

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L 'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



N° d'ordre :.....

Série :.....

Université Mohammed Seddik

BENYAHIA - Jijel

Faculté des Sciences et de la Technologie

Département : Electrotechnique

Thèse Présentée

Pour l'obtention du Diplôme de Doctorat 3ème cycle LMD
Spécialité : **Machines Electriques dans leur environnement**

Par :

BOURAHLA Kheireddine

Thème :

Adaptation des méthodes d'optimisation par intelligence artificielle pour la conception optimale des machines électriques

Soutenue le :... /.. / **2019** , Devant le Jury composé de :

Mr. MEKIDECHE Mohamed Rachid	Professeur	U.M.S.B Jijel	Président
Mme. BELLI Zoubida	Professeur	U.M.S.B Jijel	Rapporteur
Mr. HACIB Tarik	Professeur	U.M.S.B Jijel	Co-rapporteur
Mr. ALLOUI Loutfi	Professeur	U. Biskra	Examineur
Mr. ALLAG Hichem	Professeur	U. M.S.B Jijel	Examineur
Mr. REZIG Ali	Professeur	U. M.S.B Jijel	Examineur

Remerciements

Remerciements

Tout d'abord, je remercie ALLAH Le Tout Puissant qui m'a accordé la volonté et la patience pour réaliser cette thèse "El hamdo lillah".

Je voudrais exprimer ma profonde reconnaissance à mon directeur de thèse, Madame Zoubida BOULASSEL née BELLI, Maître de Conférences A à l'université de Jijel pour m'avoir encadré, orienté et guidé, tous ses précieux conseils, sa grande disponibilité, son aide sa gentillesse mais aussi ses vastes connaissances scientifiques, afin de réaliser ce travail. Je le remercie pour ses encouragements continus qui ne cessaient de me remonter le moral pendant les moments difficiles et pour l'énorme soutien scientifique et moral qu'elle a su m'accorder pendant ces cinq années et tout ses effort pour corriger cette thèse et mes travaux scientifiques.

A mon Co-encadreur monsieur Tarik HACIB professeur à l'université de Jijel, pour m'avoir Co-encadré, orienté et tous ses précieux conseils, sa grande disponibilité, son aide sa gentillesse mais aussi ses vastes connaissances scientifiques, afin de réaliser ce travail. Je le remercie pour ses encouragements continus qui ne cessaient de me remonter le moral pendant les moments difficiles et pour l'énorme soutien scientifique et moral qu'il a su m'accorder pendant ces cinq années.

Je voudrais plus particulièrement adresser mes remerciements à Monsieur Lionel VIDO Maître de Conférences à l'université de Cergy-Pontoise de m'avoir accueilli pour un stage au laboratoire SATIE, je le remercie pour ses idées, conseils, suivi au long de la durée du stage effectué.

J'adresse mes remerciements aux membres du jury qui m'ont fait honneur d'évaluer, d'examiner, et d'enrichir mon travail.

Mohamed Rachid MEKIDECHE, professeur à l'université de Jijel. Je le remercie particulièrement d'avoir accepté de présider le jury de soutenance.

Loutfi ALLOUI professeur à l'université de Biskra. Je le remercie de m'avoir honoré en acceptant d'être examinateur.

Hicham ALLAG, professeur à l'université de Jijel. Je le remercie de m'avoir honoré en acceptant d'être examinateur.

Ali REZIG Maître de Conférences à l'université de Jijel. Je le remercie de m'avoir honoré en acceptant d'être examinateur.

Je remercie aussi tout les membres de ma famille, mon père, ma mère, mes frères et mes sœurs. Aussi je ne peux pas oublier de remercier mes collègues doctorants, mes enseignants de département d'électrotechnique, mes amis et toutes personnes qui m'ont apporté un soutien moral de loin ou de près, et à tous ceux qui m'ont aidé à faire ce travail. Si j'en suis là aujourd'hui c'est principalement grâce à eux, Merci..., Je profite pour souhaiter bonne chance aux futurs docteurs.

Résumé

Résumé_ Dans cette thèse on a proposé de nouveaux algorithmes, qui appartiennent à la classe de l'intelligence artificielle, à partir de la combinaison ou l'hybridation entre différents algorithmes, dont le but est de les adapter à la conception optimale des machines électriques. On a proposé trois stratégies fondamentales d'hybridation, la première consiste à combiner les réseaux de neurones artificiels avec les deux méthodes de l'Intelligence d'Essaim, le PSO et le TLBO. Dans la deuxième stratégie on a combiné d'une part, la méthode de BA avec celle exacte basée sur la méthode du simplexe et d'autre part, la méthode de TLBO avec les deux méthodes exactes, simplexe et Quasi-Newton. Cependant, dans la troisième stratégie on a combiné entre la méthode BA et la technique de croisement, pareille à celle utilisée dans les algorithmes génétiques, et aussi entre les méthodes TLBO et la technique de la recherche locale de l'algorithme BA, d'où le nouveau algorithme TLBO-RLS. L'objectif de cette hybridation est de réduire les inconvénients des méthodes de l'Intelligence d'Essaim tels que les temps de calcul très élevés et la convergence prématurée.

Pour valider l'efficacité des algorithmes développés, ils ont été appliqués à trois fonctions test connues, après à trois problèmes de conception optimale en électrotechnique, le problème test international TEAM 25, un moteur synchrone à reluctance variable avec barrières du flux et une machine synchrone à double excitation.

Abstract

Abstract_ In this thesis a new algorithms were proposed, which belonging to the class of the artificial intelligence. They are based on the combination or hybridization between different algorithms. This hybridization aimed to adapt them for the optimal design of electrical machines. Three fundamental strategies of the hybridization have been proposed, the first one combines artificial neural networks with two methods of the swarm intelligence, PSO and TLBO. The second strategy combines, on the one hand, the BA method with an exact one using the simplex method and on the other hand, the TLBO method with two exact methods, Simplex and Quasi-Newton. However, in the third strategy, BA was combined with the crossover technique, as in genetic algorithms, and also TLBO was combined with the Random Local Search of the BA, that yields the TLBO-RLS algorithm. The objective of this hybridization is to reduce the disadvantages of swarm intelligence methods such as very high computation times and premature convergence.

To validate the effectiveness of the developed algorithms, they were applied initially to solve three known test functions ; after, to solve three problems of the optimal design in electrical engineering, the TEAM work shop international test problem 25, a Switched Reluctance Motor with flux barriers and a synchronous machine with double excitation.

ملخص

ملخص _ خلال هذه الاطروحة قمنا باقتراح خوارزميات من خلال المزج بين طرق أو خوارزميات موجودة وهذا بهدف تكيفها وجعلها ملائمة أكثر للتطبيق في مجال تصميم المركبات الكهرومغناطيسية. وقد قمنا بإتباع 3 استراتيجيات أساسية، تتمثل الأولى في مزج الشبكات العصبية مع خوارزمية مجموعة الجزئيات المتحركة أو أسراب الطيور و الخوارزمية المرتكزة على محاكاة عملية التعلم والتعليم. والإستراتيجية الثانية قامت على المزج بين خوارزميتي الخفاش و محاكاة عملية تعلم -تعليم مع طريقة نيلدر- ميد سمبلاكس و الإستراتيجية الثالثة تركز على المزج بين خوارزمية الخفاش والخوارزمية الجينية وأيضا مزج خوارزمية محاكاة عملية تعلم تعليم مع خوارزمية الخفاش، وقد جاء هذا المزج بغرض تجنب أو علاج العيوب في الطرق التقريبية مثل ضعف دقة الحل مقارنة ب الطرق الدقيقة ومشكل الوقت الكبير الذي تاخده لإيجاد الحل.

و بغرض التأكد من فعالية الخوارزميات المقترحة قمنا بتطبيقها على مشكلة تيم 25، وعلى محرك متزامن دو ممانعة مغناطيسية مع حاجز للتدفق، و محرك متزامن دو تغذية مضاعفة، بعد أن قمنا بتجربتها على ثلاث دوال اختباريه معروفة.

Table des matières

Table des matières.....	<i>i</i>
Introduction général.....	1
Chapitre I	
Optimisation: Etat de l'art	
I.1 Introduction.....	5
I.2 Définitions.....	5
I.2.1 Eléments fondamentaux d'un problème d'optimisation.....	6
<i>I.2.1.1 Fonction objectif et variables d'optimisation.....</i>	<i>6</i>
<i>I.2.1.2 Contraintes.....</i>	<i>6</i>
I.2.2 Formulation mathématique d'un problème d'optimisation.....	6
I.2.3 Optimum local et optimum global.....	7
I.2.4 Optimisation multi-objectif.....	7
I.2.5 Types de problèmes d'optimisation.....	8
<i>I.2.5.1 Problème d'optimisation sans contraintes.....</i>	<i>8</i>
<i>I.2.5.2 Problème d'optimisation avec contraintes.....</i>	<i>9</i>
I.3 Résolution d'un problème d'optimisation.....	10
I.3.1 Phase d'intensification.....	10
I.3.2 Phase de diversification.....	10
I.3.3 Méthodes de résolution.....	11
<i>I.3.3.1 Méta-heuristiques et heuristiques.....</i>	<i>11</i>
<i>I.3.3.2 Algorithmes directs et indirects.....</i>	<i>11</i>
<i>I.3.3.3 Méthodes exactes.....</i>	<i>11</i>
<i>I.3.3.4 Méthodes approchées.....</i>	<i>11</i>
I.3.4 Sensibilité d'une méthode d'optimisation.....	12
I.3.5 Robustesse d'une méthode d'optimisation.....	12
I.4 Classification des méthodes d'optimisation.....	12
I.4.1 Méthodes exactes.....	13
<i>I.4.1.1 Méthodes unidimensionnelles.....</i>	<i>13</i>
<i>I.4.1.2 Méthodes multidimensionnelles.....</i>	<i>14</i>
I.4.2 Méthodes approchées.....	17
I.5 Problème d'optimisation contraint et traitement des contraintes.....	22
I.5.1 Méthodes directes.....	22
I.5.2 Méthodes de transformation.....	23

1.5.2.1 Méthodes de pénalités.....	23
1.5.2.2 Méthode du lagrangien augmenté	24
I.6 Choix des Méthodes d'optimisation.....	25
I.7 Conclusion.....	25

