

*REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE*  
*Ministère de L'Enseignement Supérieur et de la Recherche Scientifique*

Université des frères Mentouri Constantine 1

Faculté des Sciences de la Technologie

Département d'Electronique

No d'ordre:194/DS/2018

Série :12/Elec/2018

Thèse  
de Doctorat en Sciences  
Option : Micro-ondes

Présentée Par:  
Erredir Chahrazad

Thème  
*Contribution à la modélisation et à l'optimisation de  
structures et dispositifs microondes en utilisant divers  
types de réseaux de neurones*

---

07 Novembre 2018

Devant le jury

Président:

M. Chaabi Abdelhafid      Professeur      Université des Frères Mentouri Constantine 1

Rapporteur:

M. Riabi Mohamed Lahdi,      Professeur      Université des Frères Mentouri, Constantine 1

Examineurs:

M. Belarbi Khaled,      Professeur      Ecole Nationale polytechnique de Constantine

Mr. Benatia Djamel,      Professeur      Université Batna 2

*A mes parents*

*à mon mari et mon fils*

*à mon frère, mes sœurs et leurs enfants*

*à mes deux familles*

## *Remerciements*

---

J'adresse mes remerciements en premier lieu, à **Dieu** tout puissant pour la volonté, la santé, le courage et surtout la patience qu'il m'a donné pour mener à terme ce travail.

Les travaux présentés dans cette thèse ont été effectués au sien du Laboratoire d'Electromagnétisme et Télécommunications (LET) de l'Université des Frères Mentouri Constantine 1.

Je tiens à exprimer mes sincères remerciements à mon Directeur de thèse, Monsieur Riabi Mohamed Lahdi, Professeur à l'Université des Frères Mentouri Constantine 1, qui a dirigé ce travail avec beaucoup d'intérêt et de rigueur.

Je remercie également, Monsieur Chaabi Abdelhafid, Professeur à l'Université des Frères Mentouri Constantine 1, pour l'honneur qu'il m'a fait en acceptant de présider le jury de cette thèse.

Je suis également très honoré que Monsieur Belarbi Khaled, Professeur de l'Ecole Nationale polytechnique de Constantine et Monsieur Benatia Djamel, Professeur à l'université Batna 2, Soient membres de jury.

Merci à tous ceux qui m'ont soutenu durant mes études et à ma famille pour son support continu.

# ***SOMMAIRE***

---

# *Table des Matières*

---

Table des matières .....	I
Liste des figures .....	IV
Liste des tableaux .....	VI
Liste des abréviations .....	VII
Introduction Générale .....	1
<i>Chapitre 1: Les Réseaux de Neurones Artificiels</i>	
I. Introduction .....	3
II. Historique .....	3
III. L'intérêt de L'utilisation des Réseaux de Neurones Artificiels .....	4
IV. Du neurone Biologique au Neurone Artificiel .....	5
IV.1 Neurone Biologique .....	5
IV.2 Neurone Artificiel .....	6
V. Architecture des Réseaux de Neurones Artificiels .....	8
V.1. Les Réseaux de Neurones Non Bouclés .....	9
V.1.1 Architecture monocouche .....	9
V.1.2 Architectures multicouches .....	10
V.1.3 Réseaux à base d'une fonction radiale .....	10
V.2 Les Réseaux de Neurones Bouclés .....	11
V.2.1 Réseaux entièrement connectés .....	11
V.2.2 Réseaux extérieurement connectés .....	11
VI. Processus de Formation et Propriétés D'apprentissage .....	12
VI.1 Apprentissage Supervisé .....	13
VI.2 Apprentissage Non Supervisé .....	13
VI.3 Apprentissage par Renforcement .....	14
VII. Elaboration de la Structure du Réseau .....	14
VII.1 Choix et Préparation des Echantillons .....	14
VII.1.1 Détermination des entrées/sorties du réseau de neurones .....	15
VII.1.2 Nombre de couches cachées .....	15
VII.1.3 Nombre de neurones cachés .....	15
VII.2 La Réalisation de l'Apprentissage .....	16
VII.3 Validation et Tests .....	16
VIII. Conclusion .....	17
Références Chapitre 1 .....	18

## *Table des Matières*

---

### *Chapitre 2: Optimisation par les Techniques d'Intelligence de l'Essaim*

I.	Introduction .....	20
II.	Optimisation par Essaim de Particule .....	21
II.1	Les Eléments d'OEP .....	22
II.2	Principe de la Méthode .....	22
II.3	Configuration de la Méthode .....	24
II.3.1	Nombre de particules .....	24
II.3.2	Taille et topologie de voisinage .....	24
II.3.3	Coefficients de confiance .....	25
II.3.4	Coefficient d'inertie .....	25
II.4	Algorithme d'OEP .....	26
III.	Optimisation par Araignées Sociales.....	27
III.1	Modélisation des Vibrations à Travers le Réseau Commun .....	28
III.2	Les Operations Coopératives .....	30
III.2.1	Opération coopérative femelle .....	30
III.2.2	Opération coopérative mâle .....	30
III.2.3	Opération d'accouplement .....	31
III.3	Procédure d'Algorithme .....	32
IV.	Optimisation par Singe-Araignée .....	33
IV.1	Phase Leader Local .....	34
IV.2	Phase Leader Global .....	35
IV.3	Phase D'apprentissage du Leader Global .....	36
IV.4	Phase D'apprentissage du Leader Local .....	37
IV.5	Phase de Décision du Leader Local .....	37
IV.6	Phase de Décision du Leader Global .....	38
IV.7	Paramètres de Contrôle .....	38
IV.8	Procédure d'Algorithme .....	39
V.	Optimisation a Base d'Enseignement- Apprenants .....	40
V.1	Phase Enseignant .....	40
V.2	Phase Apprenants .....	41
V.3	Procédure d'Algorithme .....	42
V.4	Algorithme OBEA Modifié .....	44
V.4.1	Nombre d'enseignants .....	44



## *Liste des Figures*

---



## *Liste des figures*

Figure I.1	Le neurone biologique .....	5
Figure I.2	Le neurone artificiel .....	7
Figure I.3	Représentation graphique, (a) de la fonction sigmoïde, (b) de la fonction tangente hyperbolique et (c) la fonction Heaviside .....	8
Figure I.4	Exemple d'un réseau monocouche .....	9
Figure I.5	Exemple d'un réseau multicouche .....	10
Figure I.6	Réseaux entièrement connectés .....	11
Figure I.7	Exemple d'un réseau récurrent .....	12
Figure II.1	Des essaims d'oiseaux et de poissons .....	22
Figure II.2	Déplacement d'une particule .....	23
Figure II.3	Voisinage étoile.....	24
Figure II.4	Voisinage anneau .....	24
Figure II.5	Voisinage rayon .....	25
Figure II.6	Différents environnements formés par les araignées sociales .....	27
Figure II.7	Configuration de chaque relation spéciale: a) $Vibc_i$ , b) $Vibb_i$ , c) $Vibf_i$ .....	29
Figure II.8	Quelques images de singes araignées .....	33
Figure II.9	Organigramme de l'algorithme OBEA.....	43
Figure II.10	Transformateur à deux paliers .....	49
Figure II.11	Filtres à iris inductif à trois cavités .....	50
Figure II.12	La convergence de la fonction objective. (a) Filtre, (b) Transformateur	52
Figure II.13	Le ROS en fonction de la fréquence pou le transformateur.....	53
Figure II.14	Le module du coefficient de réflexion S11 en fonction de la fréquence..	53
Figure III.1	Architecture de réseau de neurones .....	55
Figure III.2	pseudo-elliptique filtre, a) Vue de dessus, b) Vue de côté en coupe transversale.....	63
Figure III.3	Les conceptions du filtre plan-E à large bande .....	64
Figure III.4	structures de poteaux à ailettes .....	64
Figure III.5	Filtre plan-H considérant des coins arrondis .....	65
Figure III.6	Convergence de OEP pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3.....	68
Figure III.7	Convergence de OAS pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3.....	69

## *Liste des figures*

---

Figure III.8	Convergence de OSA pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3.....	70
Figure III.9	Convergence de OBEA pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3.....	71
Figure III.10	comparaison entre les algorithmes OEP, OAS, OSA et OBEA,(a) structure 1, (b) structure 2, (c) structure 3.....	73
Figure III.11	la coefficient de réflexion approximatif,(a) structure 1, (b) structure 2, (c) structure 3.....	74
Figure III.12	comparaison entre OEP et HEP,(a) structure 1, (b) structure 2, (c) structure 3.....	75
Figure III.13	comparaison entre OAS et HAS,(a) structure 1, (b) structure 2, (c) structure 3.....	76
Figure III.14	comparaison entre OSA et HSA,(a) structure 1, (b) structure 2, (c) structure 3.....	77
Figure III.15	comparaison entre OBEA et HOBEA,(a) structure 1, (b) structure 2, (c) structure 3.....	78
Figure III.16	coefficient de réflexion obtenu par OEP et HOEP, (a) structure 1, (b) structure 2, (c) structure 3 .....	79
Figure III.17	coefficient de réflexion obtenu par OAS et HOAS, (a) structure 1, (b) structure 2, (c) structure 3 .....	80
Figure III.18	coefficient de réflexion obtenu par OSA et HOSA, (a) structure 1, (b) structure 2, (c) structure 3 .....	81
Figure III.19	coefficient de réflexion obtenu par OBEA et HOBEA, (a) structure 1, (b) structure 2, (c) structure 3 .....	82

## *Liste des Tableaux*

---

## *Liste des Tableaux*

---

Tableau I.1	Analogie entre le neurone biologique et le neurone artificiel .....	6
Tableau II.1	Limites des paramètres à optimiser.....	49
Tableau II.2	Limites des paramètres à optimiser .....	50
Tableau II.3	La fonction objective pour une population = 50 .....	51
Tableau II.4	La fonction objective pour une Maxit = 50 .....	51
Tableau II.5	Les paramètres géométriques optimisés .....	54
Tableau III.1	Dimensions du filtre 1 (unités: millimètres).....	63
Tableau III.2	Dimensions du filtre 2 (unités: millimètres).....	64
Tableau III.3	Dimensions du filtre 3 (unités: millimètres).....	65
Tableau III.4	Paramètres du réseau de neurones .....	66
Tableau III.5	L'erreur quadratique moyenne pour maxit=300.....	67
Tableau III.6	Comparaison de performance des algorithmes OEP, OAS, OSA et OBEA.....	72
Tableau III.7	comparaison entre les algorithmes de base et leurs hybrides algorithmes .....	72

## *Liste des abréviations*

---

## *Liste des abréviations*

---

<i>RN</i>	Les reseaux de neurones
<i>RNA</i>	Les reseaux de neurones artificiels
<i>PMC</i>	Le perceptron multi-couches
<i>RBF</i>	Fonction radiale de base
<i>EQM</i>	Erreur quadratique moyenne
<i>EI</i>	Essaims Intelligents
<i>OEP</i>	Optimisation par essaim de particules
<i>OAS</i>	Optimisation par araignées sociales
<i>OSA</i>	Optimisation par singe araignée
<i>PLL</i>	Phase Leader Local
<i>PLG</i>	Phase Leader Global
<i>PALG</i>	Phase d'Apprentissage du Leader Global
<i>PALL</i>	Phase d'Apprentissage du Leader Local
<i>PDLL</i>	Phase Décision du Leader Local
<i>PDLG</i>	Phase Décision du Leader Global
<i>OBEA</i>	Optimisation a base d'enseignement-apprentissage
<i>AG</i>	Les algorithmes génétiques
<i>GD</i>	gradient décent,
<i>RP</i>	Rétro-propagation

# *Introduction Générale*

---

Avec l'utilisation des fréquences de travail de plus en plus élevées, la modélisation de structures passives microondes de plus en plus miniatures entrant dans la conception et l'optimisation de nombreux dispositifs utilisés dans les systèmes de télécommunication tels que les transitions par paliers et les filtres, a suscité ces dernières années l'intérêt sans cesse croissant des chercheurs pour le développement de nouvelles méthodes numériques très rapides et rigoureuses.

Avec la complexité croissante du matériel de Télécommunication (communication sans fil et par satellite) de plus en plus miniature, il est donc nécessaire de rendre les techniques numériques, comme la méthode de raccordement modal communément utilisée pour la modélisation des structures en guides d'ondes métalliques plus performantes.

Récemment, les réseaux de neurones artificiels (RNA) fournissent des modèles rapides et précis pour la modélisation, la simulation et l'optimisation de ce type de structures. Il a été reconnu comme un outil puissant pour prédire le comportement d'un dispositif pour lequel aucun modèle mathématique n'est pas encore disponible, ou simplement que le dispositif n'a pas encore été correctement analysé. Les RNA peut être formé pour saisir la relation d'entrée-sortie arbitraire avec n'importe quel degré de précision.

Au départ, l'apprentissage des réseaux de neurones est une tâche très importante, elle dépend de l'adaptation des paramètres de réseau, c'est-à-dire de la sélection correcte des valeurs de poids; où la formation de RN est un problème numérique d'optimisation consistant à trouver les poids optimaux des réseaux pour minimiser l'erreur quadratique moyenne (mean square error , MSE) entre la sortie du réseau et la sortie désirée.

Des algorithmes d'apprentissage spécialisés sont utilisés pour l'adaptation de ces valeurs de poids parmi lesquels le plus utilisé est l'algorithme de rétro-propagation du gradient. Dernièrement, de nombreux algorithmes basés sur la population ont été proposés pour l'apprentissage d'un réseau de neurones tels que l'optimisation des essaims de particules (OEP) les algorithmes génétiques (AG) et d'autres algorithmes d'optimisation.

Deux objectifs principaux feront l'objet de cette thèse: En premier lieu, nous parlerons de l'apprentissage de RN où de nouveaux hybrides algorithmes basés sur la



combinaison entre l'algorithme conventionnel gradient descendant et les algorithmes stochastiques basés sur l'essaim intelligent sont proposés.

Le deuxième objectif de cette étude est la modélisation des dispositifs micro-ondes par l'utilisation d'un type de réseaux de neurones qui a été adapté à notre problématique.

Ce manuscrit se divise en trois chapitres:

Le premier chapitre est consacré les bases théoriques des méthodes de réseaux de neurones artificiels avec leurs différentes architectures. Puis, nous décrivons les étapes de conception d'un modèle neuronal (choix des entrées et sorties, bases de données, structure du réseau etc....).

Dans le deuxième chapitre, nous rappelons les principes fondamentaux des méthodes d'optimisation à base des essais intelligents exploitées dans ce travail, tels que l'optimisation par essaim de particules OEP (Particle Swarm Optimization, PSO), l'optimisation par araignées sociales OAS (Social Spiders Optimization, SSO), l'optimisation par singe araignée OSA (Spider Monkey Optimization, SMO), et l'optimisation à base d'enseignement-apprentissage OBEA (Teaching-Learning-Based Optimization, TLBO). Deux applications (filtres à iris inductif à trois cavités et Transformateur à deux paliers) sont ensuite proposées.

Le dernier chapitre, composé de trois volets, concerne l'applications de RN pour la modélisation des structures micro-ondes, Le premier volet inclue l'apprentissage des RN utilisant les algorithmes d'optimisation mentionnés en deuxième chapitre, Puis, nous avons proposé de nouveaux algorithmes hybrides basés sur la combinaison entre l'algorithme d'optimisation conventionnelle (gradient décent, GD) et l'un des algorithmes d'optimisation évolutionnaire. Dans le dernier volet le réseau de neurones formés a été appliqué à la modélisation des structures de guides d'ondes (pseudo-elliptique filtre, filtre E-plan large bande et Filtres plan H considérant des coins arrondis).

*Chapitre I*  
*Les Réseaux de Neurons Artificiels*

---

## I. Introduction

Les réseaux de neurones artificiels (RNA) constituent l'une des approches d'intelligence artificielle dont le développement se fait à travers les méthodes par lesquelles l'homme essaye toujours d'imiter la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propres. Nous présentons dans ce premier chapitre un état de l'art de ces réseaux de neurones.

## II. Historique

**En 1890:** W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

**En 1943:** Mac Culloch et Pitts ont proposé le premier modèle de neurone biologique [1], communément appelé neurone formel. Ils montrèrent théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

**En 1949:** Donald. Hebb présente dans son ouvrage intitulé "The Organisation of Behavior" [2] une règle d'apprentissage d'un réseau de neurones. Il a proposé de nombreuses règles de mise à jour des connexions entre neurones dont la plus connue à cette époque est la règle de Hebb [3].

**En 1958:** le psychologue Frank Rosenblatt a développé le modèle du perceptron [4]. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception (sert à recueillir les entrées) et une couche de décision. C'est le premier modèle qui peut définir le processus d'apprentissage.

**En 1960:** B. Widrow et M. E. Hoff, développent le modèle Adaline (Adaptative Linear Element) [5]. Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Ce modèle sera par la suite le modèle de base des réseaux de neurones multicouches. Les réseaux du type Adaline restent utilisés de nos jours pour certaines applications particulières.

**En 1969:** M.Minsky et S. Papert [6], ont publié un livre dans lequel ils ont démontré les limites du perceptron Proposé par Rosenblatt ; en particulier, l'impossibilité de traiter les problèmes non linéaires par ce modèle. Cette recherche va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

**En 1982:** Hopfield développe un modèle qui utilise des réseaux totalement connectés basés sur la règle de Hebb pour définir les notions d'attracteurs et de mémoire associative. Ce modèle est aujourd'hui encore très utilisé pour des problèmes d'optimisation. La même année, T. Kohonen propose le modèle SOM (Self Organized Map) [7] pour le traitement des cartes.

**En 1983 :** La Machine de Boltzmann est le premier modèle connu capable à traiter de manière satisfaisante les limitations spécifiées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

**En 1985 et 1986:** l'algorithme de rétro-propagation du gradient est mis au point de manière indépendante par les chercheurs Parker [8], Rumelhart [9] et Lecun [10]. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Dès cette Découverte, une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires : le perceptron multicouche ne possède pas les défauts mis en évidence par Minsky.

De nos jours, l'utilisation des réseaux de neurones dans divers domaines continue de croître. Les applications sont multiples et variées.

### **III. L'Intérêt de l'Utilisation des Réseaux de Neurones Artificiels**

Les réseaux de neurones artificiels, possèdent la propriété d'approximation (toute fonction bornée régulière peut être approchée avec une précision arbitraire dans une zone limitée de ses variables). En plus, les réseaux de neurones nécessitent moins de paramètres ajustables (les poids des connexions) que d'autres outils mathématiques couramment utilisés.

Dans la pratique, on n'utilise pas les réseaux de neurones pour réaliser des approximations de fonctions connues. Le plus souvent, le problème qui se pose est de trouver une relation

entre un ensemble de sortie dans un processus donné, et l'ensemble des entrées correspondant à travers les mesures effectuées. On suppose que cette relation existe quoique :

- Les mesures sont en nombre fini,
- Elles sont certainement entachées de bruit,
- Toutes les variables qui déterminent le résultat du processus ne sont pas forcément mesurées.

## IV. Du Neurone Biologique au Neurone Artificiel

### IV.1. Neurone Biologique

Un neurone est la cellule constituante de l'unité fonctionnelle du cerveau. Il est responsable de l'émission et de la propagation du message nerveux [11]. Les parties du neurone sont : Le corps cellulaire, l'axone, les dendrites et les synapses Figure I.1.

- **les dendrites** : branches tubulaires courtes formant une espèce d'arborescence autour du corps cellulaire; ce sont les entrées principales du neurone, qui captent l'information venant d'autres neurones;
- **le corps cellulaire (soma, noyau)**: contient le noyau de la cellule nerveuse; c'est en cet endroit que prend naissance l'influx nerveux, qui représente l'état d'activité du neurone
- **les synapses** : qui transmettent l'information, en la pondérant par un poids synaptique, à un autre neurone; elle est essentielle dans le fonctionnement du système nerveux.
- **l'axone** : longue fibre nerveuse qui se ramifie à son extrémité; c'est la sortie du neurone et le support de l'information vers les autres neurones; (nerveuses, musculaires, etc.).

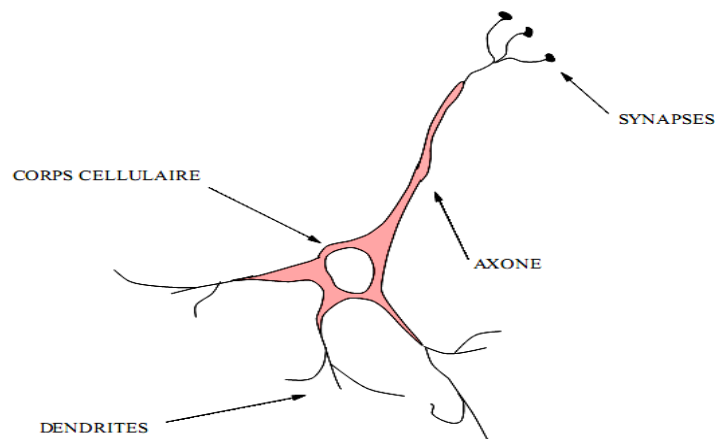


Figure I.1 : Le neurone biologique.

L'information circule entre les neurones sous forme de signaux appelés influx nerveux. L'influx nerveux se transmet sous forme électrique. Le corps cellulaire reçoit de l'influx en provenance des dendrites et réalise leur intégration. Le neurone traite cette information et renvoie le résultat au travers de son axone. Cet influx émis par le neurone peut ensuite être transmis, au travers d'une synapse, à un autre neurone, ou encore à un muscle ou à une glande.

Le poids synaptique définit alors la capacité d'une synapse à influencer le neurone récepteur. L'apprentissage est alors basé sur le mécanisme d'auto-organisation des poids synaptiques et des liaisons neuronales. L'ensemble des neurones constitue le système nerveux. Le cerveau est l'organe central de coordination et de supervision des activités du corps humain

#### IV.2. Neurone Artificiel

Un neurone artificiel est une représentation mathématique et informatique du neurone biologique. [1] ont implémenté un système de réseaux neuronaux artificiels, qui est analogue aux neurones biologiques fondée sur une structure complexe [Tableau I.1](#), comme le montre la [Figure I.2](#).

**Tableau I.1** Analogie entre le neurone biologique et le neurone artificiel

Neurone formel	Neurone biologique
Signal d'entrée	Dendrites
Poids de connexion	Synapses
Fonction d'activation	Le noyau (Soma)
Signal de sortie	Axone

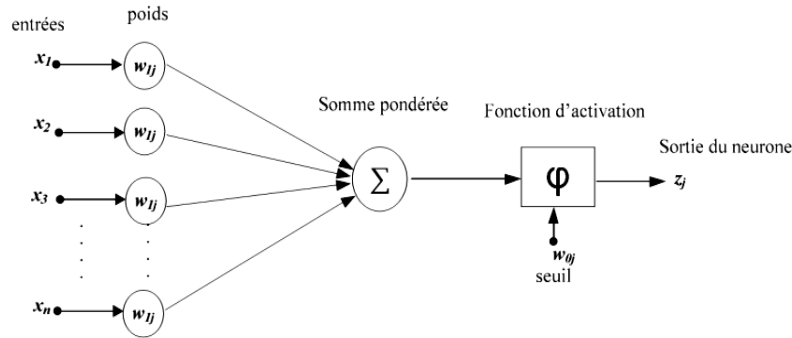


Figure I.2 Le neurone artificiel.

- **Signal d'entrée :** Un neurone formel peut recevoir des signaux  $X_i$  de plusieurs neurones. Ces signaux sont combinés en un signal d'entrée unique  $E$  :

$$E = \sum W_i \cdot X_i \quad (\text{I.1})$$

Où les  $W_i$  sont les poids affectés aux signaux extérieurs.

- **Poids de connexion :** Le poids de connexion est un nombre réel  $W_{ij}$  associé à la connexion entre le neurone  $i$  et le neurone  $j$ . Il est aussi appelé poids synaptique. Le poids de connexion permet de mesurer l'influence du neurone  $j$  sur le neurone  $i$ . Ce nombre matérialise la liaison entre l'axone du neurone  $i$  et la synapse du neurone  $j$ . La liaison est excitatrice si son poids de connexion est positif et inhibitrice s'il est négatif.
- **Fonction de transfert :** La fonction de transfert d'un neurone est la fonction qui calcule l'état de ce neurone à partir de son potentiel membranaire. Les fonctions d'activation sont généralement croissantes et bornées. A titre d'exemple, nous présentons les suivantes :

- La fonction sigmoïde (ou logistique) donnée par la formule I.2 et présentée par la Figure I.3 (a).

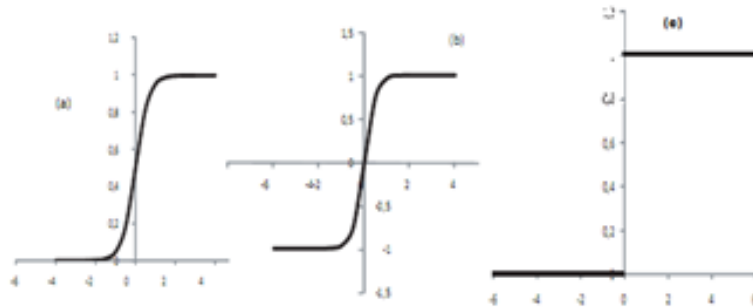
$$\varphi(x) = \frac{1}{1 + e^x} \quad (\text{I.2})$$

- La fonction tangente hyperbolique donnée par la formule I.3 et représentée graphiquement par la Figure I.3 (b).

$$\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{I.3})$$

- La fonction de Heaviside est une fonction calculée par intervalle. Elle est calculée par la formule I.4 et représentée graphiquement par la Figure I.3 (c).

$$\varphi(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (\text{I.4})$$



**Figure I.3** Représentation graphique, (a) de la fonction sigmoïde, (b) de la fonction tangente hyperbolique et (c) la fonction Heaviside.

- **Signal de sortie :** Un neurone formel est constitué essentiellement d'un intégrateur qui effectue une somme pondérée de ses entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie

$$Z = \varphi \left( \sum w_i \cdot x_i \right) = \varphi(E) \quad (\text{I.5})$$

## V. Architecture des Réseaux de Neurones Artificiels

Un réseau de neurone est constitué de plusieurs neurones interconnectés. L'architecture d'un réseau de neurones artificiel détermine comment les neurones sont disposés ou placés les uns par rapport aux autres. Ces neurones sont organisés en couches, ces couches peuvent être divisées en trois types:

- **Couche d'entrée:** Cette couche est responsable de la réception des informations (données), Ces entrées (échantillons ou modèles) sont généralement normalisées à l'intérieur des valeurs limites produites par les fonctions d'activation. Cette normalisation entraîne une meilleure précision numérique pour les opérations mathématiques effectuées par le réseau.
- **Couche cachée:** Cette couche est composée de neurones qui sont responsables d'extraire des motifs associés au processus ou système analysé. Cette couche effectue la majeure partie du traitement interne du réseau.



- **Couche de sortie:** Cette couche est responsable de la production et de la présentation des sorties du réseau final, qui sont produits par le traitement effectué par les neurones dans les couches précédentes.

Selon les connexions existantes entre les neurones de différentes couches, on peut distinguer plusieurs architectures de RNAs, que l'on peut ranger dans deux catégories :

- (1) Les réseaux de neurones statiques (ou acycliques, ou non bouclés).
- (2) Les réseaux de neurones dynamiques (ou récurrents, ou bouclés).

### V.1 Les Réseaux Non Bouclés (Feed-Forward)

Un réseau de neurones non bouclé (propagation vers l'avant de l'information) est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière" [12]. On constate que le graphe d'un réseau non bouclé est acyclique. En effet, si on se déplace dans ce type de réseau à partir d'un neurone quelconque en suivant les connexions, on ne peut pas revenir au neurone de départ. Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie. Ceux qui effectuent des calculs intermédiaires sont les neurones cachés.

#### V.1.1. Architecture monocouche (Perceptron)

Ce réseau de neurones artificiel a une seule couche d'entrée et une seule couche de sortie. La Figure I.4 illustre un réseau monocouche composé de  $n$  entrées et de  $m$  sorties. L'information circule toujours dans un seul sens (donc unidirectionnel), de la couche d'entrée vers la couche de sortie. Ces réseaux sont généralement utilisés dans les classifications de modèles et les problèmes de filtrage linéaire.

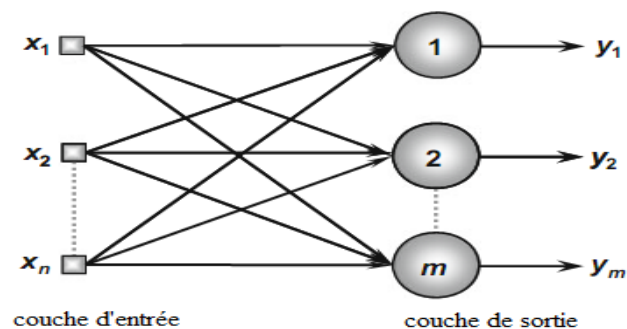


Figure I.4 Exemple d'un réseau monocouche

### V.1.2. Le Réseau multicouche (RMC)

C'est le réseau de neurones statique le plus utilisé [13]. Le RMC est une extension du perceptron monocouche qui dispose d'une ou de plusieurs couches cachées. Les neurones y sont arrangés en couches successives : la première couche qui forme le vecteur des données d'entrée est appelée couche d'entrée tandis que la dernière couche qui produit les résultats est appelée couche de sortie. Toutes les autres couches qui se trouvent au milieu sont appelées couches cachées [14], Figure I.5. Ils sont employés dans la solution de divers problèmes, tels que ceux liés à l'approximation de fonction, la classification de modèle, l'identification du système, le contrôle de processus, la robotique, et ainsi de suite.

