'avantage principal des versions avancées de la PSO est le faible nombre d'itérations nécessaires pour la convergence. On a noté que les versions améliorées de la PSO ont la capacité de converger à proximité de la meilleure solution optimale et possèdent les meilleurs temps de convergence en comparaison avec les AG's et les PSO de base. Une différence significative entre les versions sophistiquées des PSO et notre AGA réside dans le niveau d'imbrication de l'intelligence. Ils intègrent un mécanisme qui permet la diversification mais l'AGA va plus loin dans l'analyse en permanence de l'état de l'évolution et à plusieurs reprises s'auto-adapte en fonction de l'évolution de la recherche des solutions. Il est également évident, à partir des résultats obtenus et d'un grand nombre d'essais, que notre AGA est toujours apte à faire face aux convergences prématurées. Ainsi, on a montré que notre AGA peut être utilisé comme un bon optimiseur pour la recherche des coefficients optimaux des filtres RIF. Puis, nous avons mis en œuvre un algorithme génétique hybride initial avec la PSO dans le cadre de cette étude. L'idée est d'utiliser la capacité de la NPSO d'atteindre rapidement des zones intéressantes de l'espace de recherche qui peuvent être exploitées par l'AGA. Cet algorithme améliore le temps d'exécution de l'AGA par cette hybridation. En outre, il semble possible d'unifier les concepts derrière les NPSO et les AG's pour une amélioration de la coopération entre les chromosomes génétiques, une meilleure sélectivité dans la NPSO et une meilleure analyse de l'état de l'évolution et une adaptation intelligente des mécanismes intégrés. Nous pensons qu'il y a un potentiel à améliorer les versions de la PSO en intégrant des mécanismes plus intelligents comme dans l'AGA. L'hybridation approfondie entre les techniques est une direction intéressante, mais elle doit être bien gérée.

Un nouvel algorithme génétique hybride consacré à la conception de filtres est proposé dans cette thèse. L'algorithme est essentiellement une combinaison de notre algorithme génétique adaptatif et des approches locales de sorte que nous pouvons utiliser efficacement la capacité stochastique de la recherche génétique et les capacités des recherches locales. En introduisant des mécanismes spécifiques et en optimisant le rôle de la recherche locale, notre processus d'évolution peut être accéléré sans perte de précision. En particulier, la capacité de l'algorithme pour déterminer le moment approprié pour appliquer les mécanismes de l'approche locale et de diversification est très efficace dans l'optimisation des coefficients des filtres et améliore les performances de conception. Les exemples confirment que notre HGA a une puissante capacité à trouver les optimums globaux. L'utilisation de la recherche locale est d'aider les AG's à échapper aux optima locaux et à éviter la convergence prématurée. La notion de sélectivité mise en place pour l'approche locale accélère la convergence en réduisant l'espace de recherche en fonction de la diversification de la population.

Le HGA proposé apporte trois avantages en intégrant des stratégies de recherche locales dans l'évolution de la population de l'AGA. Tout d'abord, le processus de rafraîchissement contrôle la pression de la diversité de la population et de la sélection en tenant compte de la diversification et de l'intensification. Deuxièmement, cette hybridation des AGA avec une approche locale est indépendante de la sélection mère adoptée, du croisement, et de l'opérateur de mutation. Cela étend l'applicabilité du HGA à diverses variantes des AG's. Troisièmement, le HGA réduit de manière significative le coût de calcul de l'algorithme. Compte tenu de cette capacité, l'approche présentée ici pour la conception des filtres RIF offre un grand potentiel dans la conception des filtres numériques non récursifs.

La plupart des mécanismes et des opérateurs génétiques examinés dans cette thèse sont relativement bien connus par la communauté de la reconnaissance de formes. Très souvent, ils sont conceptuellement similaires mais généralement présentés par différents formalismes. Cependant, la limitation essentielle concerne l'absence de stratégie et d'organisation pour atteindre une mise en œuvre efficace tel que requis par les utilisateurs. Cette question est ouverte par nature à la façon d'intégrer une certaine expertise et de l'intelligence dans les systèmes. Nous avons prouvé dans ce document qu'une stratégie peut être formalisée et mise en œuvre. Elle donne la possibilité à notre HGA à converger vers la meilleure solution, et possède les meilleures caractéristiques de convergence par rapport aux autres algorithmes évolutionnaires. Puis, sur la base de cette nouvelle contribution, il semble possible de contester les chercheurs à travailler sur la génération de métaheuristiques capables d'unifier les progrès réalisés dans la conception des AG's pour atteindre de meilleures performances. L'idée pour la communauté de conception de filtres numériques est claire. Compte tenu de cette capacité unique, le HGA pour la conception de filtres offre un grand potentiel dans la conception améliorée de filtres numériques. Avec la poursuite du développement, l'approche présentée ici est susceptible de devenir un outil clé de la conception pour cette communauté.

