

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية



Ministère de l'Enseignement Supérieur et de la Recherche

Scientifique

Université Kasdi Merbah d'Ouargla



Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Electronique et des Télécommunications

Mémoire Pour l'obtention du diplôme de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Présenté par :

BERRA Lokmane

GASMI Abdeldjalil

Thème

Conception d'un filtre numérique utilisant les méthodes

Méta-heuristiques

Soutenue publiquement

Le : 24/06/2024

Devant le jury :

Mr. MEHAOUCHI Azeddine

MAA.

Président

UKM Ouargla

Mr. KADRI Salim

MAA.

Examineur

UKM Ouargla

Mr. BENCHABANE Abderrazak

MCA.

Encadreur

UKM Ouargla

Année Universitaire : 2023/2024



DÉDICACES

*J'ai le grand plaisir de dédier ce modeste
travail:*

A ma très chère mère

A mon très cher père

A ma très chère Grande mère

A mon Frère Salah



BERRA LOKMANE

DÉDICACES

*J'ai le grand plaisir de dédier ce modeste
travail:*

A ma très chère mère

A mon très cher père

ainsi qu'à mes frères

à tous ceux qui m'ont aimé

et à mon cher ami BERRA Lokmane.

GASMI ABDELJALIL

Remerciement

*En premier lieu, nous remercions ALLAH, le tout-Puissant pour ses faveurs et ses grâces, de nous avoir donné le courage et la patience durant ce travail. Nos remerciements et nos profondes gratitudes vont à notre encadreur **Benchabane Abdelrazzak** son soutien scientifique et psychique était de valeur et essentiel tout au long de ce mémoire. Nous remercions les honorables membres du jury pour leur précieux temps accordé à l'étude de notre mémoire. Adressons également nos remerciements, à tous nos enseignants, qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.*

Résumé

Les filtres numériques sont essentiels dans diverses applications, y compris le traitement du signal, les communications et les systèmes de contrôle. Les méthodes traditionnelles de conception de filtres rencontrent souvent des défis tels que l'enfermement dans des minima locaux et l'inefficacité computationnelle. Les algorithmes métaheuristiques offrent une alternative robuste en fournissant des capacités de recherche globale et une adaptabilité à des problèmes d'optimisation complexes. Dans ce mémoire, nous présentons la conception et la mise en œuvre de filtres numériques en utilisant des algorithmes métaheuristiques, en particulier l'optimisation par essaims particulaires (PSO), l'optimisation par loup gris (GWO) et l'algorithme de recherche de coucou amélioré (ICSA). L'objectif principal est d'explorer l'efficacité et l'efficacité de ces algorithmes pour optimiser les paramètres des filtres numériques afin d'atteindre des performances de traitement du signal souhaitées.

Mots clés : Optimisation par essaims particulaires (PSO), Optimisation par loup gris (GWO), Algorithme de recherche de coucou amélioré (ICSA), filtres numériques, optimisation complexes, conception, algorithmes métaheuristiques, traitement du signal, systèmes de contrôle

Abstract

Digital filters are essential in various applications, including signal processing, communications, and control systems. Traditional filter design methods often face challenges such as getting trapped in local minima and computational inefficiency. Metaheuristic algorithms offer a robust alternative by providing global search capabilities and adaptability to complex optimization problems. In this thesis, we present the design and implementation of digital filters using metaheuristic algorithms, particularly Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), and the Improved Cuckoo Search Algorithm (ICSA). The main objective is to explore the effectiveness and efficiency of these algorithms in optimizing the parameters of digital filters to achieve desired signal processing performance.

Keywords: Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), Improved Cuckoo Search Algorithm (ICSA), digital filters, complex optimization, design, metaheuristic algorithms, signal processing, control systems

ملخص

تعد المرشحات الرقمية أساسية في تطبيقات متنوعة، بما في ذلك معالجة الإشارات والاتصالات وأنظمة التحكم. غالبًا ما تواجه الطرق التقليدية لتصميم المرشحات تحديات مثل الوقوع في الحد الأدنى المحلي وعدم الكفاءة الحسابية. توفر الخوارزميات الميتاهيرستيكيكس بديلاً قوياً من خلال توفير قدرات البحث العالمية والقدرة على التكيف مع مشاكل التحسين المعقدة. في هذه المذكرة، نقدم تصميم وتنفيذ المرشحات الرقمية باستخدام الخوارزميات الميتاهيرستيكيكس، لا سيما تحسين السرب الجزيئي (PSO)، وتحسين الذئب الرمادي (GWO)، وخوارزمية بحث الوقواق المحسنة (ICSA). الهدف الرئيسي هو استكشاف فعالية وكفاءة هذه الخوارزميات في تحسين معلمات المرشحات الرقمية لتحقيق أداء معالجة الإشارات المرغوب.

كلمات مفتاحية: تحسين السرب الجزيئي (PSO)، تحسين الذئب الرمادي (GWO)، خوارزمية بحث الوقواق المحسنة (ICSA)، المرشحات الرقمية، التحسين المعقد، التصميم، الخوارزميات الميتاهيرستيكيكس، معالجة الإشارات، أنظمة التحكم.

Sommaire

Liste des figures

Liste des Tableaux

LISTE DES SYMBOLES

LISTE D'ABRÉVIATIONS

Introduction générale 1

Chapitre 1 : Généralités sur les filtres à réponse impulsionnelle finie

1.1 Introduction 3

1.2 Filtres à réponse impulsionnelle finie..... 3

1.3 Conception des filtres FIR par la technique des fenêtres 4

1.3.1 Méthodologie..... 4

1.3.2 Limites de la fenêtre rectangulaire 5

1.3.3 Fenêtres usuelles..... 6

1.4 Conception d'un filtre FIR par échantillonnage fréquentiel..... 8

1.5 Conception de filtres numériques FIR avec moindres carrés 9

1.6 Conception de filtres numériques FIR avec l'algorithme de Parks-McClellan 11

1.7 Conclusion..... 14

Chapitre 2 : Les Algorithmes méta-heuristique

2.1 Introduction 15

2.2 Optimisation des essaims de particules (PSO) 16

2.2.1 Principe l'algorithme PSO..... 16

2.2.2 Structure algorithmique PSO standard 16

2.2.3 Organigramme de l'algorithme PSO..... 19

2.2.4 Procédure l'algorithme PSO..... 20

2.3 Algorithme loups gris (GWO)..... 21

2.3.1 Structure algorithmique GWO standard 22

2.3.2 Organigramme de l'algorithme GWO : 26

2.4 Algorithme de recherche de coucou amélioré (ICSA) 26

2.4.1 Algorithme CSA..... 26

2.4.2 Algorithme ICSA..... 28

2.4.3 Organigramme de l'algorithme ICSA 30

2.5 Conclusion..... 30

Chapitre 3 : Les résultats de simulation

3.1	Introduction	31
3.2	Fonction objective	31
3.3	Erreur quadratique moyenne (MSE) :	33
3.3.1	Simulation par MSE1	34
3.3.2	Simulation par MSE2	36
3.4	Multi-Objectif (MO) :.....	39
3.4.1	Simulation par MO	39
3.5	Comparaison avec filtre Parks–McClellan	43
3.5.1	PSO par MSE	43
3.5.2	GWO par MSE	44
3.5.3	ICSA par MSE.....	46
3.5.4	PSO par MO	47
3.5.5	GWO par MO	48
3.5.6	ICSA par MO	48
3.6	Analyse Comparative	49
3.7	Conclusion.....	49
	Conclusion Générale.....	50
	Bibliographie	52

Liste des figures

Figure (1.1): Gabarit d'un filtre passe Bas	4
Figure (1.2): Convolution de $H_d(w)$ avec $W(w)$ réponse fréquence (rectangulaire)	5
Figure (1.3): Réponse en fréquence pour la fenêtre rectangulaire de longueur (a) $M=31$ (b) $M=61$. Effet de la fenêtre sur la bande de transition.....	6
Figure (1.4) : Réponse fréquentielle des fenêtres usuelles.	7
Figure (1.5): Les fenêtres usuelles.....	7
Figure (1.6): Réponse fréquentiel avec les échantillonnages fréquences.	8
Figure (1.7): Exemple d'illustration.	9
Figure (1.8): Exemple de synthèse d'un filtre RIF.....	9
Figure (1.9): Exemple de synthèse d'un filtre RIF par les moindres carrés.....	11
Figure (2.1) : Mouvement des particules.....	11
Figure (2.2) : Organigramme PSO.	19
Figure (2.3) : Hiérarchie sociale Loups gris..	21
Figure (2.4) : Comportement de chasse des loups gris : (A) chasser, approcher et suivre d'proies (B-D) poursuivre, harceler et encercler (E) situation stationnaire et attaque [1].....	22
Figure (2.5) : Attacking toward prey versus searching for prey.....	23
Figure (2.6) : Mise à jour de position dans l'algorithme GWO [2] [3].	25
Figure (2.7) : Organigramme de l'algorithme GWO.	26
Figure (2.8) : Organigramme de l'algorithme ICOSA.....	30
Figure (3.1) : Méthode de L'erreur quadratique 1.....	33
Figure (3.2) : Méthode de L'erreur quadratique 2.....	33
Figure (3.3) : La courbe de convergence MSE1-PSO.....	34
Figure (3.4): Réponses impulsionnelle et fréquentielle pour MSE1-PSO	34
Figure (3.5): La courbe de convergence MSE1-GWO.....	35
Figure (3.6): Réponses impulsionnelle et fréquentielle pour MSE1-GWO	35
Figure (3.7): Réponses impulsionnelle et fréquentielle pour MSE1-ICSA.....	36
Figure (3.8): La courbe de convergence MSE2-PSO	37
Figure (3.9): Réponses impulsionnelle et fréquentielle pour MSE2-PSO	37
Figure (3.10): La courbe de convergence MSE2-GWO.....	38
Figure (3.11): Réponses impulsionnelle et fréquentielle pour MSE2-GWO	38
Figure (3.12): Réponses impulsionnelle et fréquentielle pour MSE2-ICSA.....	39
Figure (3.13): Le principe de Fonction multi-objective	40

Figure (3.14): La courbe de convergence MO PSO	40
Figure (3.15): Réponses impulsionnelle et fréquentielle pour MO-PSO	41
Figure (3.16): Réponses impulsionnelle et fréquentielle pour MO-GWO	41
Figure (3.17): La courbe de convergence MO GWO.....	42
Figure (3.18): Réponses impulsionnelle et fréquentielle pour MO-ICSA	42
Figure (3.19): PSOMSE1.....	43
Figure (3.20): PSOMSE2	44
Figure (3.21): GWOMSE1	45
Figure (3.22): GWOMSE2	45
Figure (3.23): ICSAMSE1.....	46
Figure (3.24): ICSAMSE2.....	47
Figure (3.25): MO-PSO.....	47
Figure (3.26): MO-GWO.....	48
Figure (3.27): MO-ICSA.....	48

Liste des Tableaux

Tableau (1.1): Fonctions de fenêtre pour la conception de filtre de RIF :	6
Tableau (3.1): Nouvelles caractéristiques de filtre passe-bas pour MSE1-PSO :	34
Tableau (3.2): Nouvelles caractéristiques de filtre passe-bas pour MSE1-GWO:	35
Tableau (3.3): Nouvelles caractéristiques de filtre passe-bas pour MSE1-ICSA:	36
Tableau (3.4): Nouvelles caractéristiques de filtre passe-bas pour MSE2-PSO	37
Tableau (3.5): Nouvelles caractéristiques de filtre passe-bas pour MSE2-GWO :	38
Tableau (3.6): Nouvelles caractéristiques de filtre passe-bas pour MSE2-ICSA:	39
Tableau (3.7): Nouvelles caractéristiques de filtre passe-bas pour MO-PSO:	41
Tableau (3.8): Nouvelles caractéristiques de filtre passe-bas pour MO-GWO:	42
Tableau (3.9) :Nouvelles caractéristiques de filtre passe-bas pour MO-ICSA:	43

LISTE DES SYMBOLES

W_p : Bord de bande passante

W_s : Bord de bande d'arrêt

δ_1 : Ondulation de la bande passante

δ_2 : Atténuation de la bande d'arrêt

A_s : Atténuation en bande d'arrêt

Δw : Largeur de transition

W_c : Fréquence de coupure

p_b : Meilleure position personnelle

g_b : Meilleure position globale

X_i : Position de chaque particule

V_i : Vitesse de chaque particule

w : Poids d'inertie

c_1 et c_2 : Constantes d'accélération

t : Numéro d'itération.

α : Chefs de groupe

D : Vecteur de nouvelle position

AP : Probabilité de conscience

DAP : Probabilité de conscience dynamique

Z_i : Taille du pas

b : Vecteurs n-dimensionnels

X_b : Meilleure solution

$\Gamma(\cdot)$: Distribution gamma

wV : Plus faible valeur de fitness

$L(\mathbf{s}, \lambda)$: Distribution de Levy Flight

LISTE D'ABRÉVIATIONS

FIR : Finite Impulse Response

PSO : Particle Swarm Optimization

GWO : Grey Wolf Optimizer

CSA : Cuckoo Search Algorithm

ICSA : Improved Cuckoo Search Algorithm

MSE : Mean Squared Error

MO : Multi Objectif

Introduction générale

Introduction générale

Les filtres numériques à réponse impulsionnelle finie (FIR) sont essentiels dans de nombreuses applications modernes de traitement du signal, telles que les télécommunications, l'audio, le contrôle automatique et l'électronique. Ils permettent de manipuler les signaux numériques afin de les rendre plus adaptés à des fins spécifiques, comme la suppression du bruit, l'amélioration de la qualité du signal ou l'extraction d'informations pertinentes. La conception de ces filtres exige une précision et une optimisation poussées pour garantir des performances adéquates tout en respectant des contraintes spécifiques, comme la minimisation des ondulations dans la bande passante ou la maximisation de l'atténuation dans la bande d'arrêt.

Traditionnellement, la conception de filtres FIR repose sur des méthodes telles que les techniques de fenêtre, la méthode de Remez, et la conception par échantillonnage fréquentiel. Bien que ces méthodes soient efficaces, elles présentent des limitations, notamment en termes de complexité computationnelle et de difficulté à échapper aux minima locaux. Ces limitations peuvent entraîner des performances sous-optimales, en particulier pour des filtres avec des spécifications rigoureuses. Dans ce contexte, les algorithmes métaheuristiques offrent une alternative prometteuse pour surmonter ces défis.

Les algorithmes métaheuristiques sont des techniques d'optimisation inspirées de processus naturels ou de comportements sociaux. Parmi ces algorithmes, l'Optimisation par Essaims Particulaires (PSO), l'Optimisation par Loups Gris (GWO) et l'Algorithme de Recherche de Coucou Amélioré (ICSA) se distinguent par leur capacité à explorer efficacement des espaces de solution vastes et complexes, tout en évitant les pièges des minima locaux.

Notre travail consiste à faire une étude descriptive sur ce domaine important du traitement de signal (filtrage numérique). Pour ce faire, nous avons opté pour le plan qui suit :

- Le premier chapitre représente une conception de filtre FIR Par technique Fenêtres, la méthode d'échantillonnage de fréquences, la méthode de moindre carrée et l'algorithme de Parks–McClellan et simulation de ses filtres sous Matlab.
- Le deuxième chapitre une définition sur les algorithmes des métaheuristiques : PSO (L'optimisation par essaims particuliers), ICSA (algorithme de recherche de coucou amélioré) et GWO (Algorithme d'Optimisation du Loup Gris).

- Dans le troisième chapitre, nous présenterons les filtres pour métaheuristiques de trois manières différentes, et nous les comparerons avec une méthode classique efficace de Parks–McClellan et les résultats obtenus.

Chapitre 1

Chapitre 1

Généralités sur les filtres à réponse impulsionnelle finie

1.1. Introduction

Les filtres à réponse impulsionnelle finie FIR sont largement utilisés dans le traitement du signal en raison de leur stabilité inhérente et de leur capacité à avoir une réponse en phase linéaire. Ces filtres sont conçus en déterminant une séquence finie de coefficients, qui définissent leur réponse impulsionnelle. Leur conception peut être abordée par diverses méthodes, chacune ayant ses avantages spécifiques, en fonction des exigences de performance et de complexité. Les filtres FIR peuvent être utilisés dans diverses applications, notamment les filtres passe-bas, passe-haut, passe-bande et coupe-bande, en raison de leur flexibilité et de leur robustesse.

1.2. Filtres à réponse impulsionnelle finie

Les filtres à réponse impulsionnelle finie sont des systèmes linéaires discrets invariants dans le temps définis par une équation aux différences selon laquelle un nombre en sortie $y(n)$, représentant un échantillon du signal filtré, est obtenu par sommation pondérée d'un ensemble fini de nombres d'entrée, représentant les échantillons du signal à filtrer $x(n)$. Les coefficients h de la sommation pondérée constituent la réponse impulsionnelle du filtre et un ensemble fini d'entre eux seulement prennent des valeurs non nulles.

La fonction de transfert idéale d'un filtre FIR est irréalisable ; la transmittance en bande passante peut avoir des "oscillations" limitées à $\pm \delta_1$, cependant la transmittance en bande coupée peut aussi avoir des oscillations limitées à δ_2 . Entre la bande passante et la bande coupée il y a une bande de transition de largeur non Nulle δ_ω . La figure 1.1 montre la réponse fréquentielle idéale et réelle d'un filtre passe bas.

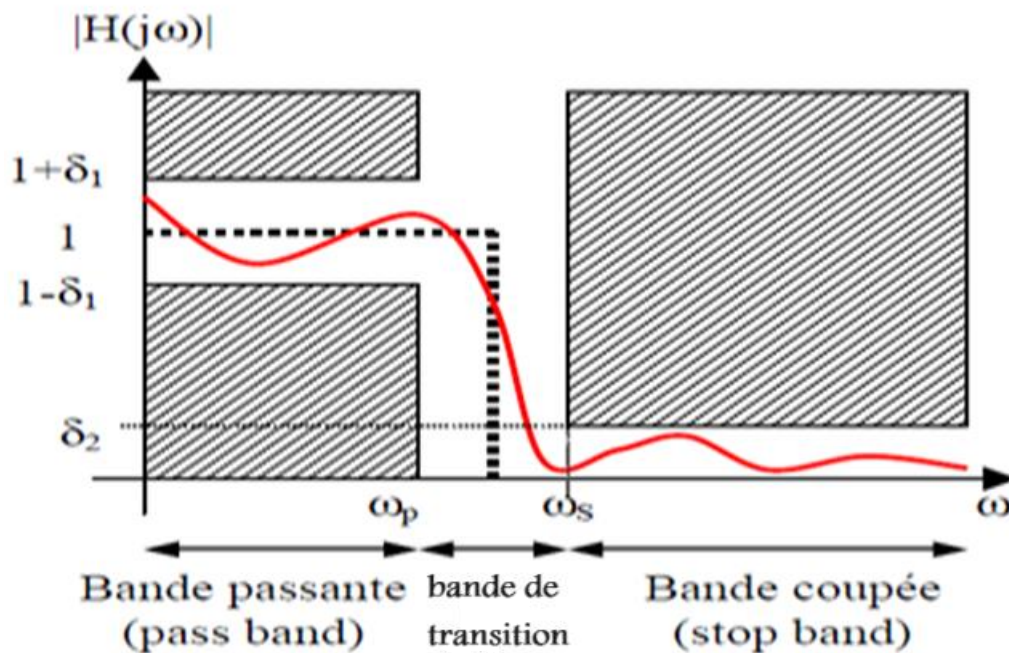


Figure (1.1) : Gabarit d'un filtre passe Bas

1.3. Conception des filtres FIR par la technique des fenêtres

1.3.1. Méthodologie

A partir des spécifications de la réponse fréquence désirée $H_d(\omega)$ on détermine la réponse impulsionnelle correspondante $h_d(n)$ par le passage du domaine fréquentiel au domaine temps discret. En effet, $h_d(n)$ n'est autre que la transformée de Fourier inverse de $H_d(\omega)$.