La Figure I.5 montre un réseau multicouche composé d'une couche d'entrée avec  $n$  neurones, deux couches cachées constituées de  $n_1$  et  $n_2$  neurones respectivement, et finalement, une couche de sortie composée de  $m$  neurones représentant les valeurs de sortie respectives du problème analysé.

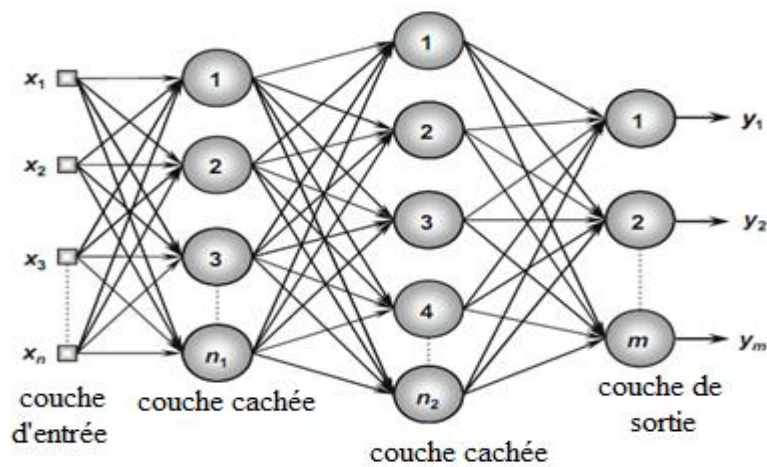


Figure I.5 Exemple d'un réseau multicouche.

### V.1.3. Réseaux à base d'une fonction radiale (RBFR)

Les réseaux à fonction radiale sont très semblables à celle des RMC mais leur particularité est qu'ils sont caractérisés par l'utilisation des fonctions Gaussiennes comme fonctions de base [15]. Les RBFR sont donc utilisés dans les mêmes genres de problèmes que les RMC à savoir, en classification et en prédiction.

## V.2. Les Réseaux Bouclés (Feed-Back)

Un réseau de neurones bouclé est schématisé par un graphe des connexions qui est cyclique. Lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de "cycle"). Ce genre de réseaux est caractérisé par le pouvoir de laisser l'information circuler récursivement d'une manière partielle ou bien total [16],[ 17]. On peut distinguer deux familles de réseaux bouclés, le réseau bouclé extérieurement et le réseau complètement bouclé.

### V.2.1 Réseaux entièrement connectés (complètement bouclé)

Ces réseaux sont caractérisés par le fait que leurs neurones sont tous connectés entre eux. Dans cette configuration, chaque neurone reçoit des informations de la part de tous les autres neurones, les traite et les envoie à sa sortie puis à tous les autres neurones Figure I.6. De plus, dans le cadre de notre application aux systèmes électronique, il s'avère que cette architecture complexe n'est pas nécessaire, compte tenu de la connaissance que nous avons des systèmes. C'est pourquoi, nous n'utiliserons pas ici la topologie complètement bouclée, cette dernière n'étant citée dans ce paragraphe qu'au titre de l'état de l'art.

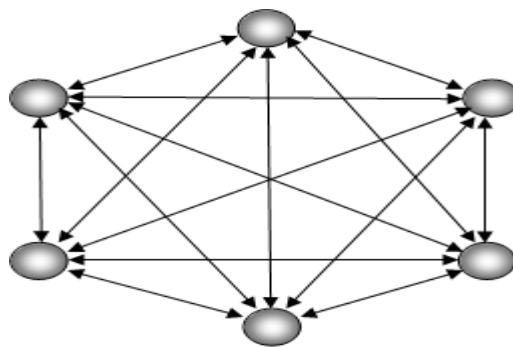


Figure. I. 6 Réseaux entièrement connectés

### V.2.1 Réseaux extérieurement connectés

L'architecture d'un réseau de neurones extérieurement connectés (le graphe des connexions est cyclique) : lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de « cycle »). La sortie d'un neurone du réseau peut donc être

fonction d'elle-même; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération.

Ainsi, à chaque connexion d'un réseau de neurones bouclé (ou à chaque arête de son graphe) est attaché, outre un poids comme pour les réseaux non bouclés, un retard, multiple entier (éventuellement nul) de l'unité de temps choisie. Une grandeur, à un instant donné, ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul. La Figure I.7 illustre un exemple de réseau Perceptron avec rétroaction, dans lequel un de ces signaux de sortie est renvoyé à la couche intermédiaire.

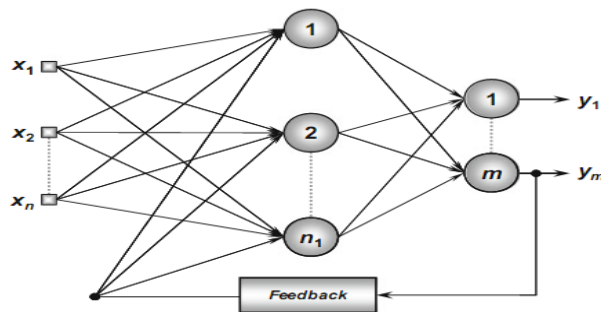


Figure I.7 Exemple d'un réseau récurrent

## VI. Processus de Formation et Propriétés d'Apprentissage

L'une des caractéristiques les plus importantes des réseaux de neurones artificiels est leur capacité d'apprentissage à partir de la présentation d'échantillons (modèles), qui exprime le comportement du système. Les neurones sont reliés entre eux via des coefficients synaptiques ou poids, l'apprentissage est le processus d'adaptation ces paramètres d'un système pour donner une réponse désirée à une entrée ou simulation quelconque, dont une seule combinaison offrira une approche optimale au sens des moindres carrés.

En effet, on utilise généralement la somme des écarts quadratiques comme critère d'ajustement, d'une part car ce dernier ne peut pas s'annuler par simple retranchement des erreurs et, d'autre part, la somme des écarts quadratiques possède des propriétés statistiques liées à la définition même de la variance.

Afin d'aboutir à cette synthèse, il est nécessaire d'effectuer une phase dite 'd'apprentissage', qui consiste à trouver les poids et les biais optimaux pour une

architecture de réseau donnée. Comme les fonctions d'activation et les fonctions à approcher sont généralement de nature non linéaire, les méthodes employées s'inspirent des méthodes d'optimisation non linéaires.

Il existe de nombreux types d'apprentissage qui peuvent être regroupées dans trois catégories (supervisé, non supervisé, et renforcé), mais l'objectif fondamental de l'apprentissage reste le même.

### **VI.1 Apprentissage Supervisé**

Cet algorithme d'apprentissage ne peut être utilisé que lorsque les combinaisons d'entrées-sorties désirés sont connues. Le réseau doit ajuster ses poids de façon à réduire l'erreur entre la réponse désirée et la sortie du réseau. Les poids sont initialisés aléatoirement avant l'apprentissage, puis modifiés itérativement jusqu'à obtention d'un compromis satisfaisant entre la précision de l'approximation sur l'ensemble d'apprentissage et la précision de l'approximation sur un ensemble de validation disjoint du précédent.

Cet apprentissage n'est possible que si un large jeu de données est disponible et si les solutions sont connues pour les exemples de la base d'apprentissage.

### **VI.2 Apprentissage Non Supervisé**

Différente de l'apprentissage supervisé, l'application d'un algorithme basé sur un apprentissage non supervisé ne nécessite aucune information sur la réponse désirée. On présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise. Ce comportement est connu sous le nom "auto organisation".

Qu'est-ce que le réseau apprend exactement dans ce cas ? L'apprentissage consiste à détecter les similarités et les différences dans l'ensemble d'apprentissage. Les poids et les sorties du réseau convergent, en théorie, vers les représentations qui capturent les régularités statistiques des données. Ce type d'apprentissage est également dit compétitif et (ou) coopératif. L'avantage de ce type l'apprentissage réside dans sa grande capacité d'adaptation reconnue comme une auto-organisation, « self-organizing » [7]. L'apprentissage non supervisé est surtout utilisé pour le traitement du signal et l'analyse factorielle.

### **VI.3 Apprentissage par Renforcement**

L'apprentissage renforcé est une technique similaire à l'apprentissage supervisé à la différence qu'au lieu de fournir des résultats désirés au réseau, on lui accorde plutôt un grade (ou score) qui est une mesure du degré de performance du réseau après quelques itérations. Plusieurs algorithmes d'apprentissage utilisés par l'apprentissage renforcé sont basés sur des méthodes stochastiques qui sélectionnent de manière probabiliste les actions d'ajustement, en considérant un ensemble fini de solutions possibles qui peuvent être récompensées si elles ont des chances de générer des résultats satisfaisants. Pendant le processus de formation, les probabilités associées à l'ajustement de l'action sont modifiées pour améliorer la performance du réseau. Les algorithmes utilisant la procédure d'apprentissage renforcé sont surtout utilisés dans le domaine des systèmes de contrôle [18].

## **VII. Elaboration de la Structure du Réseau**

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. La façon dont se présente l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation. Il faut donc déterminer les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser.

### **VII.1 Choix et Préparation des Echantillons**

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau perceptron multicouches etc...

Par exemple, dans le cas du perceptron multicouches, il faudra déterminer les entrées/sorties du réseau, aussi bien choisir le nombre de couches cachées que le nombre de neurones dans cette couche.

**VII.1.1. Détermination des entrées/sorties du réseau de neurones**

Lorsque la grandeur que l'on veut modéliser dépend de nombreux facteurs, c'est-à-dire lorsque le modèle possède de nombreuses entrées, il n'est pas possible de réaliser un « pavage » régulier dans tout le domaine de variation des entrées : il faut donc trouver une méthode permettant de réaliser uniquement des expériences qui apportent une information significative pour l'apprentissage du modèle. Cet objectif peut être obtenu en mettant en œuvre un plan d'expériences. Pour les modèles linéaires, l'élaboration de plans d'expériences est bien maîtrisée, par ailleurs, ce n'est pas le cas pour les modèles non linéaires.

Pour toute conception de modèle, la sélection des entrées doit prendre en compte deux points essentiels :

- \_ Premièrement, la dimension intrinsèque du vecteur des entrées doit être aussi petite que possible, en d'autre terme, la représentation des entrées doit être la plus compacte possible, tout en conservant pour l'essentiel la même quantité d'information, et en gardant à l'esprit que les différentes entrées doivent être indépendantes.
- \_ En second lieu, toutes les informations présentées dans les entrées doivent être pertinentes pour la grandeur que l'on cherche à modéliser : elles doivent donc avoir une influence réelle sur la valeur de la sortie.

**VII.1.2. Nombre de couches cachées**

Mis à part les couches d'entrée et de sortie, il faut décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation. Néanmoins, il a été démontré qu'un Perceptron Multicouches avec une seule couche cachée pourvue d'un nombre suffisant de neurones, peut approximer n'importe quelle fonction avec la précision souhaitée [19].

**VII.1.3 Nombre de neurones cachés**

Chaque neurone peut prendre en compte des profils spécifiques de neurones d'entrée. Un nombre plus important permet donc de mieux "coller" aux données présentées mais diminue la capacité de généralisation du réseau. Il faut alors trouver le nombre adéquat de neurones cachés nécessaire pour obtenir une approximation satisfaisante.

Il n'existe pas, à ce jour, de résultat théorique permettant de prévoir le nombre de neurones cachés nécessaires pour obtenir une performance spécifique du modèle, compte tenu des modèles disponibles. Il faut donc nécessairement mettre en œuvre une procédure numérique de conception de modèle.

## VII.2 La Réalisation de l'Apprentissage

L'apprentissage est un problème numérique d'optimisation, qui consiste à trouver les poids optimaux d'une architecture de réseau donnée pour minimiser l'erreur quadratique moyenne (mean square error: MSE) donner par:

$$EQM = \frac{1}{PT} \sum_{PT} \sum_{NS} (Y_S - Y)^2 \quad (I.6)$$

Où  $PT$  est le nombre total d'échantillons d'apprentissage,  $Y_S$  est la sortie du réseau et  $Y$  est la sortie désirée.  $NS$  est un nombre de neurones dans la couche de sortie.

Plusieurs algorithmes itératifs peuvent être mis en œuvre, parmi lesquels on note : l'algorithme de Rétro-propagation, la Méthode Quasi-Newton, Algorithme de BFGS... etc. L'apprentissage par descente de gradient et d'autres algorithmes sera détaillé dans le chapitre 3.

## VII.3 Validation et tests

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage. Une fois le réseau de neurones développé, des tests s'imposent afin de vérifier la qualité des prévisions du modèle neuronal. Cette dernière étape doit permettre d'estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage. Une validation rigoureuse du modèle développé se traduit par une proportion importante de prédictions exactes sur l'ensemble de la validation. Si les performances du réseau ne sont pas satisfaisantes, il faudra, soit modifier l'architecture du réseau, soit modifier la base d'apprentissage.



## **VIII. Conclusion**

Dans ce chapitre, nous avons introduit les définitions essentielles relatives aux réseaux de neurones. D'abord, nous avons rappelé la définition et les propriétés des réseaux de neurones, avant de décrire les architectures neuronales les plus utilisées, à savoir les modèles statiques et dynamiques. Nous poursuivrons notre exposé en citant les différents types d'apprentissage.

Nous présenterons aussi, d'une manière générale, les étapes de conception d'un réseau de neurones : le choix des entrées et sorties, l'élaboration de la base de données, de la structure du réseau etc.....

- [1]W. S. McCulloch and W. Pitts, " **A logical calculus of the ideas immanent in nervous activity**", Bulletin of mathematics biophysics, Vol.5 (4), PP.115–133, **1943**
- [2]D. O. Hebb, "**The Organization of Behavior, a neuropsychological theory** ", Lawrence Erlbaum Associates, **1949**.
- [3]E. Davalo and P. Naim, "**Des Réseaux de Neurones**", Eyrolles, Deuxième édition, **1993**.
- [4]F. Rosenblatt, "**The Perceptron: a probabilistic model for information storage and organization in the brain**", Psychological Review, Vol. 65(6), PP. 386-408, **1958**.
- [5]B. Widrow and M. E. Hoff, " **Adaptive switching circuits** ", IRE Wescon Convention Record, Vol.4, PP. 96–104, **1960**.
- [6]M. Minsky et S. Papert, "**Perceptrons**", MIT Press, Cambridge MA, **1969**.
- [7]T. Kohonen, "**Self-organized formation of topologically correct feature maps** ", Biological Cybernetics, Vol.43, PP.59–69, **1982**.
- [8]D. Parker, " **Learning Logic** ", Tech report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, **1985**.
- [9]D. E. Rumelhart, G. E. Hinton, and R. J. Williams, " **Learning internal representations by error propagation** ", In Parallel Distributed Processing, Cambridge, MA : MIT Press, PP. 318–362, **1986**.
- [10]Y. Lecun, " **Une Procédure d'Apprentissage pour Réseau à Seuil Asymétrique**", A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences, Paris, France, PP. 599–604, **1985**.
- [11]R. Rojas, "**Neural networks A systematic introduction**", Springer Verlag, New York, **1995**.
- [12] C. Bishop, "**Neural Networks for Pattern Recognition**", Oxford University Press, New York, **1995**
- [13]G. Dreyfus, J.M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S.Thiria and L. Héroult, "**Réseaux de neurones : Méthodologie et Applications**", Eyrolles, **2002**.
- [14] M.T. Hagan, H.B. Demuth and M. Beale, "**Neural network design**", PWS Publishing Company, Boston, Massachusetts, **1996**.
- [15]F. Nufiez and A. Skrivervik, "**Filter Approximation by RBF-NN and segmentation method**", IEEE MTT-S International Microwave Symposium Digest, TX, USA, PP.1561-1564. **2004**
- [16]N.K. Kasabov, "**Foundations of neural networks**", fuzzy systems, and knowledge engineering. The MIT Press, Cambridge, Massachusetts, **1996**.

- [17] J. L. Elman, "**Finding structure in time**", Cognitive Science, Vol.14 (2), PP. 179-211, **1990**.
- [18] H. White, "**Artificial neural networks**", Blackwell, New York, **1992**.
- [19] K. Hornik, M. Stinchcombe and H. White, "**Multilayer feedforward networks are universal approximators**", Neural Networks, Vol. 2 (5), PP. 359-366, **1989**.

*Chapitre II*  
*Optimisation par les Algorithmes des*  
*Essais Intelligents*

---

## **I. Introduction**

Beaucoup de problèmes d'application en ingénierie, en sciences de la décision et en recherche opérationnelle sont formulés comme des problèmes d'optimisation. De telles applications incluent le traitement du signal numérique, optimisation structurale, conception d'ingénierie, les réseaux de neurones ... etc. Une solution optimale dans ces applications a un impact économique et social important. De meilleures conceptions d'ingénierie entraînent souvent des coûts de maintenance et de maintenance plus faibles, une exécution plus rapide et un fonctionnement plus robuste dans diverses conditions de fonctionnement.

Les problèmes d'optimisation se composent de trois composants de base: un ensemble d'inconnues ou de variables, une fonction objective à perfectionner et un ensemble de contraintes qui spécifient des valeurs possibles des variables. Le problème d'optimisation consiste à trouver les valeurs des variables qui permettent d'optimiser la fonction objective tout en satisfaisant les contraintes.

Il existe de nombreux types de problèmes d'optimisation. Les variables peuvent prendre des valeurs continues, discrètes ou symboliques. La fonction objective peut être continue ou discrète et avoir des formes linéaires ou non linéaires. Les contraintes peuvent également avoir des formes linéaires ou non et être définies implicitement ou explicitement, ou n'y peuvent pas exister.

Généralement, les méthodes d'optimisation peuvent être divisées en deux catégories: méthodes déterministes et méta-heuristiques. Parmi les méthodes déterministes les plus connues: la méthode de Newton, la méthode Simplex et la méthode du Gradient. De nombreux inconvénients sont associés à l'utilisation de méthodes déterministes telles que la nécessité d'un bon point de départ, le piégeage dans des optimums locaux et la nécessité de trop de temps pour résoudre des problèmes d'optimisation complexes.

D'autre part, les méta-heuristiques sont une famille d'algorithmes stochastiques. L'adaptation à un large éventail de problèmes sans modification majeure de leurs algorithmes est considérée comme le principal avantage de ces méthodes d'optimisation. Beaucoup de ces techniques de recherche sont inspirées par les lois naturelles et des essaims intelligents (EI).

Les algorithmes de l'essaim intelligent [1, 2] (Swarm Intelligence, SI) sont des techniques d'inspiration naturelle qui impliquent l'étude du comportement collectif de systèmes décentralisés et auto-organisés. Les systèmes d'essaim intelligent contiennent un ensemble de particules (ou d'agents) qui interagissent localement les uns avec les autres et avec leur environnement. Les techniques de l'essaim intelligent peuvent être utilisées dans plusieurs applications d'ingénierie où les algorithmes EI ont été appliqués avec succès pour résoudre des problèmes d'optimisation complexes, y compris l'optimisation continue, l'optimisation contrainte et l'optimisation combinatoire.

Dans ce chapitre, nous nous concentrons sur certains algorithmes d'optimisation basés sur les essaims intelligents, tels que l'optimisation par essaim de particules OEP (Particle Swarm Optimization, PSO), l'optimisation par araignées sociales OAS (Social Spiders Optimization, SSO), l'optimisation par singe araignée OSA (Spider Monkey Optimization, SMO), et l'optimisation à base d'enseignement-apprentissage OBEA (Teaching-Learning-Based Optimization, TLBO). Nous terminons ensuite le chapitre en utilisant ces algorithmes pour l'optimisation de deux structures micro-onde (filtres à iris inductif à trois cavités et Transformateur à deux paliers).

### **II. Optimisation par Essaim de Particule** (Particle Swarm Optimization, PSO)

L'optimisation par essaim de particule a été proposée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [3]. OEP est une méthode d'optimisation stochastique basée sur la reproduction d'un comportement social des animaux qui n'ont pas de leader dans leur groupe ou leur essaim, tels que les bancs de poissons et les groupés de vols d'oiseaux [Figure II.1](#). En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim. L'information locale et la mémoire de chaque individu sont utilisées pour décider de son déplacement. Des règles simples, telles que « rester proche des autres individus », « aller dans une même direction » ou « aller à la même vitesse », suffisent pour maintenir la cohésion de l'essaim, et permettent la mise en œuvre de comportements collectifs complexes et adaptatifs.



Figure II.1 Des essaims d'oiseaux et de poissons.

## II.1 Les éléments d'OEP

Pour appliquer l'algorithme d'OEP il faut définir un espace de recherche constitué des particules et une fonction objective à optimiser. Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum. Chacune de ces particules est dotée :

- *d'une position*, c'est-à-dire ses coordonnées dans l'ensemble de définition.
- *d'une vitesse*, qui permet à la particule de se déplacer.
- *d'un voisinage*, c'est-à-dire un ensemble de particules qui interagissent directement sur la particule, en particulier celle qui a le meilleur critère.

De cette façon, au cours des itérations, chaque particule change de position. Elle évolue en fonction de son meilleur voisin, de sa meilleure position, et de sa position précédente. C'est cette évolution qui permet de tomber sur une particule optimale.

- Sa meilleure position visitée. On retient essentiellement la valeur du critère calculée ainsi que ses coordonnées.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnancement optimal.
- La valeur qu'elle donne à la fonction objective car à chaque itération il faut une comparaison entre la valeur du critère donnée par la particule courante et la valeur optimale.

## II.2 Principe de la méthode

Le principe de OEP s'est éloigné du comportement (trop complexe) des animaux pour ne conserver qu'une modélisation basée sur des agents simples appelés particules. Un essaim de particules est une solution potentielle au problème d'optimisation.

Le déplacement d'une particule est influencé par trois composantes :

- *une composante d'inertie* : la particule tend à suivre sa direction courante de déplacement;
- *une composante cognitive* : la particule tend à se fier à sa propre expérience et, ainsi, à se diriger vers le meilleur site par lequel elle est déjà passée
- *une composante sociale* : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint collectivement par l'essaim.

La Figure II.2 illustre la stratégie de déplacement d'une particule :

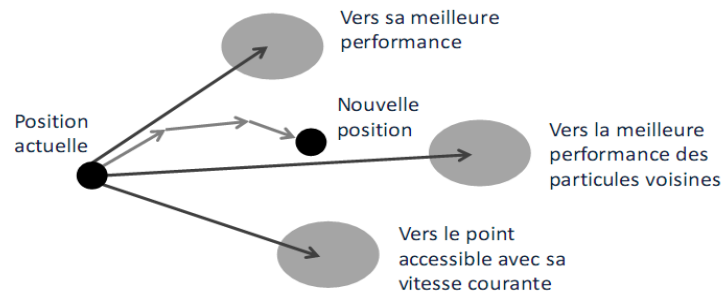


Figure II.2 Déplacement d'une particule.

L'essaim de particules est constitué de  $N$  particules et la position de chaque particule représente une solution dans l'espace de recherche. Les particules changent d'état selon les trois principes suivants: (Garder son inertie, changer d'état en fonction de sa position la plus optimale and changer d'état selon la position la plus optimale du groupe).

La position de chaque particule est affectée à la fois par la position la plus optimale lors de son mouvement (expérience individuelle) et la position de la particule la plus optimale dans ses environs (expérience globale). La mise à jour de la position  $x_i(t)$  et la vitesse  $v_i(t)$  d'une particule  $P_i$  est représentée par les équations (II.1) et (II.2)

$$v_i(t+1) = \omega * v_i(t) + c_1 r_1 [LP_i(t) - x_i(t)] + c_2 r_2 [G(t) - x_i(t)] \quad (\text{II.1})$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (\text{II.2})$$

Où  $\omega$  est l'inertie,  $c_1$  et  $c_2$  sont des coefficients constants fixés par l'utilisateur et  $r_1$  et  $r_2$  sont des nombres aléatoires dans la gamme  $[0, 1]$ .  $G(t)$  est la meilleure solution trouvée jusqu'à l'instant  $t$  et  $LP_i(t)$  la meilleure solution trouvée par la particule  $P_i$ .



## II.3 Configuration de la Méthode

### II.3.1 Nombre de particules

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre et faire de nombreux essais permettant de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

### II.3.2 Taille et topologie de voisinage

Le voisinage d'une particule est le sous-ensemble de particules de l'essaim avec lequel il a une communication directe. Ce réseau de rapports entre toutes les particules est connu comme la sociométrie, ou la topologie de l'essaim. Il existe de nombreuses combinaisons dont les suivantes sont les plus utilisées.

**Topologie en étoile** : Chaque particule est reliée à toutes les autres, l'optimum du voisinage est l'optimum global. Chaque particule est attirée vers la meilleure particule et communique avec les autres.

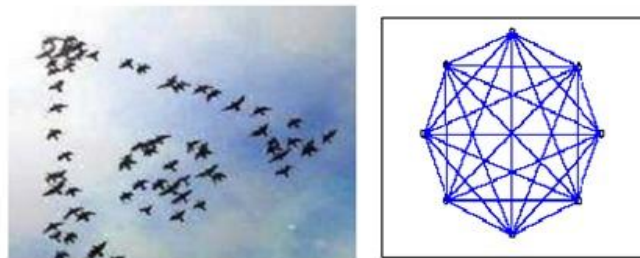


Figure II.3 Voisinage étoile

**Topologie en anneau**: Chaque particule est reliée à  $n$  particules ( $n = 3$  en général). Ce cas est souvent utilisé. Chaque particule tend à se déplacer vers la meilleure particule dans son voisinage local.

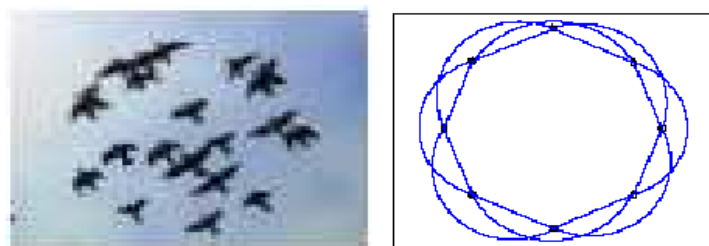


Figure II.4 Voisinage anneau.

**Topologie en rayon** : Les particules ne communiquent qu'avec une seule particule centrale.

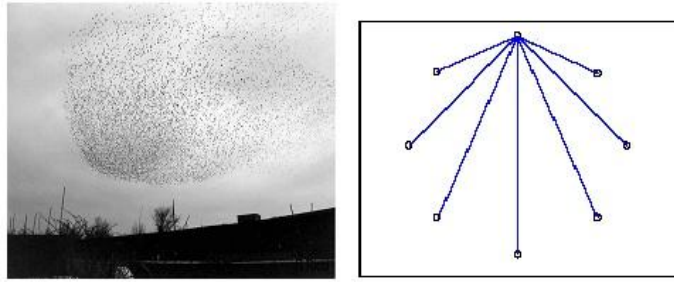


Figure II.5 Voisinage rayon.

Le choix de la topologie de voisinage a un grand effet sur la propagation de la meilleure solution trouvée par l'essaim. En utilisant le modèle étoile, la propagation est très rapide. Cependant, l'utilisation des topologies anneau et rayon ralentira la convergence car la meilleure solution trouvée doit propager par plusieurs Voisinage avant d'affecter toutes particules dans l'essaim. Cette propagation lente permettra aux particules d'explorer plus les secteurs dans la recherche espacent et diminuent ainsi la chance de la convergence.

### II.3.3 Coefficients de confiance

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme. Les variables aléatoires  $\rho_1$  et  $\rho_2$  peuvent être définis de la façon suivante :

$$\begin{cases} \rho_1 = c_1 r_1 \\ \rho_2 = c_2 r_2 \end{cases} \quad (\text{II.3})$$

Où  $r_1$  et  $r_2$  sont des nombres aléatoires compris entre 0 et 1 et  $c_1$  et  $c_2$  sont des constantes positives déterminées de façon empirique en respectant la relation  $c_1 + c_2 \leq 4$ .

### II.3.4 Coefficient d'inertie

Le coefficient d'inertie  $\omega$  contrôle l'influence de la direction de la particule sur le déplacement futur. Le but de l'introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie. Une grande valeur de  $w$  facilite l'exploration globale, sinon c'est l'exploration locale qui est concernée. Du fait de son influence sur les performances de l'algorithme OEP, le poids

d'inertie a suscité un grand intérêt de la part de la communauté des chercheurs. Dans [4], les auteurs ont proposé un coefficient d'inertie dynamique qui varie au cours du temps. Il commence par une valeur proche de 0,9 et descend linéairement pour arriver à 0,2. Cette stratégie a beaucoup amélioré les performances de l'OEP dans plusieurs problèmes d'optimisation. Le coefficient d'inertie  $\omega$  varie linéairement avec le temps selon la formule suivante :

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \frac{it}{maxit} \quad (II.4)$$

Où  $it$  est l'itération courante et  $maxit$  est le nombre maximal d'itérations. Les coefficients  $\omega_{max}$  et  $\omega_{min}$  désignent respectivement les valeurs maximum et minimum du coefficient  $\omega$ .