Chapitre II

Méthodes d'Intelligence Artificielle

II.1 Introduction.....	27
II.2 Réseaux de neurones artificiels	27
II.2.1 Neurone formel et neurone biologique	27
II.2.2 Architecture des réseaux de neurones artificiels.....	31
II.2.2.1 Réseaux de neurones non bouclés (statiques).....	31
II.2.2.2 Réseau de neurones bouclé ou dynamique	32
II.3 Apprentissage du réseau de neurones.....	32
II.4 Réseaux de neurones multicouches (MLP).....	32
II.5 Algorithme rétro-propagation d'apprentissage.....	33
II.6 Sous-apprentissage, généralisation et sur-apprentissage	35
II.7 Intelligence d'Essaim (collective) ou Swarm Intelligence.....	36
II.7.1 Optimisation par Essaim de Particules (Particle Swarm Optimisation PSO).....	41
II.7.2 Optimisation basée sur Enseignement-Apprentissage (Teaching-Learning Based Optimisation TLBO).....	43
II.7.2.1 Phase d'enseignement.....	43
II.7.2.2 Phase d'apprentissage.....	43
II.7.3 Algorithme de la Recherche Coucou (Cuckoo search CS).....	45
II.7.3.1 Vol de Lévy	46
II.7.4 Algorithme des chauves-souris (Bat Algorithm).....	48
II.8 Choix du critère d'arrêt.....	52
II.9 Conclusion.....	53

Chapitre III

Conception optimale des machines électriques

III.1 Introduction.....	55
III.2 Etapes de la conception optimale des machines électriques.....	55
III.2.1 Définition de la conception optimale	55
III.2.2 Choix de la structure de la machine à concevoir	56
III.2.3 Dimensionnement des machines électriques.....	57

III.3 Démarche de la conception optimale	60
III.4 Modélisation des dispositifs électromagnétiques	63
III.4.1 Calcul du champ électromagnétique	64
III.4.2 Formulation avec le potentiel vecteur magnétique.....	65
III.4.2.1 Cas magnétostatique	65
III.4.3 Types de modélisation.....	66
III.5 Problèmes de conception optimale étudiés	69
III.5.1 TEAM workshop problem 25	69
III.5.1.1 Formulation du problème d'optimisation	71
III.5.2 Moteur synchrone à reluctance variable avec barrières du flux (MSRV)	71
III.5.2.1 Analyse du champ électromagnétique dans la MSRV	73
III.5.2.2 Formulation du problème d'optimisation.....	74
III.5.3 Machine synchrone à double excitation	76
III.5.3.1 Analyse du champ électromagnétique dans la MSDE.....	77
III.5.3.2 Formulation du problème d'optimisation.....	78
III.6 Conclusion	79

Chapitre IV

Stratégies d'hybridation appliquées aux algorithmes d'Intelligence d'Essaim

IV.1 Introduction.....	82
IV.2 Types d'hybridation.....	82
IV.3 Hybridation des réseaux de neurones artificiels avec des algorithmes d'Intelligence d'Essaim.....	83
IV.4 Hybridation entre méthodes méta-heuristiques et méthodes exactes.....	84
IV.4.1 Hybridation entre le BA et la méthode de Nelder-Mead simplex	85
IV.4.2 Hybridation entre le TLBO et les méthodes exactes: Nelder-Mead simplexe et Quasi-Newton	86
IV.5 Hybridation entre méthodes méta-heuristiques (méta-heuristique-méta-heuristique).....	87
IV.5.1 BA avec la technique de croisement (Crossover-BA ou C-BA)	87
IV.5.2 TLBO avec la technique de recherche locale aléatoire (Radom Local Search) TLBO-RLS	91
IV.6. Résultats et discussions.....	93
IV.6.1 Algorithmes hybrides réseaux de neurones- Intelligence d'Essaim : NN-PSO et NN-TLBO	96
IV.6.2 Algorithmes hybrides méta-heuristiques avec les méthodes exactes : NM-BA, TLBO-NM et TLBO-QN.....	100
IV.6.2.1 Nelder Mead-Chauve Souris: NM-BA	100
IV.6.2.2 Algorithmes hybrides TLBO-NM et TLBO-QN	103
IV.6.3 Algorithmes C-BA et TLBO-RLS	107

IV.6.3.1 Algorithme C-BA	107
IV.6.3.1 Algorithme TLBO-RLS	110
IV.7 Conclusion	113

Chapitre V

Résultats de l'application des techniques d'Intelligence Artificielle pour la conception optimale des machines électriques

V.1 Introduction.....	116
V.2 Modèle hybride réseau de neurone avec les méta-heuristiques : NN-PSO et NN-TLBO	116
V.2.1 Application au TEAM workshop problème 25	118
V.2.2 Machine synchrone à reluctance variable avec barrières du flux	118
V.3 Résultats des algorithmes hybrides TLBO-NM, TLBO-RLS, TLBO-QN, NM-BA et C- BA	123
V.3.1 TEAM workshop problème 25	123
V.3.1.1 Etude de la robustesse des algorithmes TLBO-RLS, TLBO et NM-BA.....	126
V.3.2 Machine à reluctance variable avec barrière du flux	128
V.3.2.1 Problème 1	128
V.3.2.2 Problème 2.....	130
V.3.2.3 Problème 2 cas du nouvel intervalle.....	133
V.3.3 Machine Synchrone à Double Excitation (MSDE)	137
V.3.3.1 Normalisation des fonctions f_1 et f_2	137
V.3.3.2 Choix des poids w_1 et w_2	138
V.3.3.3 Résultats d'optimisation	139
V.4 Conclusion	144
Conclusion générale et perspectives	146
Bibliographie.....	151

Introduction générale

Ces dernières décennies la conception des machines électriques a connu une très grande évolution. Par conséquent, actuellement les industriels sont obligés de faire face à un grand défi qui consiste à satisfaire les besoins et les demandes de la clientèle concernant la qualité des produits ainsi que le temps et le coût de leur réalisation. Cela, a intensifié la concurrence économique entre les entreprises industrielles qui se trouvent forcées à accélérer la cadence de leurs fabrications et le renouvellement de leurs produits par la mise en œuvre de technologies innovantes. De leur part les ingénieurs, les chercheurs et les spécialistes de la conception sont motivés, voir obligés de répondre aux exigences et aux contraintes imposées à la fois, par les industrielles et par les consommateurs [HAMO-12] et [BRIS-07]. Dans le domaine des machines électriques, de grands progrès concernant l'amélioration de leurs performances telles que le couple, les forces, le rendement, et le volume ont été réalisés, d'où l'émergence de nouveaux produits qui répondent de mieux en mieux aux objectifs attendus. Ces innovations sont le résultat des progrès réalisés dans la démarche de la conception optimale des machines électriques, où le développement des ordinateurs de grandes puissances a permis dans une large mesure le développement des outils de conception assistée par ordinateur, C.A.O, et l'automatisation de la démarche de conception [MOKR-05], [HAMO-12] et [BRIS-07].

Le développement des outils de calcul a permis aussi une modélisation fine et rigoureuse des phénomènes physiques qui se manifestent dans la machine électrique, conduisant ainsi à une meilleure exploration de l'espace de recherche et la localisation des meilleures solutions des problèmes posés. Pour plus d'efficacité, une démarche de conception par voie d'optimisation est la meilleure alternative possible [FONT-07] et [MOKR-05].

Cet avis est consolidé par l'apparition de nouveaux types d'algorithmes d'optimisation puissants qui s'adaptent facilement à tous types de problèmes d'optimisation, évidemment ceux liés aux problèmes électrotechniques en générales et les machines électriques en particulier, réputés de problèmes difficiles à cause de la complexité de leurs géométries et la nature des phénomènes physiques qui les caractérisent [FEYE-15], [HAJJ-03] et [MOKR-05].

Ce travail de thèse est versé dans la thématique de la conception optimale des machines électriques, où la démarche basée sur les méthodes d'optimisation associées à une modélisation par le calcul du champ électromagnétique est adoptée. Cette démarche a fait l'objet de plusieurs travaux [BELA-09], [BELL-8], [BRIN-14], [BRISS-07], [DOC-10], [DOGA-13], [HACI-08], [RODR-16], [MOKR-05], [HAJJ-03] et [TRAN-09]...etc.

Dans cette optique, l'intérêt principal du présent travail est concentré sur l'adaptation des méthodes d'optimisation par intelligence artificielle pour la conception optimale des

machines électriques, où un modèle numérique pour le calcul du champ électromagnétique basé sur la méthode des éléments finis est utilisé pour la modélisation des dispositifs considérés.

Le manuscrit est organisé en cinq chapitres.

Le premier chapitre est un état de l'art sur le problème d'optimisation englobant tous ses éléments : définitions, formulations mathématiques, méthodes de résolution et traitement de contraintes.

Le deuxième chapitre est axé autour des méthodes d'optimisation appartenant à la classe de l'intelligence artificielle, divisée en deux groupes, le premier concerne les réseaux de neurones artificiels, tandis que le deuxième concerne les méthodes méta-heuristiques, inspirées des lois de la nature, connues sous le nom de méthodes de l'Intelligence d'Essaim. Ces différentes méthodes sont présentées en détail dans ce chapitre.

Les différentes démarches de la conception optimale de machines électriques font l'objet du troisième chapitre. La démarche basée sur l'utilisation conjointe des méthodes d'optimisation et le modèle éléments finis pour le calcul du champ électromagnétique est explicitée. Les cahiers des charges des applications considérées dans ce travail sont également présentés dans ce chapitre. Où la première application est un problème test international, il s'agit d'une presse à aimant, le TEAM 25. La deuxième concerne un moteur synchrone à reluctance variable avec barrières du flux, tandis que la troisième est à propos d'une machine synchrone à double excitation.

Dans le quatrième chapitre on présente les différentes stratégies d'optimisation par intelligence artificielle développées dans ce travail, où la validité des algorithmes classiques et ceux modifiés, dans le but d'améliorer leurs performances, est étudiée à travers leur application pour la résolution des fonctions test connues. Les résultats de ces applications sont présentés, comparés et discutés dans ce même chapitre.

Le cinquième chapitre est consacré à la présentation des résultats obtenus d'applications de la démarche de conception optimale adoptée dans ce travail, basée sur l'association de calcul du champ par éléments finis aux différentes stratégies d'optimisation développées, pour la résolution des trois problèmes de conception optimale considérés.

Le mémoire est clôturé par une conclusion générale et des perspectives.

***Chapitre I : Optimisation: Etat
de l'art***

I.1 Introduction

L'être humain cherche toujours à améliorer sa vie quotidienne et cour sans cesse après la perfection. On rencontre plusieurs problèmes d'aide à la décision, de planification, d'identification et de gestion dans les différents secteurs, économiques, pédagogiques, sociales, scientifiques et industriels, où souvent, la qualité de la solution fournie ainsi que le respect de certaines règles et conditions spécifiques au problème posé, sont des facteurs essentiels, voir mêmes des exigences, qui s'imposent lors de sa résolution[GHER-13]. Cela, a incité les chercheurs dans les différents temps et disciplines de proposer et développer des méthodes pour trouver des solutions optimales qui assurent à la fois, une qualité acceptable et un temps de calcul raisonnable[GHER-13]. Cet axe de recherche réfère aux problèmes d'optimisation. L'apparition des ordinateurs puissants avec des processus de calcul très forts ainsi que le développement de l'informatique[GHER-13], ont permis l'évolution rapide de cet axe de recherche et encouragé les chercheurs à proposer un grand nombre d'algorithmes d'optimisation[GHER-13], qui utilisent des stratégies et des principes différents pour aboutir à la solution souhaitée. Actuellement, l'optimisation est devenue un outil indispensable pour résoudre les différents problèmes relatifs à la vie humaine.