### *Orientations futures et perspectives*

Globalement, la préoccupation majeure dans la conception génétique est liée à l'efficacité en termes de temps nécessaire pour parvenir à une solution de la qualité souhaitée. Très souvent, le temps d'exécution pour l'évaluation d'une fonction unique peut être encore de l'ordre de quelques minutes ou heures lorsque l'AG est exécuté par rapport à moins de quelques secondes lorsque les approches traditionnelles sont utilisées. Il est nécessaire de réduire cet écart si l'on veut démocratiser l'utilisation des AG's pour les problèmes du monde réel impliquant des coûts de calcul conséquents.

Trois causes sont bien connues pour expliquer cette faiblesse : d'abord, le choix incorrect des paramètres de contrôle. C'est une source de limitation des AG's en raison de son influence néfaste sur le compromis entre l'exploitation et l'exploration. Puis, il y a la qualité et la pertinence des opérateurs génétiques, et de l'initialisation. Pour les deux premiers, nous affirmons que les progrès arrivent à maturité et que le niveau d'amélioration devient limité. En ce qui concerne l'initialisation, elle peut être obtenue de manière aléatoire, ou en utilisant des informations spécifiques au problème. Si l'influence de cette initialisation est clairement connue, il n'existe pas de règle générale pour mettre en place la meilleure initialisation pour un problème donné. Nous pensons que cette question devrait être plus étudiée profondément car elle constitue une véritable source d'amélioration.

Lors de l'analyse du temps d'exécution dans le cadre d'un AG, il devient clair que la plupart du temps est consacré à l'évaluation de la fonction fitness elle-même. Deux mécanismes complémentaires semblent intéressants à étudier, même si leur mise en œuvre semble ne pas être aussi facile qu'il y paraît : la première consiste à réduire le coût de la fonction fitness en utilisant des méthodes plus approximatives pour la mesurer. Mais : comment adapter l'approximation de telle sorte que les informations délivrées sont suffisamment pertinentes à tous les stades génétiques ? La deuxième consiste à sélectionner certains chromosomes de référence à partir de la population par des approches de type clustering. Seuls ces chromosomes seront pratiquement évalués par des approches d'interpolation. Mais : Comment être sûr de la cohérence des procédures d'interpolation.

Pour le domaine de la conception des filtres, l'objectif crucial est d'obtenir des filtres de qualité supérieure. Les performances des filtres est multi-objectifs par nature et il n'existe aucun critère universel qui s'applique à tous les filtres, même si l'erreur quadratique reste un bon indicateur. Les objectifs précis dépendent du type de filtres et des contraintes d'ingénierie imposées par leurs applications. Notre plan d'avenir est d'étendre cet algorithme général à des algorithmes plus spécialisés en intégrant les différents objectifs dans une deuxième phase de l'algorithme. Le défi consistera à améliorer les performances globales sans affecter les fonctions conviviales de l'algorithme et réduire le nombre d'itérations pour obtenir une réponse acceptable en un temps minimal.



---

# Bibliographie

- [Ababneh et al., 2008] Jehad I. Ababneh, Mohammad H. Bataineh, Linear phase FIR filter design using particle swarm optimization and genetic algorithms, *Digital Signal Processing* 18 (2008) pp. 657–668
- [Ahmad et al., 1989] M. O. Ahmad, J-D Wang, An Analytical Least Square Solution to the Design Problem of Two-Dimensional FIR Filters with Quadrantly Symmetric or Antisymmetric Frequency Response, *IEEE Transactions On Circuits And Systems*, Vol. 36, No. 7, (1989)
- [Akramifar et al., 2010] S.A. Akramifar, G.Ghassem-Sani, Fast forward planning by guide denforced hill climbing, Elsevier, *Engineering Applications of Artificial Intelligence* 23 (2010) pp. 1327–1339
- [Apostolov, 2011] Peter Apostolov, Method for FIR filters design with compressed cosine using Chebyshevs norm, *Signal Processing* 91 (2011) pp. 2589–2594
- [Bellanger, 1987] M. Bellanger, "Traitement Numérique du Signal", Masson 3ème Edition 1987
- [Bellanger, 2006] M. Bellanger, "Traitement numérique du signal : théorie et pratique", 8<sup>ème</sup> édition, Dunod 2006
- [Bertsimas et al., 1993] D. Bertsimas, J. Tsitsiklis, Simulated annealing, *Statistical Science*, vol. 8, No. 1 (1993) pp. 10–15
- [Bhattacharya et al., 1999] D. Bhattacharya, A. Antoniou, Design of 2-D FIR filters by a feedback neural network, *Multidimensional Systems and Signal Processing* 10, Boston (1999) pp. 319–330
- [Blanchet et al., 1998] G. Blanchet, M. Charbit, "Traitement numérique du signal : simulation sous Matlab", Hermes, Paris 1998
- [Boeringer et al., 2005] D.W. Boeringer, D.H. Werner, D.W. Machuga, A simultaneous parameter adaptation scheme for genetic algorithms with application to phased array synthesis, *IEEE Transactions on Antennas and Propagation*, Vol. 53 (2005) pp. 356–371
- [Bohachevsky et al., 1986] Bohachevsky I., Jonhson M. E., Stein M.L., Generalized simulated annealing for function optimization, *Technometrics* 28 (1986) pp. 209–217
- [Boudjelaba et al., 2004] K. Boudjelaba, D. Chikouche, R.E. Bekka, Application des Algorithmes Génétiques pour la Synthèse des Filtres RIF 1-D Passe-Bas, *Proceedings du 1er Congrès International sur le Génie Electrique CIGE'04*, Sétif, Algeria, 10-12 Octobre 2004, pp. 36–40
- [Boudjelaba et al., 2011] K. Boudjelaba, D. Chikouche, F. Ros, Evolutionary techniques for