#### **II.4 Algorithme d' OEP**

Les étapes essentielles de l'optimisation par essaim de particules sont présentées par l'algorithme suivant :

**Étape 1:**Initialisation aléatoire de la population et la vitesse des particules.

**Étape 2:**Trouver la meilleure valeur de la fonction objective.

**Étape 3:** Traitement

**Répéter** jusqu'à fin d'itération.

**Répéter** pour chaque particule.

Générez la nouvelle valeur de la vitesse en utilisant l'équation (II.1).

Calculez la nouvelle position en utilisant l'équation (II.2).

Evaluation de la valeur de la fonction objective.

Trouvez la meilleure position pour chaque particule.

**Fin**

Trouvez la meilleure position globale

**Fin**

En général le critère d'arrêt peut être un nombre d'itérations fixe.

### III. Optimization par Araignées Sociales (Social Spiders Optimization, SSO)

Erik Cuevas et al [5-7] ont proposé un nouvel algorithme d'optimisation d'essaims appelé Optimisation par araignées sociales OAS (Social-Spiders Optimization, SSO). OAS est basée sur le comportement des colonies d'araignées naturelles [Figure II.6](#). L'OAS suppose que l'espace de recherche entier est un réseau commun où toutes les araignées sociales interagissent les unes avec les autres. Dans l'approche proposée, chaque solution dans l'espace de recherche représente une position d'araignée sur le réseau commun (réseau social). Chaque araignée reçoit un poids en fonction de leur valeur de la fonction objective. L'algorithme modélise deux agents de recherche (araignées): mâles et femelles. En fonction du sexe, chaque individu est conduit par un ensemble d'opérateurs évolutifs différents qui représentent différents comportements coopératifs qui sont généralement pris dans la colonie.



**Figure. II.6** Différents environnements formés par les araignées sociales.

Une caractéristique intéressante des sociales araignées est la population hautement féminine. Afin d'imiter ce fait, l'algorithme commence par définir le nombre d'araignées femelles et mâles qui seront caractérisées comme des individus dans l'espace de recherche. Le nombre de femelles  $N_f$  est sélectionné au hasard dans la gamme de 65-90% de la population entière  $N$ . Par conséquent,  $N_f$  est calculé par l'équation suivante:

$$N_f = \text{floor} [(0.9 - \text{rand} * 0.25) * N] \quad (\text{II.5})$$

Où  $rand$  est un nombre aléatoire entre  $[0,1]$ , ( $floor$ ) transfère un nombre réel à un nombre entier. Le nombre d'araignées mâles  $N_m$  est calculé comme étant le complément entre  $N$  et  $N_f$ . Il est calculé comme suit:

$$N_m = N - N_f \quad (\text{II.6})$$

Dans la métaphore biologique, la taille de l'araignée est la caractéristique qui évalue la capacité individuelle à mieux performer sur ses tâches assignées. Dans l'approche proposée, chaque individu (araignée) reçoit un poids  $W_i$  qui représente la qualité de la solution qui correspond à l'araignée  $i$  (indépendamment du genre) de la population  $N$ . Pour calculer le poids de chaque araignée, la prochaine équation est utilisée:

$$W_i = \frac{pire - f(x_i)}{pire - meilleur} \quad (\text{II.7})$$

Où  $W_i$  est le poids de l'araignée  $i$ ,  $f(x_i)$  est la valeur de la fonction objective de l'araignée  $x_i$ . Les valeurs  $pire$  et  $meilleur$  sont définis comme suit (en considérant un problème de minimisation):

$$meilleur = \min_{i=1,2,\dots,N} (f(x_i)) \quad \text{and} \quad pire = \max_{i=1,2,\dots,N} (f(x_i)) \quad (\text{II.8})$$

### III.1 Modélisation des Vibrations à Travers le Réseau Commun

Le réseau commun est utilisé comme un mécanisme pour transmettre des informations parmi les membres de la colonie. Cette information est codée en tant que petites vibrations qui sont essentielles à la coordination collective de tous les individus de la population. Les vibrations dépendent du poids et de la distance de l'araignée qui les a générés. Étant donné que la distance est relative à l'individu qui provoque les vibrations et le membre qui les détecte, les membres situés près de l'individu qui provoque les vibrations, perçoivent des vibrations plus fortes par rapport aux membres situés dans des positions éloignées. Pour reproduire ce processus, les vibrations perçues par l'individu  $i$  à la suite de l'information transmise par l'élément  $j$  sont modélisées selon l'équation suivante:

$$Vib_{ij} = W_j * \exp\left(-(d_{ij})^2\right) \quad (\text{II.9})$$

Où le  $d_{ij}$  est la distance Euclidienne entre les araignées  $i$  et  $j$ . Il existe principalement trois types de vibrations Figure II.7:

- Les Vibrations  $Vibc_i$  sont perçues par l'individu  $i$  ( $S_i$ ) à la suite de l'information transmise par l'élément  $c$  ( $S_c$ ) qui est un individu qui a deux caractéristiques importantes: il est le plus proche de l'élément  $i$  et possède un poids plus élevé par rapport à  $i$  ( $W_c > W_i$ ).

$$Vibc_i = W_c * \exp(-(d_{ic})^2) \quad (II.10)$$

- Les vibrations  $Vibb_i$  sont perçues par l'individu  $i$  en tant que résultat de l'information transmise par le membre  $b$  ( $S_b$ ), avec  $b$  étant l'individu possédant le meilleur poids (meilleure valeur de la fonction objective) de l'ensemble de la population  $N$ , de telle sorte que  $W_b = \max_{k=1,2 \dots N}(W_k)$  [10].

$$Vibb_i = W_b * \exp(-(d_{ib})^2) \quad (II.11)$$

- Les vibrations  $Vibf_i$  sont perçues par l'individu  $i$  ( $S_i$ ) à la suite de l'information transmise par l'élément  $f$  ( $S_f$ ), avec  $f$  étant l'individu féminine le plus proche de  $i$ .

$$Vibf_i = W_f * \exp(-(d_{if})^2) \quad (II.12)$$

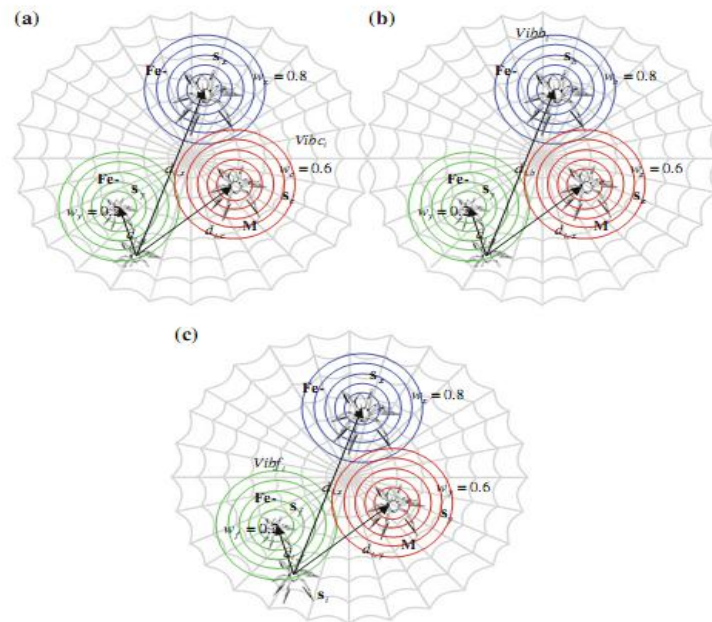


Figure II. 7 Configuration de chaque relation spéciale: a)  $Vibc_i$  , b)  $Vibb_i$  , c)  $Vibf_i$  [5].

### III.2 Les Opérations Coopératives

L'araignée sociale entretient des relations de coopération avec les autres membres de la colonie. Cette relation dépend bien entendu du sexe de l'araignée.

#### III.2.1 Opération coopérative femelle

Les araignées femelles sont déplacées soit sur un mouvement de répulsion ou sur une attraction. Un tel changement de position de l'araignée femelle  $i$  est calculé comme une combinaison de trois éléments différents. Le premier concerne le changement en ce qui concerne l'élément le plus proche de  $i$  qui détient un poids plus élevé et produit la vibration  $Vibc_i$ . La seconde considère le changement en ce qui concerne le meilleur individu de la population entière  $N$  qui produit la vibration  $Vibb_i$ . Enfin, le troisième incorpore un mouvement aléatoire.

Étant donné que le dernier mouvement d'attraction ou de répulsion dépend de plusieurs phénomènes aléatoires, la sélection est modélisée sous la forme d'une décision stochastique. Pour cette opération, un nombre aléatoire  $r_m$  uniforme est généré dans l'intervalle  $[0,1]$ . Si  $r_m$  est inférieur à un  $PF$  de seuil, un mouvement d'attraction est généré; par ailleurs, un mouvement de répulsion est produit. Par conséquent, cet opérateur peut être modélisé comme suit:

$$f_i^{k+1} = \begin{cases} f_i^k + \alpha Vibc_i(S_c - f_i^k) + \beta Vibb_i(S_b - f_i^k) + \delta \left( rand - \frac{1}{2} \right) & \text{avec probabilité } PF \\ f_i^k - \alpha Vibc_i(S_c - f_i^k) - \beta Vibb_i(S_b - f_i^k) + \delta \left( rand - \frac{1}{2} \right) & \text{avec probabilité } 1 - PF \end{cases} \quad (\text{II.13})$$

Où  $\alpha, \beta, \delta$  et  $rand$  sont des nombres aléatoires entre  $[0,1]$ ,  $k$  représente le nombre d'itérations. L'individuel  $S_c$  et  $S_b$  représentent l'élément le plus proche de  $i$  qui détient un poids plus élevé et le meilleur individu de la population totale  $N$ , respectivement.

#### III.2.2 Opération coopérative mâle

Dans ce comportement coopératif, selon le comportement biologique de l'araignée sociale, la population masculine est divisée en deux classes: les araignées dominantes et non dominantes. Les araignées masculines dominantes ont de meilleures caractéristiques de forme physique (généralement en ce qui concerne la taille) par rapport à celles qui ne sont

pas dominantes. Les mâles dominants sont attirés par l'araignée féminine la plus proche du réseau commun (social). En contraste, les araignées non dominantes ont tendance à se concentrer au centre de la population masculine en tant que stratégie pour tirer profit des ressources gaspillées par les mâles dominants.

Les membres masculins sont divisés en membres dominants  $D$  et membres non dominants  $ND$  en fonction de leur position par rapport à l'élément médian. Le poids médian est indexé par  $W_{N_f+m}$  et le changement de position de l'araignée mâle peut être modélisé comme suit:

$$m_i^{k+1} = \begin{cases} m_i^k + \alpha Vibf_i(S_f - m_i^k) + \delta \left( rand - \frac{1}{2} \right) & \text{Si } W_{N_f+i} > W_{N_f+m} \\ m_i^k + \alpha \left( \frac{\sum_{h=1}^{N_m} m_h^k W_{N_f+h}}{\sum_{h=1}^{N_m} W_{N_f+h}} - m_i^k \right) & \text{Si } W_{N_f+i} \leq W_{N_f+m} \end{cases} \quad (\text{II.14})$$

Où l'individu  $S_f$  représente l'individu femelle la plus proche de l'élément mâle  $i$  tandis que  $\left( \frac{\sum_{h=1}^{N_m} m_h^k W_{N_f+h}}{\sum_{h=1}^{N_m} W_{N_f+h}} \right)$  correspondent à la moyenne pondérée de la population masculine.

### **III.2.3 Opération d'accouplement**

L'accouplement dans une colonie sociale-araignée est effectué par les mâles dominants et les membres femelles qui se trouvent dans un certain rayon appelé rayon d'accouplement donné par:

$$R = \frac{\sum_{d=1 \dots n} (X_d^h - X_d^l)}{2 D} \quad (\text{II.15})$$

Où  $X_d^h$  et  $X_d^l$  sont respectivement la limite supérieure et inférieure pour une dimension donnée et  $D$  est la dimension du problème. Les mâles et les femelles qui sont sous le rayon d'accouplement génèrent de nouvelles araignées selon la méthode de la roulette: le poids de chaque araignée impliquée définit la probabilité d'influence pour chaque individu dans la nouvelle araignée.

$$P_{Si} = \frac{W_i}{\sum_{j=1}^{n_c} W_j} \quad (\text{II.16})$$

Où  $n_c$  le nombre d'araignées qui se trouvent dans le rayon d'accouplement.



Une fois que la nouvelle araignée est formée  $S_{new}$ , elle est comparée à la pire araignée  $S_{worst}$  de la colonie, selon leurs valeurs de poids. Si la nouvelle araignée est meilleure que la pire araignée, la pire est remplacée par la nouvelle. Sinon, la nouvelle araignée est rejetée et la population ne souffre pas de changements. En cas de remplacement, la nouvelle araignée assume le genre et l'index de l'araignée remplacée. Un tel fait assure que toute la population  $N$  maintient le taux initial entre les membres femelles et mâles.

### III.3 Procédure d'Algorithme

La procédure de calcul de l'algorithme proposé peut être résumée comme suit:

**Étape 1:** Définir le nombre de mâles  $N_m$  et de femelles  $N_f$  araignées dans toute la population  $N$  équations (II.5) et (II.6).

**Étape 2:** Initialiser au hasard les membres femelles et males et calculer le rayon d'accouplement équation (II.15).

Pour $i = 1: N_f$	Pour $k = 1: N_m$
Pour $j = 1: D$	Pour $j = 1: D$
$f_{ij}^0 = X_j^l + rand(X_j^h - X_j^l)$	$m_{ij}^0 = X_j^l + rand(X_j^h - X_j^l)$
Fin Pour	Fin Pour
Fin Pour	Fin Pour

**Étape 3:** Calculer le poids de chaque araignée équation (II.7).

**Étape 4:** Déplacer les araignées selon l'opération coopératif femelle.

Pour $i = 1: N_f$
Calculer $Vibc_i$ et $Vibb_i$ équations (II.10), (II.11)
Si ( $rand < PF$ )
$f_i^k + \alpha Vibc_i(S_c - f_i^k) + \beta Vibb_i(S_b - f_i^k) + \delta \left( rand - \frac{1}{2} \right)$
Sinon
$f_i^k - \alpha Vibc_i(S_c - f_i^k) - \beta Vibb_i(S_b - f_i^k) + \delta \left( rand - \frac{1}{2} \right)$
Fin SI
Fin Pour

**Étape 5:** Déplacer les araignées selon l'opérateur coopératif mâle.

```

Trouvez l'individu médian des mâles  $W_{N_f+m}$ 
Pour  $i = 1: N_f$ 
    Calculer  $Vibf_i$  équation (II.12)
    Si ( $W_{N_f+i} > W_{N_f+m}$ )
         $m_i^k + \alpha Vibf_i (S_f - m_i^k) + \delta \left( rand - \frac{1}{2} \right)$ 
    Sinon
         $m_i^k + \alpha \left( \frac{\sum_{h=1}^{N_m} m_h^k W_{N_f+h}}{\sum_{h=1}^{N_m} W_{N_f+h}} - m_i^k \right)$ 
    Fin SI
Fin Pour
    
```

Étape 6: Effectuer l'opération d'accouplement (Section III.2.3).

Étape 7: Si les critères d'arrêt est atteint, le processus est terminé; sinon, retourner à l'étape3

#### IV. Optimisation par Singe-Araignée (Spider Monkey Optimization, SMO)

L'optimisation par singes araignées OSA (Spider Monkey Optimization, SMO) [8,9] est un nouvel algorithme inspiré de la nature et motivé par le comportement social des singes araignées Figure II.8. L'OSA est une stratégie d'optimisation stochastique basée sur la population qui imite la structure sociale fission-fusion (FFSS) [10] basée sur un comportement de recherche de nourriture des singes araignées.



Figure II. 8 Quelques images de singes araignées

Les principales caractéristiques des singes araignées sont les suivantes :

- Survivre en groupes de 40-50 singes.

- La plus vieille singe en groupe devient généralement leader (connu sous le nom de leader global) du groupe et responsable de tous types de décision.
- Pour rechercher des nourritures, qu'ils divisent le grand groupé en petits sous-groupes (taille 3 à 8).
- Tous les sous- groupes sont également dirigés par un leader local qui sera chargé de planifier un parcours de recherche disciplinée toute la journée.
- Le comportement alimentaire des araignées dans l'algorithme OSA est classé en quatre parties:
  - le groupe commence la recherche de nourriture et évalue leur distance de la nourriture.
  - Dans la deuxième étape, en fonction de la distance des aliments, les membres du groupe mettent à jour leurs positions et encore évaluer la distance des sources de nourriture.
  - À la troisième étape, le leader local met à jour sa meilleure position au sein du groupe et si la position n'est pas mise à jour pour un nombre de fois spécifié, tous les membres de ce groupe commencent la recherche de la nourriture dans des directions différentes.
  - Ensuite, dans la quatrième étape, le leader global met à jour sa meilleure position et en cas de stagnation, il divise le groupe en sous-groupes de plus petite taille.

Le processus OSA se compose de six phases: la phase de leader local (PLL), la phase de leader global (PLG), la phase d'apprentissage du leader local (PALL), la phase d'apprentissage du leader global (PALG), la phase de décision du leader local (PDLL) et la phase de décision du leader global (PDLG). Il existe quatre paramètres de contrôle importants pour ces phases; le taux de perturbation (Pr), leader local limite (LLL), leader global limite (LGL) et le nombre maximum de groupes (MG). Les détails de chaque phase sont expliqués ci-dessous

### IV.1 Phase de Leader Locale (PLL)

Dans la phase de leader local, chaque singe araignée  $SA$  modifie sa position actuelle en fonction de l'expérience du leader local ainsi que des membres locaux du groupe comme l'expression suivante:

$$SA_{ij} = SA_{ij} + r_1(LL_{Kj} - SA_{ij}) + r_2(SA_{rj} - SA_{ij}) \quad (\text{II.17})$$

Où  $i = 1..N$ ,  $N$  est la taille de population,  $j = 1..D$ ,  $D$  est le nombre de variables d'optimisation.  $SA_{ij}$  est la  $j^{\text{ème}}$  dimension de la  $i^{\text{ème}}$  SA,  $LL_{kj}$  représente la  $j^{\text{ème}}$  dimension de la  $K^{\text{ème}}$  position de leader local de groupe.  $SA_{rj}$  est la  $j^{\text{ème}}$  dimension du  $r^{\text{ème}}$  SA qui est choisi de manière aléatoire dans le groupe  $K^{\text{ème}}$  de sorte que  $r \neq i$ .  $r_1$  est un nombre aléatoire entre 0 et 1 et  $r_2$  est un nombre aléatoire entre -1 et 1.

L'Algorithme 1 montre le processus de mise à jour de la position dans la phase de leader local. Dans l'Algorithme 1,  $MG$  est le nombre maximal de groupes dans l'essaim et  $Pr$  est le taux de perturbation qui contrôle la quantité de perturbation dans la position actuelle.  $Pr$  Varié dans l'intervalle de [0.1, 0.9] (section IV.8).

Algorithme 1: Phase de Leader Local	
Pour	Chaque $k \in \{1 \dots MG\}$
Pour	Chaque membre $SA_i \in k$ groupe
Pour	Chaque $j \in \{1 \dots D\}$
Si	$(rand \geq Pr)$ Alors
	$SA_{new_{ij}} = SA_{ij} + r_1(LL_{kj} - SA_{ij}) + r_2(SA_{rj} - SA_{ij})$
	sinon
	$SA_{new_{ij}} = SA_{ij}$
	Fin Si
	Fin Pour
	Fin Pour
	Fin Pour

**IV.2 Phase de Leader Global (PLG)**

Après avoir terminé la phase de leader local, la phase de leader global (PLG) commence. En PLG, tous les SA ont mis à jour leur position en utilisant l'expérience du leader global et des membres locaux du groupe. L'équation de mise à jour de position pour cette phase est la suivante:

$$SA_{ij} = SA_{ij} + r_1(GL_j - SA_{ij}) + r_2(SA_{rj} - SA_{ij}) \tag{II.18}$$

Où  $GL_j$  représente la  $j^{\text{ème}}$  dimension de la position de leader global et l'index  $j \in \{1, 2, \dots, D\}$  est choisi de manière aléatoire.  $r_1$  est un nombre aléatoire entre 0 et 1 et  $r_2$  est un nombre aléatoire entre -1 et 1.

Dans cette phase, les positions des singes araignées ( $SA_i$ ) sont mises à jour en fonction des probabilités  $Pro_i$  qui sont calculées en fonction de leur fonction d'objective. La probabilité  $Pro_i$  peut être calculée par l'expression suivante:

$$Pro_i = 0.9 \frac{F_i}{max\_F} + 0.1 \tag{II.19}$$

Ici  $F_i$  est la fonction objective de la  $i^{ème}$  et  $max\_F$  est la fonction objective maximale du groupe.

Algorithme 2: Phase de Leader Global
<pre> Pour k = 1: MG faire   count=1   GS Taille de Groupe k   Tant que count &lt; GS faire     Pour i = 1: GS faire       Si (rand ≤ Pi) Alors         count=count+1         sélectionner au hasard j ∈ {1 ... D}         sélectionner aléatoirement SA<sub>rj</sub> du groupe k où i ≠ r         SAnew<sub>ij</sub> = SA<sub>ij</sub> + r<sub>1</sub>(GL<sub>j</sub> - SA<sub>ij</sub>) + r<sub>2</sub>(SA<sub>rj</sub> - SA<sub>ij</sub>)       Fin Si     Fin pour     Si (i = GS) Alors       i = 1     Fin Si   Fin Tant que Fin pour           </pre>

**IV.3 Phase d'Apprentissage du Leader Global (PALG)**

Dans cette phase, la position de leader global est mise à jour en appliquant la sélection avide dans la population, c'est-à-dire que la position du SA ayant la meilleure performance dans la population est sélectionnée comme la nouvelle position de leader global. En outre,

il est vérifié que la position de leader global est en cours de mise à jour et, si ce n'est pas le cas, le Global Limite Count (GLC) est incrémenté de 1.

#### **IV.4 Phase d'Apprentissage du Leader Local (PALL)**

Dans cette phase, la position du leader local est mise à jour en appliquant la sélection avide dans chaque groupe, c'est-à-dire la position du SA ayant la meilleure fonction d'objective dans ce groupe qui est sélectionnée comme nouvelle position du leader local. Ensuite, la position mise à jour du leader local est comparée à l'ancienne et si le leader local n'est pas mis à jour, le Local Limite Count (LLC) est incrémenté de 1.

#### **IV.5 Phase de Décision du Leader Local (PDLL)**

Si la position de leader local n'est pas mise à jour jusqu'à un seuil prédéterminé appelé Leader Local Limite, tous les membres de ce groupe ont mis à jour leurs positions soit par une initialisation aléatoire, soit par l'utilisation des informations combinées de leader global et leader local équation (II.20).

$$SA_{ij} = SA_{ij} + r_1(GL_j - SA_{ij}) + r_1(SA_{ij} - LL_{Kj}) \quad (II.20)$$

Le pseudo-code de la phase DLL pour le groupe k est indiqué dans l'algorithme 3. Dans cet algorithme, *Local Limite Count<sub>k</sub>* est le compteur de la meilleure solution locale du groupe k.

<b>Algorithme 3: Phase de Décision du Leader Local</b>	
Pour	$k = \{1 \dots MG\}$ faire
Si	$(Local\ Limite\ Count_k > LLL)$ alors
	$Local\ Limite\ Count_k = 0$
	$GS =$ la taille du groupe $k$
Pour	$i \in \{1 \dots GS\}$ faire
	Pour Chaque $j \in \{1 \dots D\}$ faire
	Si $(rand \geq Pr)$ alors
	$SA_{new_{ij}} = SA_{min\ j} + rand(SA_{max\ j} - SA_{min\ j})$
	Sinon
	$SA_{new_{ij}} = SA_{ij} + r_1(GL_j - SA_{ij}) + r_1(SA_{ij} - LL_{Kj})$
	Fin Si
	Fin Pour
	Fin Pour
	Fin Si
	Fin Pour

**IV.6 Phase de Décision du Leader Global (PDLG)**

Dans cette phase, la position de leader global est surveillée et si elle n'est pas mise à jour jusqu'à un nombre prédéterminé d'itérations appelé Leader Global Limite, le leader global divise la population en petits groupes. Pour cela, la population est divisée en deux groupes, puis trois groupes, etc. jusqu'à le nombre maximal de groupes (MG). Chaque fois en phase DGL, le processus LLL est lancé pour les nouveaux groupes formés. Dans lequel le nombre maximum de groupes est formé et le leader global n'est pas mis à jour, le leader global combine tous les groupes pour former un seul groupe. Le fonctionnement de cette phase est illustré dans [Algorithme 4](#)

Algorithme II.4: Phase Décision de Leader Global	
Si	( <i>Global Limite Count</i> > <i>Global Leader Limite</i> ) alors
	<i>Global Limite Count</i> = 0
	Si (le nombre de groupes < MG) alors
	Diviser la population en groupes
	Si non
	Combiner tous les groupes pour former un seul groupe
	Fin Si
	mettre à jour la position du local leader
Fin Si	

**IV.7 Paramètres de Contrôle**

La taille de population N, le taux de perturbation (*pr*), la limite de leader local (LLL), la limite de leader global (LLG) et le nombre maximum de groupes (MG) sont les paramètres qui affectent les performances du OSA de manière significative.

- *Pr* varie de 0,1 à 0,9 tandis que MG, LLL, GLL et N sont fixée à 5, 1500, 50 et 50, respectivement.
- MG varie de 1 à 6 alors que LLL, GLL et N sont fixée à 1500, 50 et 50 respectivement. *Pr* est augmentant linéairement de 0,1 à 0,9 par itérations.

$$Pr(k + 1) = Pr(k) + \frac{(0.9 - 0.1)}{maxit} , \quad Pr(1) = 0.1 \tag{II.21}$$

- Avec  $k$  le nombre d'itérations et  $maxit$  le nombre maximal d'itérations
- LGL varie de 10 à 220 alors que LLL, MG et N sont fixées respectivement à 1500, 5 et 50.  $pr$  augmente linéairement de 0,1 à 0,9 par itérations.
- LLL varie de 100 à 2500 tandis que MG, LGL et La taille des populations est fixée à 5, 50 et 50, respectivement.  $pr$  augmente linéairement de 0,1 à 0,9 par itérations.
- La taille de population varie de 40 à 160 alors que LLL, LGL et MG sont fixés respectivement à 1500, 50 et 5.  $pr$  augmente linéairement de 0,1 à 0,9 par itérations.

Il est clair d'après la discussion ci-dessus qu'il existe quatre paramètres de contrôle dans l'algorithme OSA. Les paramètres de contrôle sont proposés comme suit:

- \_ MG =  $N / 10$ , c'est-à-dire qu'il est choisi tel que le nombre minimum de SA dans un groupe soit de 10
- \_ Leader Global Limit  $\in [N / 2, 2 \times N]$ ,
- \_ Leader Local Limit devrait être  $D \times N$ ,
- \_  $pr \in [0.1, 0.9]$ ,

### IV.8 Procédure d'Algorithme

Les procédures de calcul de l'algorithme proposé peuvent être résumées comme suit:

**Etape 1:** Initialisation de la population N, leader locale Limite (LLL), leader globale Limite(GLL),et  $Pr$ .

**Etape 2:** Calculer la fonction objective (c'est-à-dire la distance entre les individus et les sources de nourriture).

**Etape 3:** Tant que (Les critères d'arrêt ne sont pas satisfaits) faire

- \_ Générer les nouvelles positions pour tous les membres du groupe en utilisant l'expérience personnelle, l'expérience du leader local et l'expérience des membres du groupe, [Algorithme 1](#)
- \_ Calculer la probabilité pour tous les membres du groupe en utilisant l'équation (II.19).



- \_ Produire les positions de tous les membres du groupe, sélectionnés par probabilité, en utilisant l'expérience personnelle, l'expérience de leader global et les expériences des membres du groupe, [Algorithme 2](#).
- \_ Mettre à jour la position des leaders locaux et du leader global ([sections IV.3 et IV.4](#)).
- \_ S'il y a de leader local ne mettant pas à jour sa position après un certain nombre (Limite Leader Local), rediriger tous les membres de ce groupe en utilisant [algorithme 3](#).
- \_ Si le leader global ne met pas à jour sa position pour un nombre spécifié (Limite Leader Global), le leader global divise le groupe en plusieurs petits groupes par [algorithme 4](#).

Fin Tant que.