Ce présent chapitre se focalise sur la présentation du problème d'optimisation. Au début, la définition et la formulation mathématique du problème sont abordées, après on présente les différents types de problèmes d'optimisation ainsi que les différentes classes de méthodes utilisées pour leur résolution.

I.2 Définitions

Le problème d'optimisation consiste à minimiser ou maximiser une grandeur ou un critère, qui peut être économique (productivité, coût d'investissement, coût de gestion,...etc.), technique (précision, stabilité...etc.) ou physique (quantité de chaleur, énergie, couple,...etc.), en respectant quelques exigences ou contraintes liées au problème. C'est à dire trouver les valeurs des variables les plus satisfaisantes au besoin désiré (généralement maximiser les performances et minimiser le coût d'une grandeur (critère) dépanadant de ces variables) [CULI-94] et [BELL-08].

I.2.1 Eléments fondamentaux d'un problème d'optimisation

I.2.1.1 Fonction objectif et variables d'optimisation

La fonction objectif est le critère à minimiser ou à maximiser selon le besoin. Elle est mathématiquement formulée en termes de variables, dites variables d'optimisation. Ces variables sont choisies parmi celles qui influencent la valeur de la fonction objectif et au même temps peuvent prendre des valeurs optimales dans le monde réel (parce que ce n'est pas toute solution obtenue est réalisable).

I.2.1.2 Contraintes

L'optimisation est effectuée généralement sous des contraintes de différentes natures et relations avec le problème. Cela limite le domaine de recherche des solutions et permet de donner des résultats réels et applicables (faisables) [DOGA-13]. Ces contraintes sont de deux natures différentes : celles relatives aux variables d'optimisation (l'espace de recherche est limité) et celles relatives aux grandeurs de sortie du modèle (l'espace de solutions est limité) [JANN-10].

I.2.2 Formulation mathématique d'un problème d'optimisation

Le problème d'optimisation se traduit mathématiquement par une fonction objectif f définie dans l'espace R , ou dans une partie de R vers R , dans lequel on cherche la solution X^* qui minimise f . En général, ce problème est écrit sous la forme suivante [SALD-92], [VASC-94], [MONM-00] et [N'TSH-11] :

$$\left\{ \begin{array}{l} \text{Min } f(x) \in R \\ g_i(x) \leq 0 \quad i = 1 \dots p \\ h_j(x) = 0 \quad j = 1 \dots m \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1 \dots n \end{array} \right. \quad (\text{I.1})$$

où:

$f(x)$ est le critère à minimiser, s'appelle la fonction objectif ou fonction coût;

$X = [x_1, x_2, x_3, \dots, x_n]$, est le vecteur des variables d'optimisation de dimension n , il présente les paramètres du problème à optimiser;

$x_{k \min}$ et $x_{k \max}$ sont les valeurs minimale et maximale de chaque variable, elles désignent les contraintes du domaine;

Les expressions $g_i(x)$ et $h_j(x)$ sont respectivement des contraintes d'inégalité et d'égalité.

Dans le cas d'un problème de maximisation (maximiser un critère), ça revient à le transformer à un problème de minimisation par l'utilisation de l'inverse ou l'opposé de $f(x)$. C'est-à-dire : $\min (-f(x))$; cela permet de conserver toutes les généralités citées précédemment [VASC-94], [BRIS-07], [RODR-16], [TRAN-09] et [ILEA-11].

I.2.3 Optimum local et optimum global

Il est nécessaire d'appréhender la différence entre un optimum global et un autre local. Sur la figure ci-dessous, les solutions d'une fonction objectif f à une variable x sont illustrées [DOGA-13], [SALD-92], [MONM-00], [CULI-94], [VANA-15] et [FONT-07].

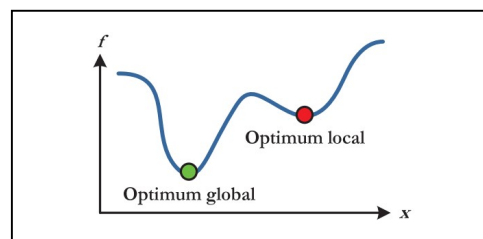


Figure I.1 Optimum local et optimum global [DOGA-13].

On peut remarquer que le vrai optimum est l'optimum global, tandis que l'optimum local est un « piège ». La connaissance de cette différence est primordiale dans un problème d'optimisation, parce que quelque soit l'algorithme utilisé, il peut tomber dans le problème de solution locale [DOGA-13], [SALD-92], [MONM-00], [CULI-94], [VANA-15] et [FONT-07].

I.2.4 Optimisation multi-objectif

L'optimisation multi-objectif consiste à chercher les valeurs des variables d'un problème qui maximisent ou minimisent une ou plusieurs fonctions objectifs, en respectant les contraintes, pour obtenir la solution. Elle procède donc par la définition au préalable des critères de qualité de la solution du problème, puis l'algorithme d'optimisation pour résoudre le problème en cherchant les meilleures solutions en fonction de ces critères.

Un problème d'optimisation multi-objectifs de dimension n peut être écrit de façon générale sous la forme suivante [DOC-10], [VASC-94], [CULI-94], [TRAN-09], [N'TSH-11], [RODR-16] et [BRIN-14]:

$$\left\{ \begin{array}{l} \text{Min } F(x) = [f_1(x), f_2(x), f_3(x), \dots, f_L(x)] \in R^n \\ g_i(x) \leq 0 \quad i = 1 \dots p \\ h_j(x) = 0 \quad j = 1 \dots m \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1 : n \end{array} \right. \quad (\text{I.2})$$

Où:

R^n est l'espace de recherche multidimensionnel (espace des fonctions objectifs $f_1(x)$, $f_2(x) \dots f_L(x)$ où $L=1 \dots n$) borné par les contraintes du domaine;

$X_k = [x_1, x_2, x_3, \dots, x_n]$, est le vecteur des variables d'optimisation de dimension n qui représentent les paramètres du problème à optimiser; $x_{k \min}$ et $x_{k \max}$ sont les valeurs minimale et maximale de chaque variable, elles désignent les contraintes du domaine;

$F(x)$ est le critère à minimiser appelé aussi fonction objectif ou fonction coût;

Les expressions $g_i(x)$ et $h_j(x)$ sont respectivement des contraintes d'inégalité et d'égalité.

Dans le problème (I.2), si L est égale à 1 le problème d'optimisation est mono objectif, dans ce cas on va chercher le minimum d'une seule fonction objectif. Dans le cas où L est égale à 2, on dit que le problème d'optimisation est bi-objectif, où multi-objectif. Si L est supérieur à 2, on va faire la minimisation de plusieurs fonctions objectifs simultanément [JANN-10] et [RODR-16].

I.2.5 Types de problèmes d'optimisation

D'après l'explication précédente, il est clair qu'il est possible de classer les problèmes d'optimisations selon l'existence des contraintes ou non en deux classes principales, le problème d'optimisation sans contraintes et celui avec ou sous contraintes. Les deux, peuvent être mono ou multi-objectif.

I.2.5.1 Problème d'optimisation sans contraintes

On dit qu'un problème d'optimisation est sans contraintes si aucune contrainte ou fonction contrainte est présente. La formulation d'un tel problème est écrite mathématiquement suivant la formule ci-dessous [CULI-94], [SALD-92], [HAJJ-03], [TRAN-09] et [VANA-15]:

$$\left\{ \begin{array}{l} \text{Min } f(x) \in R^n \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1, \dots, n \end{array} \right. \quad (\text{I.3})$$

➤ **Conditions d'optimalité**

Les conditions nécessaires pour que x^* est un minimum local d'un problème d'optimisation sans contraintes (I.3) sont données par l'équation (I.4) [CULI-94], [BRIS-07] [VANA-15] et [HACI-04]:

$$\begin{cases} H(x^*) \text{ non négatif} \\ \nabla f(x^*) = 0 \end{cases} \quad (\text{I.4})$$

avec:

∇f : est le gradient de la fonction objectif.

$H = \nabla^2 f$: est le Hessien de la fonction objectif (la matrice de dérivées secondes partielles de f).

Les conditions d'optimalité suffisantes pour que x^* soit minimum local sont données par l'équation (I.5) [CULI-94] et [DOC-10]:

$$\begin{cases} H(x^*) \text{ Positif} \\ \nabla f(x^*) = 0 \end{cases} \quad (\text{I.5})$$

Les conditions citées précédemment ne s'appliquent pas à des points situés sur les frontières de l'espace de recherche, elles sont uniquement valables pour des fonctions continûment différentiables.

I.2.5.2 Problème d'optimisation avec contraintes

Un problème d'optimisation est dit contraint ou avec contraintes, si des fonctions contraintes, d'égalité ou d'inégalité, ou les deux types au même temps, existent. La formule générale d'un problème d'optimisation contraint est donnée par l'équation (I.6) [CULI-94], [ANDR-13], [SAUV-00], [VASC-94], [SALD-92] :

$$\begin{cases} \text{Min } f(x) \in R \\ g_i(x) \leq 0 \quad i = 1 \dots p \\ h_j(x) \leq 0 \quad j = 1 \dots m \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1 \dots n \end{cases} \quad (\text{I.6})$$

Les contraintes dans un problème d'optimisation s'écrivent sous la forme d'une équation d'égalité ou d'inégalité. Généralement, dans le cas où des contraintes d'égalité, $h_j(x)=0$, existent, on peut les considérer à travers deux contraintes d'inégalité $h_j(x) \leq 0$ et $-h_j(x) \leq 0$; il en résulte qu'il est possible de remplacer les contraintes d'égalité dans un problème par des contraintes d'inégalité, et le problème (I.6) est simplifié selon la forme (I.7): [CULI-94] et [BRIS-07] :

$$\begin{cases} \text{Min } f(x) \in \mathbb{R}^n \\ g_i(x) \leq 0 \quad i = 1 \dots q, q = (p + m) \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1 \dots n \end{cases} \quad (\text{I.7})$$

La fonction (I.8) est appelée fonction de Lagrange, elle est associée au problème (I.7) [CULI-94], [BRIS-07] et [VANA-15] :

$$L(x, \lambda) = f(x) + \sum_{i=1}^q \lambda_i \cdot g_i(x) \quad (\text{I.8})$$

$\lambda_i \geq 0$ sont les multiplicateurs de Lagrange.

La condition nécessaire pour que x^* soit minimum local d'un problème d'optimisation contraint est donnée par les équations de Kuhn-Tucker [CULI-94], [VANA-15] et [VASC-94] :

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^q \lambda_i \times \nabla g_i(x^*) = 0 \\ \lambda_i \times \nabla g_i(x^*) = 0 \quad i \in [1, q] \end{cases} \quad (\text{I.9})$$

On peut écrire l'équation (I.9) à l'aide de l'équation (I.8) selon la forme suivante:

$$\begin{cases} \nabla L(x^*, \lambda^*) = 0 \\ \lambda_i \times \nabla g_i(x^*) = 0 \quad i \in [1, q] \end{cases} \quad (\text{I.10})$$

I.3 Résolution d'un problème d'optimisation

La résolution d'un problème d'optimisation consiste à chercher un optimum. Le processus de recherche se fait selon deux types ou bien passe par deux phases principales, l'intensification et la diversification.