- the synthesis of 2-D FIR filters, 2011 IEEE Workshop on Statistical Signal Processing (SSP'11), Nice, France, (28-30 June 2011) pp. 601–604
- [Boudjelaba et al., 2011] K. Boudjelaba, F. Ros, D. Chikouche, An advanced genetic algorithm for designing 2-D FIR filters, 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, (August 23-26, 2011) pp. 60–65
- [Cen, 2007] Cen L, A hybrid genetic algorithm for the design of FIR filters with SPoT coefficients. *Signal Processing* 87 (2007) pp. 528–540
- [Chan et al., 2012] Chung-Yong Chan, Paul M. Goggans, Using Bayesian inference for the design of FIR filters with signed power-of-two coefficients, *Signal Processing*, Vol. 92, N. 12 (2012) pp. 2866–2873
- [Charalambous, 1985] C. Charalambous, The performance of an algorithm of minimax design of two-dimensional linear phase FIR digital filters, *IEEE Trans. Circuits Syst.*, vol. 32 (1985) pp. 1016–1028
- [Clerc et al., 2003] Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essais particuliers (2003)
- [Coello et al., 2002] C.A. Coello, G.B. Lamont, D.A.V. Van Veldhuizen, Lamont GB, Evolutionary algorithms for solving multi-objective problems, Kluwer, New York, 2002
- [Dimitris Bertsimas et al., 1993] Dimitris Bertsimas, John Tsitsiklis, Simulated annealing, *Statistical Science*, vol. 8, No. 1 (1993) pp. 10–15
- [Dunn, 1998] S.A. Dunn, The Use Of Genetic Algorithms And Stochastic Hill-Climbing In Dynamic Finite Element Model Identification, Elsevier, *Computers & Structures* Vol. 66, No. 4 (1998) pp. 489–497
- [Eberhart et al., 2000] R. Eberhart, Y. Shi, Comparison between Genetic Algorithms and Particle Swarm Optimization, Proc. 7th Ann. Conf. on Evolutionary Computation, San Diego, 2000
- [Eberhart et al., 2001] Eberhart R. C., Shi Y., Particle swarm optimization: developments, applications and resources. Proceedings of IEEE Congress on Evolutionary Computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea., 2001a
- [Eberhart et al., 2001] Eberhart R. C., Shi Y., Tracking and optimizing dynamic systems with particle swarms. Proceedings of IEEE Congress on Evolutionary Computation 2001 pp. 94–97. IEEE service center, Piscataway, NJ., Seoul, Korea., 2001b
- [Erdozain et al., 2010] A. Erdozain, P. M. Crespo, A new stochastic algorithm inspired on genetic algorithms to estimate signals with finite rate of innovation from noisy samples, Elsevier, *Signal Processing* 90 (2010) pp. 134–144
- [Franzen et al., 1998] O. Franzen, H. Blume, H. Schroder, FIR-filter design with spatial and frequency design constraints using evolution strategies, *Signal Processing* 68 (1998) pp.