### **V. Optimisation à Base d'Enseignement- Apprentissage** (Teaching-learning-based optimization, TLBO)

L'algorithme d'optimisation à base d'enseignement apprentissage OBEA (Teaching-learning-based optimization, TLBO) est un algorithme proposé par Rao et al. [11, 12] et Rao et Savsani [13], qui est basé sur l'effet de l'influence d'un enseignant sur les résultats des élèves dans une classe. Dans cette méthode, un groupe d'étudiants (apprenants) dans une classe est considéré comme une population et les différents sujets proposés aux apprenants sont analogues avec les différentes variables de conception du problème d'optimisation. Les résultats de l'apprenant sont analogues à la valeur de la fonction objective du problème d'optimisation, la meilleure solution dans l'ensemble de la population est considérée comme l'enseignant. L'enseignant et les apprenants sont les deux composantes essentielles de l'algorithme, donc il y a deux modes d'apprentissage; par l'enseignant (connu sous le nom de la phase d'enseignant) et d'interagir avec d'autres apprenants (connu sous le nom de la phase d'apprenant).

#### **V.1 Phase Enseignant**

Dans cette phase, les apprenants prennent leurs connaissances directement par l'enseignant, où un enseignant essaie d'augmenter la valeur moyenne de résultat de la classe à une autre valeur, ce qui est mieux que, en fonction de sa capacité. Cela fait suite à un processus aléatoire en fonction de nombreux facteurs.

Dans ce travail, la valeur de la solution est représentée par  $X_{j,i,k}$ , où  $j$  désigne la variable de conception (sujet pris par les apprenants),  $j = 1, 2, \dots, D$ ;  $i$  représente l'élément de la population (l'apprenant),  $i = 1, 2, \dots, N$ ; et  $k$  représente l'itération,  $k = 1, 2, \dots, maxit$ , où  $maxit$  est le nombre maximum de générations (itérations). La solution actuelle est mise à jour en fonction de l'expression suivante

$$X'_{j,i,k} = X_{j,i,k} + DMi_{j,i,k} \quad (II.22)$$

$DMi_{j,i,k}$  la différence entre la moyenne existante et la nouvelle moyenne de chaque sujet est donnée par

$$DMi_{j,i,k} = r(X_{j,i,kbest} - TF * M_{j,k}) \quad (II.23)$$

$M_{j,k}$  le résultat moyen des apprenants dans un sujet particulier  $j$ ,  $X_{j,i,kbest}$  le nouveau moyen qui correspond au résultat du meilleur apprenant (c'est-à-dire enseignant) dans le sujet  $j$ .  $r$  est un nombre aléatoire dans l'intervalle  $[0, 1]$ .  $TF$  est le facteur d'enseignement est généré de manière aléatoire pendant l'algorithme dans l'intervalle  $[1, 2]$ , dans lequel le nombre 1 correspond à aucune augmentation du niveau de connaissance et 2 correspond au transfert complet de connaissances, les valeurs intermédiaires indiquent la quantité de niveau de transfert de connaissances. La valeur de  $TF$  n'est pas donnée en tant qu'entrée de l'algorithme; sa valeur est décidée aléatoire par l'algorithme

$$TF = round [1 + rand (2- 1)] \quad (II.24)$$

## **V.2 Phase Apprenants**

C'est la deuxième partie de l'algorithme où les apprenants améliorent leur connaissance par l'interaction entre eux. Un apprenant interagit de façon aléatoire avec d'autres apprenants pour améliorer ses connaissances. Un apprenant apprend des choses nouvelles si l'autre apprenant a plus de connaissances que lui. A toute itération  $i$ , chaque apprenant est comparé aux autres apprenants au hasard. A titre de comparaison, sélectionner aléatoirement deux apprenants  $P$  et  $Q$  tels que  $X'_{P,i} \neq X'_{Q,i}$ , (où  $X'_{P,i}$  et  $X'_{Q,i}$  sont les valeurs actualisé de  $P$  et  $Q$ , respectivement, à la fin de la phase enseignant).

$$\left\{ \begin{array}{l} X''_{j,P,k} = X'_{j,P,k} + r * (X'_{j,P,k} - X'_{j,Q,k}), \text{ si } f(X'_{j,P}) < f(X'_{j,Q}) \\ X''_{j,P,k} = X'_{j,P,k} + r * (X'_{j,Q,k} - X'_{j,P,k}), \text{ si } f(X'_{j,P}) > f(X'_{j,Q}) \end{array} \right. \quad (II.25)$$

$$\left\{ \begin{array}{l} X''_{j,P,k} = X'_{j,P,k} + r * (X'_{j,Q,k} - X'_{j,P,k}), \text{ si } f(X'_{j,P}) > f(X'_{j,Q}) \end{array} \right. \quad (II.26)$$

$X_{j,P,k}''$  est acceptée si elle donne une meilleure valeur de réponse. Les équations (II.25) et (II.26) sont pour des problèmes de minimisation.

### **V.3 Procédure d'Algorithme**

L'algorithme d'optimisation à base d'enseignement et des apprenants est un algorithme basé sur la population qui simule le processus d'enseignement-apprentissage dans une classe. Cet algorithme ne nécessite que les paramètres de contrôle commun tels que la taille de la population et le nombre de générations et ne nécessite aucun paramètre de contrôle spécifique à l'algorithme. L'algorithme OBEA introduit ici est indiqué dans l'organigramme de la [Figure II.9](#).

Les étapes suivantes donnent des explications de l'algorithme OBEA.

**Étape 1:** Initialiser la taille de la population ou le nombre des apprenants dans une classe ( $N$ ), le nombre de générations ( $maxit$ ), nombre de variables ou de sujets de conception (cours) ( $D$ ) et les limites des variables de conception ( $X_{min}$  et  $X_{max}$  de chaque variable). Puis, définir le problème d'optimisation comme suit: Minimiser  $f(X)$ , où  $f(X)$  est la fonction d'objective.  $X$  est un vecteur pour les variables de conception telles que  $xmin_j \leq x_j \leq xmax_j$ .

**Étape 2:** La population  $X$  est initialisée de manière aléatoire par une matrice de  $N$  lignes et de  $D$  colonnes en utilisant l'équation:

$$x_{j,i} = xmin_j + rand(xmax_j - xmin_j) \quad (II.27)$$

Où,  $rand$  représente une variable aléatoire uniformément répartie dans l'intervalle  $(0, 1)$ ,  $xmin_j$  et  $xmax_j$  représentent la valeur minimale et maximale pour le paramètre  $x_j$ .

**Étape 3 :** Phase d'enseignant : l'apprenant ayant la valeur de la fonction objective minimale est considéré comme l'enseignant. Puis nous calculons la moyenne de chaque sujet, et la différence entre la moyenne actuelle et le résultat correspondant de l'enseignant. Pour obtenir un nouvel ensemble d'apprenants améliorés, nous utilisons l'équation (II.22).

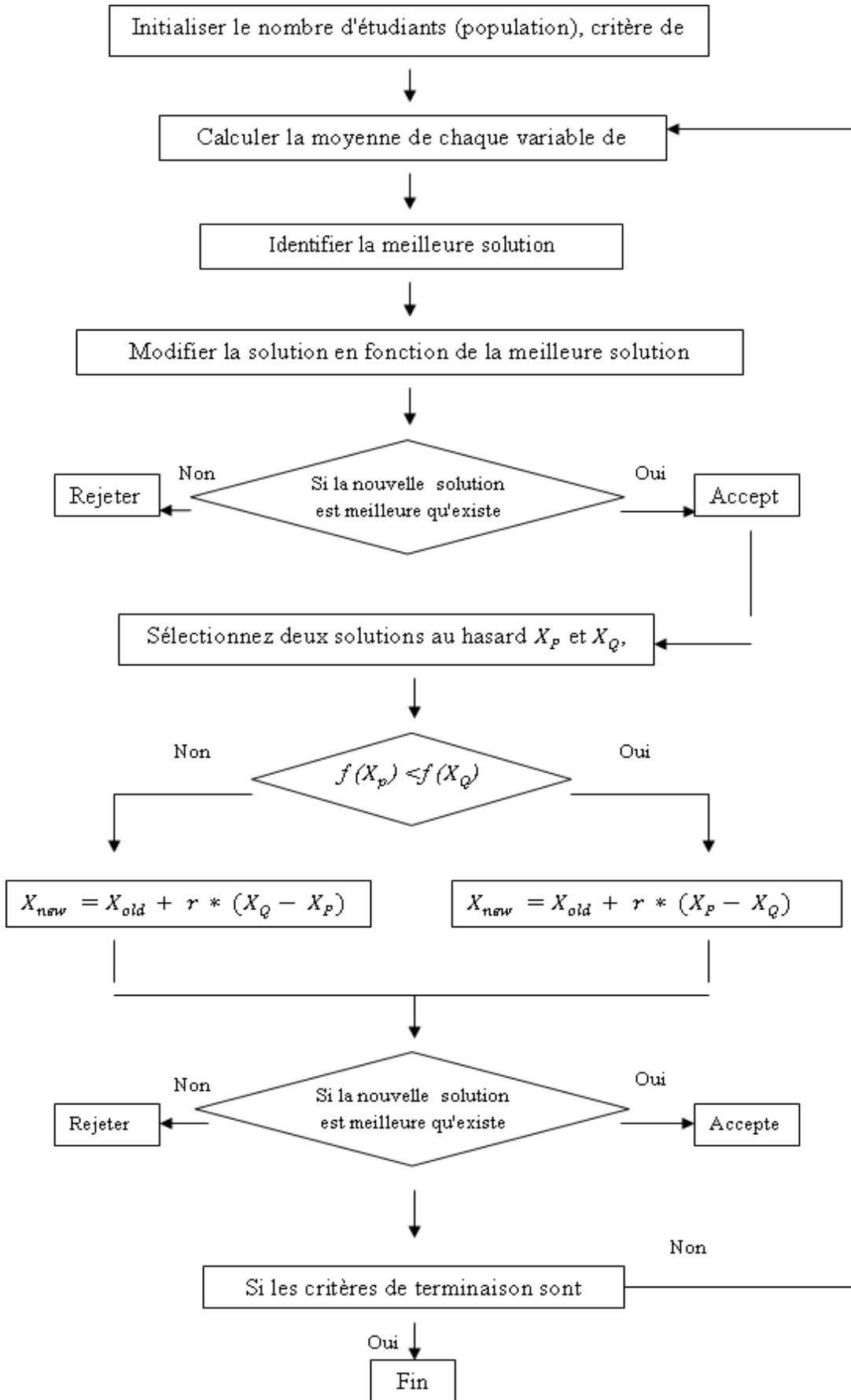


Figure II. 9 Organigramme de l'algorithme OBEA.

**Étape 4** : Phase apprenants: Comme expliqué ci-dessus, les apprenants augmentent leurs connaissances à l'aide de leur interaction mutuelle. L'expression mathématique est expliquée dans la [section V.2](#).

**Étape 5** : Arrêter si le nombre maximum de générations est atteint; sinon répéter à partir de l'étape 3

### **V.4 Algorithme OBEA Modifié**

Dans l'algorithme de base de l'OBEA, le résultat des apprenants est amélioré par un seul enseignant (par l'enseignement en classe) ou par l'interaction avec d'autres apprenants. En outre, le facteur d'enseignement dans l'algorithme OBEA de base est soit 2 ou 1, ce qui reflète deux circonstances extrêmes où un apprenant apprend tout ou rien de l'enseignant. Dans ce système, un enseignant doit faire plus d'efforts pour améliorer les résultats des apprenants. Au cours de l'optimisation, cette situation entraîne un taux de convergence plus lent du problème d'optimisation. Compte tenu de ce fait, pour améliorer les capacités d'exploration et d'exploitation, certaines améliorations ont été apportées à l'algorithme de base du TLBO [14-16].

#### ***V.4.1 Nombre d'enseignants***

Dans l'algorithme de base de l'OBEA, il n'y a qu'un enseignant qui enseigne aux apprenants et si la classe contient un plus grand nombre d'élèves, l'enseignant doit alors faire plus d'efforts pour améliorer les résultats des élèves; même si avec cet effort, il ne peut y avoir aucune amélioration apparente dans les résultats. Dans l'algorithme d'optimisation, ce résultat conduit à un nombre plus élevé d'évaluations de fonctions pour atteindre une solution optimum et donner un taux de convergence faible. Afin de surmonter ce problème, l'algorithme OBEA de base est amélioré en introduisant plus d'un enseignant pour les apprenants. Grâce à cette modification, la classe entière est divisée en différents groupes d'apprenants selon leur niveau (c'est-à-dire les résultats) et un enseignant individuel est affecté à un groupe individuel d'apprenants. Maintenant, chaque enseignant essaie d'améliorer les résultats de son groupe assigné et si le niveau (c'est-à-dire les résultats) du groupe atteint le niveau de l'enseignant assigné, alors ce groupe est assigné à un meilleur enseignant. Cette modification est expliquée dans la [section V.4.3](#). Le concept

de nombre d'enseignants consiste à effectuer le tri de la population au cours de l'optimisation et, ainsi, à éviter la convergence prématurée de l'algorithme.

### V.4.2 Facteur d'enseignement adaptatif

Une autre modification est liée au facteur d'enseignement (TF). Dans l'OBEA de base, la décision du facteur d'enseignement est une étape aléatoire et peut être 1 ou 2. Cette pratique est correspondante à une situation où les élèves n'apprennent rien de l'enseignant ou d'apprendre toutes les choses de l'enseignant, respectivement. Mais, dans un phénomène d'enseignement-apprentissage réel, cette fraction n'est pas toujours à son état final pour les apprenants mais varie également entre les deux, le facteur d'enseignement est modifié comme suit:

$$(TF)^k = \left( \frac{f(x_i)}{f(x_{enseignant})} \right)^k, \text{ if } f(x_{enseignant}) \neq 0 \quad (\text{II.28})$$

$$(TF)^k = 1, \text{ if } f(x_{enseignant}) = 0 \quad (\text{II.29})$$

Où  $f(x_i)$  est le résultat de n'importe quel apprenant,  $i$  ( $i = 1 \dots N$ ), considérant tous les sujets à l'itération,  $k$ , et  $f(x_{enseignant})$  est le résultat de l'enseignant à la même itération,  $k$ . Ainsi, dans l'algorithme OBEA modifié, le facteur d'apprentissage varie automatiquement lors de la recherche. Le réglage automatique de TF améliore les performances de l'algorithme.

### V.4.3 Apprentissage par Tutoriel (travaux pratiques)

Cette modification est basée sur le fait que les étudiants peuvent aussi apprendre en discutant avec leurs camarades de classe ou même avec l'enseignant pendant les heures de travaux pratiques tout en résolvant les problèmes et les devoirs. Puisque les étudiants peuvent augmenter leurs connaissances en discutant avec d'autres étudiants ou avec l'enseignant, nous intégrons ce mécanisme de recherche à la phase d'enseignement. L'expression mathématique de cette modification est donnée dans les étapes de l'algorithme (section V.4.4).

V.4.4 Procédure d'Algorithme

Les procédures de calcul de l'algorithme proposé peuvent être résumées comme suit:

**Étape 1:** Définir le problème d'optimisation comme Minimiser ou Maximiser  $f(X)$ , où  $f(X)$  est la valeur de la fonction objective et  $X$  est un vecteur des variables de conception.

**Étape 2:** Initialiser la population (c.-à-d. Apprenants,  $i = 1, 2, \dots, N$ ) et donner les variables du problème d'optimisation (c.-à-d., Nombre de sujets offertes aux apprenants,  $j = 1, 2, \dots, m$ ), et évaluez-les.

**Étape 3:** sélectionner la meilleure solution (c'est-à-dire la meilleure  $f(X)$ ) qui joue en tant qu'enseignant principal pour ce cycle. Attribuez-le au premier rang.

**Étape 4:** Sélectionner les autres enseignants ( $T_s, s = 2, 3, \dots, NT$ ) au hasard et les classer.  $NT$  est le nombre d'enseignants.

**Étape 5:** Attribuer les apprenants aux enseignants en fonction de leur valeur de la fonction objective comme:

Pour  $i = 1:N$   
Si  $f(X)_2 \geq f(X)_i > f(X)_1$ ,  
affecter l'apprenant  $f(X)_i$  à l'enseignant 1  
Sinon Si  $f(X)_3 \geq f(X)_i > f(X)_2$ ,  
affecter l'apprenant  $f(X)_i$  à l'enseignant 2  
.  
.  
.  
Sinon Si  $f(X)_T \geq f(X)_i > f(X)_{T-1}$ ,  
affecter l'apprenant  $f(X)_i$  à l'enseignant T -1.  
Sinon  
affecter l'apprenant  $f(X)_i$  à l'enseignant T  
Fin Pour.

(La procédure ci-dessus est pour un problème de minimisation, la procédure est inversée pour un problème de maximisation).

**Étape 6:** Calculer la moyenne de chaque groupe d'apprenants dans chaque sujet.

**Étape 7:** Pour chaque groupe, évaluer la différence entre la moyenne actuelle et le résultat correspondant de l'enseignant de ce groupe pour chaque sujet en utilisant le facteur d'enseignement adaptatif (donné par les équations (II.28) et (II.29))

**Étape 8:** Pour chaque groupe, mettre à jour les connaissances des apprenants à l'aide des connaissances des enseignants, en même temps que les connaissances acquises par les apprenants pendant les heures de travaux pratique, selon:

$$(X'_{j,i})_s = (X_{j,i} + DMi_j)_s + rand(X_i - X_{hh}) , Si f(x)_{hh} > f(x)_i \quad (II.30)$$

$$(X'_{j,i})_s = (X_{j,i} + DMi_j)_s + rand(X_{hh} - X_i) , Si f(x)_i > f(x)_{hh} \quad (II.31)$$

Où  $hh \neq i, j = 1 \dots D, i = 1 \dots N, et s = 1 \dots NT$

**Étape 9:** Pour chaque groupe, mettre à jour les connaissances des apprenants en utilisant les connaissances de certains autres apprenants, ainsi que par l'auto-apprentissage, selon:

$$\left\{ \begin{array}{l} (X''_{j,i})_s = X'_{j,i,k} + rand * (X'_{j,p} - X'_{j,i})_s + rand(X_{enseignant} - EF * X'_{j,i})_s, \text{ si } f(X'_i) > f(X'_p) \quad (II.32) \\ (X''_{j,i})_s = X'_{j,i,k} + rand * (X'_{j,i} - X'_{j,p})_s + rand(X_{enseignant} - EF * X'_{j,i})_s, \text{ si } f(X'_i) < f(X'_p) \quad (II.33) \end{array} \right.$$

Où  $EF = \text{facteur d'exploration} = \text{rond}(1 + rand)$ . (Les équations ci-dessus sont pour un problème de minimisation, l'inverse est vrai pour un problème de maximisation.)

**Étape 10:** Combiner tous les groupes.

**Étape 11:** Répéter la procédure des étapes 3 à 10 jusqu'à ce que le critère de terminaison soit respecté.

## **VI. Comparaison des Techniques d'Optimisation**

Les algorithmes de l'essaim intelligent ont été largement utilisés pour résoudre des problèmes d'optimisation complexes. Ces méthodes sont plus puissantes que les méthodes



conventionnelles basées sur la logique formelle ou la programmation mathématique. En termes de comparaison des algorithmes de l'essaim intelligent nous avons:

- \_ les quatre algorithmes étudiés dans ce chapitre sont des techniques basées sur la population qui mettent en œuvre un groupe de solutions pour procéder à la solution optimale.
- \_ Chaque méthode nécessite des paramètres qui affectent les performances de l'algorithme.
  - OEP nécessite des coefficients de confiance ( $c_1$ ,  $c_2$ ) et d'inertie  $\omega$
  - OAS nécessite le paramètre de seuil PF
  - OSA nécessite le taux de perturbation (Pr), leader local limite (LLL), leader global limite (LGL) et le nombre maximum de groupes (MG).
  - Contrairement, OBEA ne nécessite aucun paramètre, ce qui simplifie l'implémentation d'OBEA.
- \_ Les algorithmes OEP et OBEA utilisent la meilleure solution de l'itération pour changer la solution existante dans la population, ce qui augmente le taux de convergence.
- \_ OBEA et OEP ne divisent pas la population contrairement à OAS et OSA
- \_ OBEA, OAS et OSA implémentent la gourmandise pour accepter la bonne solution

### VII. Exemples d'application et Résultats

Dans cette section, la performance des algorithmes proposés est testée sur deux structures micro-ondes (Transformateur à deux paliers et filtre à iris inductifs à trois cavités) conçus par la méthode de raccordement modal.

A) *Transformateur à deux paliers ou Transition graduelle à trois discontinuités*: la Figure II.10 représente un transformateur d'impédance à transitions graduelles constitué de trois discontinuités entre guides circulaires concentriques uni-axiaux, les rayons du guide d'entrée et sortie sont fixés à  $R_0 = 11.165$  et  $R_3 = 13.4$ . La fréquence de travail  $f_0 = 9.5 \text{ GHz}$ , les paramètres à optimiser sont  $R_1$ ,  $R_2$  (rayons du guide interne) et  $l_1$ ,  $l_2$  (Longueur du guide interne) dans la bande de fréquence [8.5GHz – 10GHz].

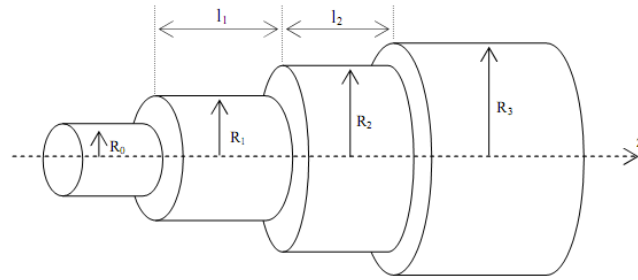


Figure II.10 Transformateur à deux paliers

Le but de l'optimisation sera de minimiser la fonction objective donnée par l'équation II.34 (le rapport des ondes stationnaires de tension, ROS équation II.35) près de la fréquence de conception (de diminuer la longueur totale du transformateur et d'augmenter sa bande passante). Les limites de chaque paramètre à optimiser sont représentées sur le Tableau II.1.

$$fitness = \min_{R_1, R_2, l_1, l_2} \left( \max_{f_1 < f < f_2} ROS(f) \right) \quad (II.34)$$

$$ROS = \frac{1 + |S_{11}|}{1 - |S_{11}|} \quad (II.35)$$

Tableau II.1 Limites des paramètres à optimiser

Paramètre	Min (mm)	Max (mm)
$R_1$	11.26	11.662
$R_2$	11.663	12.68
$l_1$	12.24	14.97
$l_2$	11.17	13.65

B) *filtre à iris inductif à trois cavités* Figure. II.11: Où le guide principal est le standard WR28 (A=7.112mm, B=3.556mm). Quatre paramètres doivent être optimisés  $W_1$ ,  $W_2$  (l'ouverture de l'iris) et  $l_1$ ,  $l_2$  (les distances entre l'iris). Les épaisseurs de l'iris sont fixées à  $t_1 = 1,45$  mm,  $t_2 = 1,1$  mm. Le Tableau II.2 sont recensées les variables géométriques et leurs gammes correspondantes. La fréquence retenue est dans la bande (34, 35.5 GHz).

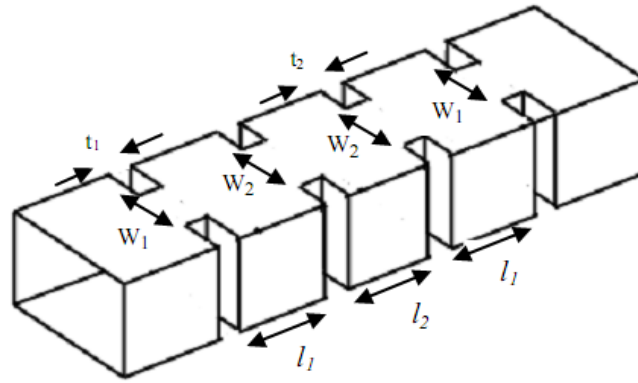


Figure II.11 Filtres à iris inductif à trois cavités

Tableau II.2 Limites des paramètres à optimiser

Paramètre	Min (mm)	Max (mm)
$W_1$	1.8	5.4
$W_2$	1.2	3.6
$l_1$	2.07	6.22
$l_2$	2.35	7.05

L'objectif est de minimiser la fonction objective (la valeur moyenne du coefficient de réflexion  $S_{11}$ ) dans la gamme de fréquences:

$$fitness = \frac{\sum_{f_1}^{f_2} S_{11}(f)}{PT} \quad (II.36)$$

Avec  $PT$  est le nombre de points dans l'intervalle  $[f_1, f_2]$ .

Pour les deux structures mentionnées ci-dessus, la convergence de la fonction d'objective de chaque algorithme (Meilleur, Pire, Moyenne) est présentée au [Tableau II.3](#) avec une taille de population 50 et le nombre maximum d'itérations prend les valeurs suivantes (30, 50, 300). Le [Tableau II.4](#) montre la convergence de la fonction objective pour un nombre d'itérations 50 et la taille de la population (30, 70 et 100). Chaque algorithme est exécuté 10 fois indépendamment. Les autres paramètres spécifiques des algorithmes sont donnés ci-dessous:

- OBEA Paramètres: pour OBEA il n'y a pas de constantes à définir.
- OEP Paramètres:  $c_1$  et  $c_2$  sont des constantes  $c_1 = c_2 = 2$ , l'inertie diminue linéairement de 0,9 à 0,2.
- OAS Paramètres: PF = 0,7.
- OSA Paramètres: LLL = 100, GLL = 50, MG = 5 et Pr diminué linéairement de 0,4 à 0,1(équation II.21).

**Tableau II.3** La fonction objective pour une population = 50

Maxit	Structure		Filtre			Transformateur		
	Algorithme	Meilleur	Pire	Moyenne	Meilleur	Pire	Moyenne	
30	OEP	0.0193	0.1588	0.0907	1.0509	1.0561	1.0531	
	OAS	0.0191	0.1405	0.0680	1.0498	1.0569	1.0522	
	OSA	0.0082	0.0100	0.0088	1.0494	1.0507	1.0499	
	OBEA	0.0082	0.0092	0.0088	1.0494	1.0499	1.0496	
50	OEP	0.0126	0.0945	0.0791	1.0498	1.0538	1.0512	
	OAS	0.0110	0.0353	0.0190	1.0495	1.0502	1.0498	
	OSA	0.0081	0.0089	0.0082	1.0494	1.0497	1.0495	
	OBEA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	
300	OEP	0.0084	0.0210	0.0108	1.0494	1.0495	1.0494	
	OAS	0.0081	0.0121	0.0091	1.0494	1.0494	1.0494	
	OSA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	
	OBEA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	

**Tableau II.4** La fonction objective pour une Maxit = 50

Population	Structure		Filtre			Transformateur		
	Algorithme	Meilleur	Pire	Moyenne	Meilleur	Pire	Moyenne	
30	OEP	0.0197	0.0849	0.0411	1.0496	1.0518	1.0503	
	OAS	0.0120	0.0536	0.0409	1.0497	1.0508	1.0503	
	OSA	0.0081	0.0140	0.0089	1.0494	1.0495	1.0494	
	OBEA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	
70	OEP	0.0126	0.0634	0.0326	1.0503	1.0513	1.0508	
	OAS	0.0119	0.0501	0.0272	1.0495	1.0519	1.0503	
	OSA	0.0081	0.0082	0.0081	1.0494	1.0494	1.0494	
	OBEA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	
100	OEP	0.0120	0.0352	0.0255	1.0497	1.0500	1.0498	
	OAS	0.0103	0.0311	0.0162	1.0494	1.0499	1.0496	
	OSA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	
	OBEA	0.0081	0.0081	0.0081	1.0494	1.0494	1.0494	

La Figure II.12 montre la convergence de la fonction fitness du meilleur individu de chaque algorithme pour les deux structures.