I.3.1 Phase d'intensification

Dite aussi phase d'exploitation, c'est un type de recherche locale qui permet de bien exploiter l'entourage d'une solution trouvée et améliorer cette solution par une bonne recherche dans son voisinage. L'inconvénient de cette recherche est qu'elle peut être attrapée dans un optimum local.

I.3.2 Phase de diversification

Elle s'appelle aussi phase d'exploration, elle permet d'explorer d'autres espaces dans le domaine de recherche, loin de la solution déjà trouvée, dont le but est de trouver une

solution meilleure que celle existante, et donc éviter de se piéger dans une solution locale [JANN-10].

I.3.3 Méthodes de résolution

La résolution d'un problème d'optimisation est effectuée par des méthodes ou des algorithmes dont on trouve beaucoup de noms dans la littérature. Généralement, ces noms sont liés au problème d'optimisation, les plus souvent rencontrés sont : méta-heuristique et heuristique, directe et indirecte, exacte et approchée. Dans ce qui suit, on explique brièvement la signification de ces appellations.

I.3.3.1 Méta-heuristiques et heuristiques

Les deux méthodes partent du même principe qui consiste à trouver des solutions optimales en un temps acceptable, la différence entre les deux, est que les heuristiques sont spécifiques au problème traité, par contre, les méta-heuristiques sont carrément indépendantes du problème à résoudre [KAMM-06].

I.3.3.2 Algorithmes directs et indirects

Un algorithme est dit direct ou d'ordre 0, dans le cas où le calcul du gradient ou le hessien de la fonction objectif ou des contraintes n'est pas nécessaire, dans son processus il n'a seulement besoin que de la fonction objectif. Ce type d'algorithmes est très lent pour la convergence, il convient aux fonctions non continues et non différentiables [JANN-10].

On trouve aussi, les algorithmes indirects d'ordre 1 ou 2, la recherche de l'optimum dans ce cas, nécessite le calcul des dérivées partielles de la fonction objectif ou des fonctions contraintes. Ils sont très rapides grâce à l'information de la direction de recherche fournie par les dérivées [JANN-10].

I.3.3.3 Méthodes exactes

Ces méthodes reposent sur des processus qui assurent la solution optimale, soit locale ou globale, c'est-à-dire, donner la même solution à chaque exécution pour le même contexte initial (solution initiale, paramètres de contrôle de l'algorithme). Généralement, elles sont basées sur des principes mathématiques [KAMM-06].

I.3.3.4 Méthodes approchées

Sont des méthodes qui reposent généralement sur des bases probabilistes et aléatoires. Elles donnent une solution satisfaisante mais qui change avec chaque exécution même en utilisant la même condition initiale.

I.3.4 Sensibilité d'une méthode d'optimisation

Une méthode est dite sensible, dans le cas où elle exige la considération des valeurs initiales précises, des variables de conception et des paramètres de contrôle, pour assurer sa convergence vers la valeur optimale du problème [MERH-13] et [HAJJ-03].

I.3.5 Robustesse d'une méthode d'optimisation

On dit qu'une méthode d'optimisation est robuste dans le cas où elle peut donner des bons résultats quelques soit les conditions initiales (variables de conception et paramètres de contrôle de l'algorithme) [MERH-13] et [HAJJ-03].

I.4 Classification des méthodes d'optimisation

Pour résoudre un problème d'optimisation on peut utiliser plusieurs méthodes ou algorithmes. Ces méthodes ou algorithmes peuvent être classés selon plusieurs critères :

- Critère de besoin de la connaissance du gradient et la deuxième dérivée de la fonction objectif, ou non : méthodes directes ou indirectes.
- Critère de nombre de fonctions objectifs à minimiser, une seule fonction ou plusieurs fonctions objectifs au même temps : algorithmes multi-objectifs ou algorithmes mono objectifs.
- Critère de l'origine de la méthode, biologique, mathématique, physique, sociale,...etc.
- Critère d'adaptation à tous les problèmes ou non : algorithmes méta-heuristiques ou heuristiques.
- Critère de la précision des résultats, la robustesse et le mécanisme du processus de l'algorithme ou le mode de recherche de l'optimum [HAJJ-03] : méthodes exactes, dites aussi déterministes, ou approchées, dites aussi stochastiques.
- Critère du nombre de solutions initiales qui développent une solution unique (dit aussi parcours de solutions), ou à base de population (population est un groupe des solutions initiales). Dans certaines références les méthodes à base de population sont désignées par le parallélisme.
- Critère d'utilisation d'un seul algorithme ou le regroupement de plusieurs algorithmes déjà existants pour former un seul algorithme dit hybride : hybridation.

D'autres critères existent, comme par exemple le nombre de variables dans la fonction objectif, méthodes unidimensionnelle ou multi dimensionnelle, le type de recherche de la méthode, locale ou globale.

I.4.1 Méthodes exactes: dites aussi déterministes

Suivant la fonction objectif, mono-variable ou multi-variables, ou suivant la dimension du problème d'optimisation, on peut classer les méthodes déterministes en deux types : unidimensionnelles et multidimensionnelles [CULI-94], [BELL-08], [QUAR-04] et [COST-01].

I.4.1.1 Méthodes unidimensionnelles

On dit qu'une méthode d'optimisation est unidimensionnelle, si elle traite spécialement les problèmes d'optimisation à une seule variable. Parmi ces méthodes, on trouve la méthode de Dichotomie [CULI-94], la méthode de la section dorée [CULI-94] et celle de Brent [COST-01]. D'autres méthodes existent dont les plus connues sont présentées dans la figure I.2.

Ces méthodes exigent généralement que la fonction objectif est uni modale. Par contre, elles n'exigent pas la continuité ou la dérivabilité de la fonction. Leur principe de minimisation ou de recherche est basé sur la localisation du point minimal par la réduction successive de l'intervalle de recherche.

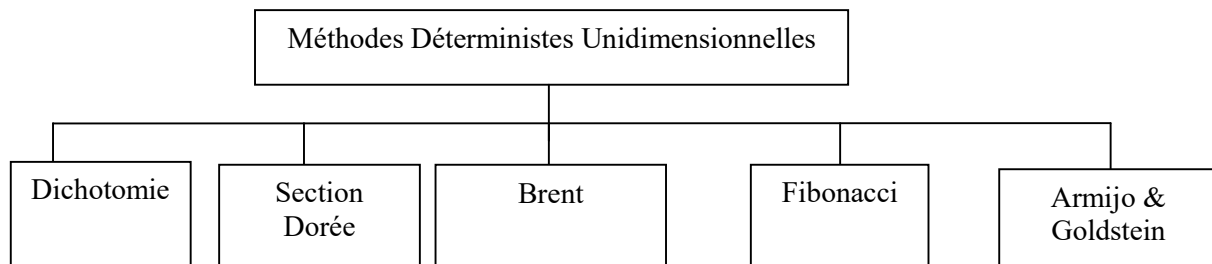


Figure I.2 : Principales méthodes déterministes unidimensionnelles.

Dans ce qui suit, on va présenter quelques méthodes unidimensionnelles.

1. Méthode de Dichotomie

Comme il a été déjà mentionné, cette méthode est unidimensionnelle, démarre le processus à partir des trois points (x_1, x_2, x_3) appartenant à l'intervalle de recherche, où x_1 est le point minimum, x_2 celui maximum et x_3 généralement le milieu de cet intervalle, avec $f(x_3) < f(x_2) < f(x_1)$. Dans la deuxième étape, on subdivise l'intervalle initial à deux autres intervalles $[x_1, x_3]$ et $[x_3, x_2]$ de façon à obtenir de nouveaux points (x_4, x_5) sur lesquels la fonction objectif sera évaluée [CULI-94], [BELL-08], [QUAR-04] et [COST-01].

A partir des valeurs de la fonction objectif en ces points, on choisit le nouveau triplet pour le prochain découpage. Ainsi, à chaque itération du processus, on utilise deux nouvelles évaluations de la fonction objectif pour réduire l'intervalle de recherche. Le processus itératif

s'arrête lorsque l'intervalle de découpage devient plus petit à une certaine tolérance ε donnée [CULI-94], [BELL-08], [QUAR-04] et [COST-01].

2. Méthode de Brent

Comme la précédente, cette méthode est aussi basée sur la réduction de l'intervalle ou l'espace de recherche, où à chaque itération on obtient un nouvel intervalle plus petit que le précédent. La différence entre cette méthode et celle de Dichotomie est que la méthode de Brent choisit le point de découpage de l'intervalle, en partant par exemple des points x_1 , x_2 et x_3 où les valeurs de la fonction objectif sont $f(x_1)$, $f(x_2)$ et $f(x_3)$, par interpolation on fait une approximation parabolique nommée F de la fonction originale f , le minimum de la fonction parabolique approximative F , nommé x_4 est choisi comme un point de découpage, (généralement le minimum de la fonction approximative F n'est pas loin du minimum de la fonction originale f), puis on calcule $f(x_4)$, et on compare sa valeur avec la valeur de la fonction objectif f précédente, et on choisit les 3 meilleures valeurs de x pour la prochaine itération, ainsi de suite jusqu'à que l'intervalle de recherche est plus petit qu'une certaine valeur epsilon [COST-01].

I.4.1.2 Méthodes multidimensionnelles

Les méthodes multidimensionnelles sont utilisées pour résoudre les problèmes d'optimisation où la fonction objectif possède plusieurs variables, on peut classer ces méthodes en deux groupes : méthodes analytiques (de descente) et méthodes géométriques (heuristiques).

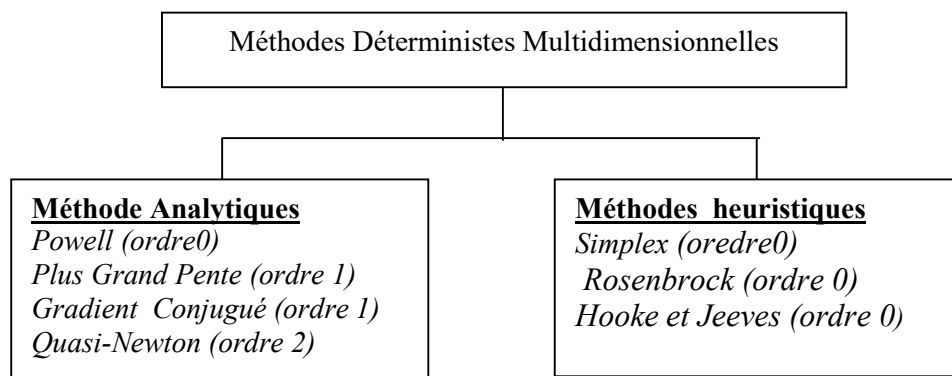


Figure I.3: Principales méthodes déterministes multidimensionnelles [COST-01]

1. Méthodes analytiques

Généralement ces méthodes sont d'ordre 1 ou 2. Elles sont basées sur la connaissance de la direction de recherche et convergent vers l'optimum le plus proche par des minimisations successives. Parmi les méthodes analytiques, on trouve la méthode de la *Plus Grande Pente* [CULI-94], [QUAR-04], Newton [CULI-94], [QUAR-04], le

Gradient Conjugué [CULI-94], [QUAR-04], la méthode de *Powell* [COST-01] et les méthodes *Quasi-Newton*, [CULI-94], [QUAR-04] et [COST-01]. On présente ci-dessous, les méthodes Quasi-Newton, considérées comme les méthodes analytiques les plus communément utilisées.