295-306

- [Glover, 1989] Glover F., Tabu search—part I, *ORSA Journal on Computing*, 1(3) (1989) pp. 190–206
- [Glover, 1990] Glover F., Tabu search—part II, *ORSA Journal on Computing*, 2(1) (1990) pp. 4–32
- [Goldberg, 1989] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989
- [Hime et al., 2009] Hime A, Oliveira Jr., Antonio Petraglia, Mariane R. Petraglia, Frequency Domain FIR Filter Design Using Fuzzy Adaptive Simulated Annealing, *Circuits Syst Signal Process* 28 (2009) pp. 899–911
- [Holland, 1973] Holland J. H., *Genetic algorithms and the optimal allocation of trials*, *SIAM Journal on Computing* 2, 1973
- [Holland, 1975] J.H. Holland (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Michigan; re-issued by MIT Press (1992)
- [Hsieh et al., 1997] C-H Hsieh, C-M Kuo, Y-D Jou, and Y-L Han, Design of Two-Dimensional FIR Digital Filters by a Two-Dimensional WLS Technique, *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, Vol. 44, No. 5, May 1997
- [Johnson et al., 2002] Alan W. Johnson, Sheldon H. Jacobson, A class of convergent generalized hill climbing algorithms, Elsevier, *Applied Mathematics and Computation* 125 (2002) pp. 359–373
- [Jun et al., 2009] Shu Jun, Li Jian, A Hybrid of Differential Evolution and Particle Swarm Optimization for Global Optimization, *Third International Symposium on Intelligent Information Technology Application* (2009) pp. 138–141
- [Kamp et al., 1975] Y. Kamp and J. P. Thiran, Chebyshev approximation for two-dimensional nonrecursive digital filters, *IEEE Trans. Circuits Syst.* 22 (1975) pp. 208–218
- [Kao et al., 2000] Yung-An Kao, Sau-Gee Chen, Design of finite-word-length FIR filters with least-squares error *Signal Processing* 80 (2000) pp. 1591–1596
- [Kar et al., 2011] Rajib Kar, Durbadal Mandal, Dibbendu Roy, Sakti Prasad Ghoshal, FIR Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach, *ACEEE Int. J. on Electrical and Power Engineering*, Vol. 02, No. 02 (2011) pp. 1–5
- [Kar et al., 2012] Rajib Kar, Durbadal Mandal, Sangeeta Mondal, Sakti Prasad Ghoshal, Crazyness based Particle Swarm Optimization algorithm for FIR band stop filter design, *Swarm and Evolutionary Computation* 7 (2012) pp. 58–64

- [Karaboğa et al., 2006] Karaboga, N., Cetinkaya, B.: Design of digital fir filters using differential evolution algorithm, *Circuits systems and signal processing* Vol. 25, No. 5 (2006) pp. 649–660
- [Karam et al., 1995] L.J. Karam, J.H. McClellan, Complex Chebyshev approximation for FIR filter design, *IEEE Trans. Circuits and Systems II* 42 (1995) pp. 207–216
- [Karam et al., 1999] Lina J. Karam, James H. McClellan, Chebyshev digital FIR filter design, *Signal Processing* 76 (1999) pp. 17–36
- [Kaur et al., 2012] Amanpreet Kaur, Ranjit Kaur, Design of FIR Filter Using Particle Swarm Optimization Algorithm for Audio Processing, *International Journal of Computer Science and Network (IJCSN)*, Vol. 1, No. 4 (2012) pp. 103–108
- [Kennedy et al., 1995] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in *Proc. IEEE int. Conf. On Neural Network* (1995)
- [Kirkpatrick et al., 1983] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, *Science* 220 (1983) pp. 671–680
- [Lee et al., 2000] Lee J.-H., Yang S.-J., Tang D.-C.: Minimax design of 2-D linear-phase FIR filters with continuous and powers-of-two coefficients, *Elsevier Signal Processing* 80 (2000) pp. 1435–1444
- [Lu et al., 1997] Lu H.-C., Yeh K.-H.: Optimal design of 2-D FIR digital filters by scaling-free McClellan transformation using least-squares estimation, *Elsevier Signal Processing* 58 (1997) pp. 303–308
- [Lu et al., 2000] Hung-Ching Lu, Shian-Tang Tzeng, Design of arbitrary FIR log filters by genetic algorithm approach, *Signal Processing* 80 (2000) pp. 497–505
- [Lu et al., 2000] H-C Lu, S-T Tzeng, Design of two-dimensional FIR digital filters for sampling structure conversion by genetic algorithm approach, *Signal Processing* 80 (2000) pp. 1445–1458
- [Lu et al., 2000] H. C. Lu and K. H. Yeh, 2-D FIR filters design using least square error with scaling-free McClellan transformation, *IEEE Trans. Circuits Syst., II*, vol. 47 (2000) pp. 1104–1107
- [Lu, 2002] W.-S. Lu, A unified approach for the design of 2-D digital filters via semidefinite programming, *IEEE Trans. Circuits Syst.*, vol. 49 (2002) pp. 814–826
- [Lu et al., 2005] W.-S. Lu, T. Hinamoto, A new minimax design for 2-D FIR filters with low group delay, *ISCAS'2005, Kobe, Japan, May 2005*, pp. 2028–2031
- [Lu et al., 2006] W.-S. Lu, T. Hinamoto, A Second-Order Cone Programming Approach for Minimax Design of 2-D FIR Filters with Low Group Delay, *ISCAS 2006, Island of Kos, Greece, May 2006*, pp. 2521–2524
- [Lu et al., 2007] W.-S. Lu, T. Hinamoto, Design of FIR filters with discrete coefficients via