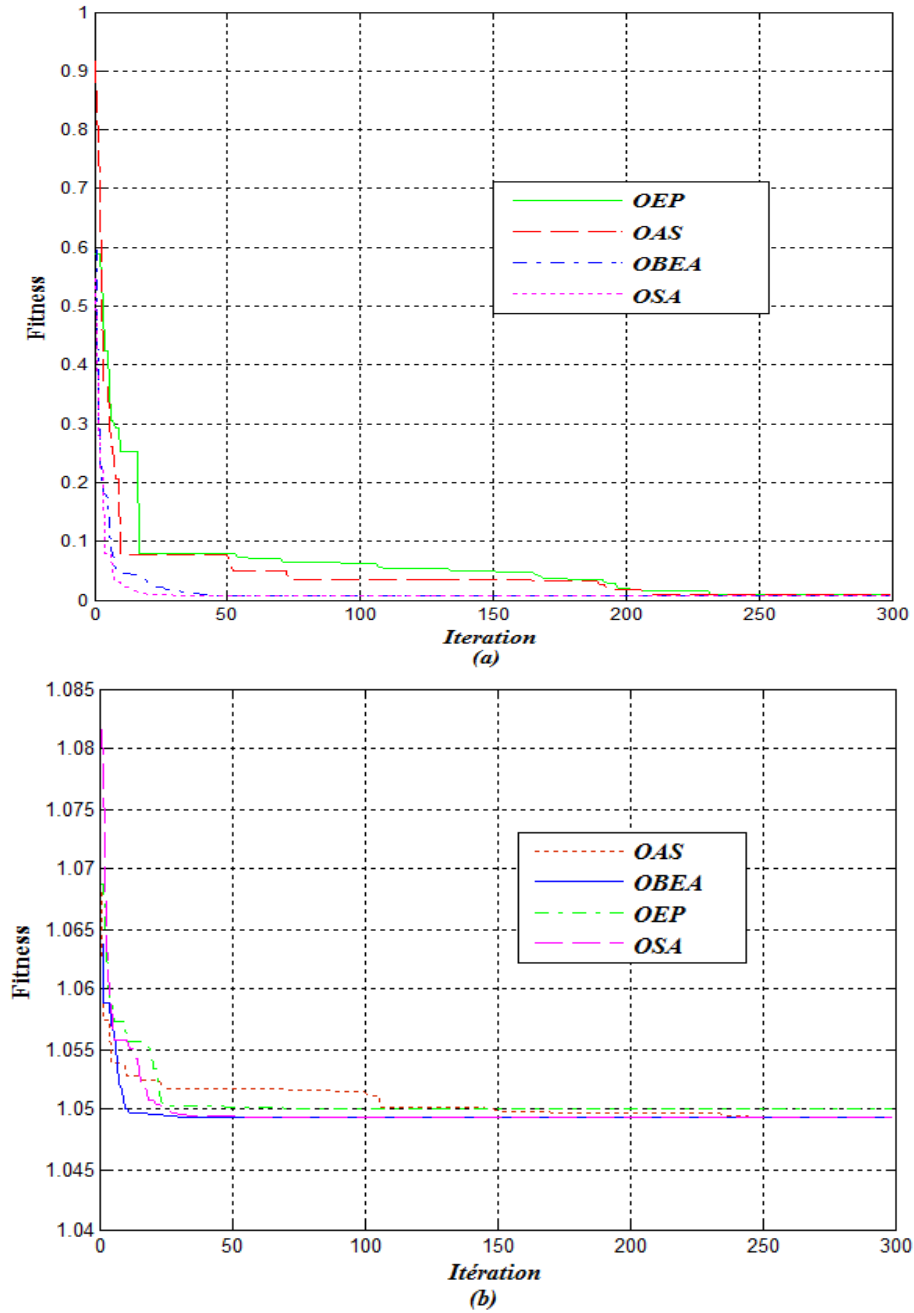


Figure II.12 La convergence de la fonction objective. (a) Filtre, (b) Transformateur

On observe à partir des Tableaux II.3, et II.4 et de la Figure II. 12 que les algorithmes OSA et OBEA sont plus performants en termes de convergence que les algorithmes OAS et OEP. Les algorithmes OSA et OBEA convergent vers le minimum optimal pour les premières itérations  $\text{maxit} = 30$ , alors que les algorithmes OAS et OEP convergent vers le minimum optimal en 300 itérations.

Les résultats d'optimisation utilisant les algorithmes (OEP, OAS, OSA et OBEA) sont présentés dans le Tableau II.5 et la figure II.13. Les résultats sont proches, ce qui prouve l'efficacité des algorithmes.

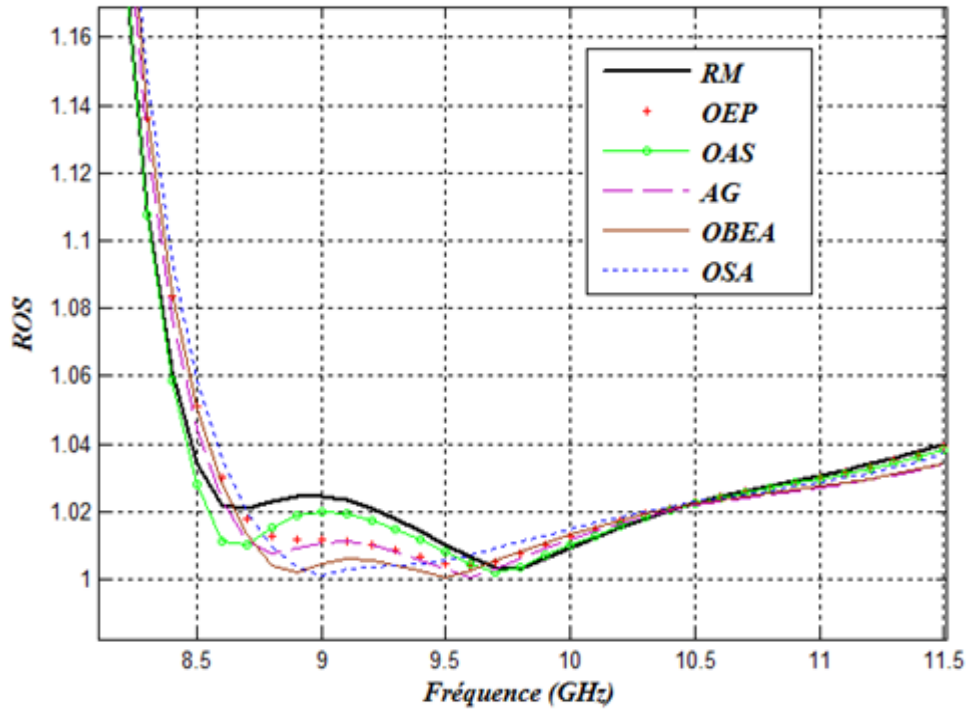


Figure II.13. Le ROS en fonction de la fréquence pour le transformateur.

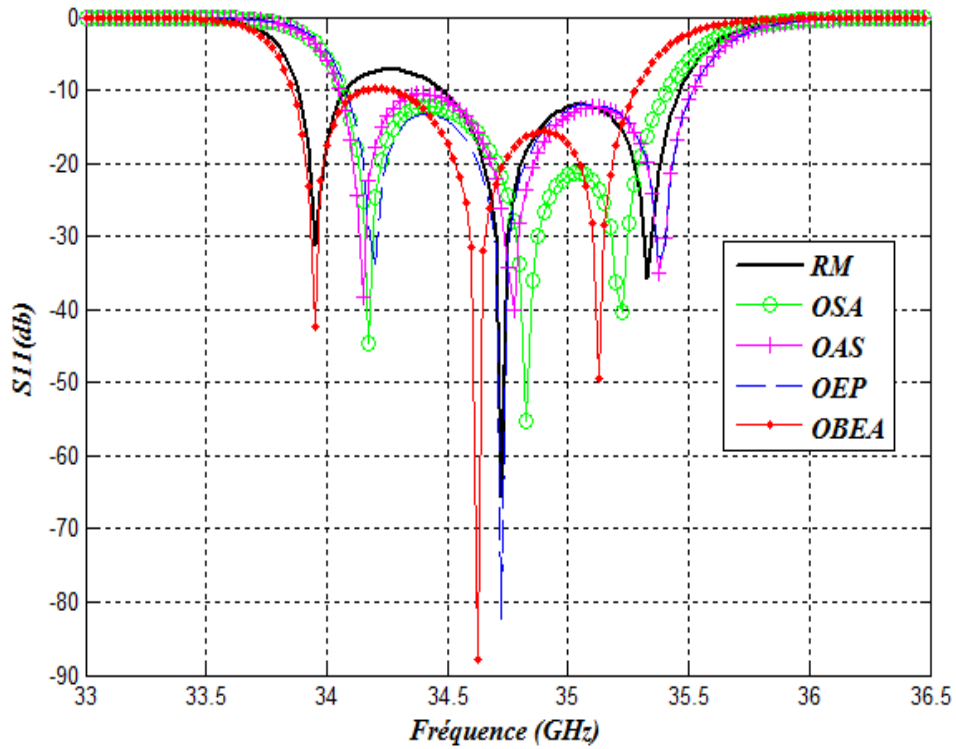


Figure II.14. Le module du coefficient de réflexion S11 en fonction de la fréquence.

**Tableau II.5** Les paramètres géométriques optimisés

Transformateur	Paramètre	Référence [17]	AG [17]	OEP	OAS	OSA	OBEA
	R <sub>1</sub> (mm)	11.36	11.424	11.34	11.38	11.35	11.40
	R <sub>2</sub> (mm)	11.956	12.172	11.99	12.02	12.08	12.50
	l <sub>1</sub> (mm)	13.61	13.462	13.60	13.58	13.55	13.62
	l <sub>2</sub> (mm)	12.41	12.152	12.38	12.40	12.44	12.01
Filtre	Paramètre	Référence [18]	OEP	OAS	OSA	OBEA	
	W <sub>1</sub> (mm)	3.6	3.633	3.613	3.670	3.640	
	W <sub>2</sub> (mm)	2.4	2.345	2.350	2.330	2.356	
	l <sub>1</sub> (mm)	4.15	4.130	4.138	4.110	4.168	
	l <sub>2</sub> (mm)	4.7	4.689	4.694	4.720	4.750	

### VIII. Conclusion

Dans ce chapitre, nous présentons l'application des trois algorithmes récents d'essaims d'intelligents dans la littérature (optimisation par les singes-araignées (OSA), optimisation par araignées sociales (OAS) et optimisation à base d'enseignement-apprentissage (OBEA). Ces trois algorithmes sont utilisés pour l'optimisation des structures micro-ondes (Transformateur à deux paliers et filtres à iris inductif à trois cavités). Les résultats obtenus sont comparés à ceux de l'algorithme d'essaim intelligent le plus populaire (l'optimisation par essaim de particules, OEP). Les résultats ont montré la validation des algorithmes proposés.

- [1] B. K. Panigrahi, Y. Shi, and M. H. Lim, "*Handbook of Swarm Intelligence. Series: Adaptation, Learning, and Optimization*", Springer-Verlag, Berlin Heidelberg, Chapter 23, PP. 517-542, **2010**.
- [2] C. Blum and D. Merkle, "*Handbook on Computational Intelligence, Swarm Intelligence – Introduction and Applications*", World Scientific, Chapter 16, PP. 587-606, **2016**.
- [3] J. Kennedy, and R. C. Eberhart, "*Particle Swarm Optimization*", In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, PP. 1942–1948, **1995**.
- [4] N. Singh, and S. B. Singh, "*A New Modified Approach of Mean Particle Swarm Optimization Algorithm*", IEEE 5th International Conference on Computational Intelligence and Communication Networks, Mathura, India, PP. 296–300, **2013**.
- [5] E. Cuevas, M. Cienfuegos, D. Zaldívar and M. P. Cisneros, "*A swarm optimization algorithm inspired in the behavior of the social-spider*", Expert Systems with Applications, Vol. 40, PP. 6374-6384, **2013**
- [6] E. Cuevas, and M. Cienfuegos, "*A new algorithm inspired in the behavior of the social-spider for constrained optimization*", Expert Systems with Applications, Vol. 41, PP. 412-425, **2014**
- [7] S. Ouadfel, and A. A. Taleb, "*Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study*", Expert Systems with Applications, Vol. 55, PP. 566-584, **2016**.
- [8] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "*Spider Monkey Optimization Algorithm for Numerical Optimization*", Memetic Computing, Vol. 6 (1), PP. 31-47, **2014**.
- [9] K. Gupta, K. Deep, and J. C. Bansal, "*Spider monkey optimization algorithm for constrained optimization problems*", Vol. 21(23), PP. 6933–6962, **2017**.
- [10] M. M. F. Symington, "*Fission-Fusion Social Organization in Ateles and Pan*", International Journal of Primatology, Vol. 11 (1), PP. 47-61. **1990**
- [11] R.V. Rao, V.J. Savsani, and D.P. Vakharia, "*Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems*", Computer-Aided Design. Vol 43 (3), PP. 303–315, **2011**.
- [12] R.V. Rao, V.J. Savsani, and D.P. Vakharia, "*Teaching-learning-based optimization: A novel optimization method for continuous non-linear large scale problems*", Information Sciences. Vol 183 (1), PP. 1–15, **2012**.
- [13] R.V. Rao, V.J. Savsani, and J. Balic, "*Teaching-learning-based optimization algorithm for unconstrained and constrained real parameter optimization problems*", Engineering Optimization. Vol. 44 (12), PP. 1447–1462, **2012**
- [14] R.V. Rao, and V. Patel, "*An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems*", Scientia Iranica D , Vol.20 (3), PP. 710–720 , **2013**
- [15] R.V. Rao, and V. Patel, "*Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm*", Engineering Application of Artificial Intelligence, Vol 26 (1), PP. 430–445, **2013**.
- [16] R.V. Rao, and V. Patel, "*Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm*", Applied Mathematical Modeling, Vol. 37 (3), PP. 1147–1162, **2013**.
- [17] R. Thabet, M.L. Riabi, and M. Belmeguenai, "*Rigorous design and efficient optimization of quarter-wave transformers in metallic circular waveguides using the mode-matching method and genetic algorithm*", Progress in Electromagnetic Research, Vol.68, PP. 15–33, **2007**.
- [18] R. Yang and A.S. Omar, "*Investigation of Multiple Rectangular Aperture Irises in Rectangular Waveguide Using TE Modes*", IEEE Transactions On Microwave Theory And Techniques, Vol. 41, PP. 1369–1374, **1993**.

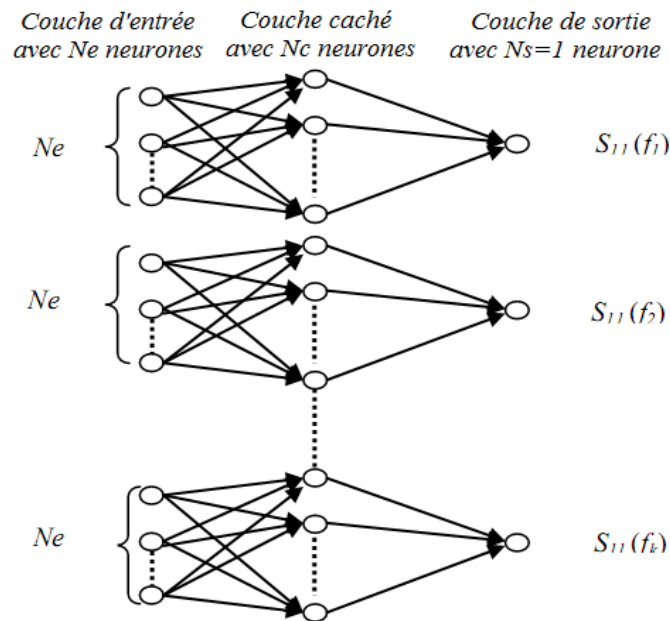


*Chapitre III*  
*Modélisation des Structures Micro-ondes*  
*Utilisant les Réseaux de Neurones*

---

## I. Introduction

Au cours des dernières années, les réseaux neuronaux (RN) feedforward, ont été largement utilisés pour la modélisation des dispositifs micro-ondes complexes [1-3]. Dans ce chapitre, nous proposons d'étudier trois filtres micro-ondes (pseudo-elliptique filtre, filtre plan-E large bande et Filtre plan-H avec cavités couplées directement) en utilisant un réseau de neurones à trois couches Figure III.1. Où chaque sous-réseau possède,  $N_e$  neurones d'entrée correspondant au nombre de paramètres géométrique de la structure,  $N_c$  neurones dans la couche cachée et la sortie est associée à la valeur du coefficient de réflexion  $S_{11}(f_k)$   $N_s=1$ . L'ensemble du réseau est constitué de  $k$ -RN distincts correspondants à une fréquence donnée, où  $k$  déterminé par le nombre approximatif de points dans l'intervalle de fréquence. Les poids de connexion des neurones de la couche d'entrée aux neurones de la couche cachée sont  $WE$  et les poids de connexion des neurones de la couche cachée aux neurones de la couche de sortie sont  $WS$ .



**Figure III.1:** Architecture de réseau de neurones

La première étape dans la modélisation des structures utilisant le RN est la formation (apprentissage) du réseau. L'algorithme le plus utilisé pour former le RN est la l'algorithme conventionnel de rétro-propagation (Back propagation) basée sur le gradient descendant [4]. Dernièrement, de nombreux algorithmes évolutionnaires basés sur la population ont été proposés pour la formation des réseaux de neurones, tels que l'optimisation par essais de particules (OEP) [5], les algorithmes génétiques(AG) [6] et autres [7].

Dans ce chapitre, nous avons d'abord formé le RN par trois algorithmes d'optimisation évolutionnaires récents (optimisation à base d'enseignement apprentissage OBEA, optimisation par les singes araignées OSA et optimisation par les araignées sociales OAS). Puis, nous avons proposé de nouveaux algorithmes hybrides basés sur la combinaison entre l'algorithme d'optimisation conventionnelle (gradient décent, GD) et les algorithmes d'optimisation évolutionnaire mentionné précédemment.

## II. Formation (Apprentissage) des Réseaux de Neurones

Les réseaux de neurones artificiels (RNAs) sont des familles de fonctions non linéaires modulaires, capables d'approximer n'importe quelle fonction bornée dans un intervalle fermé de ses entrées, aussi complexe soit-elle. La plupart des algorithmes d'apprentissage des réseaux de neurones sont des algorithmes d'optimisation : Ils cherchent à optimiser les poids des réseaux afin de minimiser l'erreur quadratique moyenne EQM (Mean Square Error) équation (III.1) entre les réponses réelles du réseau et ses réponses désirées. Cette optimisation se fait de manière itérative, Les poids sont aléatoirement initialisés avant l'apprentissage, ils sont ensuite ajustés de manière itérative jusqu'à atteindre une résolution précise.

$$EQM = \frac{1}{PT} \sum_{PT} \sum_{Ns} (Y_s - Y)^2 \quad (III.1)$$

Avec  $PT$  est le nombre total d'échantillons d'apprentissage,  $Y_s$  est la sortie du réseau et  $Y$  est la sortie désirée.

$$Y_s = f_2 \left( \sum_{Nc} WS * f_1 \left( \sum_{Ne} WE * X \right) \right) \quad (III.2)$$

$f_1$  et  $f_2$  sont les fonctions d'activation (typiquement: sigmoïde, tanh ...),  $X$  est le vecteur d'entrée de RN.

### II.1 Apprentissage par Rétro-propagation du Gradient

L'algorithme de rétropropagation (ARP) ou de propagation arrière (Back-propagation en anglais) est l'exemple d'apprentissage supervisé le plus utilisé. Dans l'algorithme de rétro-propagation du gradient [8] l'ajustement des coefficients se fait en commençant par

les poids de la couche de sortie et en terminant par les poids de la couche d'entrée, d'où le nom de rétro-propagation, où chaque nouveau poids est déterminé en fonction de l'ancien;

$$W(new) = W(old) - \mu \frac{\partial EQM}{\partial W(old)} \quad (III.3)$$

$\mu$  étant le coefficient d'apprentissage (plus  $\mu$  est grand, plus la modification des poids se fera rapidement), une fois que tous les poids ont été modifiés, on représente à nouveau le vecteur  $X$  en entrée et on recalcule le vecteur en sortie du réseau. Si l'EQM est toujours considérée trop élevée, on réitère la procédure d'ajustement précédemment décrite, jusqu'à ce qu'EQM respecte le critère de précision fixé.

Le principe de rétropropagation peut être décrit en trois étapes fondamentales:

- \_ routage de l'information à travers le réseau;
- \_ rétropropagation des sensibilités et calcul du gradient;
- \_ ajustement des paramètres par la règle du gradient approximé.

Il est important de noter que l'algorithme de RP souffre des limitations inhérentes à la technique du gradient à cause du risque d'être piégé dans un minimum local, et la convergence lente surtout lorsqu'il s'agit des réseaux de grands taille (c'est à dire pour lesquels le nombre de poids de connexion à déterminer est important).

## II.2 Apprentissage par les Algorithmes des Essaims Intelligents

Dans cette partie, Nous avons présenté l'implémentation d'apprentissage de RN utilisant les algorithmes des essaims intelligents étudiés au [chapitre II](#). Les deux premières étapes d'apprentissage de chaque algorithme sont les mêmes.

- Dans la première étape nous déterminons:
  - \_ L'architecture du réseau (nombre de couches, nombre de neurones dans chaque couche et les fonctions d'activation)
  - \_ Les paramètres d'optimisation: taille de population  $NP$ , nombre maximum d'itérations *maxit* et les variables à optimiser ( $WE$  matrice de  $Nc$  lignes et  $Ne$  colonnes et  $WS$  matrice de  $Ns$  lignes et  $Nc$  colonnes)

- \_ Le problème d'optimisation (fonction objective): trouver les **WE** et **WS** optimaux qui minimisent l'erreur quadratique moyenne (**EQM**)
- la deuxième étape: Initialiser la population et évaluer la valeur de la fonction objective correspondante. L'initialisation de population est effectuée de manière aléatoire, en fonction de la taille de population et du nombre de neurones, Cette population est exprimée comme:

$$Popu = [WE_1, WE_2, \dots, WE_{NP}; WS_1, WS_2, \dots, WS_{NP}] \quad (III.4)$$

Une nouvelle stratégie de préparation de la population a été proposée dans cette thèse, où la population est décomposée en deux groupes l'un représente la population de poids d'entrées **WE<sub>p</sub>** et l'autre représente la population de poids de sortie **WS<sub>p</sub>**. Le pseudo-code représenté dans l'Algorithme III.1 illustre cette étape:

---

**Algorithme III.1:** Initialisation et évaluation de la population

---

```

% Initialiser la population
    WEp = rand * (NP * Nc, Ne)
    WSp = rand * (Ns, NP * Nc)
% Evaluation
    Pour i=1: NP
        WE = WEp(Nc * (i - 1) + 1: Nc * i, :)
        WS = WSp(:, Nc * (i - 1) + 1: Nc * i)
        • calculer les sorties Ys et Y pour chaque vecteur
          d'entrée de réseau de neurones
        • calculer EQM
    Fin

```

---

Les autres étapes d'implémentation de chaque algorithme sont représentées dans les sections suivantes.

### II.2.1 Implémentation d'optimisation par Essaim de Particule

Pour former le RN par l'algorithme OEP, nous avons les étapes suivantes:

**Étape 1.** Déterminer les paramètres de OEP (les constantes  $c_1$ ,  $c_2$  et l'inertie  $\omega$  équation II.4) et générée les vitesses initiale **Vel<sub>1</sub>** pour la population **WE<sub>p</sub>** et **Vel<sub>2</sub>** pour la population **WS<sub>p</sub>**.

$$Vel_1 = rand * (NP * Nc, Ne) \quad (III.5)$$

$$Vel_2 = rand * (Ns, NP * Nc) \quad (III.6)$$

**Étape 2.** Une fois la fonction objective évaluée (Algorithme III.1), déterminer la meilleure position de la population (*globalpar1*, *globalpar2*). Les autres positions de particules sont enregistrées sous les noms *localpar1* et *localpar2*.

**Étape 3.** Les positions et les vitesses de toutes les particules sont mises à jour selon les équations

$$Vel_1(t+1) = \omega * Vel_1(t) + rand * c_1(localpar1 - WEp) + rand * c_2(globalpar1 - WEp) \quad (III.7)$$

$$WE_p(t+1) = WE_p(t) + Vel_1(t+1) \quad (III.8)$$

$$Vel_2(t+1) = \omega * Vel_2(t) + rand * c_1(localpar2 - WSp) + rand * c_2(globalpar2 - WSp) \quad (III.9)$$

$$WS_p(t+1) = WS_p(t) + Vel_2(t+1) \quad (III.10)$$

**Étape 4.** Trouver la meilleure position pour chaque particule *localpar1<sub>i</sub>*, *localpar2<sub>i</sub>* et la meilleure position globale *globalpar1*, *globalpar2*

**Étape 5.** Répéter la procédure de l'étape 3 jusqu'à ce que les générations itératives soient terminées.

### II.2.2. Implémentation d'optimisation par les araignées sociales

La procédure par étapes pour la mise en œuvre du OAS-RN peut être résumée comme suit:

**Étape 1.** Définir les paramètres de l'algorithme OAS (nombre de femelles *N<sub>f</sub>*, nombre de mâles *N<sub>m</sub>*, et le seuil *PF*).

**Étape 2.** Localiser la meilleure et la pire fonction d'objective, puis calculer le poids de chaque araignée en termes de son fonction objective (équation II.7)

**Étape 3.** Calculer les vibrations *Vib<sub>c</sub>* et *Vib<sub>b</sub>* (équations II.10 et II.11) et déplacer les araignées femelles selon les équations de l'opération coopératif femelles (équation II.13) pour les deux populations WEp et WSp.

**Étape 4.** Calculer la vibration *Vib<sub>f</sub>* (équation II.12), puis déplacer les araignées mâles selon les équations de l'opération coopératif mâles (équation II.14) pour les deux populations WEp et WSp.

**Étape 5.** Effectuer l'opération d'accouplement (section III.2.3, chapitre II).

**Étape 6.** Si le critère d'arrêt atteint, le processus est terminé; sinon, revenir à l'étape 2.

### II.2.3 Implémentation d'optimisation par singes araignées

La procédure par étapes pour l'implémentation de (OSA-NN) est donnée dans cette section.

**Étape 1.** Initialiser les paramètres de contrôle nécessaires: taux de perturbation (Pr), limite

de leader local (LLL), limite de leader global (LLG) et nombre maximum de groupes (MG)

**Étape 2.** Localiser les leaders global et local

**Étape 3.** La phase de leader local commence par la mise à jour de la position de tous les membres du groupe en fonction du leader local et des membres locaux du groupe (Algorithme II.1; chapitre II). Toutes les valeurs acceptées à la fin de cette phase deviennent l'entrée de la phase de leader global.

**Étape 4.** Produire de nouvelles positions pour tous les membres du groupe, sélectionnés par probabilité (*Pro*, équation II.19), en utilisant l'expérience de soi, l'expérience du leader global et les expériences des membres du groupe (Algorithme II.2; chapitre II).

**Étape 5.** Mettre à jour la position des leaders locaux et du leader global.

**Étape 6.** S'il ya un leader local ne met pas à jour sa position après un nombre prédéterminé appelé Limite Leader Local, tous les membres de ce groupe mettent à jour leurs positions soit par une initialisation aléatoire, soit en utilisant des informations combinées de leader global et leader local (Algorithme II.3; chapitre II)

**Étape 7.** Si le leader global ne met pas à jour sa position pour un certain nombre prédéterminé appelé Limite Leader Global, le leader global divise la population en petits groupes (Algorithme II.4; chapitre II).

**Étape 8.** Répétez la procédure de l'étape 2 jusqu'à ce que le critère de terminaison soit satisfait.

#### ***II.2.4 Implémentation d'optimisation à base d'enseignement- apprentissage***

La procédure par étapes pour la mise en œuvre de (OBEA-RN) est donnée dans cette section.

**Étape 1.** Déterminer le nombre d'enseignants NT,

**Étape 2.** Sélectionner la meilleure solution qui agit comme enseignant principal pour ce cycle, et sélectionner les autres enseignants au hasard.

**Étape 3.** Arranger les enseignants en fonction de leur valeur de fitness (fonction objective), puis attribue les apprenants aux enseignants (Algorithme II.5, Chapitre II)

**Étape 4.** Calculer le résultat moyen de chaque groupe d'apprenants dans chaque variable de conception: MWE, MWS (Algorithme III.2)

Algorithme III.2: Calculer MWE, MWS

Pour i=1: Ne Pour j=1: Nc MWE (j, i)= 0; Pour L=j : Nc : NP*Nc MWE (j, i)= MWE (j, i)+WEp (L, i) Fin Fin Fin MWE= MWE / NP	Pour j=1: Nc Pour k=1: Ns MWS (k, j)= 0; Pour L=j : Nc : NP*Nc MWS (k, j)= MWS (k, j)+WSp (k, L) Fin Fin Fin MWS= MWS / NP
--	--

**Étape 5.** Pour chaque groupe, évaluer la différence entre la moyenne actuelle et le résultat correspondant de l'enseignant de ce groupe de chaque variable de conception en utilisant le facteur adaptatif d'enseignement (section V.4.2, chapitre II)

**Étape 6.** Pour chaque groupe, mettre à jour la solution en utilisant les équations de la phase d'enseignant (II.22) (II.23) et les équations de la phase d'apprentissage (II.25) (II.26).

**Étape 7.** Combiner tous les groupes.

**Étape 8.** Répéter la procédure de l'étape 2 jusqu'à ce que le critère de terminaison soit satisfait.

### II.3 Apprentissage par l'Algorithme Hybride Rétropropagation - Essaim Intelligent

Dans cette section, la méthode proposée a été décrite. Elle est basée sur une nouvelle combinaison d'algorithmes d'essaims intelligents et l'algorithme de rétropropagation pour obtenir un algorithme d'apprentissage hybride qui forme des RNs. l'algorithme de RP est un algorithme basé sur les méthodes d'optimisation conventionnelles du gradient; certains problèmes existent dans ces algorithmes:

- \_ l'algorithme de RP sera facilement piégé dans les minimums locaux, de sorte que l'algorithme de RP peut conduire à l'échec de la recherche d'une solution globale.
- \_ la vitesse de convergence de l'algorithme de RP est trop lente. Le problème est que le comportement convergent de l'algorithme RP dépend des choix des valeurs initiales des poids de RN.