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega \quad (1.1)$$

En général, dans la réponse impulsionnelle $h_d(n)$ obtenu de (1.1) est infini dans le temps et doit être tronqué à un nombre fini de points. Pour un filtre d'ordre M , ce qui est équivalent à $M+1$ coefficients, on doit multiplier $h_d(n)$ par une fenêtre rectangulaire défini comme:

$$w(n) = \begin{cases} 1, & n = 0, 1, \dots, M \\ 0 & \text{Sinon} \end{cases} \quad (1.2)$$

La réponse impulsionnelle résultante devient

$$h(n) = h_d(n) \cdot w(n) = \begin{cases} h_d(n), & n = 0, 1, \dots, M \\ 0 & \text{Sinon} \end{cases} \quad (1.3)$$

On rappelle que la multiplication de $h_d(n)$ avec la fonction de fenêtre est équivalente à une

convolution de $H_d(\omega)$ avec $W(\omega)$, où $W(\omega)$ est la réponse fréquentielle de la fenêtre $w(n)$. Par conséquent la réponse fréquentielle du filtre obtenu a pour expression :

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\theta)W(\omega - \theta) d\theta \tag{1.4}$$

Cette opération de convolution entraîne l'apparition des ondulations connus sous le phénomène de GIBBS

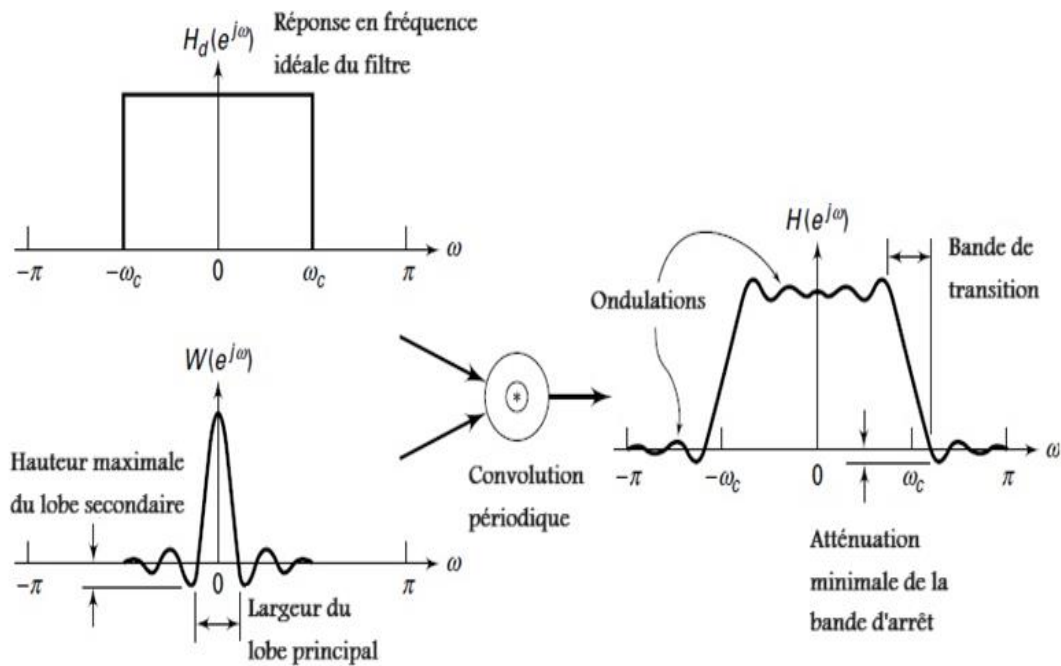


Figure (1.2) : Convolution de $H_d(\omega)$ avec $W(\omega)$ réponse fréquence (rectangulaire)

1.3.2. Limites de la fenêtre rectangulaire

La réponse fréquentielle de la fonction de fenêtre est illustrée par la fig1.3 pour une longueur de la réponse impulsionnelle $M = 31$ et 61 . La largeur du lobe principal est $4\pi/M$ par conséquent, comme l'augmentation de M , entraîne un étirement du lobe principal. En fait, même si la largeur de chaque lobe diminue avec une augmentation de M , la hauteur de chaque lobe augmente avec une augmentation M tel que les superficies de chaque lobe reste indifférent aux changements dans M . Cette situation est indésirable du fait que l'objectif est de réduire au maximum le niveau des lobes secondaires.

Il faut noter ici que la fenêtre rectangulaire a le lobe principal le plus étroit possible de toutes les fenêtres de la même longueur, cependant, ses lobes secondaires sont les plus élevés. Une atténuation de 13dB au niveau du premier lobe secondaire de la fenêtre rectangulaire par rapport au lobe principal est considérée comme indésirable dans la plupart des applications.

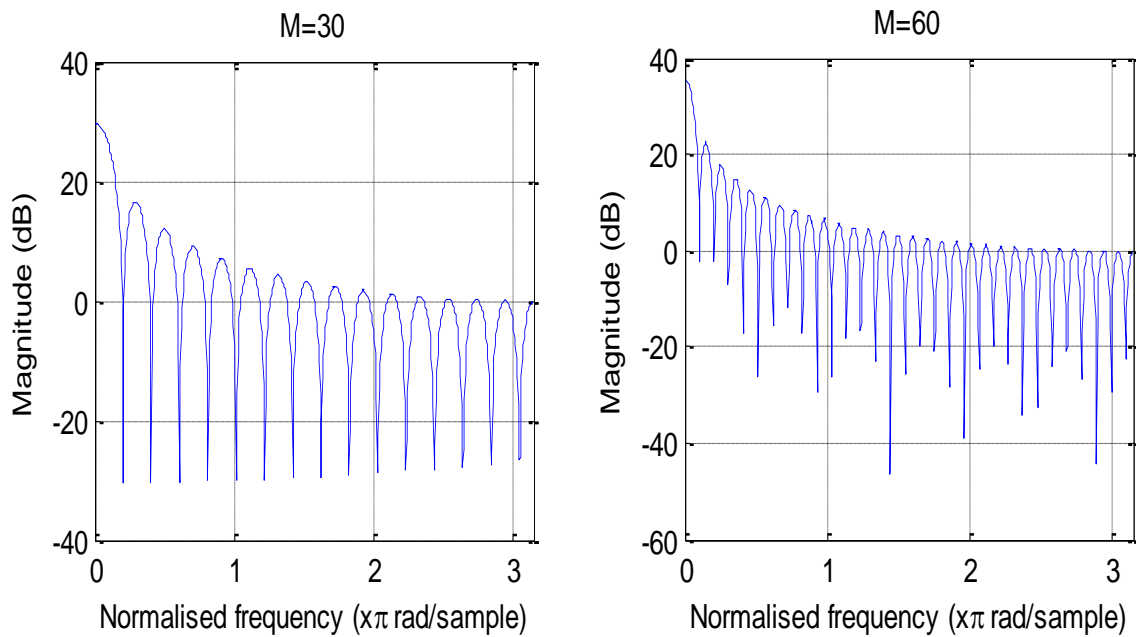


Figure (1.3) : Réponse en fréquence pour la fenêtre rectangulaire de longueur M=31 et M=61.

1.3.3. Fenêtres usuelles

Pour remédier au problème des niveaux des lobes latéraux de la fenêtre rectangulaire, plusieurs fenêtres ont été proposées mais au détriment d'un élargissement du lobe principal. Les fenêtres les plus populaires sont résumées dans le tableau (1). Il englobe la fenêtre rectangulaire, la fenêtre de Hanning, la fenêtre de Hamming et la fenêtre Blackman.

Tableau (1.1) : Fonctions de fenêtre pour la conception de filtre de FIR.

Fenêtre	Expression	lobe latéral maximal	Bonde de transition
Rectangulaire	1	21	$\frac{1.8\pi}{M}$
Hanning	$\frac{1}{2} \left(1 - \cos \frac{2\pi n}{M-1} \right)$	44	$\frac{6.2\pi}{M}$
Hamming	$0.54 - 0.46 \cos \frac{2\pi n}{M-1}$	53	$\frac{6.6\pi}{M}$
Blackman	$0.42 - 0.5 \cos \frac{2\pi n}{M-1} + 0.8 \cos \frac{4\pi n}{M-1}$	74	$\frac{11\pi}{M}$

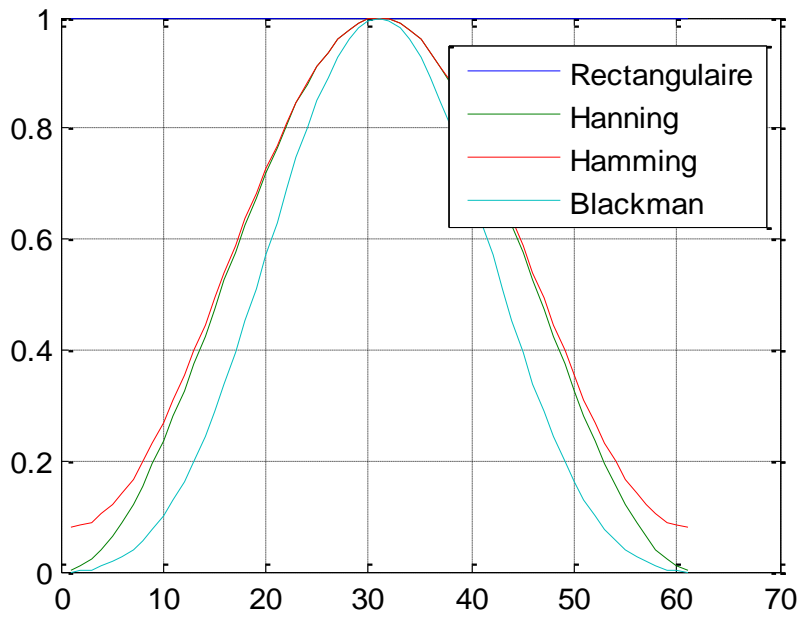


Figure (1.4) : Les fenêtres usuelles

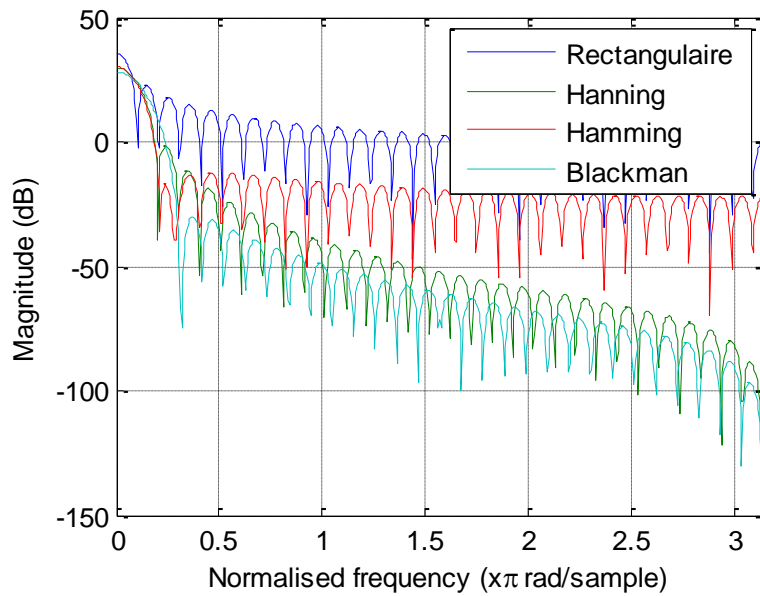


Figure (1.5) : Réponse fréquentielle des fenêtres usuelles.

Idée de conception de fenêtre de base pour les spécifications de filtre données, choisissez la longueur de filtre M et une fonction de fenêtres $w(n)$ pour la largeur de lobe principale la plus étroite et l'atténuation de lobe latérale la plus petite.

Pour cette condition on préfère la fenêtre 'Hamming' parce qu'elle choisit petite lobe latérale (-47) et la une petite largeur du pic central.

1.4. Conception d'un filtre FIR par échantillonnage fréquentiel

En plus des méthodes de conception par la transformée de Fourier et le fenêtrage discuté dans la section précédente, la méthode de l'échantillonnage fréquentiel est une autre alternative. Le principe de cette méthode est de calculer les coefficients de la réponse impulsionnelle à partir d'un nombre d'échantillons $H(k)$ de la réponse fréquentielle désirée.

Pour commencer, on spécifie la réponse fréquentielle désirée $H_d(\omega)$ à un ensemble de fréquences équidistantes, à savoir :

$$\omega_k = \frac{2\pi k}{M+1} \quad k = 0, 1, \dots, M \quad (1.5)$$

où M est l'ordre du filtre. Cet échantillonnage nous permet d'extraire les échantillons de la réponse fréquentielle désirée :

$$H_d(k) = H_d(e^{j\omega}) \Big|_{\omega_k = \frac{2\pi k}{M+1}}$$

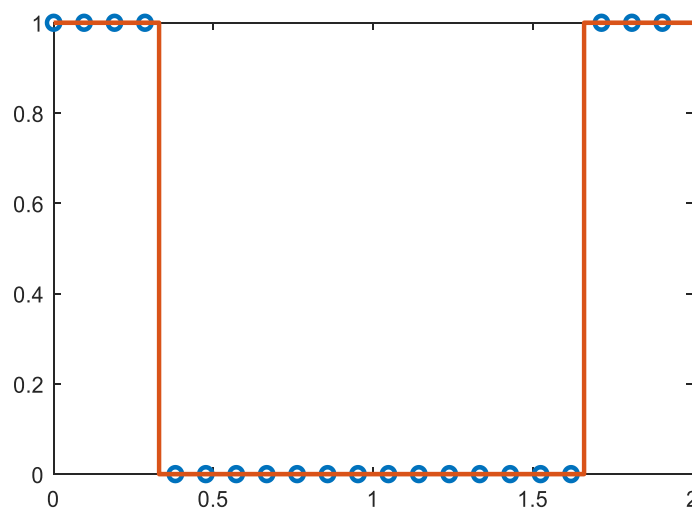


Figure (1.6) : Réponse fréquentielle avec les échantillonnages fréquences

Concernant la phase, il faut fixer le type du filtre à réaliser. Pour un filtre symétrique, la phase de la réponse fréquentielle est de la forme $\theta(\omega) = -\frac{M}{2}\omega$. Cependant pour un filtre antisymétrique la phase est de la forme $\theta(\omega) = -\frac{M}{2}\omega + \frac{\pi}{2}$.

Seulement ici, il faut noter les coefficients du filtre sont réels, ce qui entraîne une symétrie conjuguée de son spectre :

$$H(k) = H^*(M - k + 1) \tag{1.5}$$

pour $k = 1, \dots, M/2$ pour M pair et $k = 1, \dots, (M - 1)/2$ pour M impair

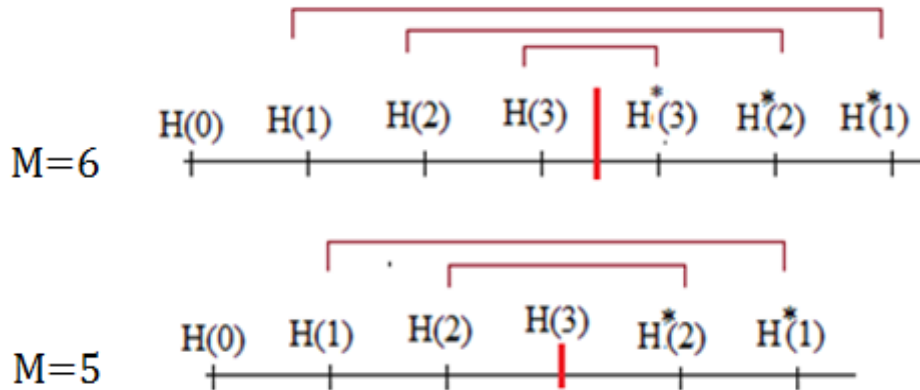


Figure (1.7) : Exemple d'illustration

Une fois les phases sont calculées, On construit les échantillons de la réponse fréquentielle point par points (chaque échantillon de la réponse réelle avec sa phase correspondante).

Finalement il suffit d'utiliser la transformée de Fourier discrète inverse pour trouver les coefficients de $h(n)$

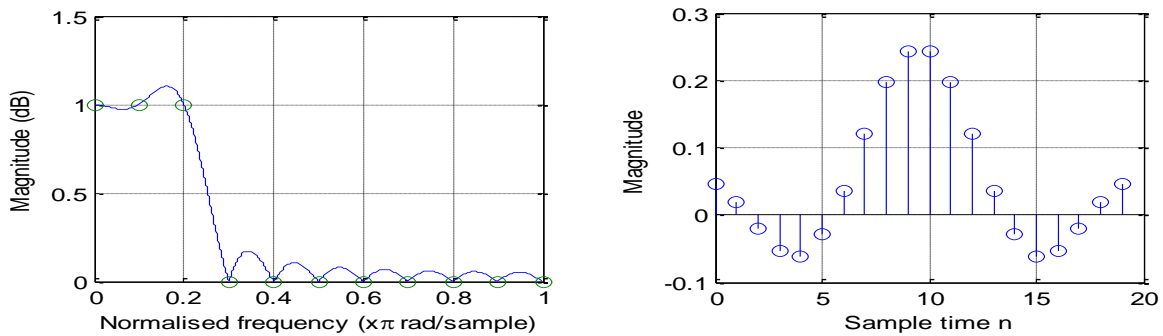


Figure (1.8) : Exemple de synthèse d'un filtre FIR

1.5. Conception de filtres numériques FIR avec moindres carrés

La synthèse des filtres FIR par les moindres carrés est une méthode utilisée dans le traitement du signal numérique pour concevoir des filtres qui minimisent l'erreur entre la réponse souhaitée et la réponse réelles du filtre sur une plage de fréquences spécifiée. Pour utiliser la méthode des moindres carrés, il faut commencer par :

- Définir la réponse désirée $H_d(\omega)$ sur la plage 0 à π .
- Choisir le type et l'ordre du filtre M selon la réponse désirée.
- Créez une grille fine de fréquences ω_k pour $k=0,1,\dots,N-1$, où N est le nombre d'échantillons de fréquence.

La réponse fréquentielle d'un filtre FIR à phase linéaire est

$$H(e^{j\omega_k}) = e^{j\theta(\omega_k)} H(\omega_k) \quad (1.6)$$

Où $H(\omega_k)$ est la réponse en amplitude du filtre. Supposons qu'on ait choisi un filtre FIR type I ayant la réponse en amplitude est :

$$H(\omega_k) = h(0) + 2 \sum_{n=1}^{M/2} h(n) \cos(n\omega_k), \quad k = 0, \dots, N-1 \quad (1.7)$$

La méthode des moindres carrés sert à formuler l'erreur entre la réponse désirée et la réponse réelle pour chaque fréquence ω_k

$$E(\omega_k) = \{H_d(\omega_k) - H(\omega_k)\} \quad (1.8)$$

L'objectif est de trouver les coefficients $h(n)$ inclus dans le terme $H(\omega_k)$ minimisant l'erreur quadratique sur la grille de fréquences :

$$J = \sum_{k=0}^{K-1} |E(\omega_k)|^2 \quad (1.9)$$

En remplace l'expression de $H(\omega_k)$ dans l'équation (1.7) :

$$E(\omega_k) = \left\{ H_d(\omega_k) - \left(h(0) + 2 \sum_{n=1}^{M/2} h(n) \cos(n\omega_k) \right) \right\} \quad (1.10)$$

Pour toutes les fréquences ω_k

$$\begin{bmatrix} E(\omega_0) \\ E(\omega_1) \\ \vdots \\ E(\omega_{N-1}) \end{bmatrix} = \begin{bmatrix} H_d(\omega_0) \\ H_d(\omega_1) \\ \vdots \\ H_d(\omega_{N-1}) \end{bmatrix} - \begin{bmatrix} 1 & 2\cos(\omega_0) & \cdots & 2\cos(\omega_0 M/2) \\ 1 & 2\cos(\omega_1) & \ddots & 2\cos(\omega_1 M/2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2\cos(\omega_{N-1}) & \cdots & 2\cos(\omega_{N-1} M/2) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M/2) \end{bmatrix}$$

Ou sous la forme compacte :

$$\mathbf{E} = \mathbf{H}_d - \mathbf{A}\mathbf{h} \tag{1.11}$$

D'où :

$$J = \|\mathbf{H}_d - \mathbf{A}\mathbf{h}\|^2 \tag{1.12}$$

La minimisation de J donne la solution de cette équation qui sont les coefficients du filtre :

$$\mathbf{h} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d} \tag{1.13}$$

\mathbf{A}^H est la transposée conjuguée de \mathbf{A} .

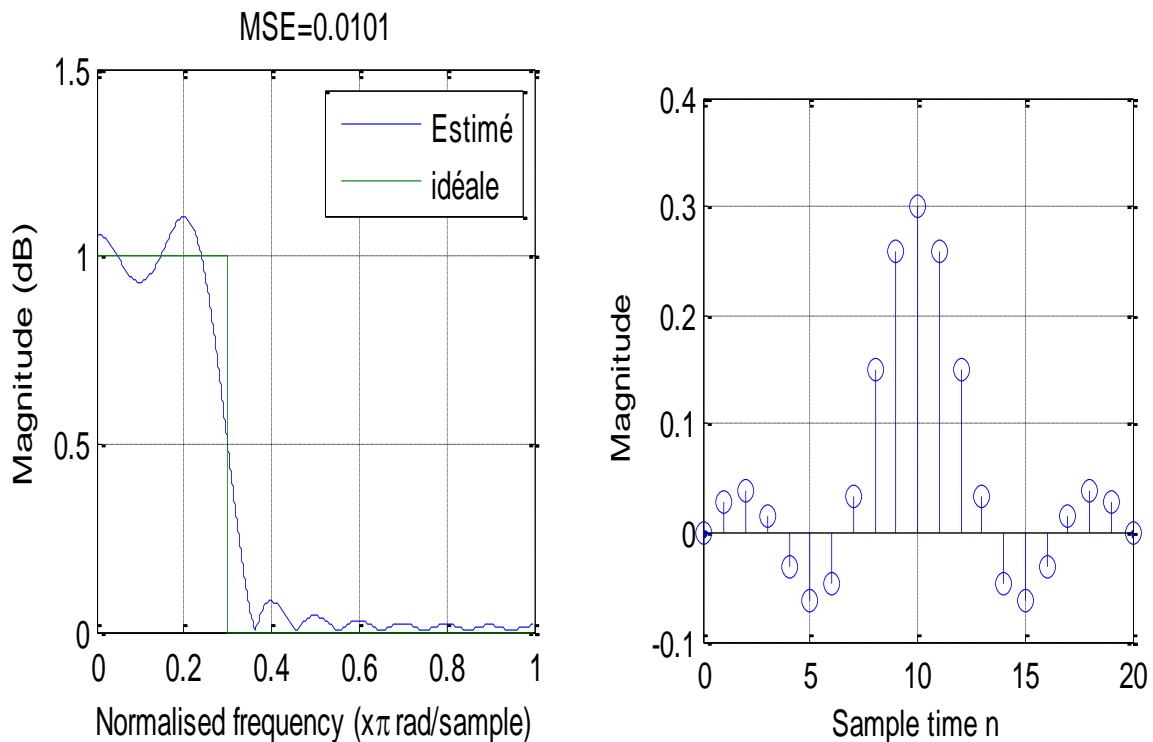


Figure (1.9) : Exemple de synthèse d'un filtre FIR par les moindres carrés

1.6. Conception de filtres numériques FIR avec l'algorithme de Parks-McClellan

La conception de filtres numériques FIR peut être réalisée de manière efficace en utilisant l'algorithme de Parks-McClellan. Cet algorithme, également connu sous le nom d'algorithme de Remez, permet de concevoir des filtres qui minimisent l'erreur de la bande passante en utilisant une approche basée sur la théorie des moindres carrés, voici les équations essentielles impliquées dans la conception d'un filtre FIR en utilisant l'algorithme de Parks-McClellan :

Réécrivant $H_r(\omega)$ comme :

$$H_r(\omega) = Q(\omega)P(\omega) \quad (1.14)$$

Où $P(\omega)$ a une forme unique pour les quatre types

$$P(\omega) = \sum_{n=1}^{M/2} a(n) \cos(n\omega) \quad (1.15)$$

et $Q(\omega)$ dépend du type du filtre considéré. Pour un filtre de type I, $Q(\omega) = 1$. En plus $H_d(\omega)$ prend la valeur 1 dans la bande passante et zéro dans la bande atténuée.