- polynomial programming: Towards the global solution, ISCAS 2007, New Orleans, May 2007, pp. 2048–2051
- [Lu et al., 2009] W. S. Lu and T. Hinamoto, Optimal Design of Nonlinear-Phase FIR Filters With Prescribed Phase Error, *IEEE Trans. On Sig. Processing* Vol. 57, No. 9 (2009) pp. 3399–3410
- [Luitel et al., 2008] B. Luitel, G.K. Venayagamoorthy, Differential Evolution Particle Swarm Optimization for Digital Filter Design, *IEEE Congress on Evolutionary Computation* (2008) pp. 3954–3961
- [Mastorakis et al., 2003] Mastorakis N. E., Gonos I. F., Swamy M. N. S., Design of two-dimensional recursive filters using genetic algorithms. *IEEE Transactions On Circuits and Systems I: Fundamental Theory And Applications* Vol. 50, No. 5 (2003) pp. 634–639
- [McClellan et al., 1973] J.H. McClellan, T.W. Parks, L.R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust.*, 21 (1973) pp. 506–526
- [Mondal et al., 2011] Sangeeta Mondal, Vasundhara, Rajib Kar, Durbadal Mandal, S. P. Ghoshal, Linear Phase High Pass FIR Filter Design using Improved Particle Swarm Optimization, *World Academy of Science, Engineering and Technology* 60 (2011) pp. 1620–1627
- [Mondal et al., 2012] Sangeeta Mondal, S. P. Ghoshal, Rajib Kar, Durbadal Mandal, Novel Particle Swarm Optimization for Low Pass FIR Filter Design, *Wseas Transactions On Signal Processing*, Issue 3, Volume 8 (2012) pp. 111–120
- [Nguyen et al., 1986] D. T. Nguyen, M. N. S. Swamy, Approximation design of 2-D digital filters with elliptical magnitude response of arbitrary orientation, *IEEE Trans. Circuits Syst.*, vol. 33 (1986) pp. 597–603
- [Nordebo et al., 1997] S. Nordebo, I. Claesson, Minimum Norm Design of Two-Dimensional Weighted Chebyshev FIR Filters, *IEEE Trans. Circuits Syst.-II*, vol. 44, No. 3 (1997) pp. 251–253
- [Omran et al., 2007] Mahamed G.H. Omran, Andries P. Engelbrecht, Ayed Salman, Differential Evolution Based Particle Swarm Optimization, *IEEE Swarm Intelligence Symposium, SIS 2007* (2007) pp. 112–119
- [Parcks et McClellan, 1972] T.W. Parks, J.H. McClellan, Chebyshev approximation for non-recursive digital filters with linear phase, *IEEE Trans. Circuits Theory* CT-19 (1972) pp. 189–194
- [Pei et al., 1995] S-C Pei, J-J Shyu, Design of two-dimensional FIR digital filters by McClellan transformation and least-squares contour mapping, *Signal Processing* 44 (1995) pp. 19-26