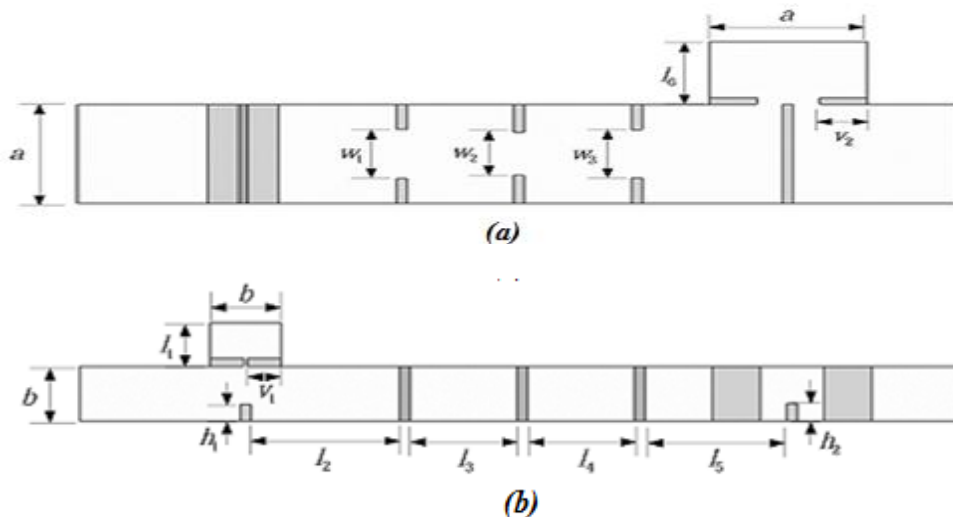
De manière générale, les algorithmes d'essaim intelligent ont une forte capacité de trouver le résultat optimum. Cependant, comme les algorithmes d'essaim intelligent ont plusieurs paramètres à ajuster par une approche empirique, si ces derniers ne sont pas correctement définis, la recherche deviendra très lente près de l'optimum global.

Pour l'algorithme hybride proposé, l'algorithme évolutionnaire a été utilisé pour analyser l'espace de recherche dans un premier temps, puis l'algorithme conventionnel est utilisé, de sorte que l'essaim obtenu est pris comme estimation initiale. Cette méthode permet d'utiliser séparément les capacités des deux algorithmes. Par contre, lorsque nous les utilisons séparément, nous ne pouvons pas acquérir leurs capacités simultanément. Dans la méthode proposée, à chaque itération après avoir utilisé l'algorithme évolutionnaire, l'essaim doit être utilisé dans l'algorithme conventionnel.

### III. Exemples d'application

Dans cette partie, le réseau de neurones formés a été appliqué à la modélisation des structures de guides d'ondes décrites ci-dessous.

➤ **Filtre Pseudo-Elliptique:** sur la [Figure III.2](#) les coupes d'un filtre pseudo-elliptique sont représentés. Il est composé de cavités multicellulaires plan-E et plan-H. Les dimensions de ce filtre [9] sont consignées dans le [Tableau III.1](#).

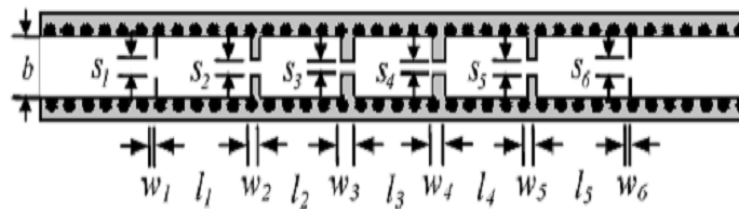


**Figure III.2.** Pseudo-elliptique filtre : (a) Vue de dessus, (b) Vue de côté en coupe transversale.

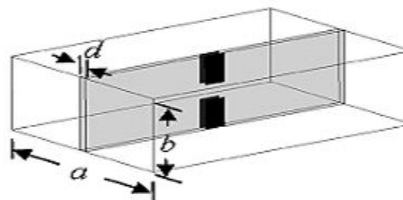
**Tableau III.1.** Dimensions du filtre conçu (unités: millimètres).

<b>a</b>	<b>b</b>	<b>l<sub>1</sub></b>	<b>l<sub>2</sub></b>	<b>l<sub>3</sub></b>
22.86	10.16	3.18	21.43	15.39
<b>l<sub>4</sub></b>	<b>l<sub>5</sub></b>	<b>l<sub>6</sub></b>	<b>V<sub>1</sub></b>	<b>V<sub>2</sub></b>
15.56	20.43	14.31	4.78	6.86
<b>h<sub>1</sub></b>	<b>h<sub>2</sub></b>	<b>W<sub>1</sub></b>	<b>W<sub>2</sub></b>	<b>W<sub>3</sub></b>
3.10	3.28	11.37	9.93	11.04

➤ **Filtres plan-E à large bande:** La conception du filtre plan-E à large bande est illustrée à la Figure III.3. Différent du filtre plan-E passe-bande traditionnel, les structures de poteaux à ailettes de la Figure III.4 sont utilisées comme éléments de couplage dans ce filtre. Les dimensions de cet issue [10] sont présentées dans le Tableau III.2.



**Figure III.3:** Les conceptions du filtre plan-E à large bande.



**Figure III.4:** structures de poteaux à ailettes

**Tableau III. 2.** Dimensions du filtre (unités: millimètres).

<b>a</b>	<b>b</b>	<b>W<sub>1</sub></b>	<b>W<sub>2</sub></b>	<b>W<sub>3</sub></b>
7.112	3.556	0.254	0.889	1.219
<b>W<sub>4</sub></b>	<b>W<sub>5</sub></b>	<b>W<sub>6</sub></b>	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>
W3	W2	W1	1.092	0.610
<b>S<sub>3</sub></b>	<b>S<sub>4</sub></b>	<b>S<sub>5</sub></b>	<b>S<sub>6</sub></b>	<b>l<sub>1</sub></b>
0.508	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	7.696
<b>l<sub>2</sub></b>	<b>l<sub>3</sub></b>	<b>l<sub>4</sub></b>	<b>l<sub>5</sub></b>	<b>d</b>
6.452	6.706	l2	l1	0.254

➤ Filtre plan-H avec cavités directement couplées dans un guide d'ondes rectangulaire Figure III.5, en considérant les coins arrondis. Ces coins sont typiquement présents dans les fenêtres ou dans les cavités des filtres avec des cavités couplées en raison de défauts dans les processus de fabrication les plus courants. Les dimensions du filtre [11] sont énumérées dans le Tableau III.3.

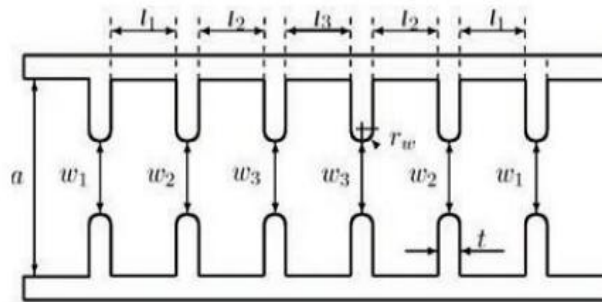


Figure III.5: Filtre plan-H considérant des coins arrondis

Tableau III.3: Dimensions du filtre conçu (unités: millimètres).

<b>a</b>	<b>b</b>	<b>t</b>
34.85	15.79	1.88
<b>l<sub>1</sub></b>	<b>l<sub>2</sub></b>	<b>l<sub>3</sub></b>
22.88	25.33	25.57
<b>w<sub>1</sub></b>	<b>w<sub>2</sub></b>	<b>w<sub>3</sub></b>
15.10	9.03	7.98

#### IV. Résultats et Discussion

Dans cette section, pour modéliser toutes les structures mentionnées ci-dessus nous proposons une architecture de RN multicouche avec une seule couche cachée. Ce réseau est capable d'approximer n'importe quelle relation d'entrée-sortie non linéaire à un degré de précision arbitraire, à condition qu'un nombre suffisant de neurones de couche cachée soit utilisé [12,13]. Les couches d'entrée et de sortie sont déterminées par les structures à modéliser: La couche d'entrée a un nombre de neurones égal au paramètre géométrique de la structure. La couche de sortie correspondant au paramètre physique  $S_{11}(f)$  dans l'intervalle de fréquence proposé. Le choix du nombre de neurones cachés est fortement lié à la nature de la non-linéarité à modéliser. Dans nos exemples d'application, le Tableau III.4 montre les paramètres du réseau de neurones ainsi que le nombre de neurones cachés avec une bonne convergence de l'algorithme et une bonne précision du modèle neuronal

formé. Les fonctions d'activation sont la fonction tangente hyperbolique (Tansig) et la fonction linéaire (Purelin) respectivement. Les autres paramètres spécifiques des algorithmes sont:

- \_ Paramètres OEP:  $c_1$  et  $c_2$  sont des constants  $c_1 = c_2 = 2$ , l'inertie diminue linéairement de 0,9 à 0,2
- \_ Paramètres OAS: le paramètre de seuil PF = 0.7
- \_ Paramètres OBEA: pour OBEA il n'y a pas des constantes à définir
- \_ Paramètres GD: le coefficient d'apprentissage  $\mu=0.8$
- \_ Paramètres OSA: le paramètre de OSA dépend de la taille de la population (section IV.7 chapitre II)

**Tableau III. 4.** Paramètres du réseau de neurones.

Structure	1	2	3
<b>Neurones d'entrée et son intervalle (mm)</b>	9.66<W1<13.07 8.44<W2<11.42 9.38<W3<12.70 4.10<V1<5.50 5.83<V2<7.89	0.20<W1<0.30 0.71<W2<1.07 0.97<W3<1.46 0.87<S1<1.31 0.48<S2<0.74 0.41<S3<0.61	13.95<W1<16.61 8.13<W2<9.93 7.18<W3<8.78
<b>Neurones caché</b>	8	10	6
<b>Neurones de sortie</b>	Abs (S11(f))	Abs (S11(f))	Abs (S11(f))
<b>Intervalle de fréquence (GHz)</b>	[8-12]	[26-36]	[6.6 – 7.4]
<b>K</b>	41	34	54

Les résultats obtenus pour les structures étudiées sont présentées .Dans ce qui suit:

1. la première partie, nous représentons l'influence d'augmentation de la population sur chaque algorithme (NP=20, 50 et 100), le nombre d'itération maximale fixé par maxit=300 Tableau III.5 et Figures III.6, III.7, III.8 et III.9.
2. dans la deuxième partie les différents algorithmes sont comparés pour une taille de population NP= 50 et le nombre maximal d'itérations maxit= 500,1000 et 1500 Tableau III.6. La Figure III.10 représente les résultats du meilleur individu de chaque algorithme après 10 exécutés indépendamment (NP=50 et maxit=1000).

Les résultats de modélisation (S11(f) coefficient de réflexion avec NP=50 et maxit=500) sont comparés à ceux obtenus par HFSS Figure III.11

3. la dernière partie, représente la comparaison entre chaque algorithme et l'algorithme hybride (NP=50 et maxit=300) (Tableau III.7, Figure III.12, III.13, III.14 et III.15), les résultats de modélisation sont illustrés à la Figure III.16.

**Tableau III.5:** L'erreur quadratique moyenne pour maxit=300

Algorithme	NP	Structure 1			Structure 2			Structure 3		
		EQM			EQM			EQM		
		Meilleur	Pire	Moy	Meilleur	Pire	Moy	Meilleur	Pire	Moy
OEP	20	0.1600	0.3727	0.2689	0.1019	0.2417	0.1981	0.1339	0.3547	0.2357
	50	0.0956	0.4296	0.2371	0.0502	0.3093	0.1560	0.0655	0.3145	0.1687
	100	0.1490	0.5452	0.3058	0.0413	0.1837	0.1043	0.0540	0.2352	0.1158
OAS	20	0.1033	0.1590	0.1296	0.0219	0.0852	0.0569	0.0257	0.2032	0.0762
	50	0.0145	0.1526	0.0750	0.0094	0.0385	0.0275	0.0197	0.1934	0.0670
	100	0.0062	0.0653	0.0438	0.0019	0.0410	0.0232	0.0067	0.1155	0.0440
OSA	20	0.2030	0.4319	0.2925	0.1825	0.2759	0.2141	0.2887	0.9349	0.6144
	50	0.0199	0.0354	0.0277	0.0114	0.0253	0.0169	0.0347	0.0721	0.0582
	100	0.0029	0.0063	0.0046	0.0015	0.0030	0.0020	0.0114	0.0168	0.0142
OBEA	20	0.0201	0.2015	0.0735	0.0123	0.0548	0.03056	0.3131	0.6708	0.4373
	50	0.0032	0.0078	0.0055	0.0008	0.0092	0.0032	0.0032	0.0850	0.0248
	100	0.0003	0.0009	0.0006	0.0002	0.0009	0.0005	0.0005	0.0116	0.0057

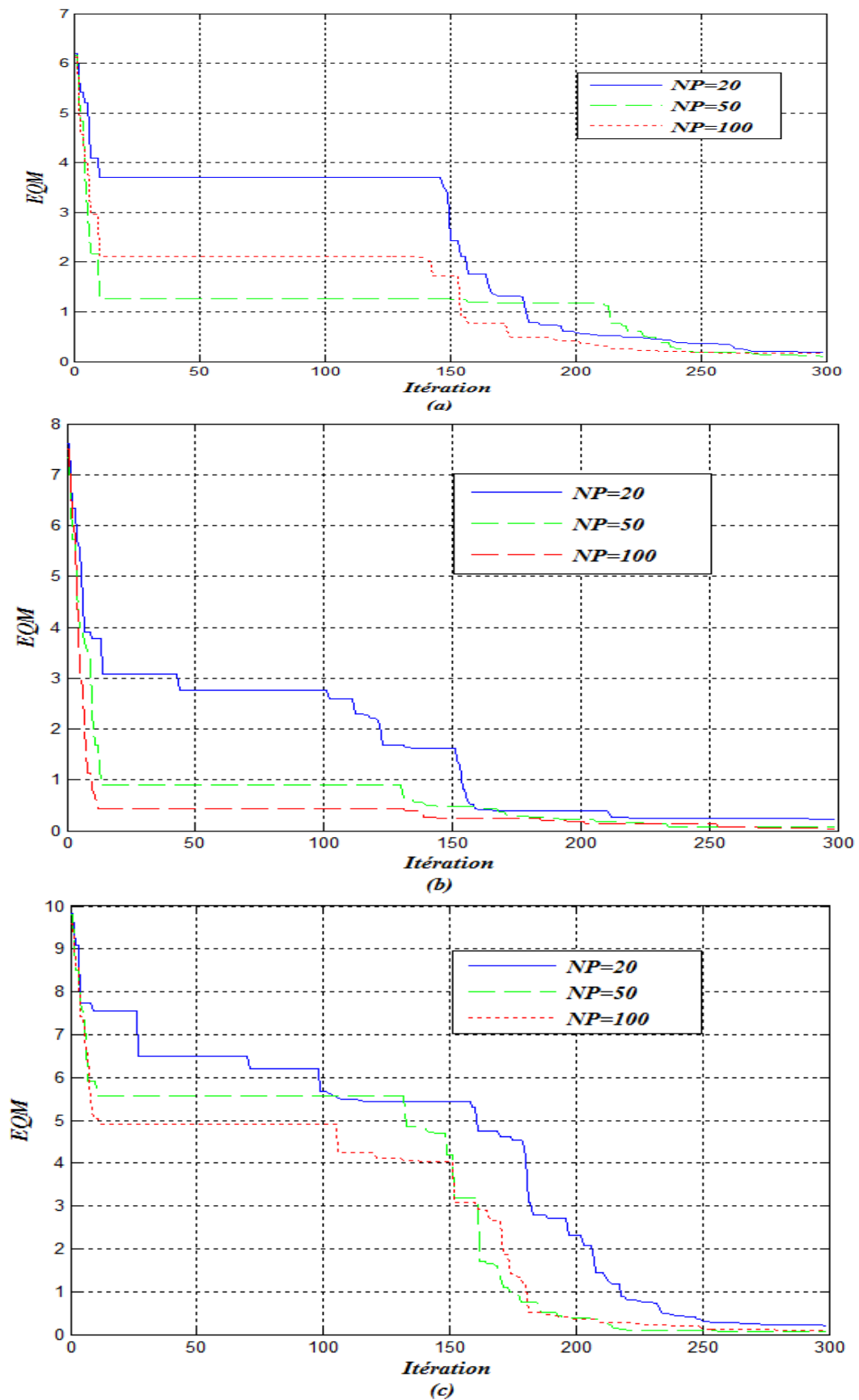


Figure III.6: Convergence de OEP pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3

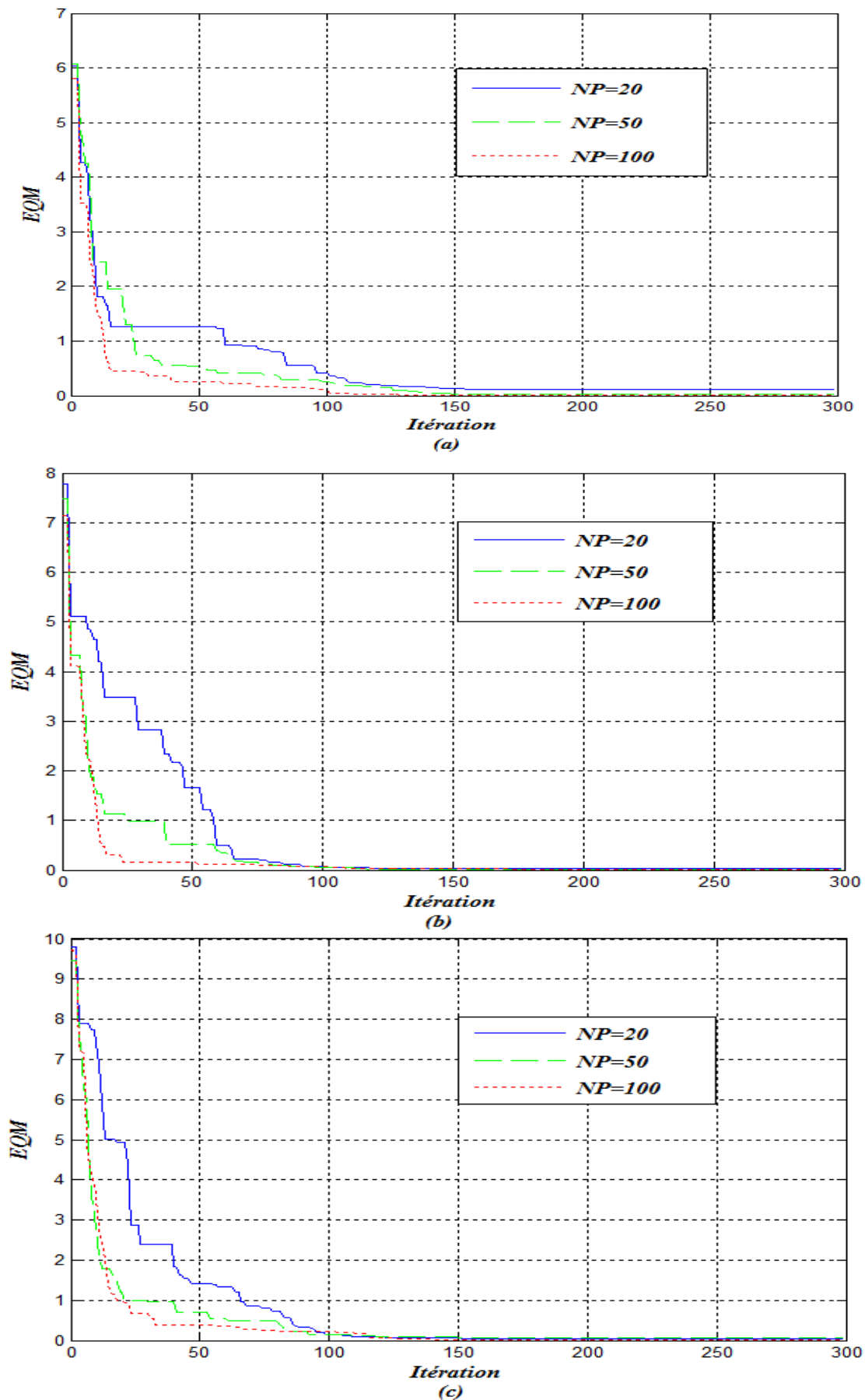


Figure III.7: Convergence de OAS pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3

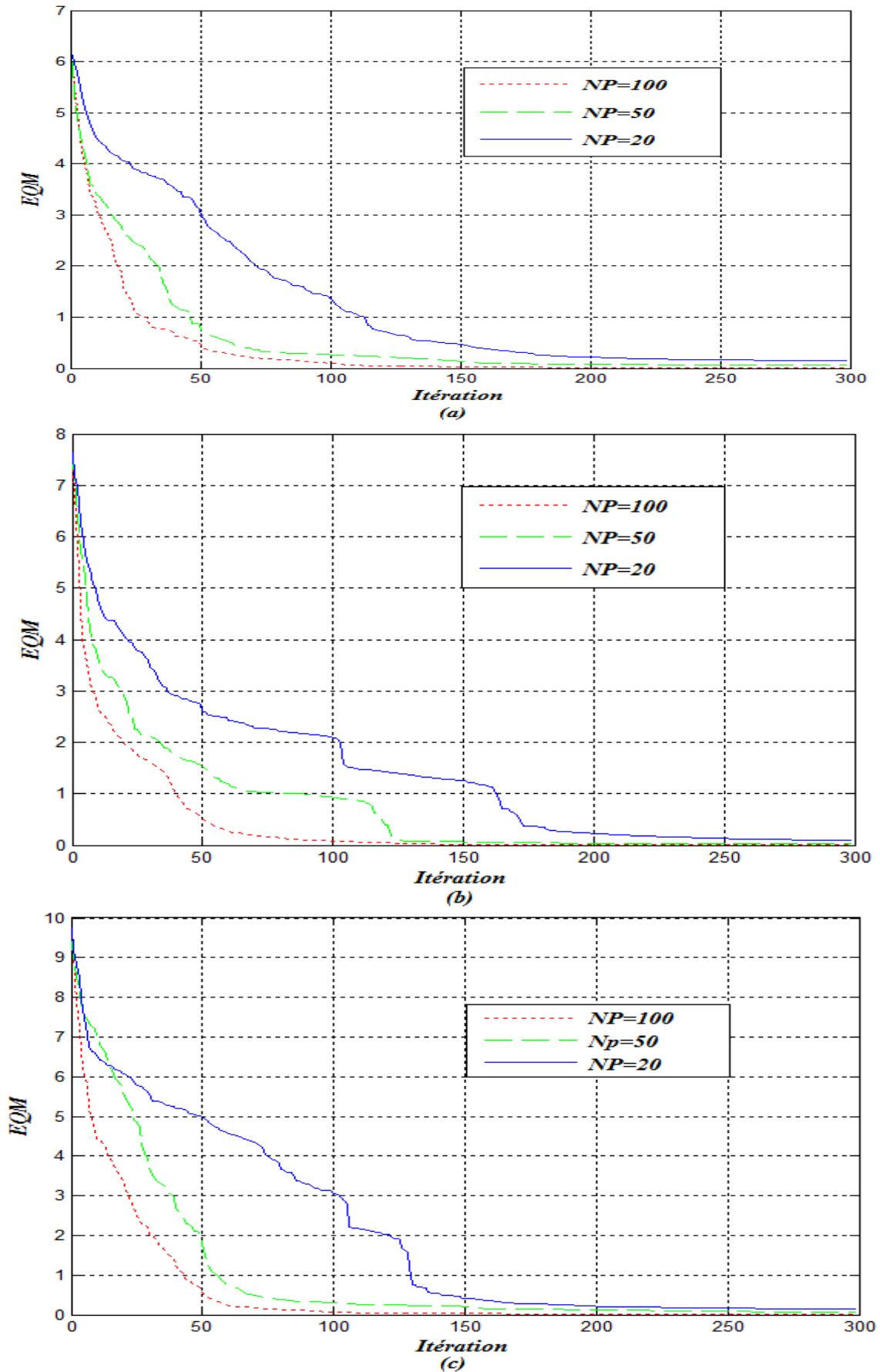


Figure III.8: Convergence de OSA pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3



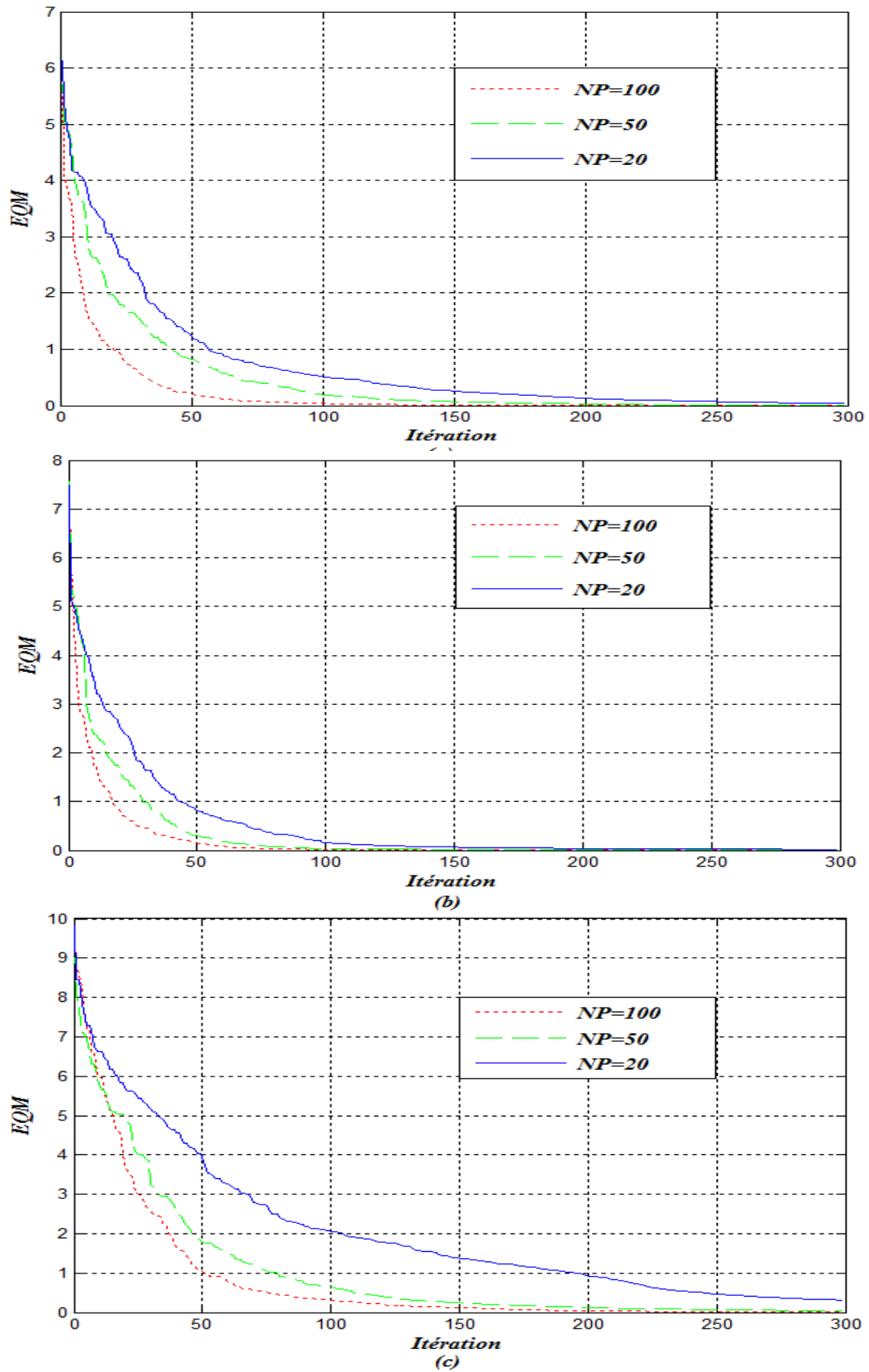


Figure III.9: Convergence de OBEA pour minimiser EQM, (a) structure 1, (b) structure 2, (c) structure 3