Définissant maintenant l'erreur d'approximation pondérée :

$$E(\omega) = W(\omega)(H_d(\omega) - H_r(\omega)) \quad (1.16)$$

$W(\omega)$ est une fonction de pondération qui nous permet de prendre une erreur d'approximation relative dans les deux bandes du filtre. Si $W(\omega) = 1$ dans la bande atténuée, $W(\omega) = \delta_2/\delta_1$

$$W(\omega) = \begin{cases} \frac{\delta_2}{\delta_1} & 0 \leq \omega \leq \omega_p \\ 1 & \omega_s \leq \omega \leq \pi \end{cases} \quad (1.17)$$

$$E(\omega) = W(\omega)[H_d(\omega) - Q(\omega)P(\omega)] = W(\omega)Q(\omega) \left[\frac{H_d(\omega)}{Q(\omega)} - P(\omega) \right] \quad (1.18)$$

Ou bien sous la forme :

$$E(\omega) = \widehat{W}(\omega)[\widehat{H}_d(\omega) - P(\omega)] \quad (1.19)$$

Le problème maintenant est de trouver les paramètres $a(n)$ qui minimisent la valeur maximale de l'erreur d'approximation sur les bandes considérées.

$$\begin{aligned} & \min_{a(n)} [\max_{\omega \in S} |E(\omega)|] \\ & = \min_{a(n)} \left[\max_{\omega \in S} \left| \widehat{W}(\omega) \left[\widehat{H}_d(\omega) - \sum_{n=1}^{M/2} a(n) \cos(n\omega) \right] \right| \right] \quad (1.20) \end{aligned}$$

S représente la bande passante et la bande d'arrêt du filtre désiré.

La solution de ce problème est proposée par Parks-McClellan qui a appliqué un théorème dit théorème d'alternation.

Théorème d'alternation :

Pour que $P(\omega)$ soit la meilleure approximation de $\hat{H}_d(\omega)$, L'erreur $E(\omega)$ présente au moins $L + 2$ fréquences extrêmes de S ($L = M/2$). C.-à-d. qu'il existe au moins $L + 2$ fréquence de S telle que $\omega_1 < \omega_2 < \dots < \omega_{L+2}$ et $E(\omega_i) = -E(\omega_{i+1})$

$$|E(\omega_i)| = \max_{\omega \in S} |E(\omega)|, \quad i = 1, 2, \dots, L + 2 \quad (1.21)$$

Puisque $H_d(\omega)$ et $E(\omega)$ sont constantes dans les deux bandes :

$$\frac{dE(\omega)}{d\omega} = \frac{d}{d\omega} [W(\omega)(H_d(\omega) - H_r(\omega))] = -W(\omega) \frac{dH_r(\omega)}{d\omega} \quad (1.22)$$

Donc les fréquences ω_i qui correspondent au pics de $E(\omega)$ sont aussi des pics de $H_r(\omega)$ ($\frac{dE(\omega)}{d\omega} = 0$ entraîne $\frac{dH_r(\omega)}{d\omega} = 0$).

De meme :

$$H_r(\omega) = \sum_{n=1}^L a(n) \cos(n\omega) = \sum_{n=1}^L a(n) \sum_{k=0}^n \beta_{kn} (\cos(\omega))^n = \sum_{n=1}^L \hat{a}(n) (\cos(\omega))^n \quad (1.23)$$

Donc $H_r(\omega)$ est un polynome de degré L qui aura au plus $L - 1$ minimum ou maximum local. De meme, $\omega = 0$ et $\omega = \pi$ sont souvent des fréquences extrêmes pour $H_r(\omega)$ et pour $E(\omega)$. Par conséquent il y aura $L + 1$ fréquences extrêmes. Il faut noter aussi que $\omega = \omega_p$ et $\omega = \omega_s$ sont aussi des fréquences extrêmes ce qui entraîne que $E(\omega)$ présente au plus $L + 3$ fréquence extrêmes.

Dans les fréquences extrêmes ω_n , $n = 0, 1, \dots, L + 1$

$$\hat{W}(\omega_n) [\hat{H}_d(\omega_n) - P(\omega_n)] = (-1)^n \delta, \quad n = 0, 1, \dots, L + 1 \quad (1.24)$$

où δ est la valeur maximale de $E(\omega)$. ($\delta = \delta_2$ selon $W(\omega)$).

L'équation (1.24) peut se mettre sous la forme :

$$P(\omega_n) + \frac{(-1)^n \delta}{\hat{W}(\omega_n)} = \hat{H}_d(\omega_n), \quad n = 0, 1, \dots, L + 1 \quad (1.25)$$

$$\sum_{k=1}^L a(k) \cos(\omega_n k) + \frac{(-1)^n \delta}{\hat{W}(\omega_n)} = \hat{H}_d(\omega_n) \quad n = 0, 1, \dots, L + 1 \quad (1.26)$$

$$\begin{bmatrix} 1 & \cos(2\omega_0) & \cdots & \cos(\omega_0 L) & \frac{1}{\widehat{W}(\omega_0)} \\ 1 & \cos(2\omega_1) & \ddots & \cos(\omega_1 L) & \frac{-1}{\widehat{W}(\omega_1)} \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ 1 & \cos(2\omega_{L+1}) & \cdots & \cos(\omega_{L+1} L) & \frac{(-1)^{L+1}}{\widehat{W}(\omega_{L+1})} \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(L) \\ \delta \end{bmatrix} = \begin{bmatrix} \widehat{H}_a(\omega_0) \\ \widehat{H}_a(\omega_1) \\ \vdots \\ \widehat{H}_a(\omega_{L+1}) \end{bmatrix}$$

Initialement ω_n , $a(k)$ et δ sont tous inconnus. Pour résoudre ce problème on utilise l'algorithme de Remez [] où on commence par une initialisation des ω_n , détermine δ et $P(\omega)$ puis on calcule $E(\omega)$. à partir de $E(\omega)$ on détermine des nouveaux ω_n et on répète le processus jusqu'à la convergence.

1.7. Conclusion

La conception des filtres FIR peut être réalisée par différentes méthodes, chacune ayant ses propres avantages et inconvénients. La méthode de troncature de la réponse impulsionnelle est la plus simple mais peut introduire des artefacts indésirables. L'utilisation de fenêtres non rectangulaires et l'échantillonnage de la fréquence offrent des solutions pour minimiser ces effets et apporter plus de flexibilité dans la conception. L'algorithme des moindres carrés est particulièrement utile dans le contexte des applications de filtrage adaptatif, car il équilibre bien la performance et la complexité. L'algorithme de Parks-McClellan, quant à lui, est optimal pour minimiser l'erreur maximale dans la réponse en fréquence du filtre. En plus de leur stabilité intrinsèque, les filtres FIR peuvent atteindre une réponse en phase linéaire, ce qui est essentiel pour certaines applications, en ayant des coefficients symétriques ou antisymétriques. Ainsi, le choix de la méthode de conception dépend largement des exigences spécifiques de l'application visée.

Chapitre 2

Chapitre 2

Les Algorithmes méta-heuristiques

2.1. Introduction

Une puissante collection de stratégies d'optimisation connues sous le nom de méta-heuristiques est utilisée pour résoudre des problèmes difficiles pour lesquels des approches plus conventionnelles peuvent s'avérer improductives ou inefficaces. Ces algorithmes, qui fonctionnent le mieux avec des espaces de recherche vastes et coûteux en termes de calcul, sont conçus pour produire des solutions décentes (et souvent presque optimales) en un temps raisonnable.

Les algorithmes, en particulier les méta-heuristiques, sont bien adaptés aux tâches de modélisation non linéaire et d'optimisation globale. Ils offrent une approche efficace pour générer des solutions satisfaisantes grâce à des processus itératifs d'essais et d'erreurs, en particulier lorsqu'il s'agit de problèmes complexes qui défient une recherche exhaustive en raison de leur complexité. Compte tenu de la complexité de ces problèmes, il est souvent peu pratique d'explorer toutes les solutions ou combinaisons possibles. L'objectif est plutôt de trouver des solutions réalisables et de bonne qualité dans un délai raisonnable. Cependant, rien ne garantit que les meilleures solutions soient identifiées et l'efficacité d'un algorithme n'est pas toujours prévisible. L'objectif principal est de développer des algorithmes pratiques et efficaces capables de fournir systématiquement des solutions de haute qualité, même si l'optimalité ne peut être garantie. En pratique, les méta-heuristiques excellent dans la navigation dans des espaces de solutions complexes, produisant souvent des solutions presque optimales. Bien qu'il n'y ait aucune garantie d'atteindre l'optimalité, les méta-heuristiques offrent un équilibre entre exploration et exploitation, conduisant à la découverte de solutions satisfaisantes répondant aux exigences du problème. Ces solutions ne sont pas toujours les meilleurs possibles, mais elles sont généralement de bonne qualité et fiables à des fins pratiques. Les algorithmes méta-heuristiques algorithmes PSO (Optimisation essaims particules), GWO (Optimiseur de loup gris) et l'algorithme ICSA (Algorithme de recherche de coucou amélioré) sont des méthodologies avancées utilisées pour résoudre des problèmes d'optimisation difficiles. Chacun de ces les algorithmes s'inspirent des observations naturels comportements

naturels pour orienter la recherche de solutions optimales dans des espaces de recherche complexes. Diriger la recherche de solutions optimales dans des situations complexes les espaces.

2.2. Optimisation essaims particules (PSO)

2.2.1. Principe l'algorithme PSO

L'algorithme d'optimisation par essaim de particules (PSO), s'inspire des comportements collectifs observés chez divers animaux tels que les bancs de poissons, les essaims d'insectes et les troupes d'oiseaux. Cette technique vise à trouver l'optimum global de divers ensembles de problèmes. Initialement proposé par Kennedy et Eberhart en 1995, le concept d'optimisation des essaims de particules était motivé par le désir d'imiter les mouvements gracieux mais imprévisibles observés dans les volées d'oiseaux. L'objectif était de découvrir les schémas sous-jacents régissant les capacités collectives de ces animaux. Quant à l'introduction du PSO, PSO est un algorithme de recherche aléatoire proposé à partir de la recherche sur la recherche de nourriture des troupes d'oiseaux. L'algorithme PSO est un processus de simulation d'un troupeau d'oiseaux à la recherche de nourriture, et chaque oiseau du troupeau est considéré comme une particule de la population dans l'algorithme. Autrement dit, une solution possible au problème de la différence minimale entre pic et vallée est résolue par le mouvement des particules. Ces oiseaux sont mobiles lorsqu'ils recherchent de la nourriture. En changeant la position et la vitesse de leur vol, ils continuent de s'approcher de la nourriture jusqu'à ce qu'ils la trouvent, et il en va de même pour le PSO. La particule continue de s'approcher de la solution optimale en mettant à jour sa position et sa vitesse jusqu'à ce qu'elle soit trouvée.

2.2.2. Structure algorithmique PSO standard

Dans l'algorithme PSO, chaque particule représente une solution potentielle au problème d'optimisation. L'algorithme met à jour de manière itérative les emplacements des particules en fonction de l'emplacement précédent de chaque particule, du meilleur emplacement qu'elle a rencontré jusqu'à présent (meilleur emplacement personnel) et du meilleur emplacement qu'une particule de l'essaim a trouvé (meilleur emplacement global) [5].

$$P_{b_i}^t = x_i^* \mid (f(x_i^k) \mid = \min(\{f(x_i^*)\}) \quad (2.1)$$

$$k = 1, 2, \dots, t$$

où $i \in \{1, 2, \dots, N\}$, and

$$g_b^t = x_*^t \mid (f(x_*^t) \mid = \min(\{f(x_i^k)\}) \quad (2.2)$$

$i=1,2,\dots,N$ et $k=1,2,\dots,t$

- i est désigne l'indice de la particule.
- t est le numéro d'itération actuel.
- f est la fonction objectif à optimiser (minimiser).
- p_b est la meilleure position personnelle de la particule.
- g_b est la meilleure position globale trouvée par n'importe quelle particule de l'essaim.

Le mouvement de chaque particule dans le PSO est contrôlé par deux facteurs principaux : l'élan et l'impact de sa meilleure position personnelle et de sa meilleure position globale. La mise à jour de la vitesse et de la position de chaque particule dans PSO est représentée par les équations suivantes [5].

$$v_i^{t+1} = wv_i^t + c_1r_1(P_{b_i}^t - x_i^t) + c_2r_2(g_b^t - x_i^t) \quad (2.3)$$

$$x_i^{(t+1)} = x_i^t + v_i^{(t+1)} \quad (2.4)$$

x_i est le vecteur de position (ou une solution potentielle) de la particule.

- c_1 et c_2 sont des coefficients d'accélération.
- r_1 et r_2 sont des valeurs aléatoires comprises entre 0 et 1.
- P est le poids d'inertie.

La formule (2.3) représente une formule de mise à jour de la vitesse de l'essaim de particules, comprenant trois composantes distinctes. La première partie est la composante "inertie", qui reflète la tendance de la particule à maintenir sa vitesse précédente, indiquant l'inertie. La deuxième partie est la composante "cognition", tandis que la troisième partie est la composante "sociale", signifiant la collaboration entre les particules. La formule (2.4), en revanche, représente une formule de mise à jour de la position de la particule. Le mouvement des particules dans l'algorithme PSO est illustré dans la Figure (2.1) [6].

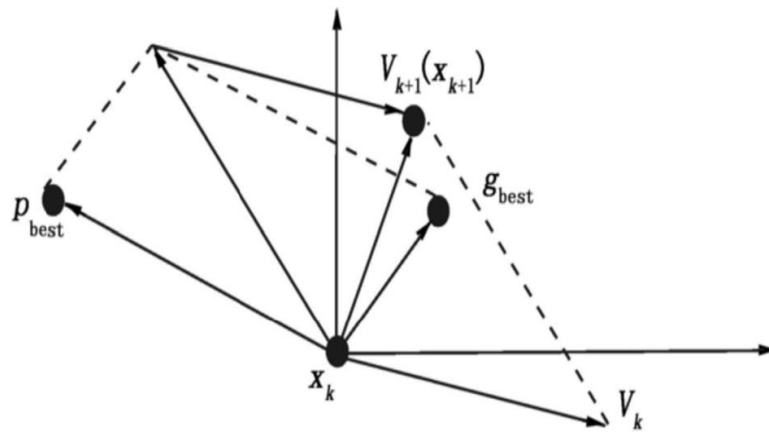


Figure (2.1): Mouvement des particules.

Pour réguler le mouvement des particules dans l'algorithme d'Optimisation essaims particules (PSO), deux techniques courantes sont utilisées : La méthode du « Serrage par vitesse » et la stratégie du « coefficient de constriction ».

Serrage de Vitesse

Les vitesses des particules sont limitées par la définition d'une limite supérieure. Cela empêche les particules de traverser l'espace de recherche trop rapidement, garantissant ainsi qu'elles explorent l'ensemble du domaine de manière approfondie. En plafonnant la vitesse, les particules maintiennent une taille de pas appropriée, les empêchant de dépasser ou de se retrouver piégées dans des optima locaux.

Stratégie du coefficient constriction

Introduite par Clerc et Kennedy, cette méthode contraint les vitesses en incorporant un coefficient de constriction dans l'équation de mise à jour de la vitesse. Ce coefficient met à l'échelle les mises à jour de vitesse sur la base d'observations théoriques de la dynamique des essaims. En ajustant les vitesses en fonction de ce coefficient, on évite aux particules de se déplacer trop rapidement, ce qui améliore l'efficacité du processus de recherche. Les deux techniques servent à contrôler la vitesse des particules et favorisent une exploration et une exploitation efficaces de l'espace des solutions dans les algorithmes PSO. Ils empêchent les particules de « raser » des parties de l'espace de recherche et garantissent qu'elles traversent l'ensemble du domaine de manière appropriée.

2.2.3. Organigramme de l'algorithme PSO

Le schéma fondamental de l'algorithme PSO peut être décomposé en plusieurs étapes. Tout d'abord, il y a l'étape d'initialisation, où les vitesses et les positions des particules sont générées de manière aléatoire. Ensuite, vient l'étape d'évaluation de la population, où la valeur de la fonction d'aptitude de chaque particule est calculée. Après cela, il y a l'étape d'apprentissage des mises à jour des échantillons, où chaque particule compare sa propre valeur de fitness avec celle de son meilleur historique personnel (p_b). Si sa valeur de fitness actuelle est meilleure que son pbest, elle met à jour son pbest avec sa position actuelle. Ensuite, la valeur de fitness du meilleur historique personnel (p_b) est comparée à celle du meilleur de la population (g_b). Si pbest est meilleur, il devient le gbest de la population. Cependant, s'il s'agit d'un problème de grande taille, on peut choisir d'utiliser la meilleure position de pbest plutôt que celle de gbest. Enfin, vient l'étape de révision des particules, où les vitesses et les positions des particules sont mises à jour en fonction des meilleures positions historiques personnelles (p_b) et de la meilleure position de la population (g_b). L'organigramme permettant de démontrer le flux de PSO est représenté sur la figure (2.2) [7].

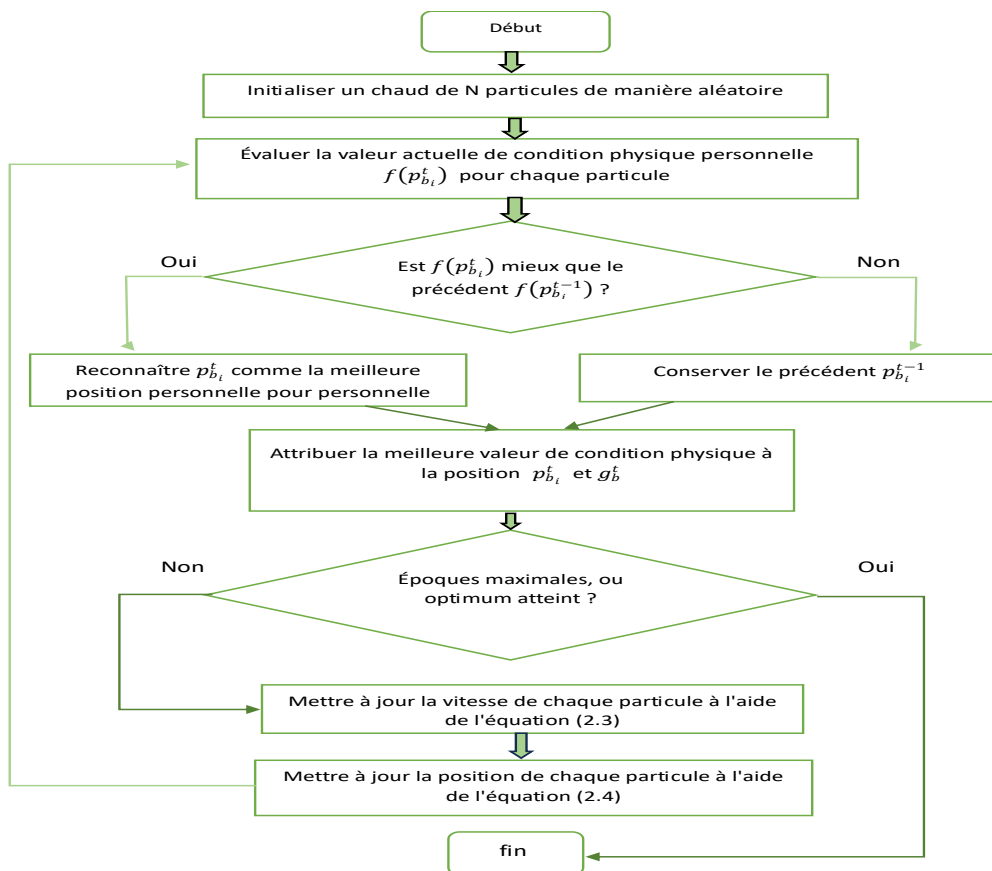


Figure (2.2): Organigramme de l'algorithme PSO.