- [Pei et al., 1995] S-C Pei, J-J Shyub, Symmetric properties of 2-D sequences and their applications for designing linear-phase 2-D FIR digital filters, *Signal Processing* 42 (1995) pp. 261-271
- [Pei et al., 1996] Pei S.-C., Shyu J.J.: General form for designing two-dimensional quadrantally symmetric linear-phase FIR digital filters by analytical least-squares method, *Elsevier Signal Process.* 48 (1996) pp. 165–174
- [Pei et al., 1996] Soo-Chang Pei, Jong-Jy Shyu, Design of two-dimensional FIR digital filters with specified magnitude and group delay responses by analytical least-squares method, *Signal Processing* 54 (1996) pp. 273–283
- [Pei et al., 1999] Pei S.-C., Shyu J.-J.: Design of two-dimensional FIR digital filters by McClellan transformation and least-squares contour mapping, *Elsevier Signal Process.* 44 (1999) pp. 19–26
- [Pérez et al., 2012] E. Pérez, M. Posada, F. Herrera, Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling, *J Intell Manuf* 23 (2012) pp. 341–356
- [Pham et al., 2010] D. T. Pham, E Koç, Design of a Two-dimensional Recursive Filter Using the Bees Algorithm, *International Journal of Automation and Computing*, Vol. 7 No. 3 (2010) pp. 399–402, DOI: 10.1007/s11633-010-0520-x
- [Potchinkov, 1997] Alexander W. Potchinkov, Design of optimal linear FIR filters by a semi-infinite programming technique, *Signal Processing* 58 (1997) pp. 165–180
- [Pei et al., 2000] S. C. Pei, P. H. Wang, Design of Arbitrary Cutoff 2-D Diamond-Shaped FIR Filters Using the Bernstein Polynomial, *IEEE Sig. Processing Letters*, vol. 7, No. 11 (2000) pp. 310–313
- [Quinquis, 2000] A. Quinquis, "Le Traitement du Signal sous Matlab, pratique et applications", Hermès, Paris, 2000
- [Rajan et al., 1983] P. K. Rajan, and M. N. S. Swamy, Design of circularly symmetric two-dimensional FIR digital filters employing transformations with variable parameters, *IEEE Trans. On Acoust., Speech, Signal Processing*, Vol. 31, No. 3 (1983) pp. 637–642
- [Reeves, 1995] C. R. Reeves, A Genetic Algorithm For Flowshop Sequencing, *Computers and Operations Research*, Vol. 22, No. 1 (1995) pp. 5–13
- [Ros et al., 2007] F. Ros, S. Guillaume, An efficient nearest classifier, Book Chapter of *Hybrid Evolutionary Systems, Studies in Computational Intelligence*, Vol. 75, Springer Verlag (2007) pp. 131–147
- [Rusu et al., 2012] C Rusu, B Dumitrescu, Iterative reweighted l1 design of sparse FIR filters, *Signal Processing* 92 (2012) pp. 905–911
- [Salman et al., 1982] W. P. Salman et M. S. Solotareff, *Le filtrage numérique*, Eyrolles 1982

- [Sheeba et al., 2007] V.S. Sheeba, E. Elizabeth, Two-dimensional FIR signal adapted filter banks: Optimality and design, Elsevier, Signal Processing 87 (2007) pp. 2381–2391
- [Shyu et al., 2008] J. J. Shyu, S-C. Pei, A generalized approach to the design of variable fractional-delay FIR digital filters, Elsevier, Signal Processing 88 (2008) pp. 1428–1435
- [Speake et al., 1979] T. C. Speake, R. M. Mersereau, A comparison of different window formulations for twodimensional FIR filter design, IEEE Acoust., Speech, Signal Processing Conf. Rec., Apr. 1979
- [Speake et al., 1981] T. C. Speake, R. M. Mersereau, A Note on the use of windows for two-dimensional FIR filter design, IEEE Trans. On Acoust., Speech, Signal Processing, vol. 29, No 1 (1981) pp. 125–127
- [Sriranganathan et al., 1995] S. Sriranganathan, D.R. Bull, D.W. Redmill, Design of 2-D multiplierless FIR filters using genetic algorithms, First Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications GALEZIA 1995, Sheffield, UK, Sep. 1995, pp. 282–286
- [Suckley, 1991] Suckley D., Genetic algorithm in the design of FIR filters. IEE Proceedings G, Circuits, Devices and Systems, Vol. 138, No. 2 (1991) pp 234–238
- [Tzeng et al., 2000] Tzeng S-T, Lu H-C, Complex genetic algorithm approach for designing equiripple complex FIR digital filters with weighting function. Signal Processing 80 (2000) pp. 197–204
- [Tzeng, 2004] S-T Tzeng, Genetic algorithm approach for designing 2-D FIR digital filters with 2-D symmetric properties, Signal Processing 84 (2004) pp. 1883–1893
- [Tzeng, 2007] S-T. Tzeng, Design of 2-D FIR digital filters with specified magnitude and group delay responses by GA approach, Signal Processing 87 (2007) pp. 2036–2044
- [van der Merwe et al., 2003] van der Merwe D.W., Engelbrecht A.P., Data clustering using particle swarm optimization, The 2003 Congress on Evolutionary Computation, CEC'03, Vol. 1 (2003) pp. 215–220
- [Venkataraman et al., 1994] S. Venkataraman, B. C. Levy, State Space Representations of 2-D FIR Lossless Transfer Matrices, IEEE Transactions On Circuits And Systems-11: Analog And Digital Signal Processing, Vol. 41, No. 2 (1994)
- [Wang et al., 2007] Wang X-H, He Y-G, Liu M-R, Four optimal design approaches of high-order finite-impulse response filters based on neural network. Journal of Central South University of Technology Vol. 14, No. 1 (2007) pp. 94–99
- [Wang et al., 2008] Wang X.-H., He Y.-G.: A neural network approach to FIR filter design using frequency-response masking technique, Elsevier Signal Process. 88 (2008) pp. 2917–2926
- [Williams et al., 2001] T. Williams, M. Ahmadi, R. Hashemian, W. C. Miller, Design of High Throughput 2-D FIR Filters Using Singular Value Decomposition (SVD) and