➤ **Méthodes Quasi-Newton**

La méthode de Newton est basée sur le calcul de la deuxième dérivée (hessien) de la fonction objectif à chaque itération. Dans la pratique, l'approximation de la matrice hessienne ou parfois son inverse par une matrice symétrique positive est plus efficace que le calcul exact. Cette approximation est connue sous le nom de la méthode Quasi-Newton.

On considère H_k l'approximation de la matrice hessienne à l'itération k_{th} . La forme générale de la méthode Quasi-Newton peut s'écrire comme suit:

Étape 0: Choisir un point initial x_1 de n dimensions et la définition de la matrice symétrique positive H_1 . On pose $\mathcal{E} > 0$ et $k = 1$.

Étape 1: Si le gradient $g(x)$ à x_k satisfait $\|g_k\| \leq \mathcal{E}$ stop.

Étape 2. Calculer la direction de recherche d_k à travers la résolution de l'équation linéaire suivante :

$$H_k d_k + g_k = 0 \quad (\text{I.11})$$

Étape 3: Chercher suivant la direction d_k le pas α_k et poser

$$x_{k+1} = x_k + \alpha_k d_k = 0 \quad (\text{I.12})$$

Étape 4: Sélectionner la nouvelle matrice symétrique positive H_{k+1} qui satisfasse l'équation originale de Quasi-Newton suivante:

$$H_{k+1} S_k = Y_k \quad (\text{I.13})$$

avec: $S_k = x_{k+1} - x_k$, $Y_k = g_{k+1} - g_k$ où g_j est le gradient de f à x_j .

Étape 5 : Poser $k = k + 1$ et retourner à l'étape 1.

Un facteur important pour l'utilisation de la méthode Quasi-Newton est l'adaptation de la formule de la matrice H_{k+1} en partant de la matrice courante H_k et utilisant l'information disponible à l'itération k_{th} . Les deux formules les plus utilisées pour l'adaptation de H_{k+1} sont données ci-dessous [Wei-06] :

-Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$H_{k+1} = H_k - \frac{H_k S_k S_k^T H_k}{S_k^T H_k S_k} + \frac{Y_k Y_k^T}{S_k^T Y_k} \quad (\text{I.14})$$

-Davidon-Fletcher-Powell (DFP)

$$H_{k+1} = H_k - \left(1 + \frac{S_k^T H_k S_k}{S_k^T Y_k}\right) \frac{Y_k Y_k^T}{S_k^T Y_k} - \frac{Y_k S_k^T H_k + H_k S_k Y_k^T}{S_k^T Y_k} \quad (\text{I.15})$$

2. Méthodes géométriques

Généralement se sont des méthodes d'ordre 0, ne demandent pas la connaissance du gradient de la fonction objectif. Elles sont basées sur la connaissance de la direction favorable et la plus prometteuse qui améliore la solution courante par exploration du domaine (l'espace de recherche) par des essais successifs. Les méthodes géométriques les plus utilisées sont, la méthode du *Simplex* [NELD-65] et [CULI-94], la méthode de *Rosenbrock* [CULI-94] et [MOKR-05], et la méthode de variations locales de *Hooke et Jeeves* [COST-01].

➤ Algorithme du Simplex (Nelder Mead) : NM

La méthode du simplex était proposée par Nelder et Mead (1965), elle vient de la méthode de recherche linéaire libre [FAN-06], en général cette méthode est considérée directe [LAGA-98], parce qu'elle n'utilise pas l'information du gradient, elle est basée sur la recherche locale par quatre opérations pour trouver l'optimum: réflexion, expansion, contraction et raccourcissement, ces étapes sont données en détail dans [WANG-11].

Soit $f(x)$ la fonction à optimiser, et x_i la liste des points du simplex courant, $i=1, \dots, n+1$.

Les étapes de la technique NM sont les suivantes:

1. **Ordonnement** : Ordonner $n+1$ points du simplexe selon la formule suivante : $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$.

2. **Réflexion** : calculer x_r , et $f_r = f(x_r)$ par :

$$x_r = x_b + \rho (x_b - x_{n+1}) \quad \text{avec, } \rho > 0 \quad (\text{I.16})$$

avec : $x_b = \frac{1}{n} \times \text{somme}(x_i)$, x_b : centre de gravité de n meilleurs points

Si $f_l \leq f_r \leq f_n$, on accepte le point de réflexion x_r et terminer l'itération.

3. **Expansion**. Si $f_r < f_l$, calculer le point expansion x_e et $f_e = f(x_e)$ par:

$$x_e = x_b + \chi (x_r - x_b); \quad \text{avec } (\chi > 1 \text{ et } \chi < \rho) \quad (\text{I.17})$$

Si $f_e < f_r$, accepter x_e , et terminer l'itération; sinon accepter x_r et terminer l'itération.

4. **Contraction**: Si $f_r \geq f_n$, exécuter la contraction entre x_b et la meilleure parmi x_{n+1} et x_r .

a. **Contraction extérieure**. Si $f_n \leq f_r < f_{n+1}$, calculer la contraction extérieure x_c et $f_c = f(x_c)$. par

$$x_c = x_b + \gamma (x_r - x_b), \quad (0 < \gamma < \rho) \quad (\text{I.18})$$

si $f_c < f_r$, accepter x_c et terminer l'itération; sinon, continuer à 5.

b. Contraction intérieure. Si $f_r \geq f_{n+1}$, calculer la contraction intérieure du point x_{cc} et $f_{cc} = f(x_{cc})$ par :

$$x_{cc} = x_b - \gamma (x_b - x_{n+1}) \quad (\text{I.19})$$

si $f_{cc} < f_{n+1}$, accepter x_{cc} et terminer l'itération; sinon, continuer à 5.

4. Raccourcir. Calculer les n points et $f(v_i)$ par :

$$v_i = x_i - \sigma (x_i - x_1) \quad , 0 < \sigma < 1 \quad i = 2, \dots, n+1 \quad (\text{I.20})$$

Les résultats des itérations est un nouveau sommet ou point unique, qui remplace le plus mauvais point actuel (x_{n+1}) dans l'ensemble des sommets de la prochaine itération, ou un ensemble de n nouveaux points du simplexe à la prochaine itération associé au meilleur point actuel (x_1), si le raccourcissement est exécuté. Les itérations se poursuivent jusqu'à ce qu'elles répondent aux critères d'arrêt [BARA-11] et [TFAI-07].

➤ **Méthode de relaxation de Rosenbrock**

Cette méthode est basée sur l'idée de transformer le problème d'optimisation de n variables (multi variables) à la résolution de n problèmes mono-variables. Le processus de transformation s'effectue par la fixation de toutes les variables, sauf une restante, à une valeur, par exemple la première, donc on obtient un problème mono variable, après on minimise la fonction objectif de notre problème avec une seule variable par l'une des méthodes citées précédemment, par exemple Brent ou Quasi-Newton, puis on fixe la première variable, qui a déjà minimisé la fonction objectif, et aussi les autres variables restent fixées sauf la deuxième qui a pris le rôle de minimiser la fonction objectif, ainsi de suite avec le reste des variables, et on continue jusqu'à un nombre d'itérations maximal ou un critère d'arrêt vitrifié [CULI-94] et [MOKR-05].

I.4.2 Méthodes approchées

Aussi appelées méthodes stochastiques, basées sur un mécanisme aléatoire, c'est à dire plusieurs exécutions pour la même solution initiale donnent des résultats différents, la plus part de ces méthodes sont des méthodes globales, et n'ont pas besoin ni du gradient, ni de la continuité ou la dérivabilité de la fonction objectif ou les fonctions contraintes. Elles peuvent être classées en deux groupes : les méthodes à base de population et les méthodes par parcours ou à solution unique.

➤ **méthodes à base de population**

Ce type de méthodes démarre par un groupe ou une population des solutions initiales. Dans ce type, on trouve les méthodes évolutionnistes et les méthodes d'Intelligence d'Essaim ou *Swarm Intelligence*, comme l'Optimisation par Essaim de

Particule ou (*Particle Swarm Optimization*) PSO, l'Optimisation par Colonie de Fourmis ou (*Ant Colony Optimization*) ACO, l'Optimisation Basée sur l'Enseignement_Apprentissage ou (*Teaching _Learning based Optimization*) TLBO, et Algorithme de Chauves -Souris ou (*Bat Algorithm*) BA.

1. Méthodes évolutionnistes

Sont des méthodes stochastiques à base de population, elles font appel à un ensemble ou un groupe des méthodes pour simuler le mécanisme de l'évolution des espèces ou la théorie de Darwin, parmi ces méthodes, on trouve les *Algorithmes Génétiques*, les *Stratégies d'Évolution*, la *Programmation Évolutionniste* et la *Programmation Génétique* [COST-01] et [MOKR-05]. Ces méthodes utilisent des mécanismes composés permettant de conduire une nouvelle génération ou une nouvelle population par quelques opérations comme la sélection et l'élimination [MOKR-05]. La méthode des algorithmes génétiques est la plus utilisée, dans ce qui suit on donne une présentation sommaire de la méthode.

Algorithme génétique

Cet algorithme était développé par Holland en 1975, inspiré de l'évolution naturelle de Darwin, qui dit que les êtres vivants les plus adaptés à leur environnement ont plus de chance pour vivre, tandis que les autres meurent. Aussi la science de la génétique qui explique la méthode du transfère de l'information génétique par les parents vers les enfants. Pour bien expliquer cet algorithme, premièrement on présente l'analogie existante entre l'algorithme génétique et le théorème de sélection de Darwin, tel est montré dans le tableau I.1.

Théorème de sélection naturelle de Darwin ou d'évolution naturelle	L'algorithme génétique
Population	Ensemble de solutions
Environnement	Espaces de recherche
Génération	Itérations
Gènes	Paramètres d'optimisation
Chromosomes ou individus	Individus (Ensemble de paramètres d'optimisation)
Adaptation de l'individu	Evaluation de la fonction objectif

Tableau I.1 Analogie entre les algorithmes génétiques et la théorie de l'évolution Darwinienne [COST-01].