2.2.4. Procédure l'algorithme PSO

L'algorithme (PSO) est un algorithme d'optimisation méta-heuristique inspiré du comportement social des oiseaux ou des poissons. Il a gagné en popularité en raison de sa simplicité, de son efficacité et de sa capacité à résoudre un large éventail de problèmes d'optimisation. Voici une description détaillée de l'algorithme PSO :

Initialisation

- Définir une taille de population « N » représentant le nombre de particules dans l'essaim.
- Initialisez la position (X_i) et la vitesse (V_i) pour chaque particule « i » dans l'espace de recherche.

Évaluation de la condition physique

Calculez la valeur de fitness « $Fitness[i]$ » pour chaque particule en fonction de la fonction objectif. La valeur de fitness indique à quel point la position d'une particule est « bonne » dans le problème d'optimisation [8].

Mettre à jour votre record personnel

- Comparez la valeur de condition physique actuelle « $Fitness[i]$ » de chaque particule avec sa meilleure position personnelle « $p_b [i]$ ». Ce record personnel représente la meilleure solution que la particule ait trouvée jusqu'à présent.
- Si la valeur de condition physique actuelle est meilleure (pour les problèmes de minimisation : " $Fitness[i] < p_b [i]$ ") ou pire (pour les problèmes de maximisation : " $Fitness[i] > p_b [i]$ ") que le record personnel, mettez à jour le record personnel avec la position actuelle et la valeur de forme physique.

Mettre à jour le meilleur mondial :

- Identifiez la particule ayant la meilleure valeur de fitness dans l'ensemble de l'essaim. Cela représente la meilleure position globale trouvée jusqu'à présent.
- Mettez à jour la meilleure position globale « g_b » avec la position et la valeur de forme physique de cette particule si elle est meilleure que la meilleure position mondiale actuelle (comparaison similaire à celle de l'étape3).

Mise à jour de la vitesse et de la position :

- Mettez à jour la vitesse « V_i » et la position « X_i » de chaque particule en fonction de son

propre meilleur historique (p_b) et du meilleur global (g_b) à l'aide des formules suivantes.

$$v_i^{(t+1)} = wv_i^t + c_{(1)}r_1(p_{b_i}^t - x_i^t) + c_2r_2(g_b^t - x_i^t)$$

$$x_i^{(t+1)} = x_i^t + v_i^{(t+1)}$$

w : poids d'inertie, contrôlant l'influence des vitesses précédentes.

c_1 et c_2 : constantes d'accélération, influençant l'attraction vers le meilleur personnel et mondial.

r_1 et r_2 : nombres aléatoires compris entre 0 et 1.

t : numéro d'itération.

Boucle et terminaison :

Répétez les étapes 2 à 5 jusqu'à ce qu'un critère d'arrêt soit rempli. Ce critère peut être un nombre maximum d'itérations, un niveau de convergence souhaité ou une autre condition prédéfinie.

2.3. Algorithme loups gris (GWO)

C'est Mirjalili qui suggère à l'origine le GWO. La façon dont les loups gris chassent dans la nature et leur nature démocratique ont servi de modèles pour l'algorithme. À l'intérieur d'un système, le leadership social des loups gris est extrêmement rigoureux. Une meute typique de loups gris se compose de cinq à douze individus. Chaque groupe comprend quatre types différents. Selon cette figure (2.3), les loups gris sont alpha (α), bêta (β), delta (δ) et oméga (ω).

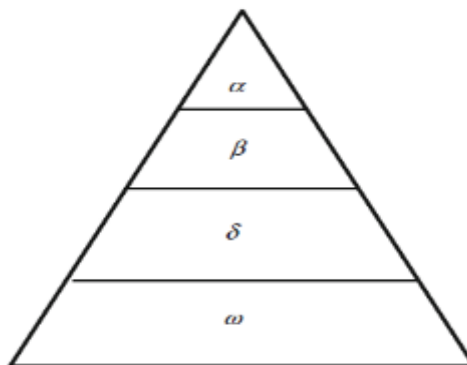


Figure (2.3) : Hiérarchie sociale Loups gris.

Un mâle et une femelle, connus sous le nom d'alpha (α), sont les chefs de groupe. Les loups subordonnés qui soutiennent les dirigeants constituent le deuxième niveau des loups gris, connu sous le nom de bêta (β). Les alphas et les bêtas doivent être subordonnés aux deltas (δ), le troisième niveau des loups gris, mais ils doivent dominer les omegas. Le rang le plus bas du loup gris est l'oméga (ω), ce qui signifie qu'il doit se soumettre à tous les autres loups dominants.

2.3.1. Structure algorithmique GWO standard

Les loups gris chassent en paquets et coexistent ensemble. Voici une description du processus de chasse et de recherche :

- S'ils localisent une proie, ils la suivent, la poursuivent et s'approchent d'elle en premier.
- Les loups gris vont chasser, encercler, et déranger leur proie jusqu'à ce qu'elle cesse de se déplacer si elle s'enfuit.
- Enfin, l'attaque commence.



Figure (2.4) : Comportement de chasse des loups gris : (A) chasser, approcher et suivre des proies (B-D) poursuivre, harceler et encercler (E) situation stationnaire et attaque [1].

La technologie d'optimisation a été créée par Mirghellili en imitant la façon dont les loups gris chassent et chassent. Le modèle mathématique identifie alpha (α) comme le meilleur ajustement, bêta (β) comme le deuxième meilleur et delta (δ) comme le troisième meilleur.

Les solutions candidates restantes sont supposées être des oméga (ω). Ces trois loups gris dirigeront tous les omégas dans leur quête (amélioration) et chasse.

Tourner autour proie

Pendant la chasse, des loups gris entourent leur victime. L’approche circulaire utilisée par les loups gris pour poursuivre leur proie peut être décrite comme.

$$D \vec{r} = | C \vec{r} \cdot (Xp) \vec{r}(t) - X \vec{r}(t) | \tag{2.7}$$

$$\overrightarrow{X(t + 1)} = \overrightarrow{X}p(t) - \vec{A} \cdot \vec{D} \tag{2.8}$$

- t est le numéro d’itération.
- Les vecteurs de coefficients sont A et C.
- Le vecteur X p représente les positions de la proie.
- Les positions des loups gris sont vectorielles par X.
- Le vecteur calculé D est utilisé pour indiquer la nouvelle position du loup gris. Vous pouvez calculer A et C avec les formules suivantes :

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{2.9}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{2.10}$$

On suppose que les vecteurs aléatoires r1 et r2 dans [0, 1] et un vecteur (A) \vec{r} dont la valeur doit diminuer linéairement de 2 à 0 à travers les itérations. La position d’un loup gris à (x, y) peut varier en fonction de l’emplacement de la proie (x', y'), comme le montre la figure (2.5). En manipulant W et W, par exemple, en réglant (A) \vec{r} à (1, 0) et C \vec{r} à (1, 1) plusieurs emplacements pour l’agent le plus idéal peuvent être atteints par rapport à la position actuelle. Le nouvel emplacement du loup gris est (x' - x, y').

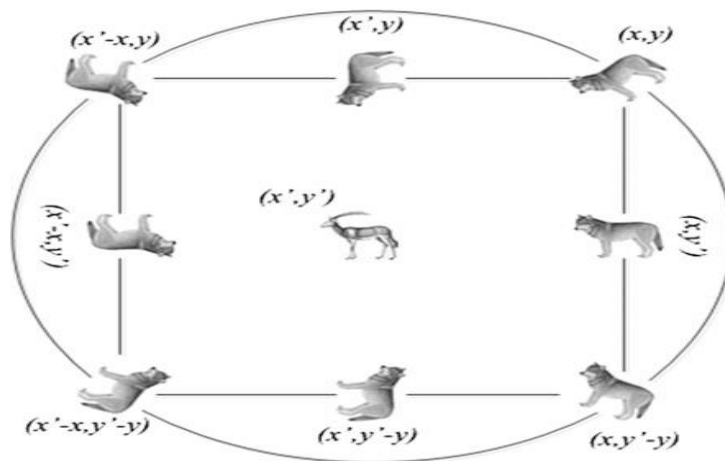


Figure (2.5) : Attaquer une proie ou rechercher une proie

Le loup gris peut sélectionner n'importe quelle position ou noeud grâce aux vecteurs aléatoires r_1 et r_2 . En conséquence, un loup gris peut être placé dans n'importe quel emplacement aléatoire près de sa proie. Les équations 2.9 et 2.10 sont utilisées dans le calcul de la position. Les loups gris peuvent également se déplacer à n'importe quel noeud d'un hypercube dans un espace de décision n-dimensionnel qui est proche de la meilleure option, qui est l'emplacement de la proie. Ils sont capables de distinguer l'emplacement de la proie des autres et de l'entourer. α et β conduisent souvent le processus de chasse, avec δ fournissant un soutien pour α . Par conséquent, afin d'émuler le comportement de persécution des loups gris, on suppose que α (le choix le plus probable pour la solution), β et δ :

Chasse

Les loups gris sont capables de localiser leur proie et de tourner autour d'eux. L'alpha conduit généralement la chasse. De temps en temps, la beta et le delta peuvent aussi aller à la chasse. D'autre part, nous ne savons pas où se trouve l'optimum (preuve) dans un espace de recherche abstrait. Nous supposons que l'alpha (meilleure solution candidate), la bêta et le delta ont des connaissances supérieures concernant l'emplacement possible de la proie afin de recréer mathématiquement le comportement de chasse des loups gris.

En conséquence, nous réservons les trois meilleurs résultats jusqu'à présent et demandons aux autres agents de recherche, y compris les omégas, d'ajuster leurs positions en fonction de la position du meilleur agent de recherche. Dans ce contexte, les formules suivantes sont proposées.

$$\begin{aligned} D_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ D_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ D_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \quad (2.11)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - A_j \cdot (D_\alpha) \\ \vec{X}_2 &= \vec{X}_\beta - A_2 \cdot (D_\beta) \\ \vec{X}_3 &= \vec{X}_\delta - A_2 \cdot (D_\delta) \end{aligned} \quad (2.12)$$

$$\vec{X}(t + 1) = \frac{(\vec{X}_1 + \vec{X}_2 + \vec{X}_3)}{3} \quad (2.13)$$

Loup gris développé attaquant proie

L'attaque de la proie après qu'elle s'est arrêtée est la dernière étape du processus de chasse. Cela peut être réalisé en abaissant progressivement un de 2 à 0. En conséquence, A est modifié

au hasard entre la plage de $[-1, 1]$. On peut l'exprimer ainsi :

$$a \rightarrow = 2 - t \cdot \frac{2}{\text{maxiter}} \quad (2.14)$$

Où t est le nombre d'itérations et maxiter est le total des itérations complétées tout au long de l'optimisation. Dès que l'intervalle $[1 ; 1]$ contient la valeur stochastique de A .

L'emplacement suivant d'un loup peut être quelque part entre son emplacement actuel et celui de sa proie. Lorsque $|A| < 1$, les loups gris deviennent agressifs vis-à-vis de leur proie, comme le montre la figure 2.6.

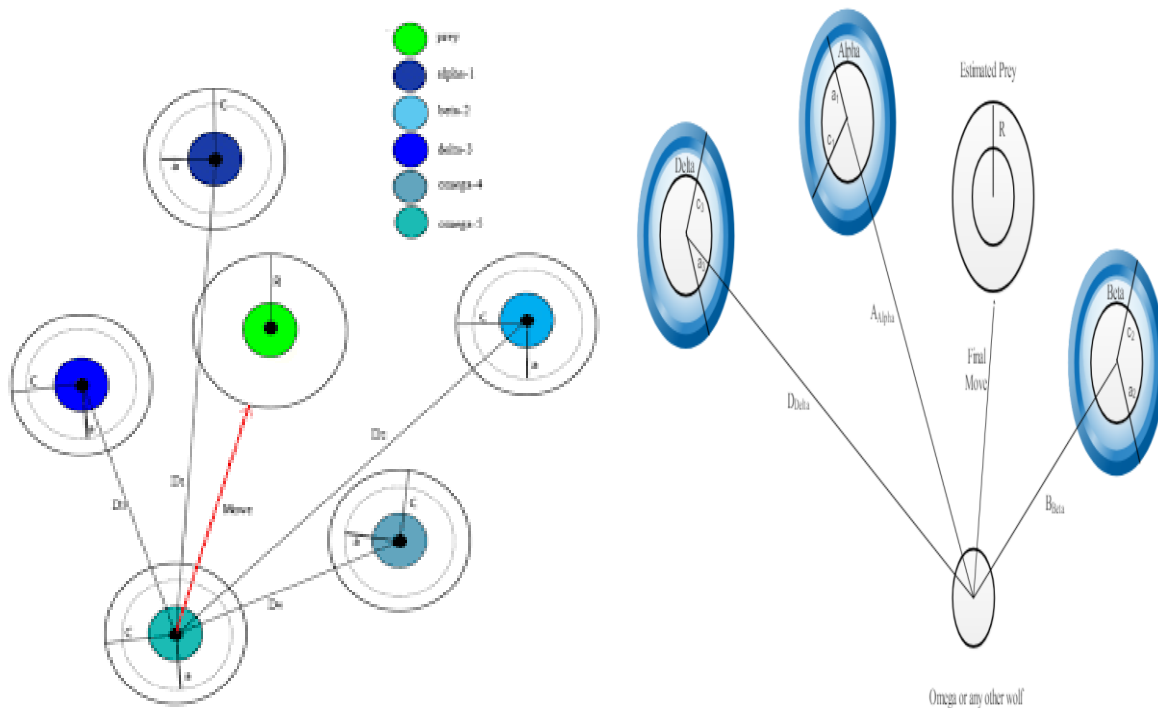


Figure (2.6) : Mise à jour de position dans l'algorithme GWO.

2.3.2. Organigramme de l'algorithme GWO

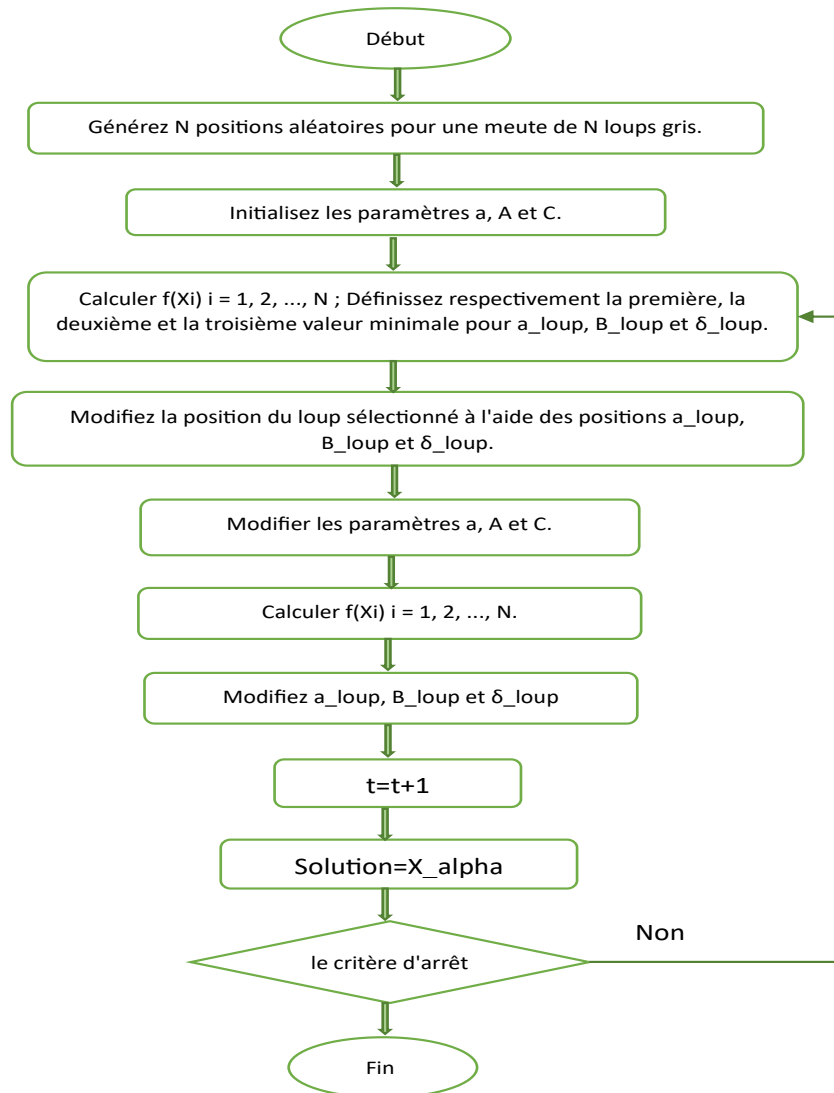


Figure (2.7) : Organigramme de l'algorithme GWO.

2.4. Algorithme recherche coucou amélioré (ICSA)

2.4.1. Algorithme CSA

Dans cette partie, nous donnons un aperçu général du CSA conventionnel. L'algorithme de recherche Crow a été créé récemment par Alireza Askarzadeh. Il s'agit d'un algorithme métaheuristique qui s'inspire des actions de vrais corbeaux. Les corbeaux font preuve d'intelligence tout au long de leur cycle de vie sous la forme de compétences de communication sophistiquées, d'utilisation d'outils, de reconnaissance faciale, de conscience de soi et d'avertissement de groupes de personnes qui pourraient être dangereux. L'endroit où la

nourriture était cachée est rappelé à la fin. Parmi les animaux les plus intellectuels du monde naturel figurent les corbeaux. Pour des raisons de simplicité et de mise en œuvre, une partie, Pa (probabilité de changement), de la population de nids est remplacée par des nids générés aléatoirement, introduisant de nouvelles solutions potentielles. Ce paramètre contrôle l'équilibre entre l'exploration (recherche de nouvelles zones de l'espace de solutions) et l'exploitation (affinement des solutions prometteuses existantes) au sein du processus de recherche de l'algorithme de stimulation des colonies (CSA).

L'étape de recherche locale de l'algorithme utilise une combinaison de marches aléatoires (exploration locale et globale) guidées par le paramètre de commutation Pa. L'équation (2.15) détaille la mise en œuvre mathématique de la marche aléatoire locale [13].

$$x_i^{(t+1)} = x_i^t + \alpha s \odot H(Pa - \epsilon) \odot (x_j^t - x_k^t) \quad (2.15)$$

Où : x_j^t et x_k^t Désigne deux différents Solutions sélectionnées au hasard par hasard permutation, s désigne la taille du pas, $H(u)$ est une fonction Heaviside définie comme une unité fonction discontinue échelonnée dont la valeur est zéro et un pour négatif et positif argument respectivement, ϵ Désigne un nombre aléatoire Sélectionné à partir d'une distribution uniforme. La marche aléatoire globale pour explorer L'espace de recherche de solutions a utilisé un Lévy vols pour générer de nouvelles solutions. C'est modélisé mathématiquement comme l'équation (2.16) :

$$x_i^{(t+1)} = x_i^t + \alpha \oplus L(s, \lambda) \quad (2.16)$$

Où $L(s, \lambda)$ est obtenu en utilisant l'équation (2.17) :

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{S_1 + \lambda}, (S \gg S_0 > 0) \quad (2.17)$$

Dans l'équation (2.16), de nouvelles solutions $x_i^{(t+1)}$ sont généré quand pour un coucou i utilise un vol de Lévy avec un pas $\alpha > 0$ modélisé comme les échelles du problème des intérêts. \oplus désigne une entrée sage multiplication similaire à celle utilisée dans PSO, mais la marche aléatoire basée sur Lévy vol ici explore la solution rechercher espacer plus efficacement à long terme lorsque la longueur du pas est beaucoup plus longue.

2.4.2. Algorithme ICOSA

La capacité de l'CSA à identifier la meilleure option pour des configurations d'espace de recherche particulières a été démontrée. Cependant, en raison de l'inefficacité de sa stratégie de recherche, sa convergence n'est pas garantie. Dans ce scénario, les formulations multimodales élevées posent des défis importants à son approche de recherche. La probabilité de conscience (AP) et la mobilité aléatoire (évasion) sont les deux composantes fondamentales du processus de recherche dans l'approche originale de l'ACS. La clé pour maintenir un équilibre entre l'intensité et la diversification est la valeur de l'AP.

Cependant, en réinitialisant les solutions potentielles, le mouvement aléatoire a un impact direct sur le processus d'exploration. Tant le mouvement aléatoire que la probabilité de conscience (AP) sont réformés dans la technique ICOSA suggérée.

Probabilité conscience dynamique (DAP)

Dans l'approche CSA originale, le paramètre AP est initialement déterminé et n'est pas modifié pendant le processus d'optimisation. Le ratio de la diversité à la densification est négativement affecté par ce phénomène. La probabilité de conscience (AP) a été remplacée par la probabilité de conscience dynamique (DAP), une valeur de probabilité ajustée en fonction de la pertinence de chaque solution potentielle, afin de renforcer cette relation.