- Genetic Algorithms, Proc. of the IEEE Pacific RIM Conf. on Signals, Communications and Computers, Victoria, BC, August 2001, pp. 571–574
- [Wu et al., 2009] Tai-Hsi Wu, Jinn-Yi Yeh, Chin-Chih Chang, A hybrid Tabu Search Algorithm to Cell Formation Problem and its Variants, World Academy of Science, Engineering and Technology 53 (2009) pp. 1090–1094
- [Yeh et al., 1996] Yeh K.-H., Lu H.-C.: Shape parameter in the two-dimensional low-pass FIR digital filters design by transformation, Elsevier Signal Process. 53 (1996) pp. 65–74
- [Zeng et al., 2006] Zeng Zhe-zhao ; Chen Ye ; Zhu Wei ; Wang Yao-nan, Optimal Design on FIR Digital Filters Using the Parallel Algorithm of Neural Networks, Proceedings of International Conference on Communications Circuits and Systems, Vol. 1 (2006) pp. 191–195
- [Zhao et al., 1997] H. Zhao, J. Yu, A novel neural network-based approach for designing 2-D FIR filters, IEEE Transactions on Circuits and Systems Society, Vol. 44, Issue: 11 (1997) pp. 1095 - 1099
- [Zhao et al., 2007] Lingling Zhao, Zhejiang Zhou, Wanping Huang, Satisfactory Optimization Design Of FIR Digital Filter Based On Adaptive Particle Swarm Optimization, IEEE International Conference on Control and Automation (2007) pp. 1662–1666
- [Zhu et al., 1997] W.-P. Zhu, M. O. Ahmad, and M. N. S. Swamy, A closed-form solution to the least-square design problem of 2-D linear-phase FIR filters, IEEE Trans. Circuits Syst., II, vol. 44 (1997) pp. 1032–1039
- [Zhu et al., 1999] W.-P. Zhu, M. O. Ahmad, and M. N. S. Swamy, Realization of 2-D Linear-Phase FIR Filters by Using the Singular-Value Decomposition, IEEE Trans. On Sig. Processing, vol. 47, No 5 (1999) pp. 1349–1358

# Bibliographie de l'auteur

## *Articles publiés dans des journaux avec impact factor*

1. K. Boudjelaba, F. Ros, D. Chikouche, Adaptive genetic algorithm-based approach to improve the synthesis of two-dimensional finite impulse response filters, *IET Signal Processing*, DOI: 10.1049/iet-spr.2013.0005
2. K. Boudjelaba, F. Ros, D. Chikouche, An efficient hybrid genetic algorithm to design finite impulse response filters, *Elsevier – Expert Systems with Applications*, Vol. 41, Issue 13 (2014) pp. 5917–5937, DOI: 10.1016/j.eswa.2014.03.034
3. K. Boudjelaba, F. Ros, D. Chikouche, Potential of Particle Swarm Optimization and Genetic Algorithms for FIR Filter Design, *Circuits, Systems, and Signal Processing – Springer*, DOI: 10.1007/s00034-014-9800-y

## *Articles de journaux avec impact factor en cours d'évaluation*

1. K. Boudjelaba, F. Ros, D. Chikouche, An efficient hybrid genetic algorithm to design FIR filters, *Pattern Analysis and Applications – Springer* (soumis)

## *Communications internationales*

1. K. Boudjelaba, D. Chikouche, F. Ros, Evolutionary techniques for the synthesis of 2-D FIR filters, 2011 IEEE Workshop on Statistical Signal Processing (SSP'11), Nice, France, (28-30 June 2011) pp. 601–604
2. K. Boudjelaba, F. Ros, D. Chikouche, An advanced genetic algorithm for designing 2-D FIR filters, 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, (August 23-26, 2011) pp. 60–65

## *Séminaires*

1. Boudjelaba k, Ros F, Chikouche D., Contribution to the design of two-dimensional non-recursive filters using the techniques of artificial intelligence: application to image processing, Journées des Jeunes chercheurs du Laboratoire Prisme, 2 Juillet 2012, France
2. Boudjelaba k, Ros F, Chikouche D., Contribution à la conception des filtres bidimensionnels non récursifs en utilisant les techniques de l'intelligence artificielle : application au traitement d'images, Journées des doctorants de l'université de Sétif, 2011, Algérie