Tableau III.6: Comparaison de performance des algorithmes OEP, OAS, OSA et OBEA

Algorithme	Maxit	structure 1			structure 2			structure 3		
		EQM			EQM			EQM		
		Meilleur	Pire	Moy	Meilleur	Pire	Moy	Meilleur	Pire	Moy
OEP	500	$7.214e^{-2}$	$5.919e^{-1}$	$2.398e^{-1}$	$5.055e^{-2}$	$3.115e^{-1}$	$1.679e^{-1}$	$7.158e^{-2}$	$2.654e^{-1}$	$1.404e^{-1}$
	1000	$4.904e^{-2}$	$3.447e^{-1}$	$1.605e^{-1}$	$7.036e^{-2}$	$2.515e^{-1}$	$1.422e^{-1}$	$1.961e^{-2}$	$2.624e^{-1}$	$1.218e^{-1}$
	1500	$2.315e^{-2}$	$2.671e^{-1}$	$1.218e^{-1}$	$3.280e^{-2}$	$1.767e^{-1}$	$9.442e^{-2}$	$1.936e^{-2}$	$1.506e^{-1}$	$8.268e^{-2}$
OAS	500	$2.111e^{-2}$	$1.843e^{-1}$	$8.229e^{-2}$	$1.222e^{-2}$	$6.379e^{-2}$	$4.217e^{-2}$	$2.213e^{-2}$	$1.760e^{-1}$	$5.973e^{-2}$
	1000	$1.396e^{-2}$	$1.542e^{-1}$	$7.468e^{-2}$	$7.718e^{-3}$	$4.699e^{-2}$	$2.776e^{-2}$	$8.599e^{-3}$	$5.606e^{-2}$	$4.742e^{-2}$
	1500	$1.323e^{-2}$	$1.403e^{-1}$	$5.335e^{-2}$	$3.466e^{-3}$	$3.760e^{-2}$	$2.011e^{-2}$	$7.549e^{-3}$	$3.079e^{-2}$	$1.950e^{-2}$
OSA	500	$1.269e^{-2}$	$6.886e^{-2}$	$2.936e^{-2}$	$4.463e^{-3}$	$5.186e^{-2}$	$1.674e^{-2}$	$1.038e^{-1}$	$9.966e^{-1}$	$2.523e^{-1}$
	1000	$3.319e^{-3}$	$6.333e^{-2}$	$1.628e^{-2}$	$3.739e^{-4}$	$7.421e^{-3}$	$2.789e^{-3}$	$1.291e^{-2}$	$7.528e^{-1}$	$1.684e^{-1}$
	1500	$1.738e^{-4}$	$2.879e^{-2}$	$9.776e^{-3}$	$1.501e^{-4}$	$1.190e^{-3}$	$5.710e^{-4}$	$1.281e^{-2}$	$2.092e^{-1}$	$1.388e^{-1}$
OBEA	500	$8.722e^{-3}$	$3.888e^{-4}$	$2.071e^{-4}$	$1.809e^{-3}$	$5.226e^{-3}$	$3.315e^{-3}$	$2.820e^{-3}$	$4.385e^{-3}$	$3.517e^{-3}$
	1000	$1.217e^{-8}$	$7.858e^{-6}$	$1.593e^{-6}$	$1.204e^{-9}$	$3.668e^{-8}$	$1.171e^{-8}$	$2.163e^{-3}$	$6.423e^{-3}$	$4.010e^{-3}$
	1500	$5.999e^{-12}$	$3.560e^{-10}$	$1.209e^{-10}$	$5.931e^{-14}$	$5.387e^{-12}$	$1.622e^{-12}$	$1.074e^{-7}$	$1.813e^{-6}$	$9.320e^{-7}$

Tableau III.7: comparaison entre les algorithmes de base et leurs hybrides algorithmes

Algorithme	structure 1			structure 2			structure 3		
	EQM			EQM			EQM		
	Meilleur	Pire	Moy	Meilleur	Pire	Moy	Meilleur	Pire	Moy
OEP	$9.560e^{-2}$	$4.296e^{-1}$	$2.371e^{-1}$	$5.02e^{-2}$	$3.093e^{-1}$	$1.560e^{-1}$	$6.55e^{-2}$	$3.145e^{-1}$	$1.687e^{-1}$
HOEP	$9.666e^{-3}$	$1.541e^{-2}$	$1.220e^{-2}$	$9.939e^{-3}$	$1.584e^{-2}$	$1.211e^{-2}$	$2.990e^{-2}$	$3.962e^{-2}$	$3.585e^{-2}$
OAS	$1.45e^{-2}$	$1.526e^{-1}$	$7.50e^{-2}$	$9.400e^{-3}$	$3.850e^{-2}$	$2.750e^{-2}$	$1.97e^{-2}$	$1.934e^{-1}$	$6.700e^{-2}$
HOAS	$1.786e^{-3}$	$9.033e^{-3}$	$4.301e^{-3}$	$2.107e^{-3}$	$1.217e^{-3}$	$2.573e^{-3}$	$1.134e^{-2}$	$9.192e^{-2}$	$4.551e^{-2}$
OSA	$1.99e^{-2}$	$3.54e^{-2}$	$2.77e^{-2}$	$1.14e^{-2}$	$2.53e^{-2}$	$1.69e^{-2}$	$3.47e^{-2}$	$7.21e^{-2}$	$5.82e^{-2}$
HOSA	$8.691e^{-4}$	$2.674e^{-3}$	$1.476e^{-3}$	$1.921e^{-3}$	$4.248e^{-3}$	$3.107e^{-3}$	$1.531e^{-3}$	$7.609e^{-3}$	$4.828e^{-3}$
OBEA	$3.2e^{-3}$	$7.8e^{-3}$	$5.5e^{-3}$	$8.001e^{-4}$	$9.200e^{-3}$	$3.200e^{-3}$	$3.200e^{-3}$	$8.500e^{-2}$	$2.480e^{-2}$
HOBEA	$1.408e^{-4}$	$4.033e^{-4}$	$2.605e^{-4}$	$8.489e^{-5}$	$5.612e^{-4}$	$2.876e^{-4}$	$1.796e^{-3}$	$2.419e^{-3}$	$2.211e^{-3}$

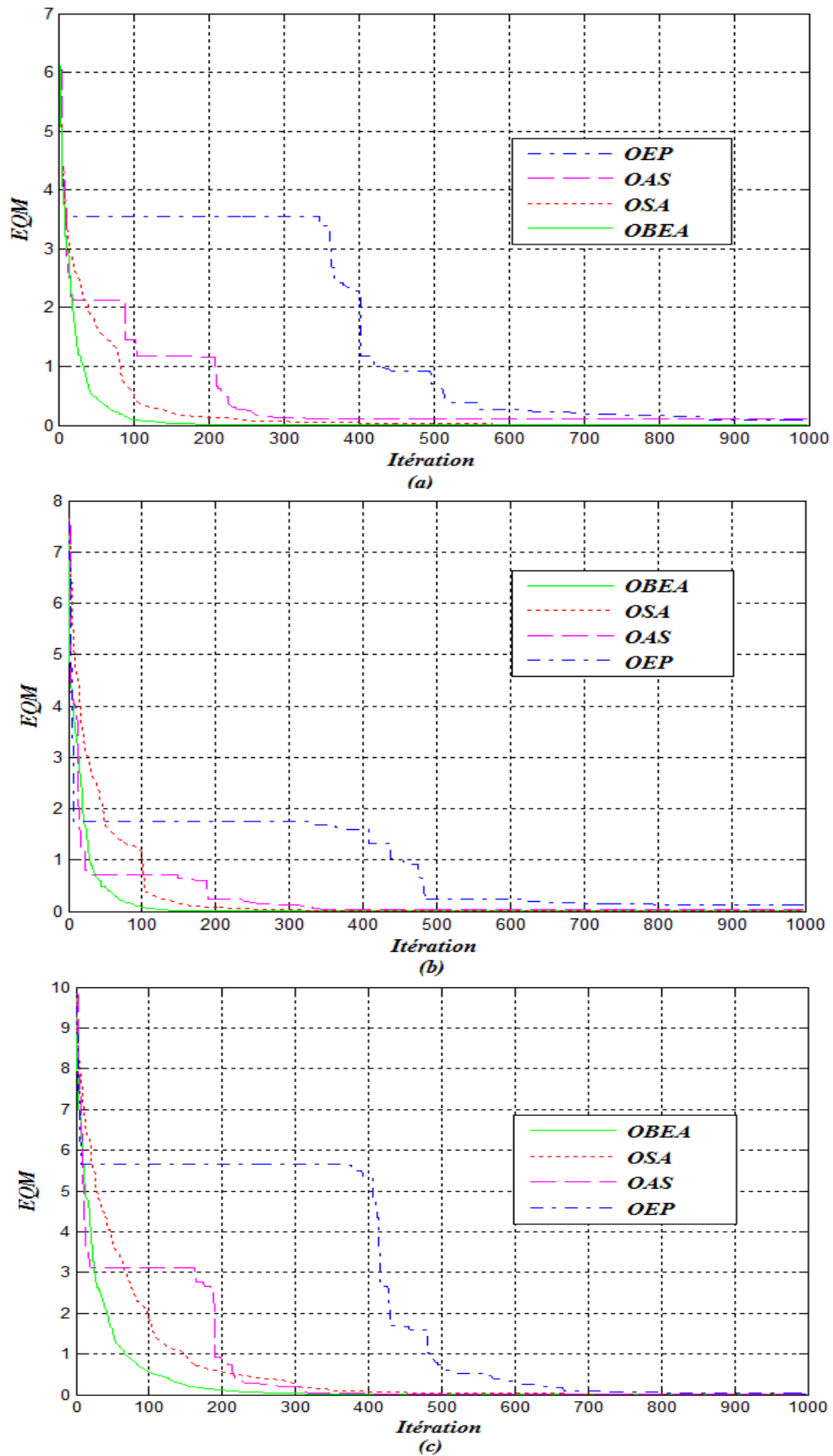


Figure III.10: comparaison entre les algorithmes OEP, OAS, OSA et OBEA, (a) structure 1, (b) structure 2, (c) structure 3

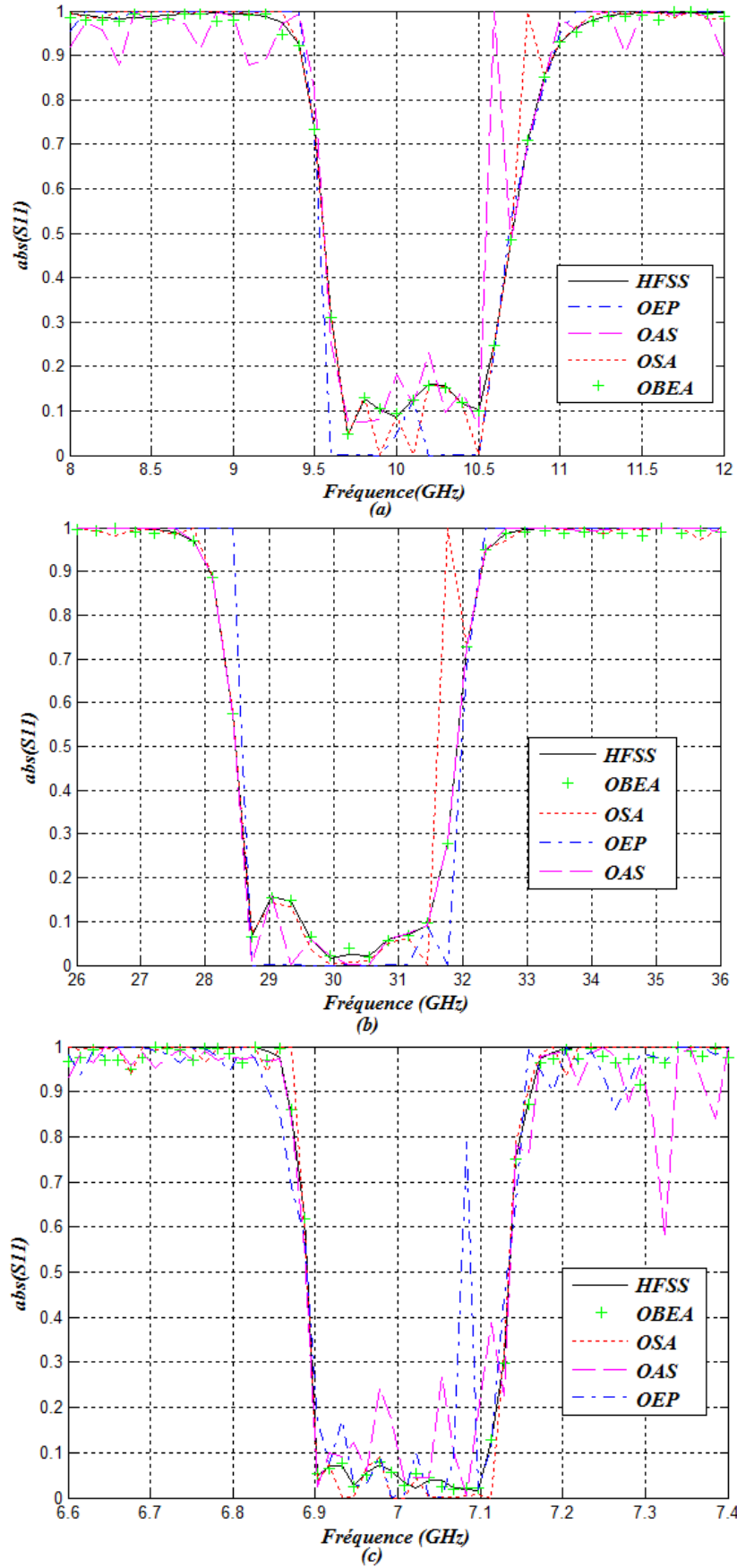
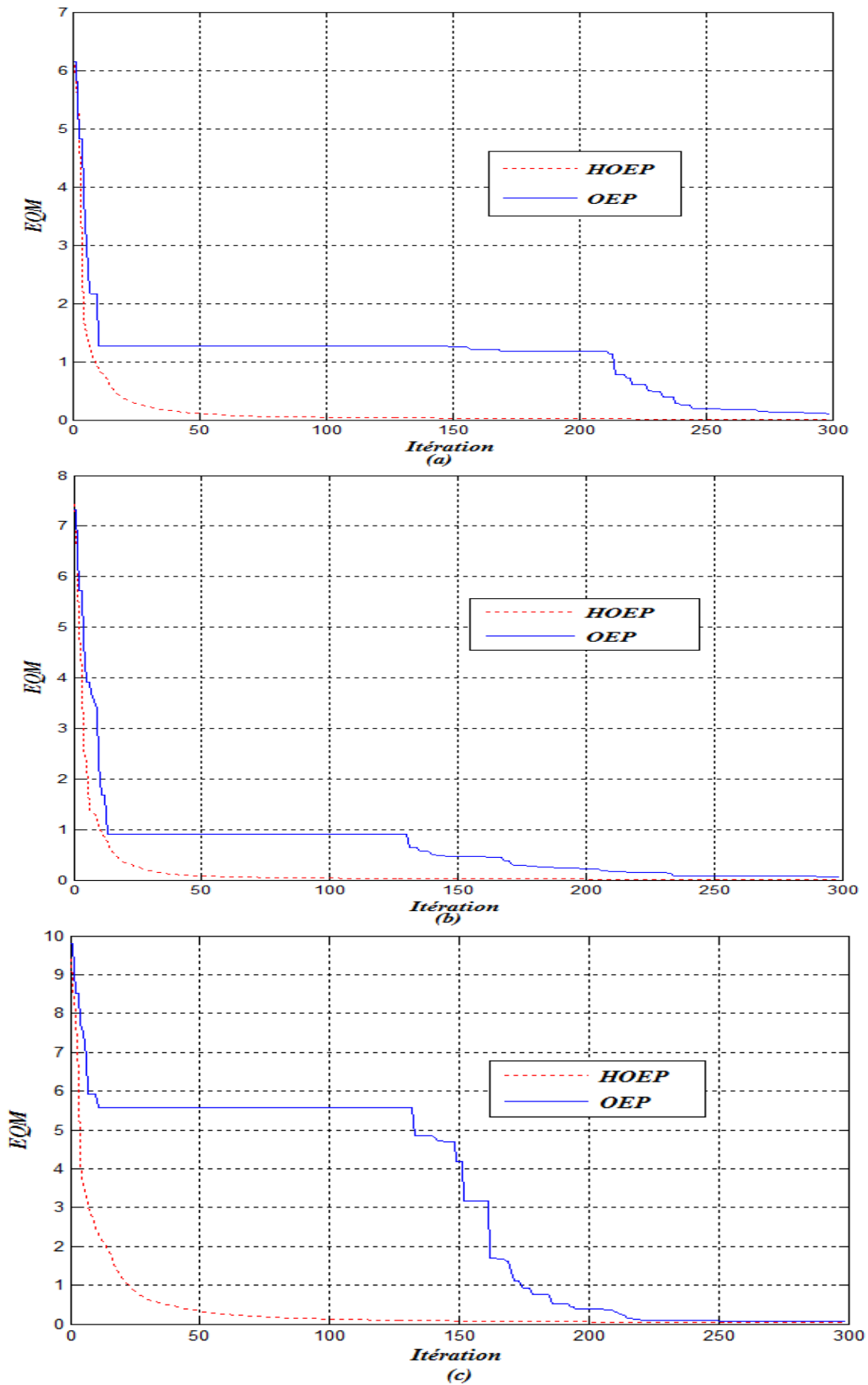
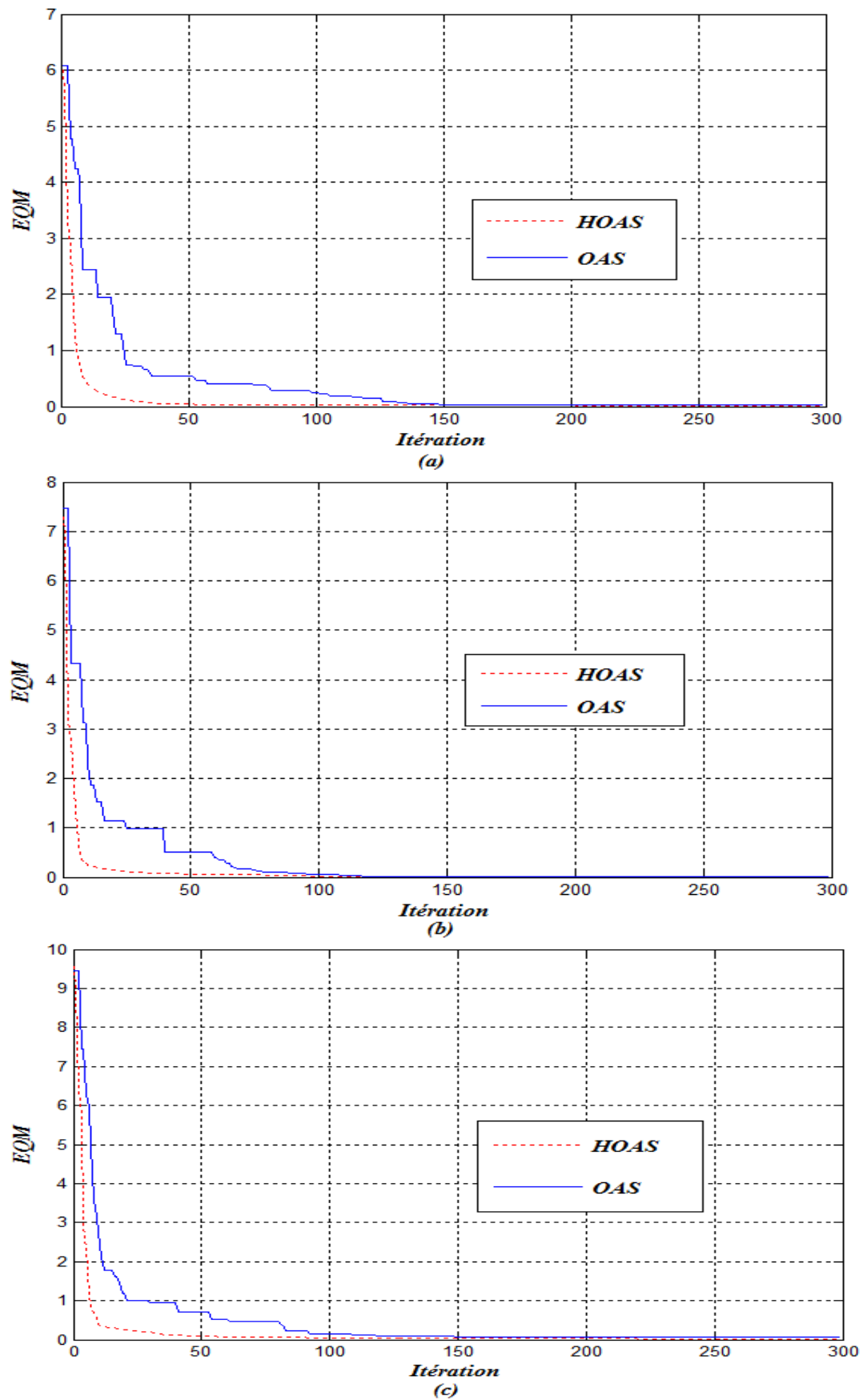


Figure III.11: le coefficient de réflexion approximatif, (a) structure 1, (b) structure 2, (c) structure 3



**Figure III.12:** comparaison entre OEP et HOEP,(a) structure 1, (b) structure 2, (c) structure 3



**Figure III.13:** comparaison entre OAS et HOAS, (a) structure 1, (b) structure 2, (c) structure 3

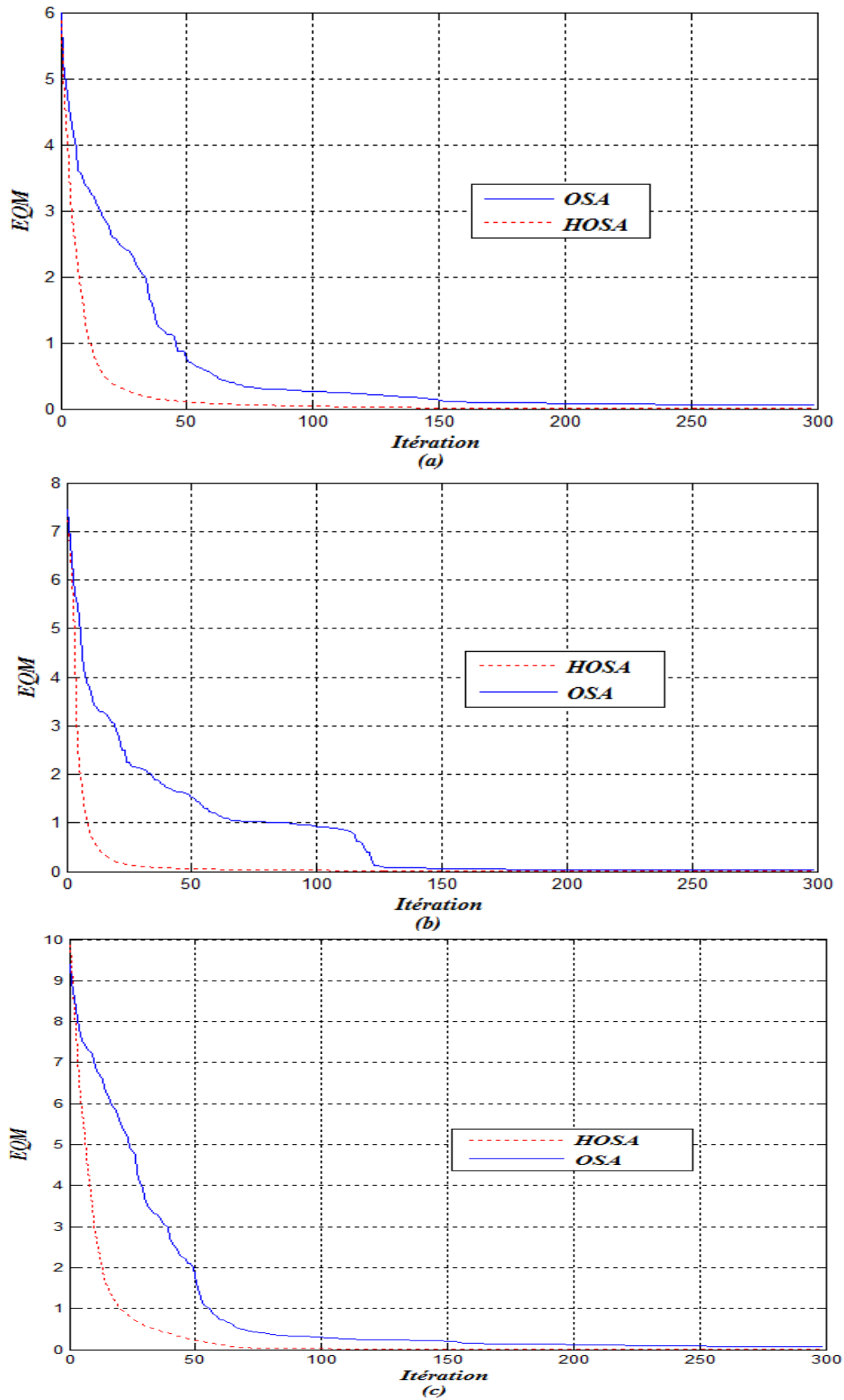


Figure III.14. comparaison entre OSA et HOSA,(a) structure 1, (b) structure 2, (c) structure 3

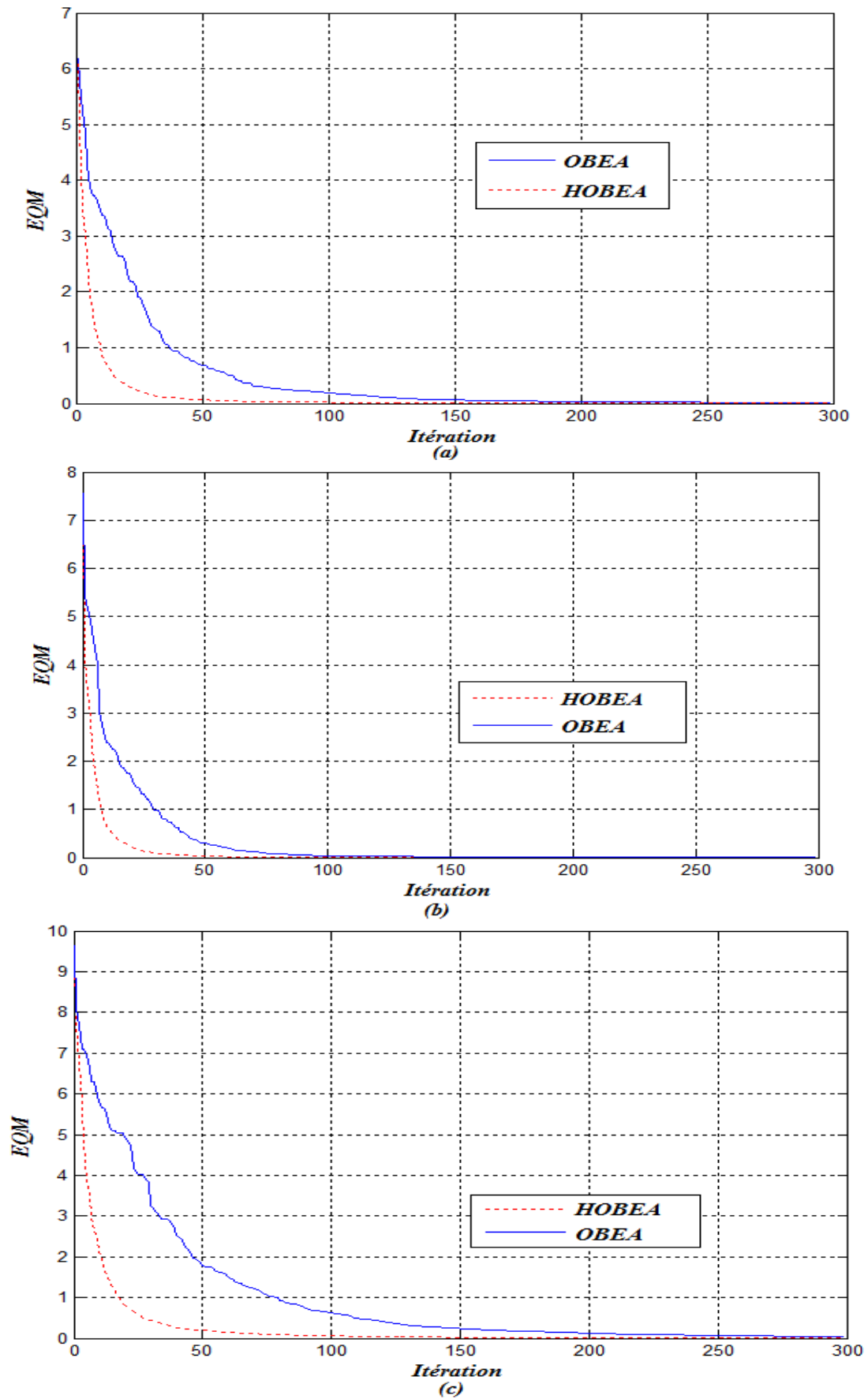


Figure III.15: comparaison entre OBEA et HOBEA,(a) structure 1, (b) structure 2, (c) structure 3