Dans le domaine d'optimisation, le critère à minimiser est formulé sous forme d'une fonction mathématique, dite fonction objectif, et selon ses valeurs on peut connaître ce

que l'individu, contenant les paramètres d'optimisation, est adapté ou non à l'objectif d'optimisation. A chaque génération les plus mauvais individus (fonction objectif maximale) sont éliminés. Dans la théorie de la sélection naturelle les meilleurs individus, construits par des gènes, possèdent plus de chance pour se marier et créer des enfants de la nouvelle génération, en optimisation l'algorithme génétique et par des opérateurs tels que la sélection, le croisement et la mutation simule ce comportement et permet d'obtenir une nouvelle population (solutions) [MENG-15]. Le processus d'optimisation par l'algorithme génétique est illustré dans la figure I.4.

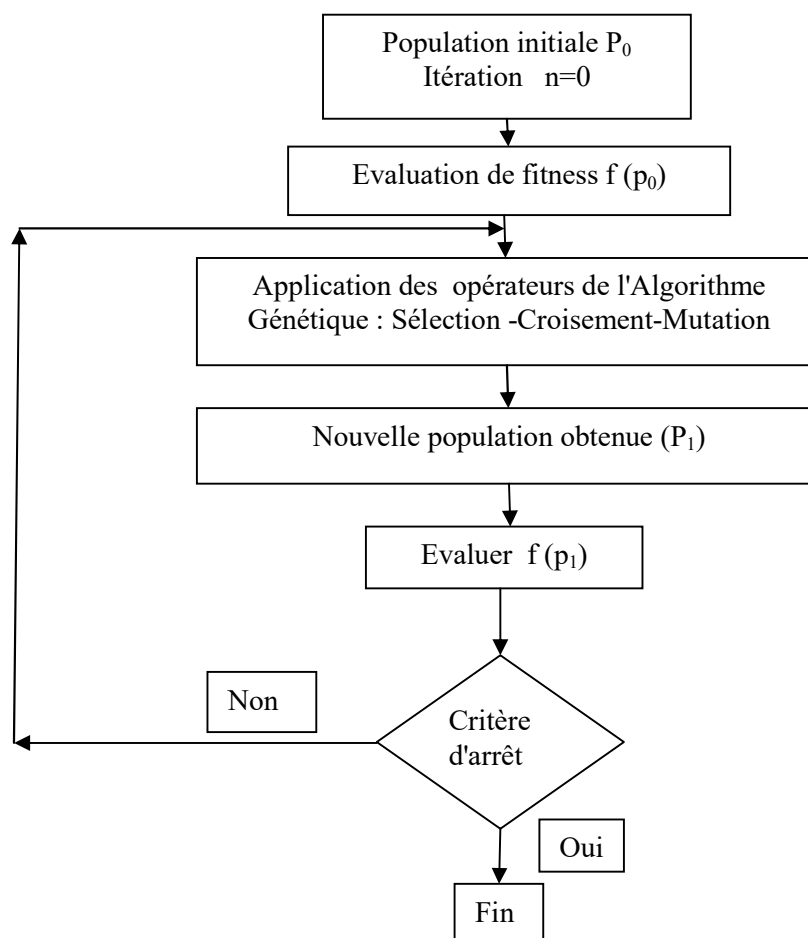


Figure I.4: Processus d'optimisation par l'algorithme génétique

2. Optimisation par essaim de particules (Particle Swarm Optimization) PSO

C'est une méthode méta-heuristique, basée sur le comportement social de quelques espèces comme les nuages d'oiseaux et les bancs de poissons, où par exemple, chaque oiseau dans un espace de recherche est influencé dans sa recherche par deux facteurs, sa position et la position du meilleur oiseau dans le groupe pour trouver la nourriture. Le PSO simule ce comportement collectif par la notion de coopération, contrairement au comportement d'élimination dans l'algorithme génétique. Cet algorithme apparait en 1995, il est très

populaire, son processus commence par une population initiale distribuée aléatoirement dans un espace de recherche, chaque individu dans la population d'oiseaux est évalué, ce qui permet de connaître les solutions initiales, locales et globales, puis il est déplacé dans l'espace de recherche avec une certaine vitesse, cette vitesse utilise la solution globale temporaire et la meilleure position locale de chaque individu. On répète le processus jusqu'à ce que toute la population converge vers la solution globale, ou un critère d'arrêt est vérifié, ou le nombre maximal d'itérations est atteint [ILEA-11] et [FEYE-16].

3. *Algorithme de colonie de fourmis (Ant Colony Optimization) ACO*

Cet algorithme est basé sur la coopération ou le comportement collectif des fourmis dans la mission de chercher la nourriture, et leur interaction avec l'environnement, où on peut voir que les fourmis sont capables de trouver le chemin le plus court entre leur maison et l'origine de la nourriture. Malgré qu'il n'existe pas de chef qui dirige les fourmis, elles utilisent l'environnement comme milieu permettant de transformer ou échanger les informations entre elles, par dépôt d'un produit chimique, dit phéromone, sur la route. Au début, les fourmis sont distribuées aléatoirement autour de la colonie pour trouver la nourriture, après la localisation de nourriture, chaque fourmi revient à la colonie et pose la phéromone sur la route, où la ligne la plus concentrée à la phéromone attire les fourmis de la colonie, et représente le chemin le plus court [KAMI-08], [DORI-04] et [GHER-13].

L'algorithme ACO, simule ce comportement de recherche de la nourriture chez les fourmis. Où, à partir d'une population initiale de fourmis distribuée aléatoirement dans l'espace de recherche (chaque fourmi représente une solution), la quantité de phéromone artificielle est déposée selon la qualité de la solution, où la meilleure solution prend la grande quantité de phéromone artificielle déposée (exprimée par des coefficients), donc les autres fourmis artificielles (solutions) suivent cette direction, où existe la meilleure solution, pour bien exploiter l'entourage de cette solution à chaque itération jusqu'à l'arriver à l'optimum ou un critère d'arrêt est vérifié [KAMI-08], [DORI-04] et [GHER-13].

4. *Algorithme des chauves souris (Bat Algorithm)*

C'est un algorithme méta-heuristique inspiré de la biologie, plus précisément du comportement d'écholocation des chauves souris, initialement proposé par Xin Yang en 2010, cet algorithme simule la méthode des micro chauves souris (micro-bats) pour trouver la proie et éviter les obstacles par l'émission d'un cri ultrason dans l'environnement, ces ultrasons tombent la proie, et leur réflexion aux oreilles des chauves souris par le biais des obstacles, leur permettre d'évaluer le volume, la forme et la distance de cette proie.

Dans cet algorithme, basé sur la notion de population, chaque chauve souris représente une solution locale dans les espaces de recherche où, bien sûr la qualité de chaque solution (chauve souris) est évaluée par la fonction objectif. Les chauves souris changent leur position ou leur place dans l'espace de recherche en se basant sur les résultats locaux précédents. La meilleure chauve souris dans la population est celle qui donne la meilleure valeur de la fonction objectif. En utilisant quelques paramètres comme la fréquence et la vitesse, à chaque itération de l'algorithme, la meilleure chauve souris passe par une étape de recherche locale pour bien exploiter l'entourage de cette solution, il faut dire aussi que cet algorithme possède quelques degrés de ressemblance avec le PSO [BENM-12], [AMRA-15] et [MENG-15].

➤ **Méthode par parcours (à solution unique)**

Comme les méthodes déterministes, ces méthodes nécessitent un point initial sauf qu'elles sont des méta-heuristiques basées sur un mécanisme aléatoire, parmi elles on trouve la recherche taboue et le recuit simulé.

1. Recherche Taboue

C'est une méta-heuristique basée sur la recherche locale, cet algorithme qui simule la mémoire du cerveau humain était initialement proposé par Glover en 1986, il démarre la recherche par un point initial, considéré comme meilleure solution, son processus de recherche de l'optimum est basé sur la création de quelques points ou des solutions au voisinage de ce point, successivement dans chaque itération, où chaque solution de l'ensemble des points est testée et ajoutée à une liste dite liste taboue, c'est à dire, interdire son calcul une autre fois tant que cette solution se trouve dans la liste interdite. La recherche taboue permet des fois de dégrader la meilleure solution courante pendant le processus, par exemple prendre une solution de qualité inférieure à celle de la solution précédente dans le même ensemble des solutions voisines, mais elle conserve toujours la meilleure solution en mémoire, chose qui permet d'éviter de tomber dans une solution locale. La taille de la liste taboue est limitée, chaque fois qu'une nouvelle solution rentre dans la liste l'ancienne sorte, le processus itératif continu jusqu'à qu'un nombre maximal d'itérations soit atteint ou un critère d'arrêt soit vérifié [KAMI-08] et [HAJJ-03].

2. Recuit Simulé

C'est une méthode méta-heuristique proposée en 1983 par Kirkpatrick, ce chercheur, et sur la base des travaux de Metropolis, observait qu'en métallurgie, le processus pour obtenir un métal très stable, c'est à dire les atomes sont bien organisés donc leur énergie est minimale, démarre par une température maximale, puis faire un

refroidissement lent, dans le cas contraire où le refroidissement est brusque ou rapide, le métal obtenu n'est pas stable et son énergie reste grande. A partir de cette observation, le chercheur a proposé l'algorithme du Recuit Simulé, qui simule les problèmes d'optimisation par un processus métallurgique, où la fonction objectif représente l'énergie du système et les paramètres d'optimisation représentent les atomes. Cet algorithme possède deux paramètres de contrôle qui sont la température T et le nombre d'itération n .

Le processus démarre par une solution initiale choisie aléatoirement, puis génère une autre solution dans le voisinage en utilisant un pas adapté au problèmes traité, après, il calcule la différence entre les valeurs de la fonction objectif pour la solution actuelle et pour la précédente, si la différence est négative $\Delta f < 0$, on remplace la solution précédente par la solution actuelle, dans le cas contraire on accepte cette solution avec une certaine probabilité, dite la probabilité de Boltzmann. Le processus itératif continu jusqu'à que la température devient minimal ou le nombre des itérations maximal soit atteint ou un autre critère d'arrêt soit vérifié [COST-01] et [KAMI-08].

I.5 Problème d'optimisation contraint et traitement des contraintes

Les méthodes d'optimisation citées précédemment sont appliquées à des problèmes sans contraintes, dans le cas d'un problème contraint, il faut faire une attention spéciale pour résoudre ce type de problèmes, où la solution n'est pas valable sauf si elle respecte les contraintes existantes [SALD-92], [CULI-94] et [COST-01].

Deux méthodes existent pour traiter le problème des contraintes, la première transforme le problème contraint à un autre non contraint et la deuxième traite le problème sans modification. La première catégorie s'appelle méthodes de transformation et la deuxième s'appelle méthodes directes;

I.5.1 Méthodes directes

Ces méthodes sont capables de prendre en compte les contraintes sans aucune modification du problème d'optimisation original, soit par une suite des traitements unidirectionnels, soit par le remplacement par une suite de sous problèmes approchés. Parmi ces méthodes on trouve, la *Programmation Quadratique Réursive*, la *Méthode de l'Ellipsoïde*, la *Méthode des Directions Admissibles*, la *Méthode du Gradient Réduit* et la *Méthode du Gradient Projeté* [SALD-92], [CULI-94] et [COST-01].