## ملخص

تصميم المرشحات ذات استجابة محدودة الاندفاع (FIR) يمكن أن تصاغ كمشكلة التحسين غير الخطية سمعته ليكون من الصعب على الأساليب التقليدية تصميمه. القيود مرتفعة و عدد كبير من المعاملات يجب أن تقدر، وخاصة عند التعامل مع مرشحات ثنائي البعد (D2). من أجل تحسين أداء الأساليب التقليدية، و نقوم بإستكشاف العديد من المنهجيات العشوائية قادرة على التعامل مع المساحات الكبيرة.

نقترح الخوارزمية الجينية الجديدة التي يتم تقديمها بعض المفاهيم المبتكرة لتحسين التقارب و جعل استخدامها أسهل للممارسين. النقطة الرئيسية لنهجنا ينبع من قدرة الخوارزمية الجينية (GA) على التكيف مع مشغلي الجينية خلال الحياة الجينية في حين تبقى بسيطة و سهلة التنفيذ. يتم تغيير معظم المعاملات والمشغليين من قبل GA نفسها. ثم، يقترح تحسين سرب الجسيمات (PSO) لتصميم المرشحات. وتؤكد الدراسة على أهمية إدخال المخابرات في metaheuristics لجعلها أكثر كفاءة عن طريق إدراج استراتيجيات ضبط النفس. أخيراً، يقترح الخوارزمية الجينية الهجينة (HGA) لتصميم المرشحات الرقمية. وتتألف الخوارزمية لعملية الجينية النقية و نهج المحلية المخصص. ويخصص عملية الجينية النقية إلى الجزء استكشافية. GA النهج المحلية تتعلق تقارب الخوارزمية و الأمتل للغاية وذلك ليكون لين العريكة.

علينا أن نبرهن على حالات مختلفة من تصميم المرشحات تحسن كبير في الأداء. مساهمتنا تتعلق أساساً قابليتها للاستعمال GA لمنطقة محددة من تصميم المرشحات منطقة معلومات الاستكشاف. انها تسعى الى مواجهة التحدي الحالي لدمقرطة استخدام GA لحل مشاكل العالم الحقيقي. التجارب أجريت مع أنواع مختلفة من المرشحات لتسليط الضوء على مساهمة التهجين في تحسين الأداء. التجارب تكشف أيضاً عن مزايا اقتراحنا بالمقارنة مع نهج التصميم التقليدية و بعض GA المرجعية في هذا المجال من التطبيق.

**المفاتيح :** المرشحات FIR أحادي (D1) و ثنائي (D2) البعد، الخوارزميات الجينية (Genetic Algorithm)، التحسين بسرب الجسيمات (Particles Swarm Optimization)، التهجين (Hybridization)، Tabu Search، Simulated Annealing، Hill Climbing، معالجة الصور.

## Abstract

The design of finite impulse response (FIR) filters can be formulated as a non-linear optimization problem reputed to be difficult for conventional approaches. The constraints are high and a large number of parameters have to be estimated, especially when dealing with 2-D FIR filters. In order to improve the performance of conventional approaches, we explore several stochastic methodologies capable of handling large spaces.

We propose a new genetic algorithm in which some innovative concepts are introduced to improve the convergence and make its use easier for practitioners. The key point of our approach stems from the capacity of the genetic algorithm (GA) to adapt the genetic operators during the genetic life while remaining simple and easy to implement. Most of parameters and operators are changed by the GA itself. Then, the Particle Swarm Optimization (PSO) is proposed for FIR filter design. The study underlines the importance of introducing intelligence in metaheuristics to make them more efficient by inserting self-tuning strategies. Finally, a hybrid genetic algorithm (HGA) is proposed for the design of digital filters. The algorithm is composed of a pure genetic process and a dedicated local approach. The pure genetic process is devoted to the exploratory part of the GA. The local approach concerns convergence of the algorithm and is highly optimized so as to be tractable.

We demonstrate on various cases of filter design a significant improvement in performance. Our contribution mainly concerns the usability of GAs for the specific area of FIR filter design. It seeks to address the current challenge of democratizing the use of GAs for real world problems. Experiments performed with various types of filters highlight the recurrent contribution of hybridization in improving performance. The experiments also reveal the advantages of our proposal compared to more conventional filter design approaches and some reference GAs in this field of application.

**Keywords:** 1D and 2D FIR filters, Genetic Algorithm, Particles Swarm Optimization, Hybridization, Tabu Search, Simulated Annealing, Hill Climbing, Image Processing.