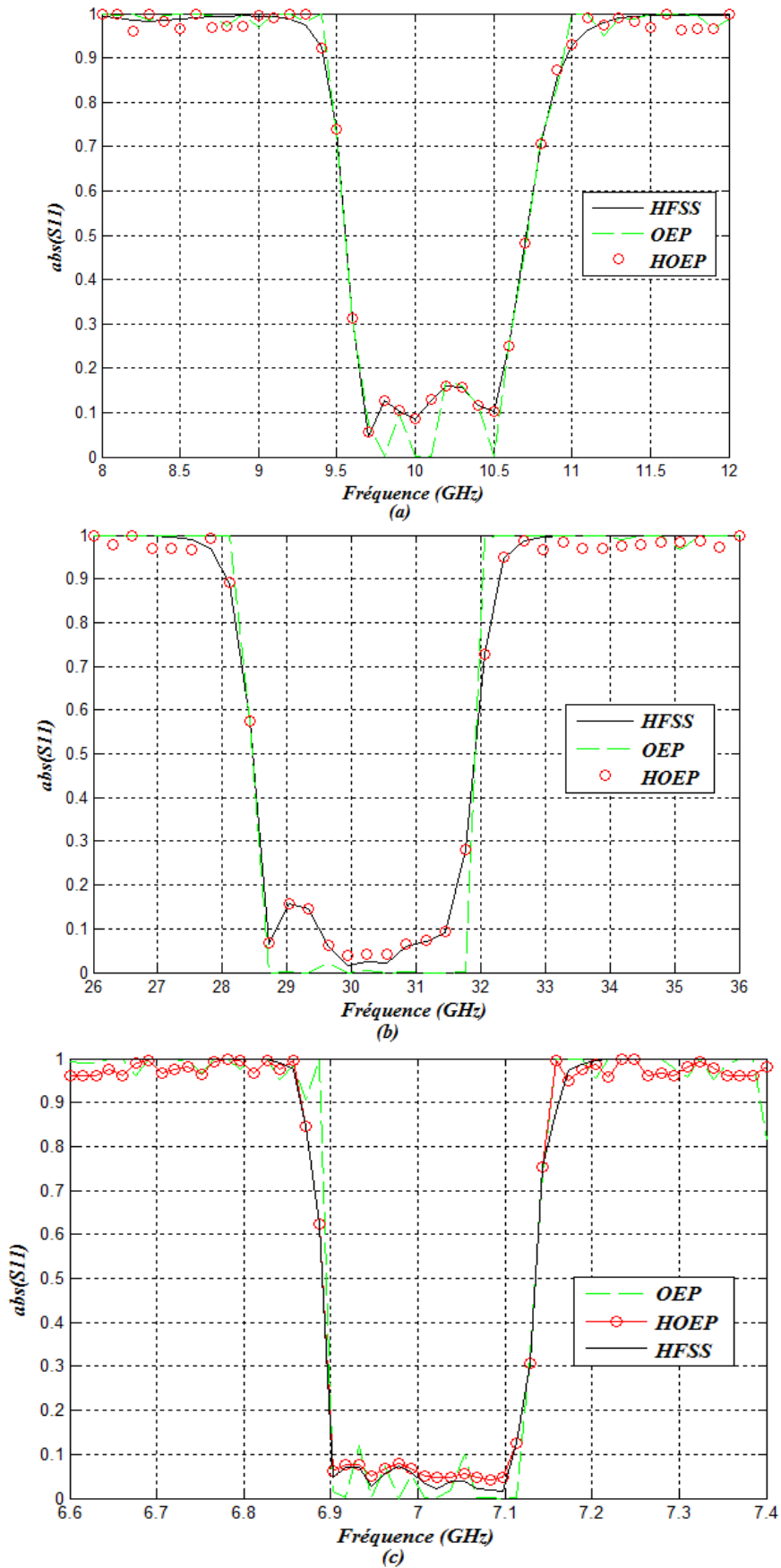
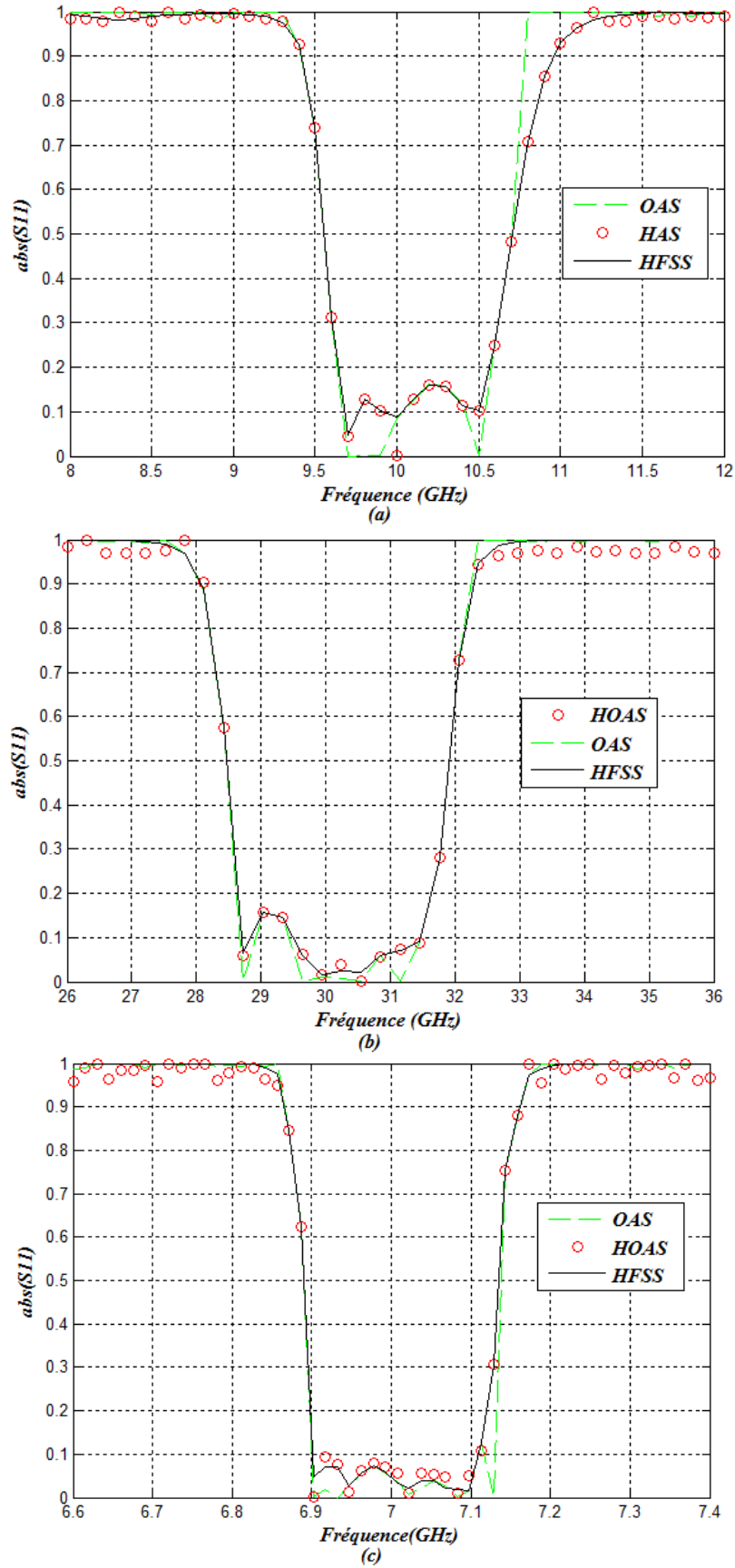


Figure III.16: coefficient de réflexion obtenu par OEP et HOEP, (a) structure 1, (b) structure 2, (c) structure 3



**Figure III.17:** coefficient de réflexion obtenu par OAS et HOAS, (a) structure 1, (b) structure 2, (c) structure 3

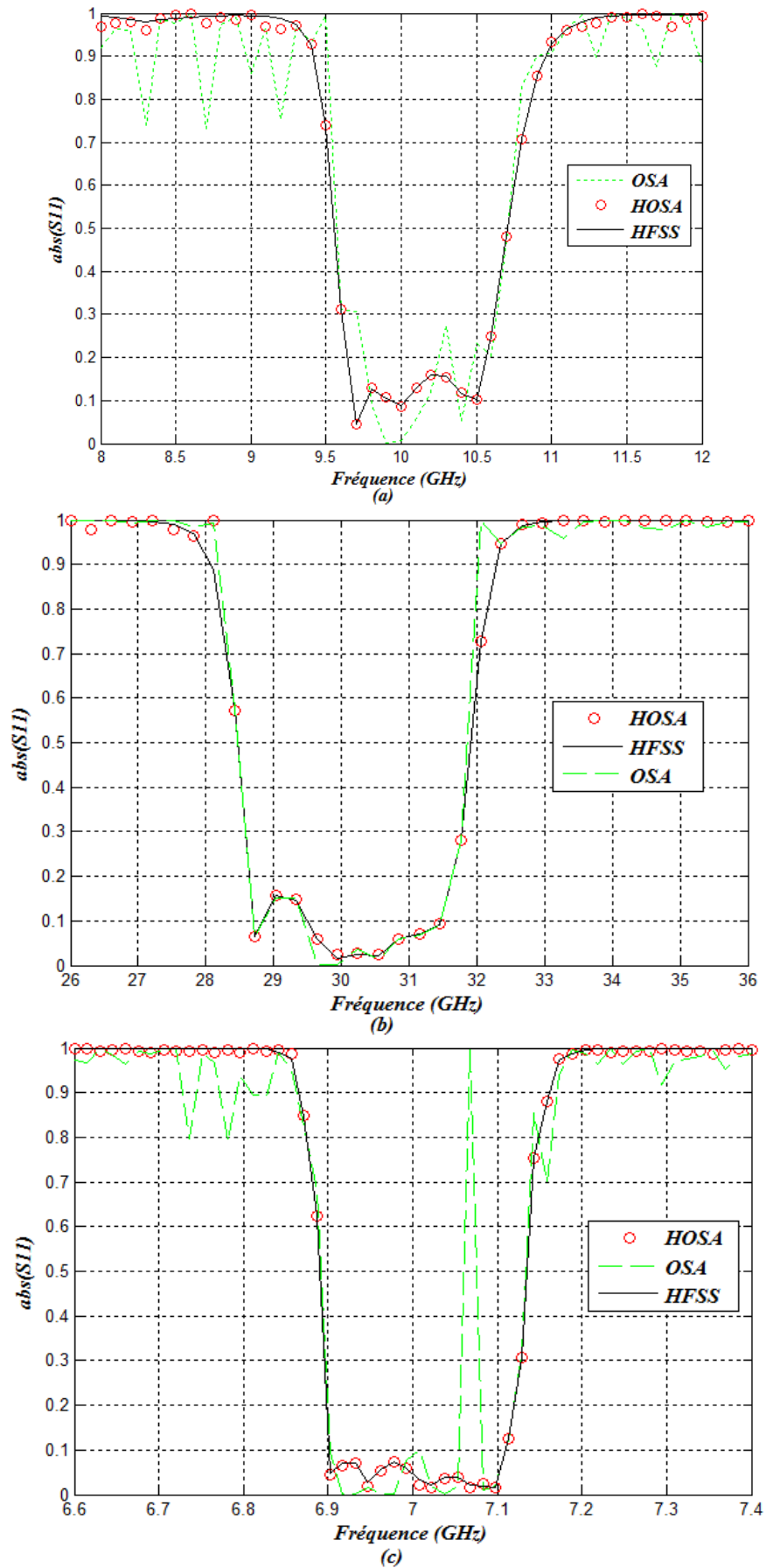


Figure III.18: coefficient de réflexion obtenu par OSA et HOSA, (a) structure 1, (b) structure 2, (c) structur3

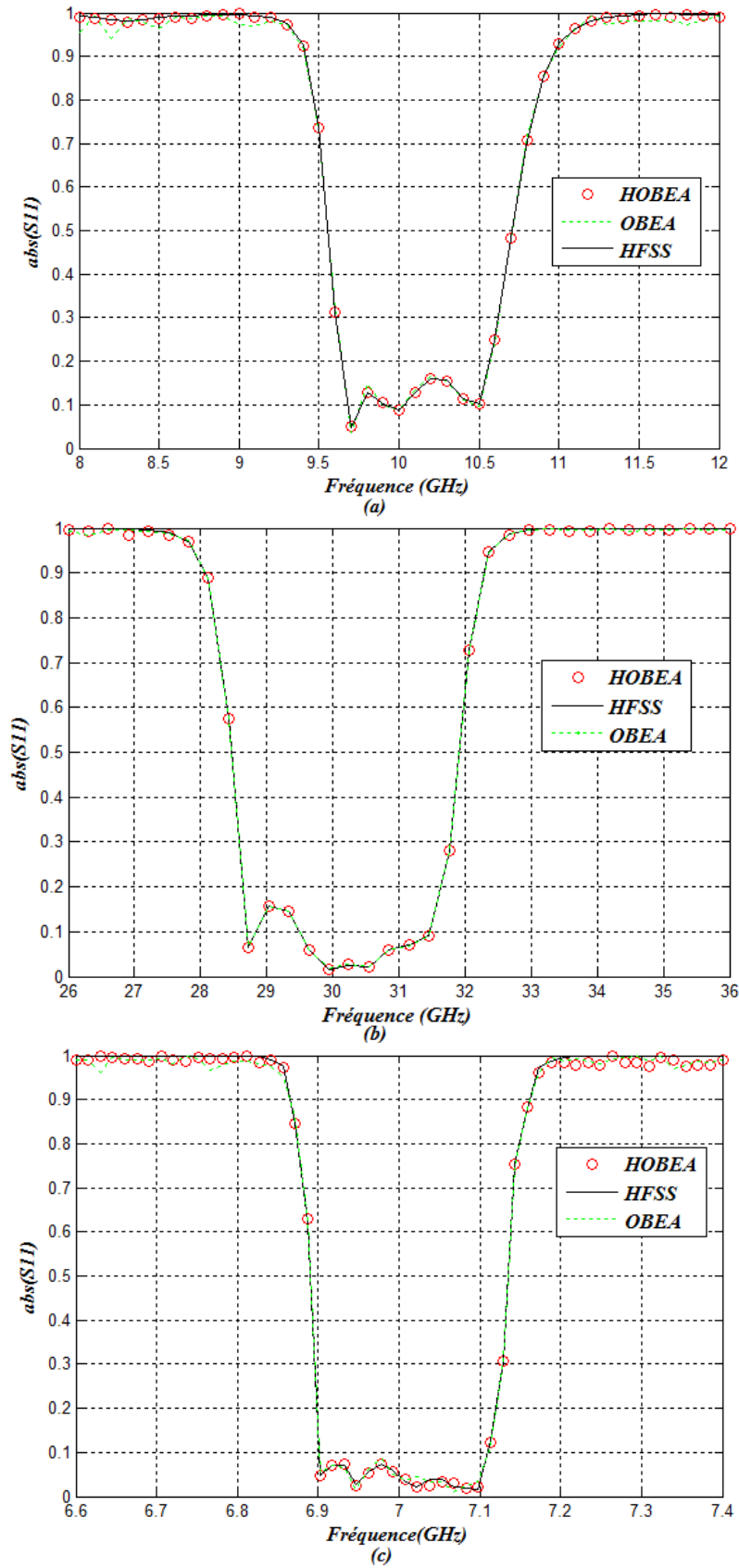


Figure III.19: coefficient de réflexion obtenu par OBEA et HOBEA, (a) structure 1, (b) structure 2, (c) structur3

L'examen des résultats obtenus, nous permet de faire des remarques suivantes:

- \_ l'algorithme OBEA donnent de meilleurs résultats en termes de convergence et d'approximation que les autres algorithmes, Cela est dû au principal avantage de l'algorithme OBEA qui ne nécessite pas la sélection de paramètres spécifiques.
- \_ les algorithmes hybrides proposés donnent une excellente convergence et modélisation

## **V. Conclusion**

Dans ce chapitre, un algorithme hybride combinant l'algorithme d'optimisation essaim d'intelligence (SI) avec l'algorithme de rétropropagation (BP), est proposé pour former les poids du réseau de neurones feedforward (FNN). Puis, les réseaux formés sont ensuite appliqués à la modélisation des structures hyperfréquences (pseudo-elliptique filtre, filtre E-plan large bande et Filtres plan H considérant des coins arrondis). Les résultats obtenus par les algorithmes hybride montrent que ces méthodes ont non seulement une plus grande vitesse de convergence que les autres algorithmes de base (SSO, SMO, TLBO et PSO) mais présentent également une excellente approximation.

- [1] Y. Wang, M. Yu, H. Kabir, and Q.J. Zhang, “**Application of Neural Networks in Space Mapping Optimization of Microwave Filters,**” International Journal of RF and Microwave Computer Aided Engineering, Vol. 22 (2), PP. 159–166, **2012**.
- [2] J. S. Sivia, A. P. S. Pharwaha, and T. S. Kamal, “**Analysis and Design of Circular Fractal Antenna Using Artificial Neural Networks,**” Progress in Electromagnetics Research B, Vol. 56, PP. 251–267, **2013**.
- [3] A. A. Deshmukh, S.D. Kulkarni, A.P.C. Venkata, and N.V. Phatak, “**Artificial Neural Network Model for Suspended Rectangular Microstrip Antennas,**” Procedia Computer Science, Vol. 49, PP. 332–339, **2015**
- [4] D.J. Jwo, and K.P. Chin, “**Applying Back-propagation Neural Networks to GDOP Approximation,**” The Journal of Navigation, Vol. 55 (2), PP. 97–108, **2002**.
- [5] D. Gyanesh, K. P. Prasant, and K. P. Sasmita, “**Artificial Neural Network Trained by Particle Swarm Optimization for Non-Linear Channel Equalization,**” Expert Systems with Applications, Vol. 41(7), PP. 3491–3496, **2014**.
- [6] S. Ding, Y. Zhang, J. Chen, and J. Weikuan, “**Research on Using Genetic Algorithms to Optimize Elman Neural Networks,**” Neural Computing and Applications, vol.23 (2), PP. 293–297, **2013**.
- [7] K. Khan, and A. Sahai, “**A Comparison of BA, GA, PSO, BP and LM for Training Feed Forward Neural Networks in E-Learning Context,**” International Journal of Intelligent Systems and Applications, Vol.4, PP. PP. 23 – 29, **2012**.
- [8] B. Rabindranath, S.S. Kumar, and C. Swastika, “**Design and Analysis of Microstrip Antenna Using Multilayer Feed-Forward Back-Propagation Neural Network (MLPFFBP-ANN) ,**” Lecture Notes in Electrical Engineering; Advances in Communication, Devices and Networking Volume 462, Chapter 43, PP. 393–398, **2018**
- [9] Q. Zhang, and Y. Lu, “**Design of Wide-Band Pseudo-Elliptic Waveguide Filters with Cavity-Backed Inverters,**” IEEE Microwave and Wireless Components Letters, Vol.20 (11), PP. 604–606, **2010**.
- [10] Z. Xu, J. Guo, Ch. Qian, and W. Dou, “**Broad-Band E-Plane Filters With Improved Stop-Band Performance,**” IEEE Microwave and Wireless Components Letters, Vol. 21(7), PP. 350–352, **2011**.
- [11] E. Diaz Caballero, J. Morro, A. Belenguer, H. Esteban, and V. Boria, “**CAD Technique for Designing H-Plane Waveguide Filters Considering Rounded Corners,**” IEEE

- MTT-S International Microwave Symposium Digest, Seattle, WA, USA, PP. 1–3, June **2013**.
- [12] K. Hornik , M. Stinchcombe, and H. White , ‘**Multilayer feedforward networks are universal approximators**’, Neural Networks, Vol. 2 (5), PP. 359-366, **1989**.
- [13] M. Sarevska, ‘**Signal Detection for Neural Network-Based Antenna Array**’, Conf.NAUN’08 on Circuits, Systems, and Signals, Marathon, Attica, Greece, PP.115-119, June **2008**.

## *Conclusion Générale*

---



Nous rappelons que l'objectif de ce manuscrit est de modéliser des dispositifs hyperfréquence en utilisant les réseaux de neurones (RN) artificiels.

Pour traiter cet objectif, nous avons proposé une architecture de RN non bouclée (Feed-Forward) à trois couches. La première couche de  $N_e$  neurones correspondant au nombre de paramètres géométrie des structures, la deuxième couche est la couche cachée de  $N_c$  neurones, la dernière couche est la couche de sortie associée à la valeur du coefficient de réflexion  $S_{11}$ .

Une fois l'architecture du réseau a été choisie, on passe à l'étape d'apprentissage du réseau. Tout d'abord, nous avons formé le RN en utilisant trois algorithmes d'optimisation récents:

- Optimisation par les araignées sociales basée sur le comportement des colonies d'araignées naturelles.
- Optimisation par singes araignées inspiré de la nature motivé par le comportement social des singes araignées
- Optimisation à base d'enseignement- apprentissage, qui est basé sur l'effet de l'influence d'un enseignant sur les résultats des apprenants dans une classe.

Une étude détaillée de chaque algorithme a été présentée, et nous avons ensuite testé la performance de ces algorithmes en deux structures micro-ondes (Transformateur à deux paliers et filtres à iris inductif à trois cavités) pour différentes paramètres de contrôle (nombre de population et le maximum d'itération). Les résultats obtenus montrent que les algorithmes OSA et OBEA sont plus performants en termes de convergence que les algorithmes OAS et OEP. Dans lequel les algorithmes OSA et OBEA convergent vers le minimum optimal pour les premières itérations  $max_{it} = 30$ , d'autre part les algorithmes OAS et OEP convergent vers le minimum optimal en 300 itérations.

Le RN traité par les trois algorithmes mentionnés a été utilisé pour la modélisation des dispositifs microondes (pseudo-elliptique filtre, filtre E-plan large bande et Filtres plan H considérant des coins arrondis), puis les résultats obtenus sont comparés avec ceux obtenus par le RN traité par l'algorithme d'optimisation des essaims de

particules, où l'algorithme OBEA donnent de meilleurs résultats en termes de convergence et d'approximation que les autres algorithmes, Cela est dû au principal avantage de l'algorithme OBEA qui ne nécessite pas la sélection de paramètres spécifiques.

La partie récente de ce travail est la proposition d'une nouvelle stratégie d'apprentissage du RN, basée sur la combinaison entre deux types d'algorithmes d'optimisation les algorithmes conventionnels (Gradient descendant) et les algorithmes essais intelligents (OAS, OSA et OBEA). Les résultats de convergence et de modélisation sont très satisfaisants.

## *Résumé*

---

**Résumé** — dans ce travail, une nouvelle stratégie de réseaux de neurones (RN) est proposée pour la modélisation de structures en guides d'ondes hyperfréquences (pseudo-elliptique filtre, filtre plan-E large bande et Filtres plan-H considérant des coins arrondis). Afin de renforcer les capacités du RN, nous avons formé RN par les algorithmes hybrides basés sur la combinaison entre l'algorithme de rétro-propagation (RP) et les algorithmes des essais d'intelligents (l'optimisation par araignées sociales OAS, l'optimisation par singe araignée OSA et l'optimisation a basée d'enseignement apprentissage OBEA). Pour valider la formation des réseaux neuronaux en utilisant les algorithmes proposés, nous avons comparé les résultats de convergence et de modélisation obtenus avec ceux obtenus avec les algorithmes de base (OAS, OSA et OBEA) et aussi avec l'algorithme basé sur la population, qui est largement utilisé dans la formation RN appelé l'optimisation par l'essaim de particules (OEP). Les résultats prouvent que les algorithmes hybrides proposés ont donné de résultats satisfaisants.

**Mots-clés:** Réseaux de neurones, Structures hyperfréquences, Modélisation, Algorithmes des essais d'intelligents.

**Abstract** — In this work, a new strategy of neural networks (NN) is proposed to modeling microwave waveguide structures (Pseudo-Elliptic filter, Broad-band E-plane filters and H-plane waveguide filters considering rounded corners). In order to enhance the capacities of the NN, we trained NN by the hybrids algorithms based on combining between back propagation (BP) algorithm and swarm intelligence algorithms (Social-Spider optimization SSO, spider monkey optimization SMO and Teaching–Learning-Based Optimization TLBO). To validate the training of neural networks using the proposed algorithms, we compared the results of convergence and modeling obtained with the results obtained using basic algorithms (SSO, SMO and TLBO) and also compared with population based algorithm, which is widely used in training NN namely particle swarm optimization (PSO). The results prove that the proposed hybrids algorithms have given better results.

**Keywords:** Neural Networks, Microwave Structures, Modeling, Swarm Intelligence Algorithms.

**ملخص** – في هذا العمل ، تم اقتراح إستراتيجية جديدة لشبكات العصبية (ش ع) من أجل نمذجة هياكل الدليل الموجي للترددات الجد عالية (مرشح شبه إهليلجي(بيضاوي الشكل), مرشح E-plans ذو شريط عريض و مرشح H-plans ذو زوايا دائرية ) .من أجل تعزيز قدرات (ش ع) ، قمنا بتدريب (ش ع) من خلال خوارزميات الهجينة على أساس الجمع بين خوارزمية الانتشار الخلفي و الخوارزميات التي تعتمد على الأسراب الذكية (التحسين القائم على العناكب الاجتماعية OAS ، التحسين القائم على القروود العناكب OSA و التحسين القائم على التدريس - التعلم OBEA ). للتحقق من صحة تشكيل الشبكات العصبية باستخدام الخوارزميات المقترحة ، قارنا نتائج التقارب والنمذجة التي تم الحصول عليها مع النتائج التي تم الحصول عليها مع الخوارزميات الأساسية (OSA, OAS, OBEA) وكذلك مع الخوارزمية المبنية على السكان ، والتي تستخدم على نطاق واسع في التدريب (ش ع) التحسين من خلال سرب الجسيمات (OEP) . أثبتت النتائج أن الخوارزميات الهجينة المقترحة قد أعطت نتائج أفضل.

**كلمات البحث:** الشبكات العصبية ، هياكل الميكروويف ، النمذجة ، خوارزميات الأسراب الذكية.

# *Contributions Scientifiques*

---

## I- Publications

Chahrazad Erredir, Mohamed Lahdi Riabi, Halima Ammari, Emir Bouarroudj (**Design of Waveguide Structures Using Improved Neural Networks**), Journal of Microwaves, Optoelectronics and Electromagnetic Applications, Vol. 16, No. 4, PP. 900-907, 2017.  
<http://dx.doi.org/10.1590/2179-10742017v16i4960>

## II- Communication internationales

### *Communications internationales indexé*

1- Chahrazad Erredir, Emir Bouarroudj , Mohamed Lahdi Riabi (**Social-Spider Optimization Neural Networks for Microwave Filters Modeling**), Springer International Publishing, 6th IFIP International Conference on Computational Intelligence and its Applications **IFIP CIA'2018** , PP. 364 – 372, 2018, Oran, Algeria

[https://doi.org/10.1007/978-3-319-89743-1\\_32](https://doi.org/10.1007/978-3-319-89743-1_32)

2- Chahrazad Erredir, Emir Bouarroudj , Mohamed Lahdi Riabi (**Swarm Intelligence Algorithm for Microwave Filter Optimization**), Springer International Publishing, 6th IFIP International Conference on Computational Intelligence and Its Applications **IFIP CIA'2018** , PP. 161-172, 2018, Oran, Algeria

[https://doi.org/10.1007/978-3-319-89743-1\\_15](https://doi.org/10.1007/978-3-319-89743-1_15)

3- Ch. Erredir, M. L. Riabi, H. Ammari, E. Bouarroudj (**Modeling of Microwave Filters Using Gradient Particle Swarm Optimization Neural Networks**), IEEE xplore, 7th seminar on detection systems : architectures and technologies (**DAT'2017**) , 2017, Algiers, Algeria

<https://ieeexplore.ieee.org/document/7889160>

### *Communications internationales avec comité programme et comité de lecture internationaux*

1- Chahrazad Erredir, Mohamed Lahdi Riabi, Emir Bouarroudj, Halima Ammari, (**Swarm Optimization Algorithm Inspired in the Behavior of the Social-Spider for Microwave Filters Optimization**) 7th African conference on non destructive testing ACNDT and 5th International conference on welding, non destructive testing and materials and alloys industry **IC-WNDT-MI, 2016, Oran, Algeria**

<https://library.crti.dz/cf1669>

2- Chahrazad Erredir, Mohamed Lahdi Riabi, Halima Ammari, Emir Bouarroudj (**Teaching Learning Based Optimization Neural Networks for Waveguide Filter Modeling**) 7th African conference on non destructive testing ACNDT and 5th International conference on welding, non destructive testing and materials and alloys industry **IC-WNDT-MI, 2016, Oran, Algeria**

<https://library.crti.dz/cf1669>

3- Chahrazad Erredir, Mohamed Lahdi Riabi (**Design of Microwave Filter Using Segmentation, Coupling Matrix and Neural network**) 1st International Conference on Nanoelectronics, communications and Renewable Energy « **ICNCRE'13** », PP. 180-183, 2013 in Jijel, Algeria , ISBN: 978-81-925233-8-5

[www.edlib.asdf.res.in](http://www.edlib.asdf.res.in)

## III. Communications nationales

1- Ch. Erredir, M. L. Riabi (**Modélisation de filtres à iris capacitifs en guides d'onde rectangulaires par utilisation des réseaux de neurones**) La première conférence nationale sur les télécommunications « **CNT'2012**» 2012, Guelma, Algérie