I.5.2 Méthodes de transformation

L'idée de base de ce type de méthodes est de transformer le problème d'optimisation avec contraintes à un problème sans contraintes, cette transformation permet l'utilisation directe des méthodes d'optimisation citées précédemment sans aucune modification. Parmi ces méthodes on trouve: les *méthodes de pénalités*, la *méthode du Lagrangien augmenté*, la *méthode des variables mixtes* et la *méthode des asymptotes mobiles*. Les méthodes de pénalités sont les plus utilisées [SALD-92], [CULI-94] et [COST-01].

I.5.2.1 Méthodes de pénalités

Ces méthodes sont basées sur la transformation du problème avec contraintes à un autre sans contraintes par l'intégration des contraintes dans la fonction objectif à optimiser toute en utilisant la fonction de pénalité $h(x)$. La nouvelle fonction à minimiser φ est donnée sous la forme suivante [SALD-92], [VASC-94], [CULI-94], [JANN-13] et [MOKR-05] :

$$\varphi(x, r) = f(x) + r \cdot h(x) \quad (\text{I.21})$$

où:

$h(x)$: Fonction de pénalités, continue et dépendante des contraintes $g(x)$.

r : Coefficient de pénalité, toujours positif.

On distingue deux types de méthodes de pénalités, méthode de pénalités intérieures et méthode de pénalités extérieures.

➤ Méthode de pénalités extérieures

La fonction de pénalité extérieure est continue et à dérivées continues, elle est écrite sous la forme suivante [SALD-92], [VANC-94], [CULI-94], [JANN-13] et [MOKR-05] :

$$h(x) = \sum_{i=1}^q \max^2(0, g_i(x)) \quad (\text{I.22})$$

Cette méthode permet de démarrer la recherche par des points qui n'appartiennent pas obligatoirement au domaine admissible, et se rapproche à la solution optimale par une suite de solutions irréalisables, mais pour assurer la solution de ce problème et éviter des problèmes numériques lors de la résolution, on utilise un processus itératif, où à chaque itération k , une suite de valeurs croissantes de r est générée. Le problème s'écrit sous la forme suivante [SALD-92], [VASC-94], [CULI-94], [JANN-13] et [MOKR-05] :

$$\varphi(x, r, k) = f(x) + r^k \sum_{i=1}^q \max^2(0, g_i(x)) \quad (\text{I.23})$$

Le coefficient r est choisi assez grand pour assurer la convergence du problème.

➤ **Méthode de pénalités intérieures**

Cette méthode ne convient pas aux problèmes qui possèdent des contraintes d'égalité, et contrairement à la méthode de pénalités extérieures, propose que la solution initiale doive obligatoirement appartenir au domaine admissible et assurer aussi que toutes les solutions générées appartiennent au domaine admissible durant le temps du processus d'optimisation, chose qui demande l'utilisation des autres algorithmes pour trouver le point initial [SALD-92], [VASC-94], [CULI-94], [HAJJ-03] et [MOKR-05]. Les fonctions de pénalités les plus utilisées sont:

- La fonction inverse:

$$h(x) = \sum_{i=1}^q \left(-\frac{1}{g_i(x)} \right) \quad (\text{I.24})$$

- La fonction logarithmique:

$$h(x) = -\log(-g_i(x)) \quad (\text{I.25})$$

Dans le cas de la fonction inverse, la fonction objectif du problème d'optimisation sans contraintes (I.21) est remplacée par la fonction suivante:

$$\varphi(x, r) = f(x) + r^{-k} \sum_{i=1}^q \left(-\frac{1}{g_i(x)} \right) \quad (\text{I.26})$$

Le coefficient r est choisi assez grand.

L'inconvénient de cette méthode, comme l'illustre le paragraphe passé, c'est la discontinuité entre les intervalles ou espaces admissibles et non admissibles et aussi le problème de trouver un point initial respectant les contraintes [SALD-92],[VASC-94], [CULI-94], [COST-01], [HAJJ-03] et [ANDR-13].

I.5.2.2 Méthode du lagrangien augmenté

Cette méthode de transformation, comme l'indique son nom, utilise la fonction lagrangienne classique $L(x, \lambda)$ comme dans (I.8), en ajoutant de nouveaux termes qui représentent la fonction de pénalités dépendantes des contraintes [ANDR-13] et [MOKR-05] :

$$L(x, \lambda, r) = f(x) + \sum_{i=1}^q \lambda_i \cdot g_i(x) + r \sum_{i=1}^q g_i^2(x) \quad (\text{I.27})$$

où:

$r \sum_{i=1}^q g_i^2(x)$: représente la fonction de pénalités.

Cette méthode nécessite le calcul des dérivées des contraintes et de la fonction objectif. Donc, elle est difficile à mettre en œuvre dans le cas de l'utilisation des éléments finis [HAJJ-03].

I.6 Choix des Méthodes d'optimisation

Chaque classe ou chaque méthode possède des avantages et des inconvénients, il n'y a pas une méthode d'optimisation meilleure que d'autres. Le choix d'une méthode par rapport aux autres est fait selon plusieurs critères, parmi eux on trouve : le problème d'optimisation, la précision souhaitée, la présence des contraintes ou non, le temps requis pour trouver la solution du problème considéré, le type de la fonction objectif et des contraintes : continues, dérivables ou non, le nombre de fonctions objectifs, caractéristiques de chaque méthode d'optimisation, et le nombre de variables ou paramètres d'optimisation.

I.7 Conclusion

La résolution d'un problème d'optimisation, contraint ou non contraint, passe tout d'abord par sa transformation en un problème de minimisation, puis chercher les valeurs des paramètres qui donnent la valeur de la fonction objectif la plus basse parmi un grand nombre de valeurs testées, solution optimale. Cette solution est obtenue par l'une des méthodes d'optimisation, exactes ou approchées. La première catégorie est adaptée à un espace de recherche restreint et complexe. Par contre, la deuxième catégorie est plus avantageuse dans le cas de grands espaces de recherche.

Chapitre II : Méthodes d'Intelligence Artificielle

II.1 Introduction

L'intelligence artificielle est un champ de recherche très attractif qui attire les chercheurs de tous les domaines scientifiques, surtout les informaticiens, les automaticiens, et les électrotechniciens, sans oublier les spécialistes des secteurs industriels et économiques, par la facilité et la rapidité de résoudre les différents problèmes avec un effort minimal de l'être humain [BENE-07]. Ces avantages sont en réalité le résultat, d'une part, des essais appliqués sur la machine pour lui donner un certain pourcentage de l'intelligence du cerveau humain à travers les techniques d'apprentissage, et d'autre part, la simulation du comportement social de certaines espèces vivantes pour chercher la nourriture. Ainsi, deux grandes branches de l'intelligence artificielle peuvent être définies. La première est celle connue sous le nom des réseaux de neurones artificiels, tandis que la deuxième réfère à l'Intelligence d'Essaim, ou l'intelligence collective, ou aussi l'intelligence du groupe [GUER-04] [LE RO-08].

Dans ce chapitre, on s'intéresse aux deux branches de l'intelligence artificielle. En commençant tout d'abord par les réseaux de neurones, l'historique et l'origine de la méthode ainsi que la modélisation, les types de réseaux, les techniques d'apprentissage, le réseau de neurones multicouches (Multi-Layer MLP) et les caractéristiques d'approximation sont abordés. Ensuite, la présentation de certaines méthodes d'Intelligence d'Essaim et les détails de leurs algorithmes et caractéristiques sont discutés.

II.2 Réseaux de neurones artificiels

II.2.1 Neurone formel et neurone biologique

Le neurone biologique c'est l'élément de base du système nerveux humain, où le cerveau humain est constitué de 10^{11} à 10^{15} cellules nerveuses ou neurones interconnectées fortement entre elles, chaque neurone fait de 1000 à 10000 connexions avec d'autres neurones. La cellule nerveuse est constituée d'un soma ou d'un corps cellulaire qui contient un noyau et une fibre longue nommée l'axone de neurone qui se termine par ramification. Le neurone (cellule nerveuse) contient un plongement sous forme d'un arbre lié au corps cellulaire. La zone, là où se rencontrent la fin de l'axone d'un neurone et les dendrites d'un autre neurone est dite : synapse, la figure II.1 illustre l'analogie entre un neurone biologique et un autre artificiel. Le neurone ou la cellule nerveuse réceptionne par son propre dendrite l'information qui vient des terminaisons des axones des autres neurones sous forme d'un signal électrique par un moyen chimique au niveau du synapse, après un assemblage de ces

signaux ou informations est effectué dans son corps cellulaire (soma), si la somme dépasse un seuil prédéfini, le neurone devient actif, et transfère cette information qui sort du corps cellulaire par son axone, sous forme d'un signal, aux autres neurones. Ainsi, les dendrites sont considérées ou représentées comme des entrées et l'axone comme une sortie [CHAM-07], [BOUH-12], [BURE-91], [GUER-04] et [ZEMO-03].

Le réseau de neurones artificiels est inspiré du réseau de neurones biologiques de l'être humain. L'élément de base d'un réseau de neurones artificiels est le neurone formel (artificiel), considéré comme unité ou cellule de calcul. Mc Culloch et Pitts sont les premiers qui ont fait une modélisation du neurone biologique et ont proposé le modèle de neurone formel (artificiel) en se basant sur l'analogie donnée dans le tableau II.1, elle éclairci la ressemblance entre un neurone biologique et un autre artificiel [LE RO-08].

Neurone biologique	Neurone formel ou artificiel
Synapse	Poids de connexion
Axone	Signal de sortie
Dendrite	Signal d'entrée
Somma	Fonction d'activation

Tableau II.1 Analogie entre neurone biologique et neurone

Chaque neurone artificiel reçoit des signaux d'entrées ou des variables d'entrées qui viennent de plusieurs éléments de calcul (neurones artificiels) en aval, chaque entrée est liée à un poids w qui représente la force de connexion. Après le neurone traite les entrées et distribue le résultat obtenu (sa sortie) aux autres neurones en amont.

Les figures II.1 et II.2 clarifient respectivement les modèles migmatique et mathématique d'un neurone artificiel. Après réception des entrées X_i pondérées par les poids W_i , et aussi l'entrée X_0 , égale à "1" et liée au poids W_0 qui représente un seuil, le neurone artificiel fait deux opérations principales, la première est la sommation de ces entrées pondérées et la deuxième c'est l'application d'une fonction d'activation de transfert f au résultat de la somme précédente [LE RO-08], [FORG-09] et [ZEMO-03].