La littérature évolutive a efficacement adopté l'utilisation de paramètres de probabilité basés sur les valeurs de forme physique. La formule suivante est utilisée pour calculer la probabilité de conscience dynamique (DAP) :

$$DAP_{i,k} = 0.9 * \frac{F(X_{i,k})}{wV} + 0.1 \quad (2.18)$$

La variable wV désigne la plus faible valeur de fitness observée jusqu'à présent. Cette valeur est calculée comme :

$$wV = \max(F(X_{i,k}))$$

En supposant un problème de minimisation. Les solutions prometteuses seront très probablement exploitées dans cette stratégie probabiliste. En revanche, les solutions de faible qualité sont susceptibles d'être réinitialisées avec une position aléatoire.

Mouvement aléatoire - Vol de Lévy

L'utilisation de moyens pour trouver de la nourriture est vitale pour la survie dans le monde naturel. Une stratégie de recherche inefficace pourrait s'avérer désastreuse pour l'animal. Paul Lévy a décrit les vols de Lévy en 1937. Il s'agit d'une sorte de promenade aléatoire que de nombreuses espèces utilisent comme stratégie de recherche de nourriture.

La distribution de Lévy, qui est une distribution de probabilité à queue lourde, régit la taille du pas dans les vols de Lévy. En ce qui concerne la recherche dans l'espace de recherche, les vols de Lévy sont plus efficaces qu'une distribution aléatoire uniforme.

Au lieu de simuler un comportement d'évitement à travers des déplacements aléatoires uniformes, ICSA suggère d'utiliser des vols de Lévy afin d'améliorer la diversification de l'espace de recherche. Par conséquent, le vol de Lévy calculé L est ajouté à la position existante $X_{i,j}$ pour créer une nouvelle position aléatoire $X_{i,k+1}$

En utilisant la procédure de Mantegna, une distribution symétrique stable de Lévy pour L est obtenue. La taille du pas Z_i est d'abord déterminée en utilisant la méthode de Mantegna de la manière indiquée ci-dessous :

$$Z_i = \frac{a}{|b|^{\frac{1}{\beta}}} \quad (2.19)$$

où $\beta = 3/2$ et a et b sont des vecteurs n -dimensionnels. Les caractéristiques suivantes décrivent la distribution normale à partir de laquelle les éléments de chaque vecteur, a et b , sont échantillonnés :

$$\begin{aligned} a &\sim N(0, \sigma^2 a) \\ \sigma a &= \left\{ \Gamma(1 + \beta) \sin\left(\frac{\pi\beta}{2}\right) \Gamma\left[\frac{1 + \beta}{2}\right] \beta^2 \frac{\beta - 1}{2} \right\}^{\frac{1}{\beta}} \\ b &\sim N(0, \sigma^2 b) \\ \sigma b &= 1 \end{aligned} \quad (2.20)$$

Lorsqu'une distribution gamma est indiquée par $\Gamma(\cdot)$. Le modèle suivant calcule le facteur L après avoir acquis la valeur de Z_i :

$$L = 0.01 - Z_i \odot (X_{i,k} - X_b) \quad (2.21)$$

- X_b représente la meilleure solution
- la nouvelle position $X_{i,k+1}$ est donnée par :

$$X_{i,k+1} = X_{i,k} + L \quad (2.22)$$

2.4.3. Organigramme de l'algorithme IC-SA

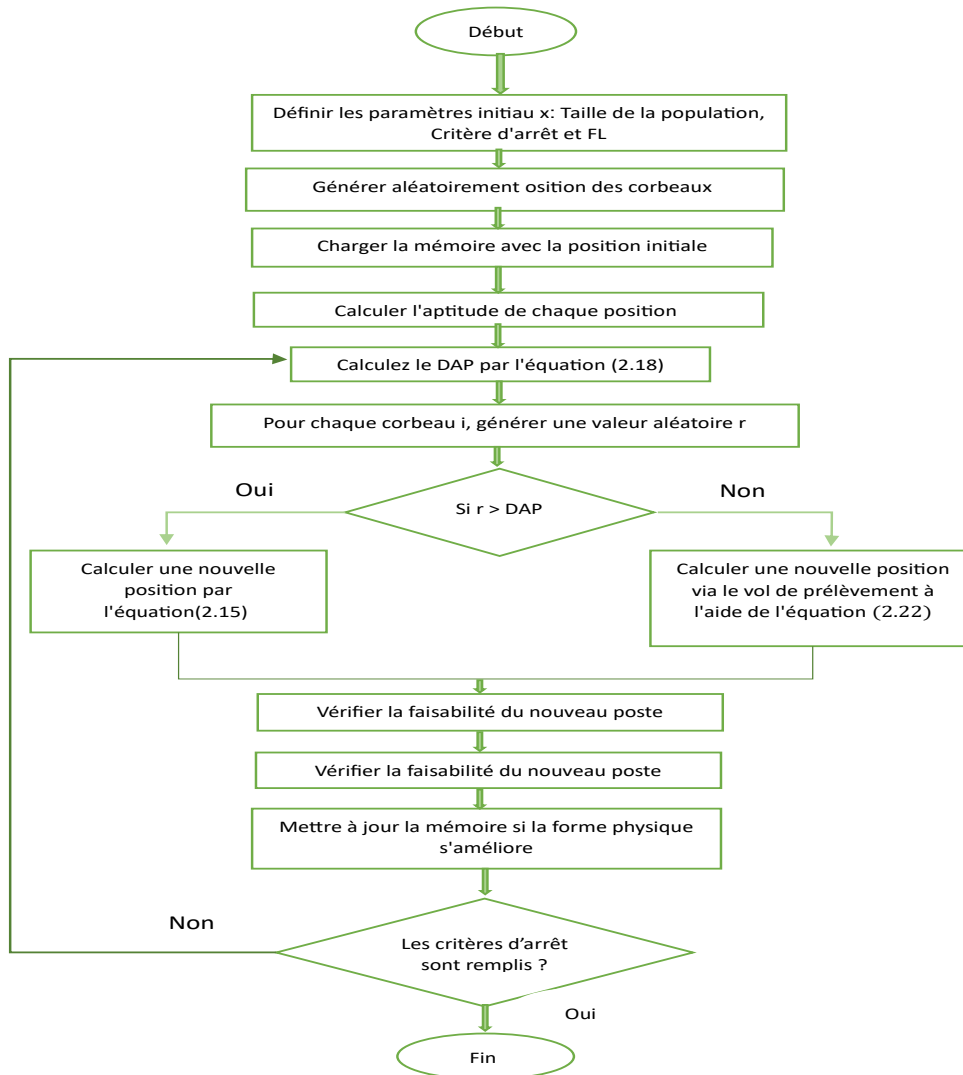


Figure (2.8) : Organigramme de l'algorithme IC-SA.

2.5. Conclusion

Les algorithmes méta-heuristiques PSO, GWO et IC-SA proposent des approches solides pour résoudre des problèmes complexes en s'inspirant des phénomènes naturels pour orienter la recherche des meilleures solutions possibles. Leur polyvalence, leur robustesse et leur efficacité les rendent adaptés à un large éventail d'applications, de l'apprentissage automatique à la gestion des ressources et à l'optimisation des idées.

Le meilleur algorithme à utiliser dépendra de la nature du problème, des ressources disponibles et des résultats souhaités. La détermination de la meilleure stratégie pour un problème particulier peut être accomplie par l'investigation et l'expérimentation de divers algorithmes.

Chapitre 3

Chapitre 3

Les résultats de simulation

3.1. Introduction

Dans ce chapitre, nous détaillerons les étapes de la simulation et de l'implémentation de ces algorithmes pour la conception de filtres passe-bas. Nous évaluerons les performances des filtres en termes de critères tels que la minimisation des ondulations dans la bande passante, la maximisation de l'atténuation dans la bande d'arrêt, la vitesse de convergence des algorithmes, et la complexité computationnelle.

Les résultats des simulations seront comparés à ceux obtenus avec la méthode de Parks-McClellan pour mettre en évidence les avantages et les inconvénients de chaque approche. Cette comparaison nous permettra de déterminer la pertinence et l'efficacité des algorithmes métaheuristiques dans la conception de filtres numériques par rapport aux méthodes classiques.

3.2. Fonction objective

Type 1 FIR filtre :

$$H(e^{jw}) = e^{-j\frac{N}{2}w} \sum_{n=0}^{M/2} a(n) \cos nw \quad (3.1)$$

$$\text{Avec} \quad a(0) = n\left(\frac{M}{2}\right), a(n) = 2n\left(\frac{M}{2} - k\right), k = 1, \dots, \frac{M}{2} \quad (3.2)$$

$$E_s = \frac{1}{\pi} \int_{ws}^{\pi} H^2(w) dw = \frac{1}{\pi} \int_{ws}^{\pi} \left(\sum_{n=0}^{M/2} (a(n) \cos nw) \right)^2 dw \quad (3.3)$$

$$= \frac{1}{\pi} \int_{ws}^{\pi} \{ [a(0)a(1) \dots a\left(\frac{M}{2}\right)] \begin{pmatrix} \cos 0 \\ \cos w \\ \vdots \\ \cos \frac{M}{2}w \end{pmatrix} \}^2 dw \quad (3.4)$$

$$= \frac{1}{\pi} \int_{ws}^{\pi} \begin{bmatrix} a(0) & \dots & a\left(\frac{M}{2}\right) \end{bmatrix} \begin{bmatrix} \cos 0 \\ \cos w \\ \vdots \\ \cos \frac{M}{2}w \end{bmatrix} \begin{bmatrix} \cos 0 & \cos w & \dots & \cos \frac{M}{2}w \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a\left(\frac{M}{2}\right) \end{bmatrix} \quad (3.5)$$

$$= \left[a(0) \dots a\left(\frac{M}{2}\right) \right] \frac{1}{\pi} \int \begin{bmatrix} \cos 0 & \cos w & \dots & \dots & \cos 0 & \cos \frac{M}{2} \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \cos 0 & \cos \frac{M}{2} w & \dots & \dots & \cos \frac{M}{2} w & \cos \frac{M}{2} w \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ \vdots \\ \vdots \\ a\left(\frac{M}{2}\right) \end{bmatrix} \quad (3.6)$$

Cela signifie $(xy)^2 = xy(x^T y^T) = xyx^T y^T$ (3.7)

$$= a^T \frac{1}{\pi} \int_{ws}^{\pi} \cos n w \cos m w dw \quad a = a^T c a \quad (3.8)$$

Avec $c = \frac{1}{\pi} \int_{ws}^{\pi} \cos n w \cos m w \quad n = 0, \dots, \frac{M}{2}$ (3.9)

$$m = 0, \dots, \frac{M}{2}$$

$$\cos a \cos b = \frac{1}{2} \cos(a + b) + \cos(a - b) \quad (3.10)$$

$$\rightarrow c(m, n) = \frac{1}{2\pi} \int_{ws}^{\pi} \cos(m + n) w dw + \frac{1}{2\pi} \int_{ws}^{\pi} \cos(m - n) w dw \quad (3.11)$$

$$\rightarrow \frac{1}{2\pi} \cdot \frac{1}{m+n} \sin(m + n) w \Big| + \frac{1}{2\pi} \cdot \frac{1}{m-n} \sin(m - n) w \Big| \quad (3.12)$$

$$Ep = \frac{1}{\pi} \int_0^{wp} (1 - H(w))^2 dw \quad (3.13)$$

$$= \frac{1}{\pi} \int_0^{wp} (1 - 2H(w) + H^2(w)) dw \quad (3.14)$$

$$= \frac{1}{\pi} \int_0^{wp} dw - \frac{2}{\pi} \int_0^{wp} H(w) dw + \frac{1}{\pi} \int_0^{wp} H^2(w) dw \dots \text{deja calculé}$$

$$= \frac{wp}{\pi} - \frac{2}{\pi} \int \left[a(0) \dots a\left(\frac{M}{2}\right) \right] \begin{bmatrix} \cos 0 \\ \cos w \\ \vdots \\ \vdots \\ \vdots \\ \cos \frac{M}{2} w \end{bmatrix} dw + \frac{1}{\pi} \int H^2(w) dw \quad (3.15)$$

$$= \frac{wp}{\pi} - \frac{2}{\pi} a^T \int_0^{wp} \cos n w dw + \frac{1}{\pi} \int H^2(w) dw \quad (3.16)$$

$$= \frac{wp}{\pi} - \frac{2}{\pi} a^T \int_0^{wp} \frac{1}{\pi} \sin n w + \frac{1}{\pi} \int H^2(w) dw \quad (3.17)$$

$$= \frac{wp}{\pi} - 2a^T p + a^T c a \quad (3.18)$$

Avec
$$p = \frac{1}{\pi} \int_0^{wp} \cos n\omega \, d\omega, n = 0, \dots, \frac{M}{2}$$

$$c = \frac{1}{\pi} \int_0^{wp} \cos n\omega \cos m\omega \, d\omega \quad \begin{cases} n = 0 \dots, \frac{M}{2} \\ m = 0 \dots, \frac{M}{2} \end{cases} \quad (3.19)$$

$$E = u \cdot Ep + (1 - u)Es, \quad 0 < u < 1 \quad (3.20)$$

3.3. L'erreur quadratique moyenne (MSE)

Ya deux méthode de L'erreur quadratique moyenne entre la réponse fréquentielle idéale (désirée), et celle obtenue

$$J_1(\omega) = \frac{1}{L} \sum_{k=0}^{L-1} |H(e^{j\omega_k}) - H_d(e^{j\omega_k})|^2 \quad (3.21)$$

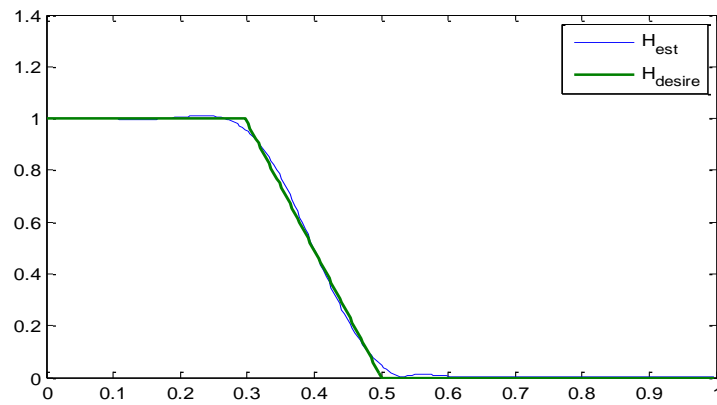


Figure (3.1) : Méthode de L'erreur quadratique 1

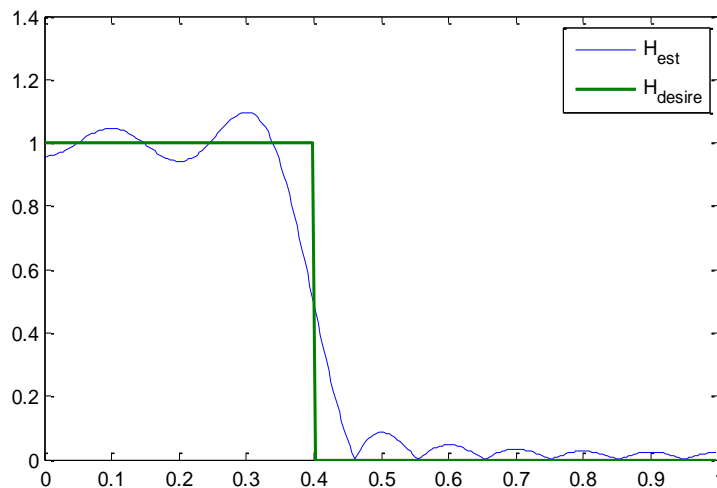


Figure (3.2) : Méthode de L'erreur quadratique 2

3.3.1. Simulation Par MSE1 :

- L'algorithme MSE1-PSO

Le paramètre du filtre qui a été dérivé, pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objective J_1 (3.21) utilisant PSO. Le nombre d'itérations est de 250. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

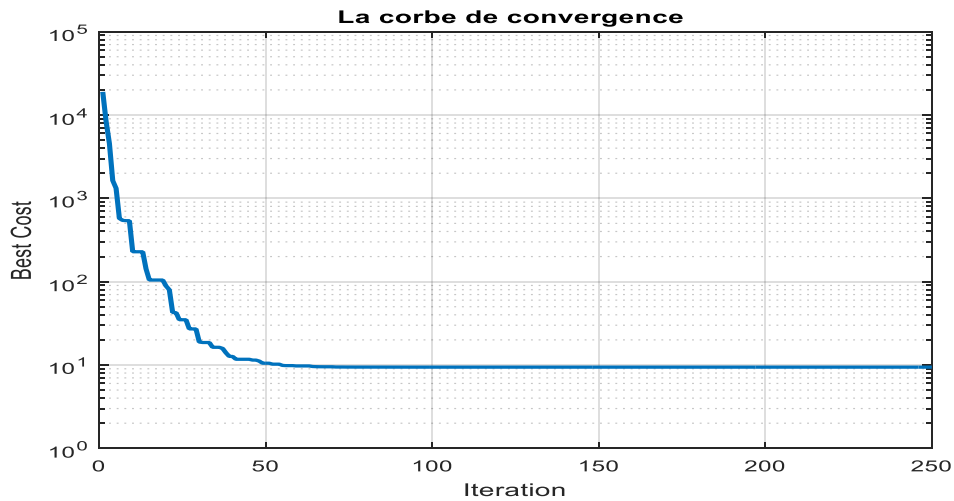


Figure (3.3) : La courbe de convergence MSE1-PSO

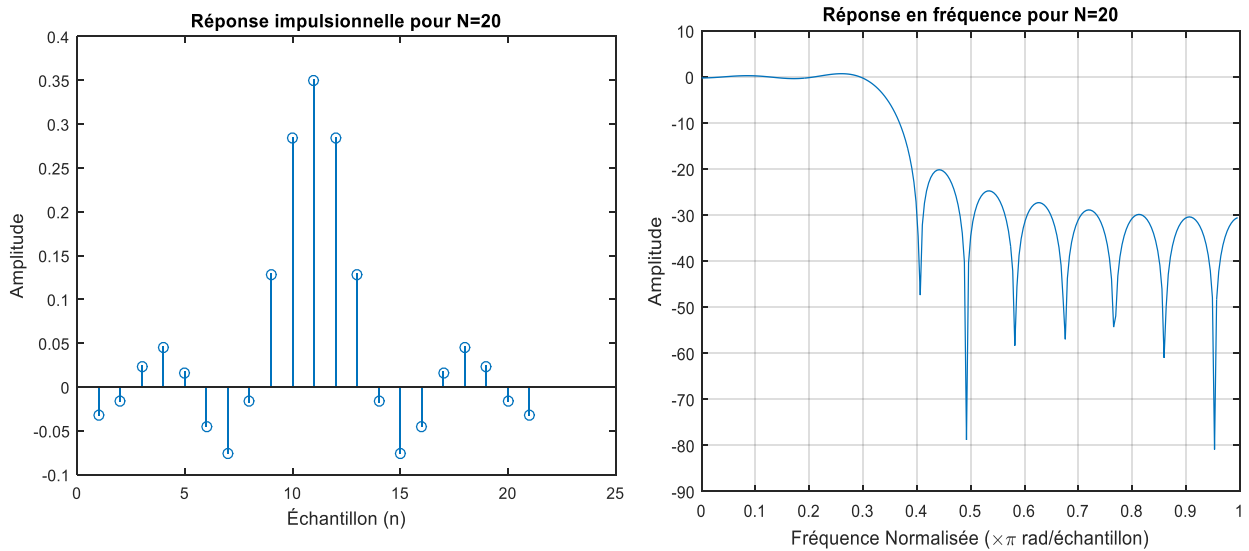


Figure (3.4) : Réponses impulsionnelle et fréquentielle pour MSE1-PSO

Tableau (3.1) : Nouvelles caractéristiques de filtre passe-bas pour MSE1-PSO

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	3.0628	31.9040	30.0769	0.3242	0.5859	0.4551	0.2617
40	2.7002	46.9464	33.4320	0.3281	0.4570	0.3926	0.1289
60	2.8934	43.2942	32.7286	0.3281	0.4687	0.3984	0.1406
80	2.9183	29.8250	29.4916	0.3281	0.5508	0.4395	0.2227

• L’algorithme MSE1-GWO

Le paramètre du filtre qui a été dérivé pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant le fonction objectif J1 (3.21) utilisant GWO. Le nombre d'itérations est de 500. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas

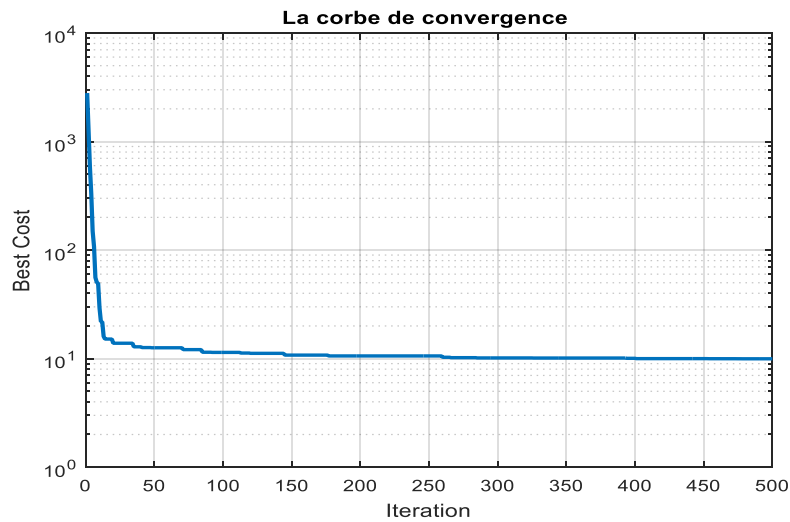


Figure (3.5) : La courbe de convergence MSE1-GWO

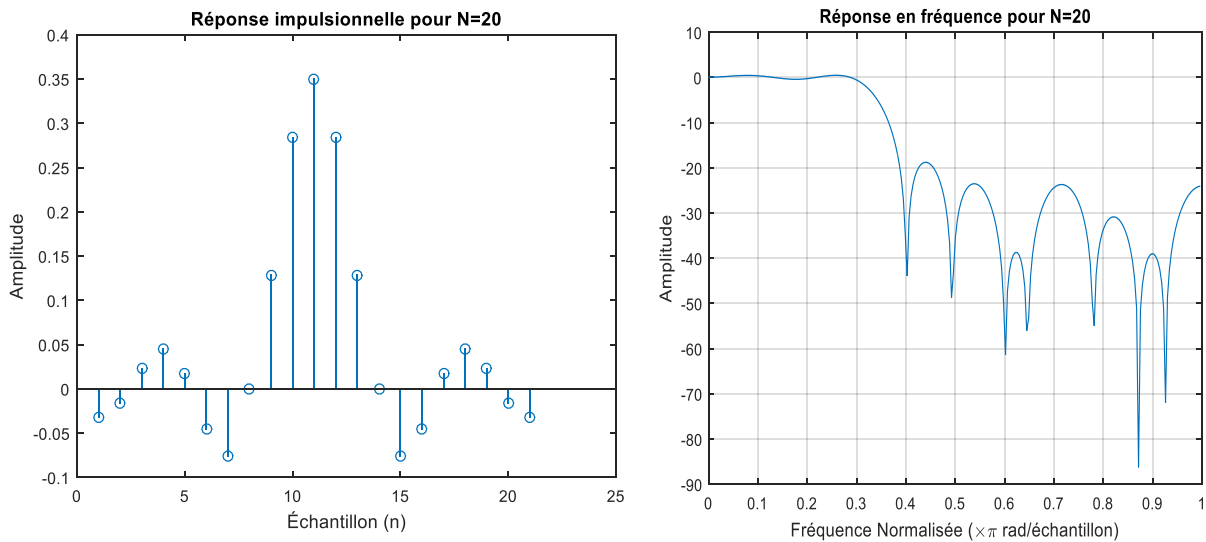


Figure (3.6) : Réponses impulsionnelle et fréquentielle pour MSE1-GWO

Tableau (3.2) : Nouvelles caractéristiques de filtre passe-bas pour MSE1-GWO

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	3.3470	22.6380	27.0968	0.3203	0.4961	0.4082	0.1758
40	3.1857	21.7632	26.7545	0.3242	0.6172	0.4707	0.2930
60	3.2116	21.3359	26.5822	0.3203	0.4414	0.3809	0.1211
80	3.4076	21.7157	26.7355	0.3203	0.4336	0.3770	0.1133

- **L’algorithme MSE1 ICSA**

Le paramètre du filtre qui a été dérivé pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,5 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objectif J1 (3.21) utilisant ICSA. Le nombre de population est de 500. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas

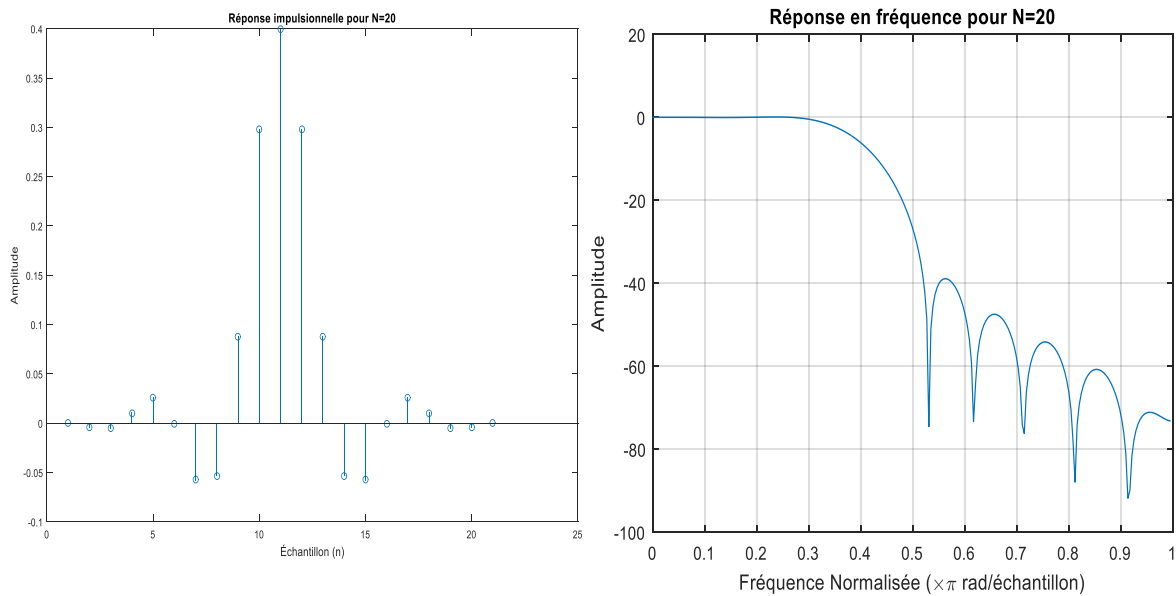


Figure (3.7) : Réponses impulsionnelle et fréquentielle pour MSE1-ICSA

Tableau (3.3) : Nouvelles caractéristiques de filtre passe-bas pour MSE1-ICSA

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	2.8644	38.8784	31.7942	0.3594	0.5313	0.4453	0.1719
40	2.9227	48.4244	33.7013	0.3555	0.5547	0.4551	0.1992
60	2.9960	36.1973	31.1735	0.3555	0.5313	0.4434	0.1758
80	3.8461	22.1188	26.8952	0.3555	0.6172	0.4863	0.2617

3.3.2.Simulation par MSE2 :

- **L’algorithme MSE2-PSO**

Le paramètre du filtre qui a été dérivé, pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objective J1 (3.21) utilisant PSO. Le nombre d'itérations est de 250. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

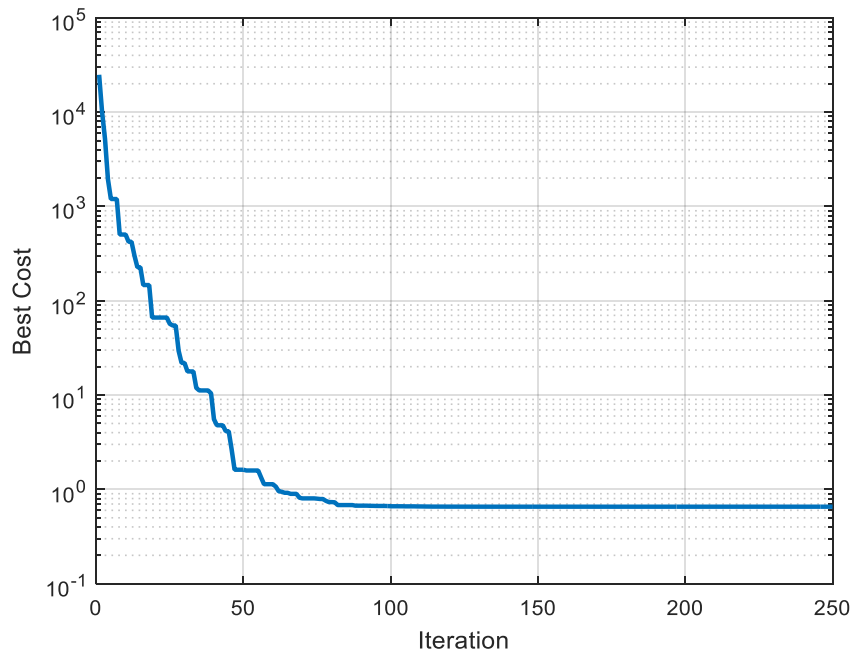


Figure (3.8) : La courbe de convergence MSE2-PSO

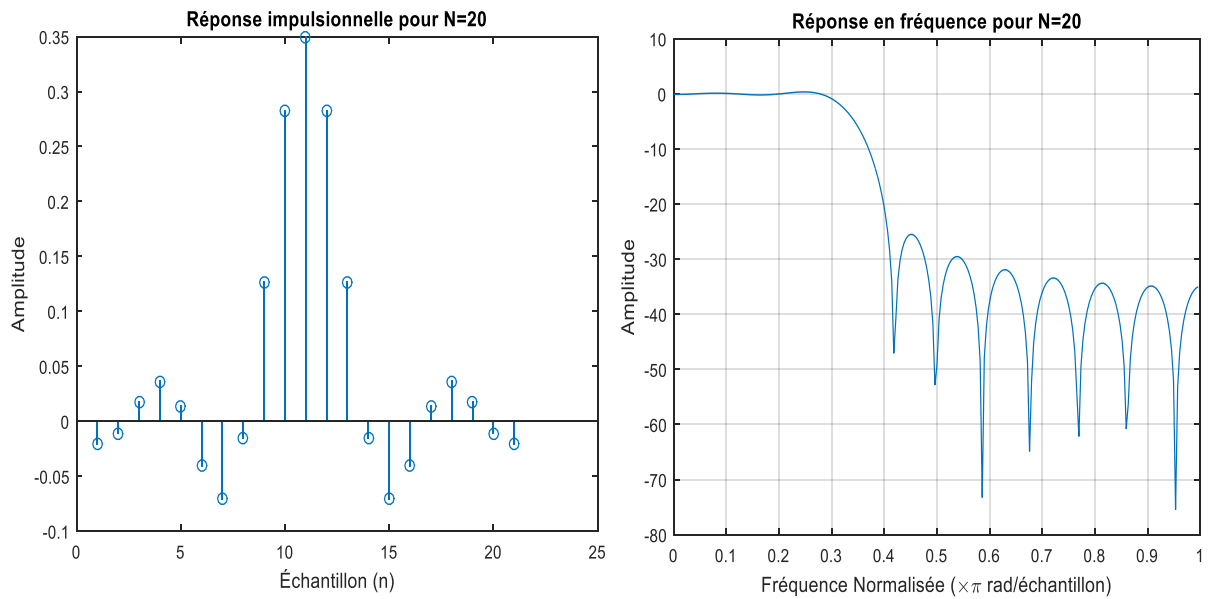


Figure (3.9) : Réponses impulsionnelle et fréquentielle pour MSE2-PSO

Tableau (3.4) : Nouvelles caractéristiques de filtre passe-bas pour MSE2-PSO

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	3.2987	24.7580	27.8743	0.3281	0.4922	0.4102	0.1641
40	3.0674	36.0882	31.1473	0.3359	0.6250	0.4805	0.2891
60	2.6697	33.1305	30.4046	0.3398	0.5273	0.4336	0.1875
80	2.4011	24.1616	27.6625	0.3398	0.4961	0.4180	0.1562

• L’algorithme MSE2 GWO

Le paramètre du filtre qui a été dérivé, pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objective J1 (3.21) utilisant GWO. Le nombre d'itérations est de 500. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

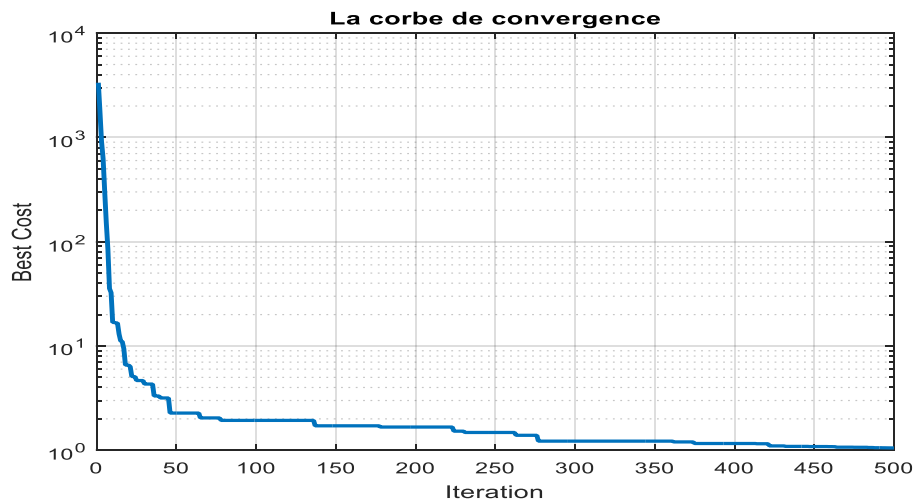


Figure (3.10) : La courbe de convergence MSE2-GWO

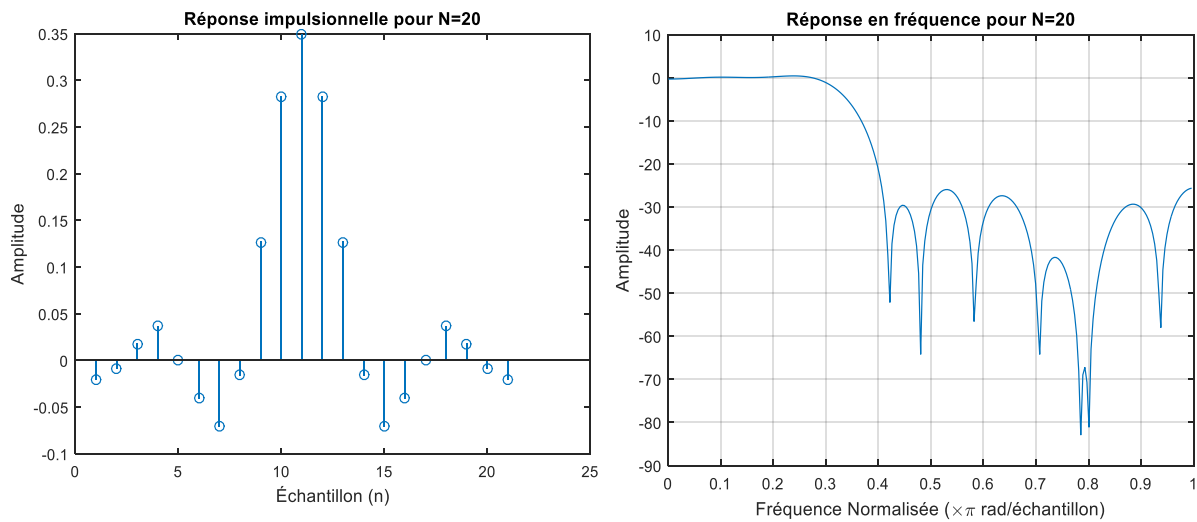


Figure (3.11) : Réponses impulsionnelle et fréquentielle pour MSE2-GWO

Tableau (3.5) : Nouvelles caractéristiques de filtre passe-bas pour MSE2-GWO

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	3.4768	23.2152	27.3154	0.3281	0.4922	0.4102	0.1641
40	3.5234	21.1850	26.5206	0.3320	0.3945	0.3633	0.0625
60	3.3608	16.5735	24.3883	0.3320	0.4609	0.3965	0.1289
80	3.7460	18.6355	25.4068	0.3398	0.6016	0.4707	0.2617

• **L’algorithme MSE2 ICSA**

Le paramètre du filtre qui a été dérivé, pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,5 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objective J1 (3.21) utilisant ICSA. Le nombre d'itérations est de 500. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

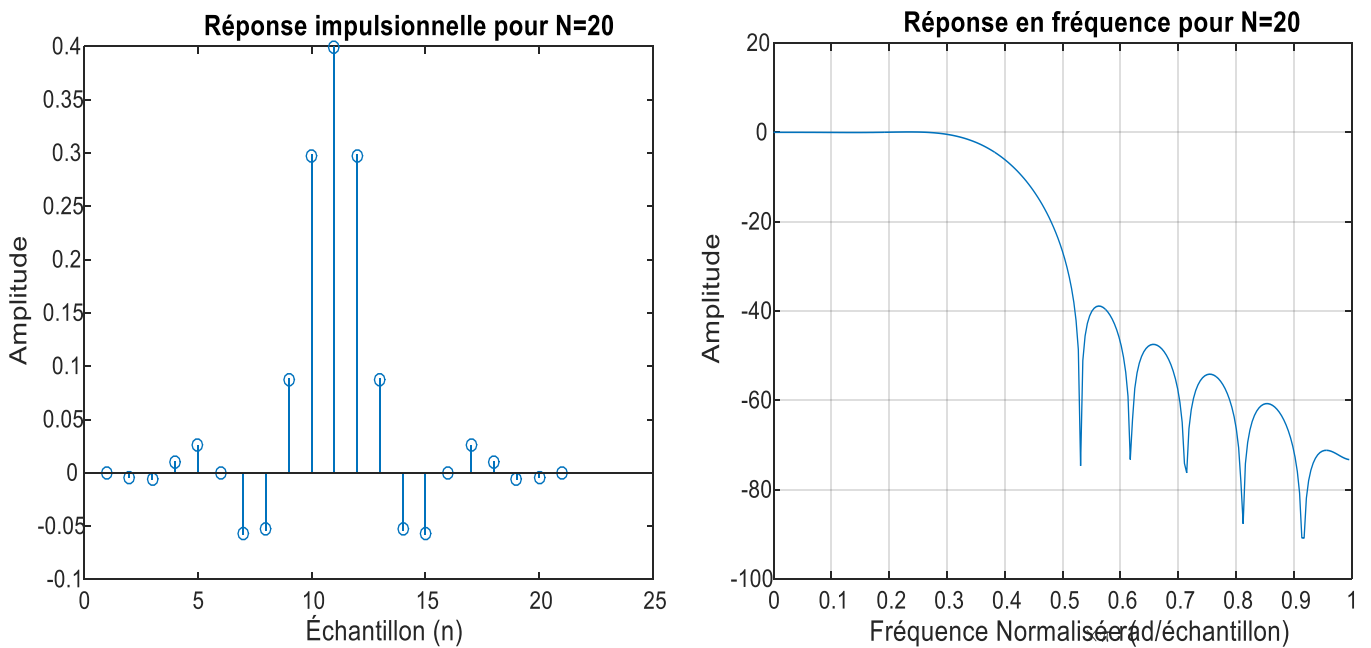


Figure (3.12) : Réponses impulsionnelle et fréquentielle pour MSE2-ICSA

Tableau (3.6) : Nouvelles caractéristiques de filtre passe-bas pour MSE2-ICSA

Ordre	δ_1 (dB)	δ_2 (dB)	A_s (dB)	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	2.8644	38.8767	31.7938	0.3594	0.5313	0.4453	0.1719
40	2.9225	44.8561	33.0364	0.3555	0.5156	0.4355	0.1602
60	2.9265	35.5032	31.0053	0.3555	0.5117	0.4336	0.1563
80	3.3476	24.3309	27.7232	0.3594	0.5469	0.4531	0.1875

3.4. Multi-Objectif (MO) :

3.4.1. Simulation par MO :

Fonction multi-objective :

$$E_p = \frac{1}{\pi} \int_{\omega_p}^{\pi} (1 - H(w))^2 dw \tag{3.22}$$

$$E_s = \frac{1}{\pi} \int_0^{\omega_s} (H(w))^2 dw \tag{3.23}$$

$$\varphi = \mu E_p + (1 - \mu) E_s \quad 0 < \mu \leq 1 \tag{3.24}$$

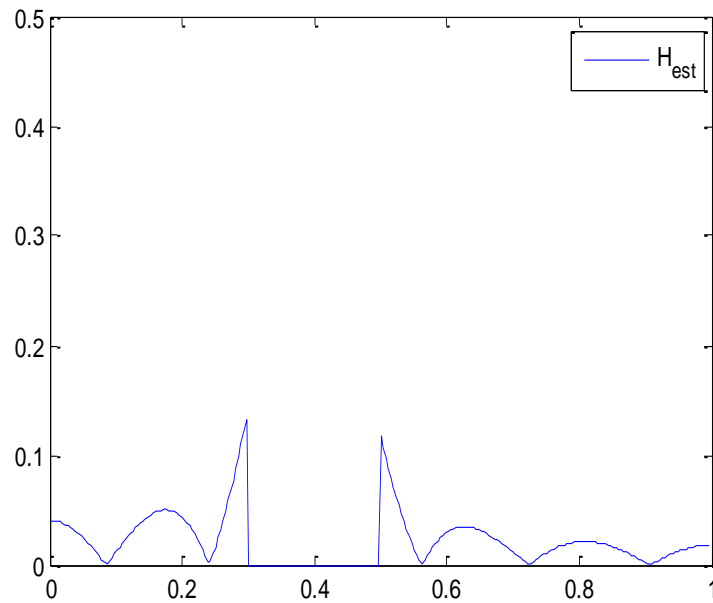


Figure (3.13) : Le principe de Fonction multi-objective

- **L’algorithme MO PSO**

Le paramètre du filtre qui a été dérivé pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d’arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant la fonction objective (3.24) utilisant PSO. Le nombre d’itérations est de 200. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas

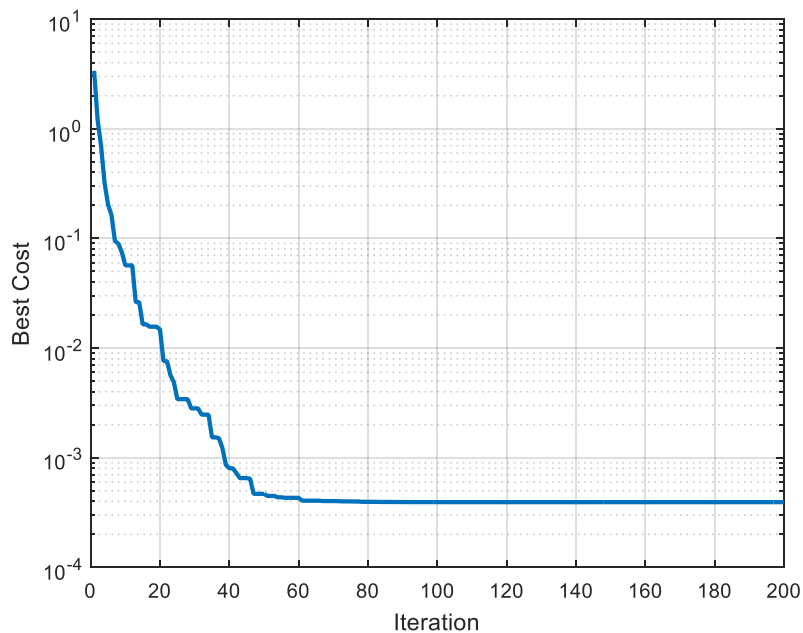


Figure (3.14) : La courbe de convergence MO PSO

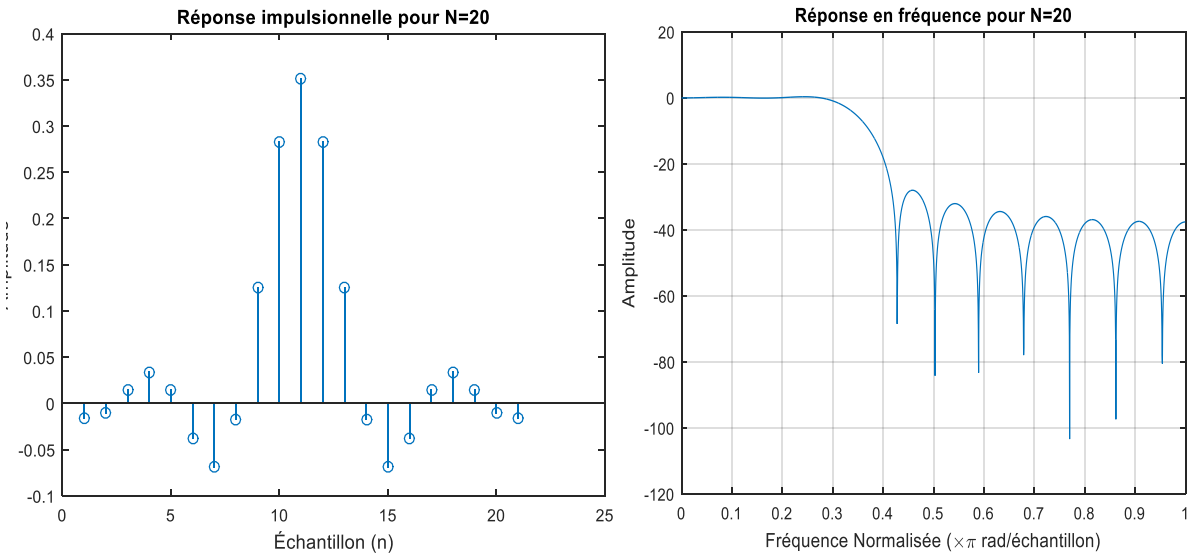


Figure (3.15) : Réponses impulsionnelle et fréquentielle pour MO-PSO

Tableau (3.7) : Nouvelles caractéristiques de filtre passe-bas pour MO-PSO

Ordre	$\delta_1(\text{dB})$	$\delta_2(\text{dB})$	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	2.9838	34.5086	0.3242	0.5898	0.4570	0.2656
40	6.0600	29.4277	0.3789	0.5898	0.4844	0.2109
60	6.6757	34.0938	0.3594	0.5039	0.4316	0.1445
80	29.6623	28.7308	0.3789	0.4766	0.4277	0.0977

• L’algorithme MO GWO

Le paramètre du filtre qui a été dérivé pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant le fonction objective (3.24) utilisant GWO. Le nombre d’itérations est de 500. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

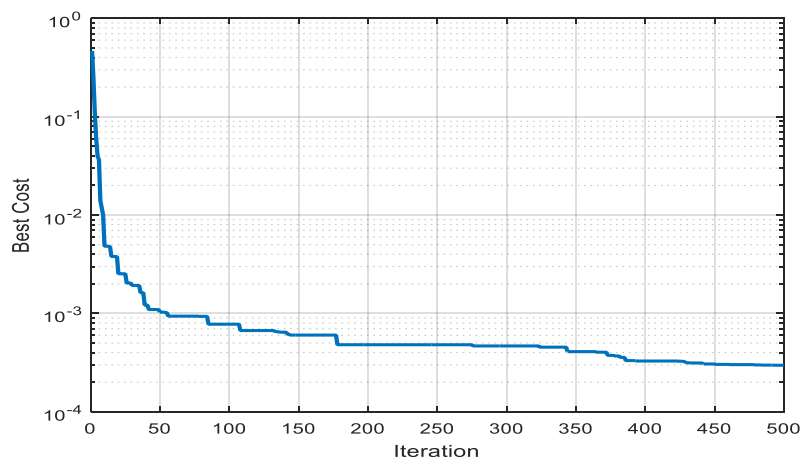


Figure (3.16) : La courbe de convergence MO GWO

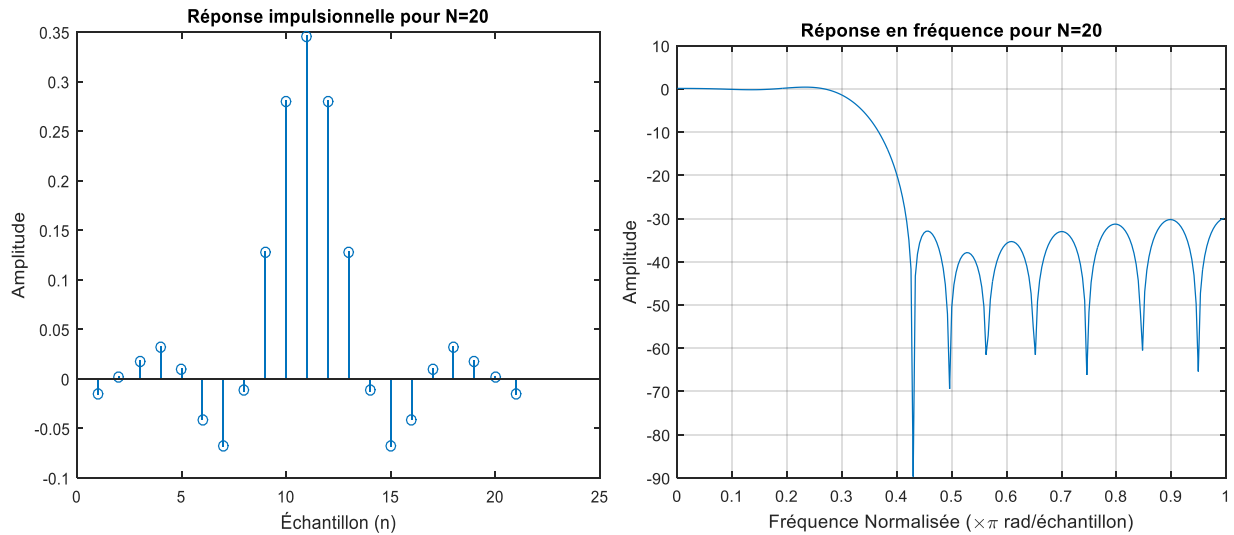


Figure (3.17) : Réponses impulsionnelle et fréquentielle pour MO-GWO

Tableau (3.8) : Nouvelles caractéristiques de filtre passe-bas pour MO-GWO

Ordre	$\delta_1(\text{dB})$	$\delta_2(\text{dB})$	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	3.0571	28.8161	0.3086	0.4961	0.4023	0.1875
40	2.7919	24.6088	0.3711	0.4609	0.4160	0.0898
60	3.3809	25.4233	0.3047	0.5547	0.4297	0.2500
80	8.1613	10.4798	0.3984	0.9258	0.6621	0.5273

• L’algorithme MO ICOSA

Le paramètre du filtre qui a été dérivé pour filtre passe-bas, bord de bande passante fréquence (ω_p) = 0,3, bord de bande d'arrêt (ω_s) = 0,4 et largeur de transition = 0,1. Le filtre est conçu en utilisant le fonction objectif (3.24) utilisant ICOSA. Le nombre de population est de 50. Le MATLAB une simulation a été effectuée pour réaliser le passe-bas.

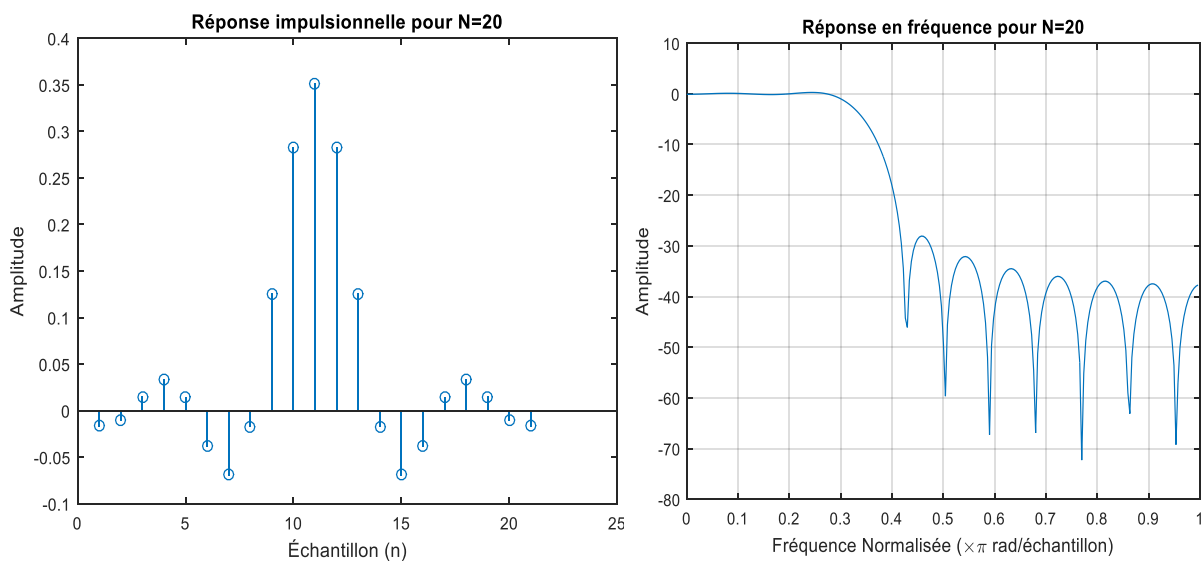


Figure (3.18) : Réponses impulsionnelle et fréquentielle pour MO-ICSA

Tableau (3.9) : Nouvelles caractéristiques de filtre passe-bas pour MO-ICSA

Ordre	$\delta_1(\text{dB})$	$\delta_2(\text{dB})$	$\omega_p \times \pi$	$\omega_s \times \pi$	$\omega_c \times \pi$	$\Delta\omega \times \pi$
20	2.9838	34.5087	0.3242	0.5898	0.4570	0.2656
40	3.0379	40.8906	0.3320	0.4102	0.3711	0.0781
60	2.7276	54.0262	0.3359	0.4063	0.3711	0.0703
80	1.9655	48.3753	0.3125	0.4180	0.3652	0.1055

3.5. Comparaison avec la méthode de Parks–McClellan :

3.5.1. PSO par MSE :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,3$ et une fréquence de coupure $\omega_s=0,4$, les deux graphiques montrent la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et PSO par les deux méthodes de MSE. Les performances des deux méthodes sont visiblement similaires avec des différences notables dans les lobes de la bande d'arrêt.

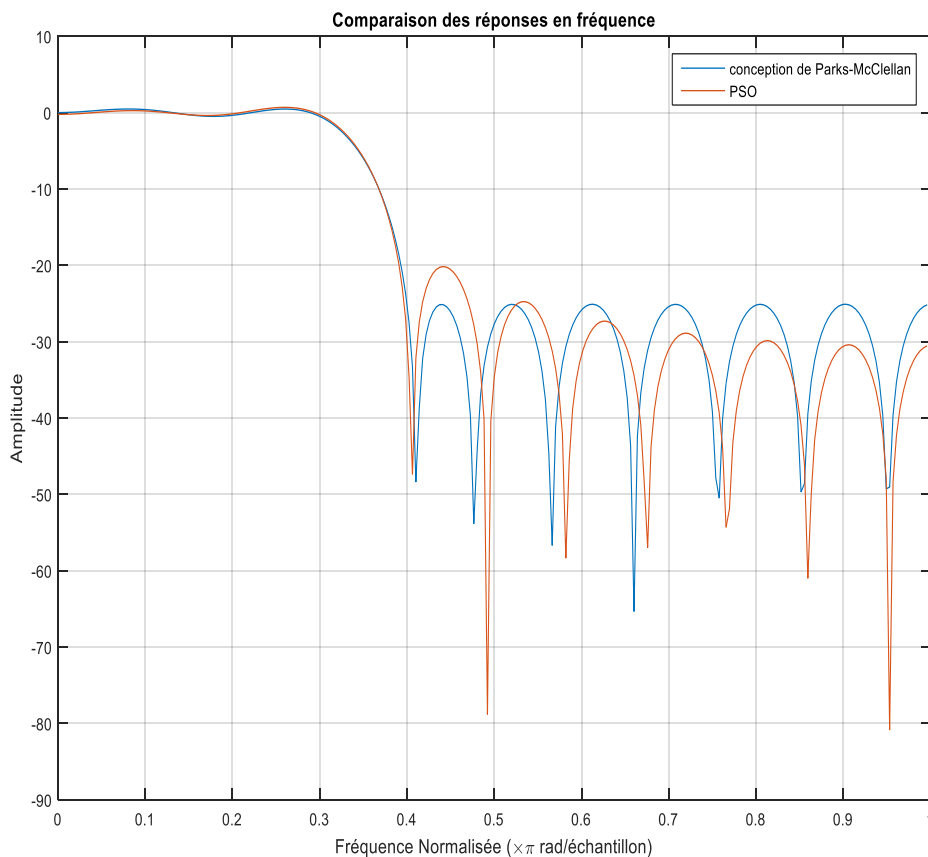


Figure (3.19) : PSOMSE1

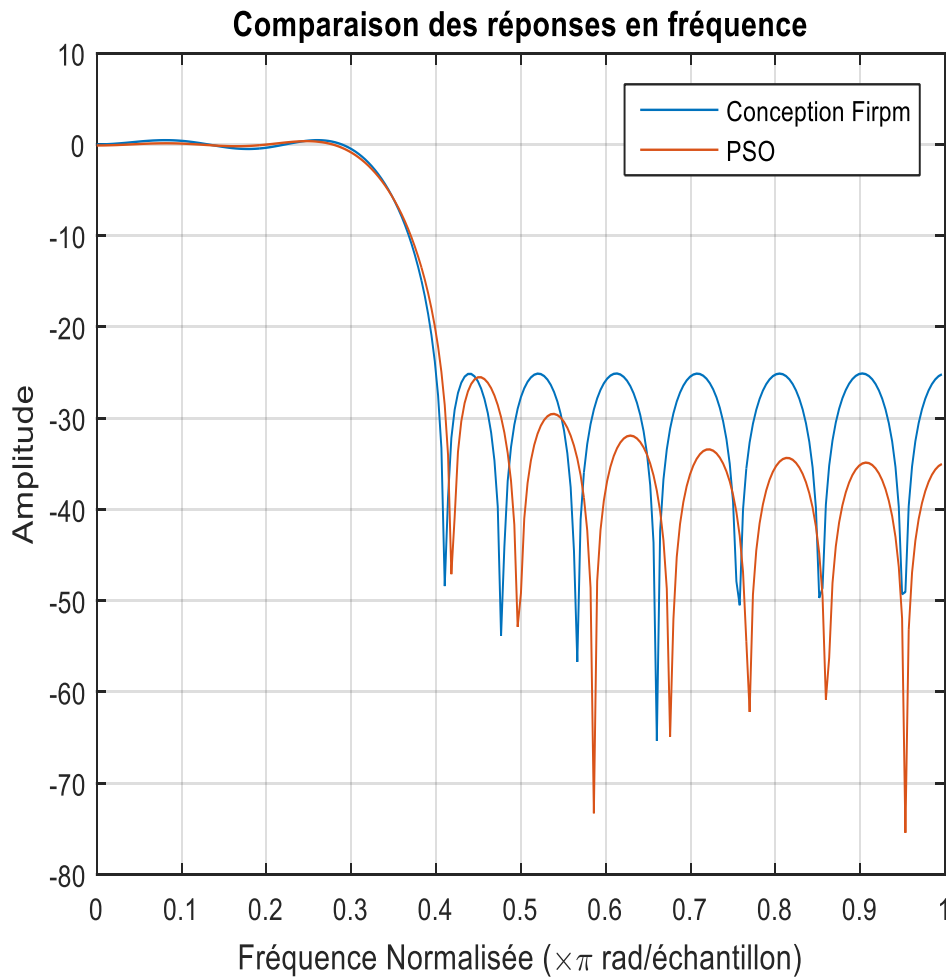


Figure (3.20) : PSOMSE2

3.5.2.GWO par MSE :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,3$ et une fréquence de coupure $\omega_s = 0,4$, les deux graphiques montrent la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et GWO par les deux méthodes de MSE. Les performances des deux méthodes diffèrent légèrement, en particulier dans les lobes de la bande d'arrêt, où l'algorithme GWO MSE2 semble offrir une meilleure atténuation dans certaines régions.

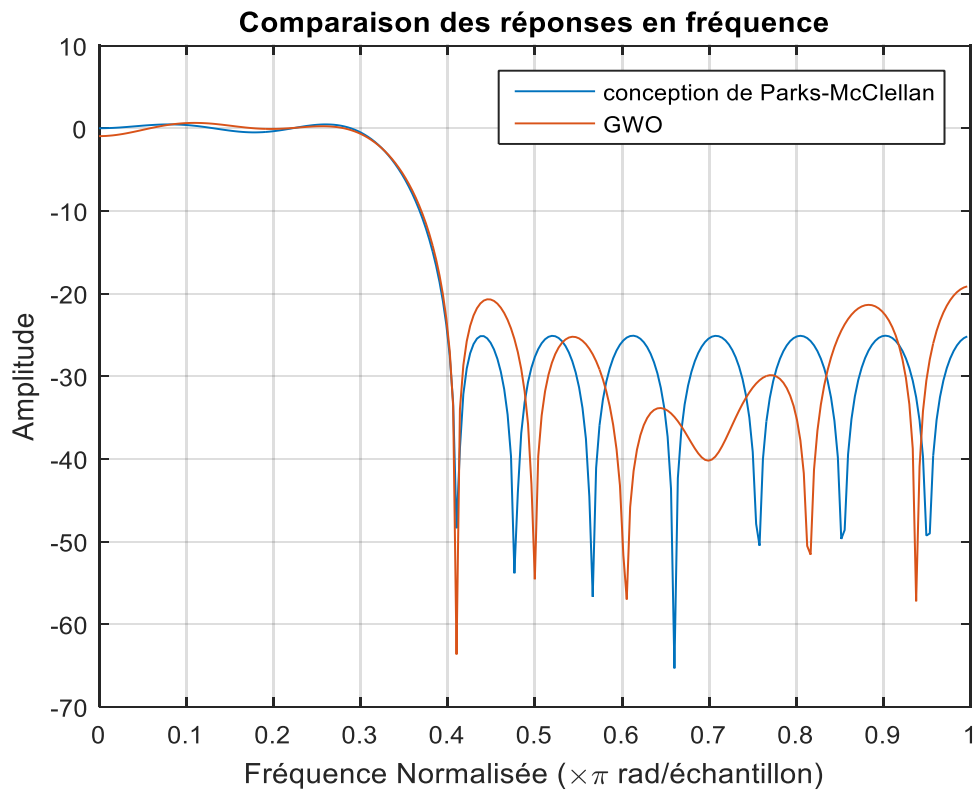


Figure (3.21) : GWOMSE1

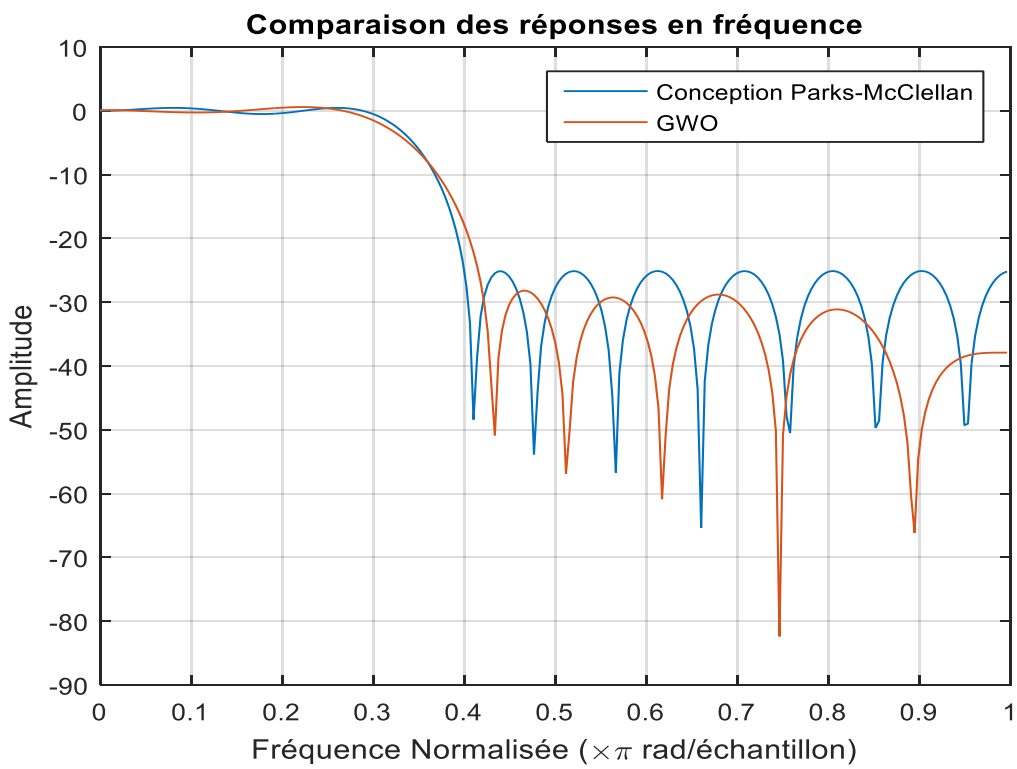


Figure (3.22) : GWOMSE2

3.5.2. ICSA par MSE :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,4$ et une fréquence de coupure $\omega_s = 0,5$, les deux graphiques montrent la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et ICSA par les deux méthodes de MSE. Les performances des deux méthodes diffèrent légèrement, en particulier dans les lobes de la bande d'arrêt.

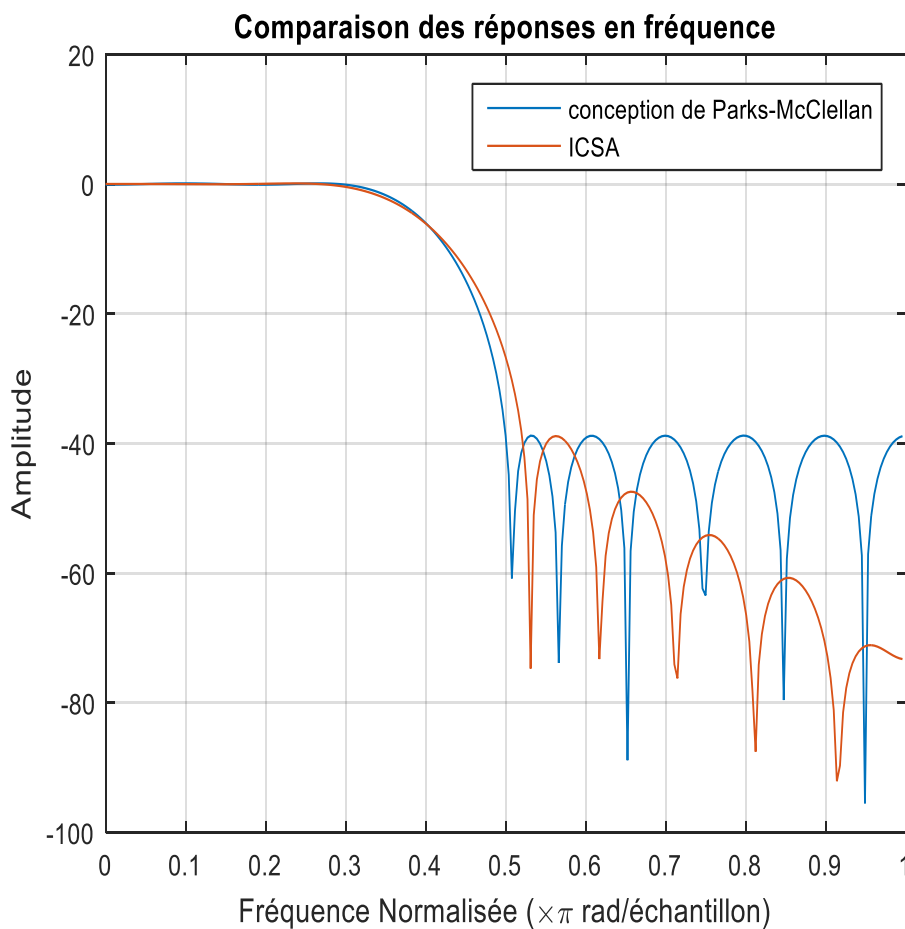


Figure (3.23) : ICSAMSE1

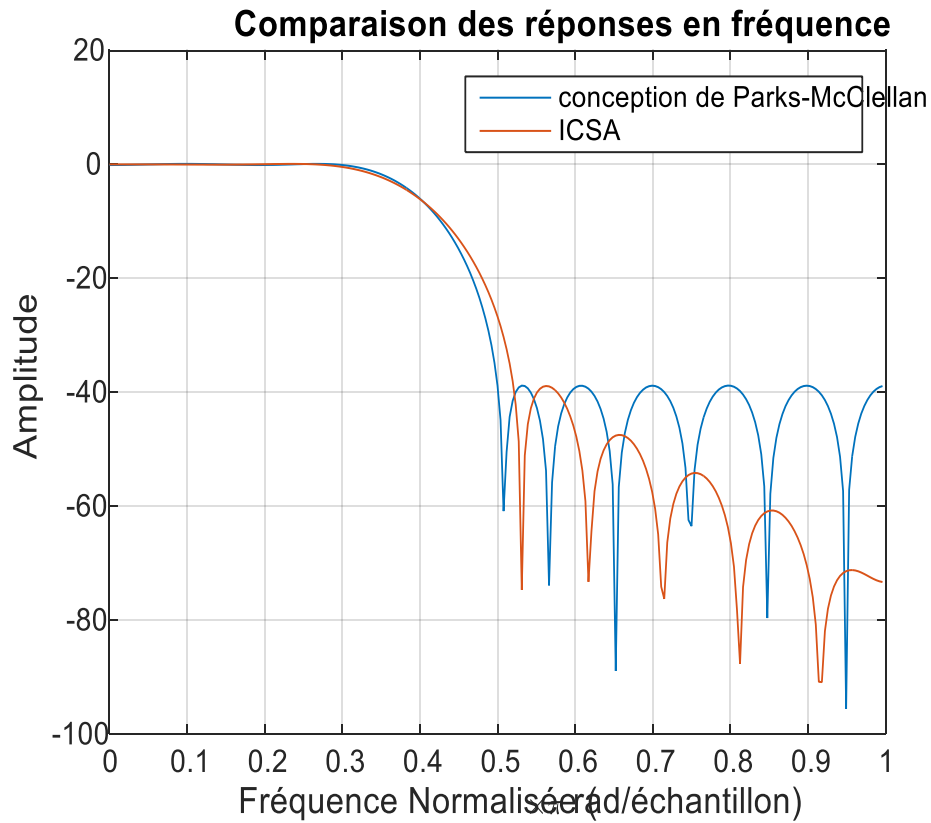


Figure (3.24) : ICSAMSE2

3.5.4.PSO par MO :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,3$ et une fréquence de coupure $\omega_s = 0,4$, le graphique montre la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et PSO par la méthode de MO.

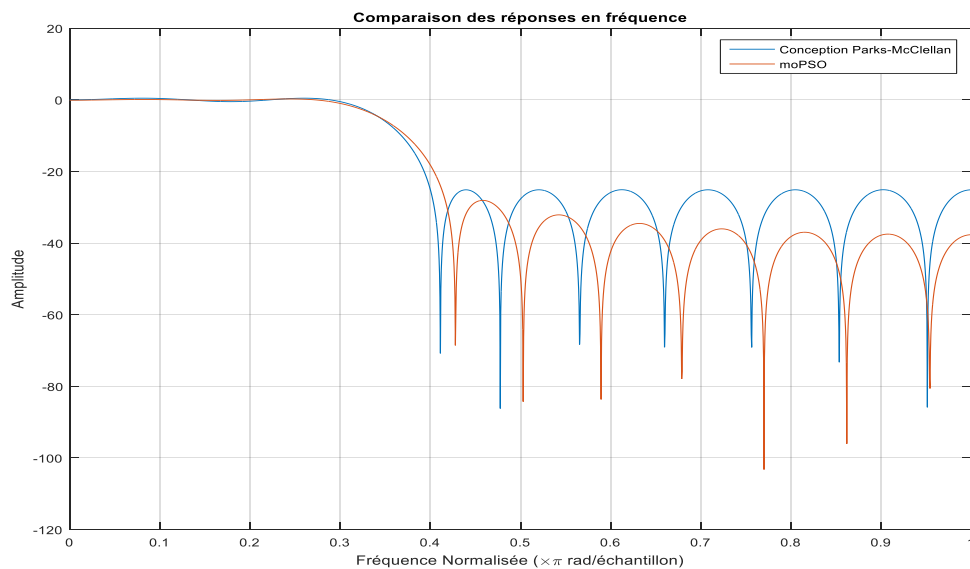


Figure (3.25) : MO-PSO

3.5.5. GWO par MO :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,3$ et une fréquence de coupure $\omega_s = 0,4$, le graphique montre la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et GWO par la méthode de MO.

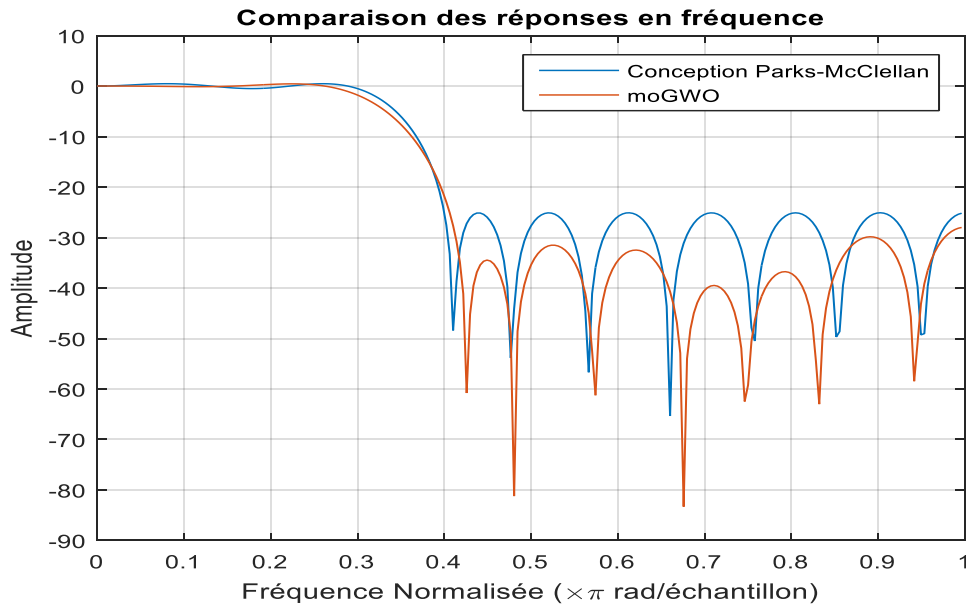


Figure (3.26) : MO-GWO

3.5.6. ICSA par MO :

Pour un filtre d'ordre 20 avec une fréquence de passage $\omega_p=0,3$ et une fréquence de coupure $\omega_s = 0,4$, le graphique montre la comparaison entre la conception de filtres utilisant la méthode de Parks-McClellan et ICSA par la méthode de MO.

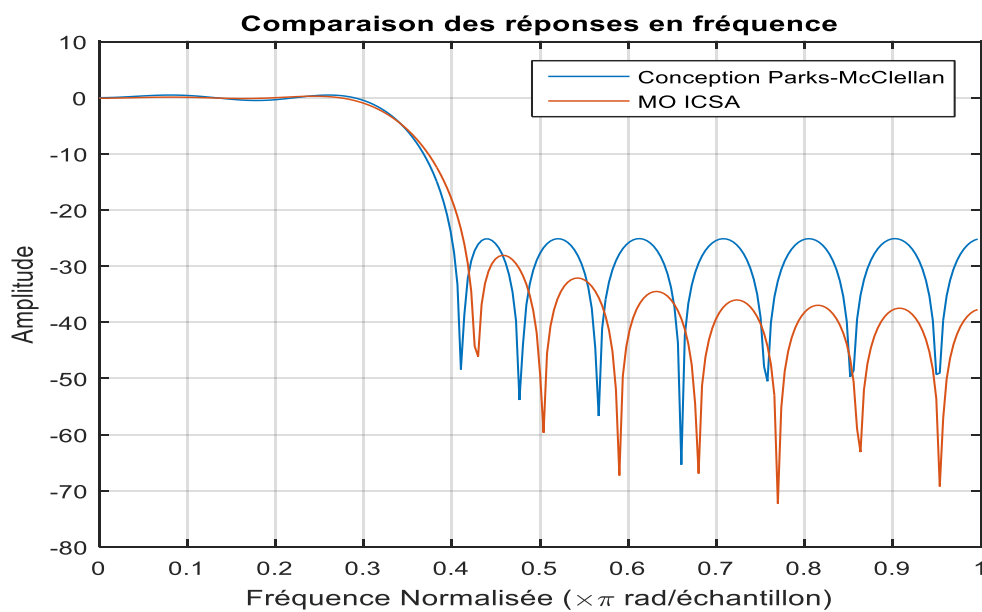


Figure (3.27) : MO-ICSA

3.6. Analyse Comparative

- **Performance :**

- Les méthodes d'optimisation comme PSO, GWO et ICSA peuvent potentiellement offrir des solutions plus adaptées à des exigences spécifiques grâce à leur capacité à explorer l'espace des solutions de manière plus exhaustive que la méthode de Parks-McClellan.
- Les optimisations multi-objectives fournissent un avantage supplémentaire en équilibrant plusieurs critères de performance, ce qui peut ne pas être directement géré par l'algorithme de Parks-McClellan.

- **Complexité :**

- Les algorithmes évolutionnaires et bio-inspirés (PSO, GWO, ICSA) peuvent être plus complexes à mettre en œuvre et à paramétrer correctement par rapport à la méthode de Parks-McClellan.
- La méthode de Parks-McClellan est bien documentée et standardisée, ce qui peut la rendre plus accessible pour des implémentations rapides et des résultats reproductibles.

- **Adaptabilité**

- Les techniques d'optimisation modernes sont plus adaptables à des problèmes complexes et à des contraintes spécifiques, ce qui les rend idéales pour des applications nécessitant une personnalisation poussée.
- La méthode de Parks-McClellan est plus rigide mais très efficace pour des conceptions de filtres standard.

3.7. Conclusion

Chacune des méthodes d'optimisation et l'algorithme de Parks-McClellan ont leurs propres avantages et inconvénients. Le choix de la méthode dépend fortement des exigences spécifiques de l'application, de la complexité du filtre souhaité et des critères de performance. Les méthodes d'optimisation modernes offrent une flexibilité et une adaptabilité accrues, tandis que l'algorithme de Parks-McClellan reste une référence pour la conception de filtres FIR standard et optimaux.

Conclusion générale

Conclusion Générale

Ce mémoire a exploré l'efficacité et l'efficience de plusieurs algorithmes métaheuristiques, à savoir l'Optimisation par Essaims Particulaires (PSO), l'Optimisation par Loups Gris (GWO) et l'Algorithme de Recherche de Coucou Amélioré (ICSA), dans le cadre de la conception de filtres numériques à réponse impulsionnelle finie (FIR). Ces filtres, essentiels pour une multitude d'applications de traitement du signal, nécessitent une optimisation précise pour atteindre des performances optimales tout en respectant des contraintes spécifiques.

Nous avons commencé par une revue approfondie des concepts théoriques relatifs aux filtres numériques et aux algorithmes métaheuristiques. Cette revue a établi les bases nécessaires pour comprendre les avantages potentiels de l'application de ces algorithmes à la conception de filtres FIR. Ensuite, nous avons développé une méthodologie pour implémenter PSO, GWO et ICSA dans ce contexte, et avons conduit des simulations pour évaluer leurs performances.

Les résultats des simulations montrent que les algorithmes métaheuristiques offrent une alternative robuste et efficace aux méthodes classiques, telles que la méthode de Parks-McClellan. En particulier, les principaux avantages observés sont les suivants :

- ✓ **Rapidité de Convergence** : Les algorithmes PSO et GWO se sont distingués par leur rapidité de convergence, permettant une conception de filtres efficace en termes de temps de calcul.
- ✓ **Équilibre entre Exploration et Exploitation** : L'algorithme GWO a démontré une capacité supérieure à équilibrer l'exploration de l'espace de solution et l'exploitation des solutions prometteuses, ce qui se traduit par une meilleure capacité à éviter les minima locaux.
- ✓ **Capacités de Recherche Globale** : L'ICSA a montré une excellente aptitude à effectuer une recherche globale, garantissant que les solutions obtenues soient proches de l'optimal global même dans des espaces de solution complexes.

En comparaison avec la méthode de Parks-McClellan, les algorithmes métaheuristiques se sont avérés plus flexibles et capables de s'adapter à des exigences spécifiques de conception de filtres. Cependant, il est important de noter que chaque méthode a ses propres avantages et inconvénients, et le choix de l'algorithme peut dépendre des contraintes spécifiques et des critères de performance souhaités.

Ce mémoire démontre la pertinence et l'efficacité des algorithmes métaheuristiques pour la conception de filtres numériques à réponse impulsionnelle finie. Les résultats obtenus confirment que ces algorithmes peuvent surmonter les limitations des méthodes classiques et offrir des performances optimales pour diverses applications de traitement du signal. Les perspectives futures incluent l'amélioration continue de ces algorithmes et l'exploration de leur applicabilité à d'autres problèmes complexes d'optimisation en ingénierie.

Bibliographie

- [1] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in Engineering Software* 69: 46-61, 2014.
- [2] Seyyedabbasi, Farzad, Amir Kiani. "I-gwo et ex-gwo: algorithmes améliorés de l'optimiseur du loup gris pour résoudre des problèmes d'optimisation globale." *Ingénierie avec les ordinateurs* 37: 509-532.
- [3] Oyewola, Stephen David, Fadele Alaba Ayotunde, Chiroma Haruna, Dada Emmanuel, et Joseph. "Application de l'optimiseur du loup gris: tendances récentes, enjeux et horizons possibles." *Journal des sciences de l'Université Gazi* 35: 485-504.
- [4] Reddad, Hakima, Maria Zemzami, Norelislam El Hami, et Nabil Hmina. *Optimisation métaheuristique et en application mécatronique*. 2022.
- [5] Bozorg-Haddad, Omid. *Optimisation avancée par des algorithmes inspirés par la nature*, volume 720. Springer, 2018.
- [6] Gomes, Herbert Martins. "Optimisation de treillis avec des contraintes dynamiques en utilisant un algorithme de banc de particules." *Expert Systems with Applications* 38(1): 957-968, 2011.
- [7] Hannan, MA, Mahmuda Akhtar, RA Begum, H Basri, A Hussain, et Edgar Scavino. "Modèle de problème de routage de véhicules capacités pour la collecte planifiée des déchets solides et optimisation des itinéraires à l'aide de l'algorithme PSO." *Gestion des déchets* 71: 31-41, 2018.
- [8] Shaw, Binod, et Jyoti Ranjan Nayak. "Optimisation du régulateur PD-FPID en cascade optimisé par l'optimiseur du lion fourmi dans un environnement déréglementé." *ACE* 1(1): 1, 2017.
- [9] Gad, Ahmed G. "Algorithme d'optimisation par essaim de particules et ses applications: une revue systématique."
- [10] Zhou, Zhou, Fangmin Li, Jemal H Abawajy, et Chaochao Gao. "Algorithme PSO amélioré intégré à l'apprentissage basé sur l'opposition et à la perception provisoire dans les centres de données en réseau." *IEEE Access* 8: 55872-55880, 2020.
- [11] Pereira, Gonçalo. "Particle swarm optimization." *INESCID Inst. Super. Techno Porto Salvo Port*, 2011.
- [12] Kennedy, James, et Russell Eberhart. "Optimisation par essaim de particules." Dans les actes de la conférence internationale sur les réseaux neuronaux ICNN'95, volume 4, pages 1942-1948. IEEE, 1995.

- [13] Yang, Xin-She, et Suash Deb. "Recherche du coucou: avancées récentes et applications." *Neural Computing and Applications* 24: 169-174, 2014.
- [14] Yang, Xin-She, et Suash Deb. "Recherche du coucou via les vols de Lévy." Dans le congrès mondial de 2009 sur l'informatique inspirée par la nature et la biologie (NaBIC), pages 210-214. IEEE, 2009.
- [15] Proakis, John G. *Digital Signal Processing Principles, Algorithms and Applications*. 4th edition.
- [16] Tan, Li, et Jean Jiang. *Digital Signal Processing: Fundamentals and Applications*. Academic Press, 2018.
- [17] Porat, Boaz. *A Course on Digital Signal Processing*.
- [18] Ingle, Vinay K., et John G. Proakis. *Digital Signal Processing Using MATLAB*.
- [19] Schilling, Robert J., et Sandra L. Harris. *Fundamentals of Digital Signal Processing Using MATLAB® Second Edition*.