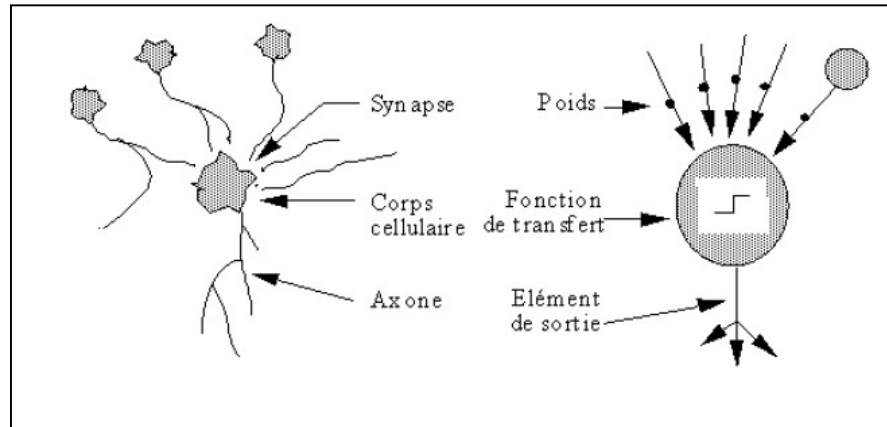


Figure II.1 Neurone biologique et neurone artificiel [ZEMO-03].

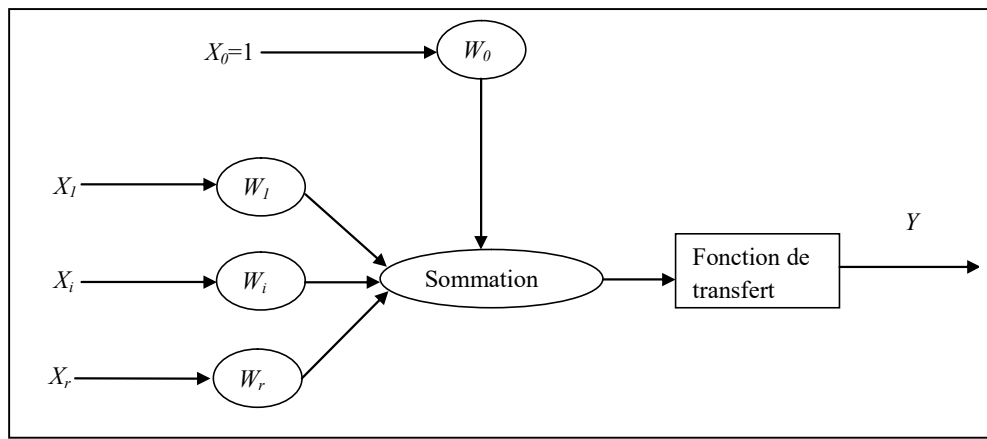


Figure II.2 Modèle mathématique d'un neurone artificiel [BOUH-12].

L'activation est faite selon l'expression ci-dessous [BOUH-12]:

$$Y = f(W_0 + \sum_{i=1}^r W_i \cdot X_i) \quad (\text{II.1})$$

Ou par la forme suivante :

$$Y = f(\sum_{i=0}^r W_i \cdot X_i) \quad (\text{II.2})$$

avec:

Y : la sortie du neurone ;

f : fonction de transfert ou d'activation ;

$i: i=1 \dots r$, nombre des variables d'entrées du neurone ;

X : variable d'entrée.

Il y a plusieurs formes de fonctions de transfert ou fonctions d'activation qui possèdent généralement des valeurs comprises dans des intervalles limités entre $[-1, 1]$ et $[0, 1]$. Quelques fonctions sont données dans le tableau II. 3 [BOUH-12].

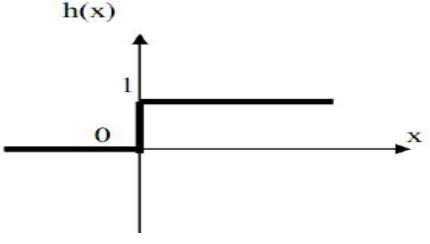
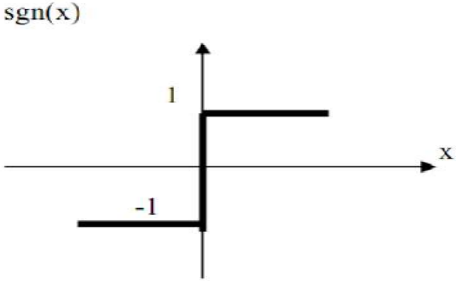
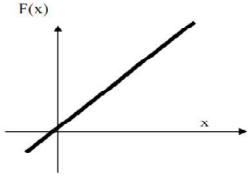
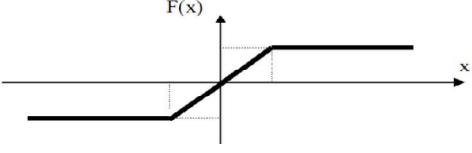
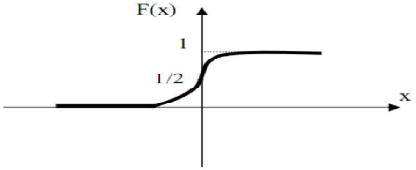
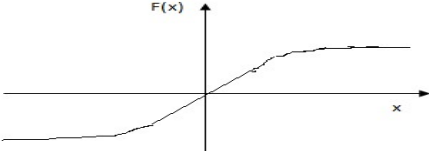
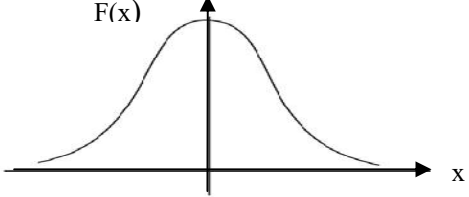
<p>Fonction de Heaviside</p> $h(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{sinon} \end{cases}$	
<p>Fonction de signe</p> $\text{sgn}(x) = \begin{cases} +1 & \text{si } x = 0 \\ -1 & \text{sinon} \end{cases}$	
<p>Fonction linéaire</p> $F(x) = x$	
<p>Fonction à seuil ou à multi seuils linéaires saturés</p> $f(x) = \begin{cases} x & \text{si } x \in [u, v] \\ v & \text{si } x \geq v \\ u & \text{si } x \leq u \end{cases}$	
<p>Fonction sigmoïde</p> $F(x) = \frac{1}{1+e^{-x}}$	
<p>Tangente sigmoïde hyperbolique [CHAM-07]</p> $F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
<p>Fonction à base radial [HACI-08]</p> $F(x) = \exp\left(-\frac{\ x - c\ ^2}{2\sigma^2}\right)$	

Tableau II.3 Exemples de fonctions de transfert ou d'activation.

II.2.2 Architecture des réseaux de neurones artificiels

Le réseau de neurones artificiels est un groupe de neurones élémentaires, chacun d'eux est considéré comme unité ou cellule de calcul, il fait des calculs parallèles. Ils sont rassemblés entre eux sous différentes formes ou topologies, généralement organisés en une seule couche, monocouche ou en plusieurs couches, multicouches dits aussi MLP (Multi-layer). Les réseaux de neurones sont utilisés pour effectuer plusieurs tâches dans différents domaines, l'approximation des fonctions, la classification des objets, la connaissance de forme, d'écriture ou de parole...etc.

Le nombre de couches et les types de connexion entre les neurones dans le réseau définissent son architecture et nous permettent ainsi de classer ou faire la différence entre topologies ou architectures. A partir de ça, on peut distinguer deux grandes familles de réseaux de neurones : réseaux de neurones bouclés (dynamiques) et réseaux de neurones non bouclés (statiques) [CHAM-07] et [AMMA-07].

II.2.2.1 Réseaux de neurones non bouclés (statiques)

Dans ce type de réseaux, les connexions entre les couches et le transfert d'informations entre les neurones de différentes couches sont effectués par des neurones d'une couche inférieure vers ceux d'une couche supérieure, et ne permettent pas le retour de l'information d'un neurone de couche supérieure vers un autre d'une couche inférieure, le transfert de l'information est unidirectionnel permettant ainsi de distinguer entre les couches (couches d'entrées, couches cachées et couches de sorties). Les connexions entre une couche et la couche suivante peuvent être totales ou partielles [CHAM-07] et [AMMA-07].

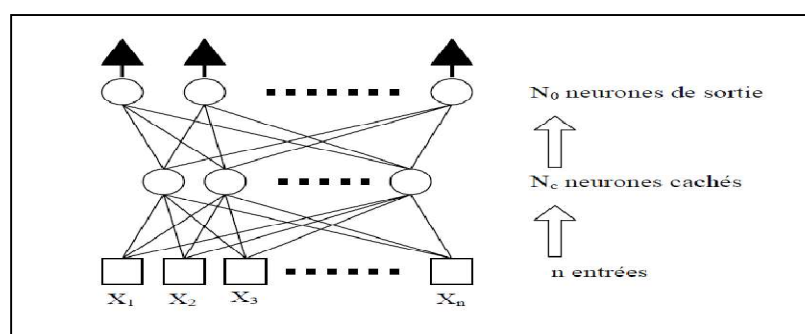


Figure II.4 Réseaux de neurones non Bouclés (statiques)[AMMA-07].

II.2.2.2 Réseau de neurones bouclé ou dynamique

Contrairement au réseau statique, il permet au moins un retour de l'information d'un neurone de la couche supérieure vers un neurone d'une couche inférieure, il est donc impossible de différencier entre les couches d'entrée et celle de sortie [CHAM-07] et [AMMA-07].

II.3 Apprentissage du réseau de neurones

L'apprentissage du réseau de neurones est une caractéristique très importante, un réseau de neurone sans apprentissage est inutile et sans importance. L'apprentissage est la tâche de réglage des poids des connexions synaptiques d'un réseau de neurones (nommés aussi les paramètres de réseaux) ou autrement, c'est l'opération qui permette de trouver les poids optimaux pour un but visé, qui représente la minimisation de l'écart entre la sortie du réseau et celle désirée, et qui permette aussi aux réseaux de donner la sortie désirée selon une entrée donnée. L'apprentissage est effectué à partir des exemples ou des entrées imposées à ce réseau, il permet au réseau de faire la mission pour laquelle il a été modélisé avec une efficacité raisonnable.

L'apprentissage d'un réseau peut être supervisé, non supervisé, ou par renforcement (semi-supervisé) [CHAM-07] et [AMMA-07].

- **Apprentissage supervisé:** Le réseau apprend à partir du couple (entrée, sortie) fourni [CHAM-07], [ZEMO-03] et [AMMA-07].
- **Apprentissage non supervisé:** Dans ce type, le réseau apprend sans aucune information sur les sorties, il n'a besoin que des entrées seulement, dans ce cas le réseau cherche la similarité entre les entrées pour connaître la ressemblance entre elles [ZEMO-03], [CHAM-07] et [AMMA-07].
- **Apprentissage par renforcement :** Ce type d'apprentissage est demi supervisé, où la donnée n'est pas un couple (entrée, sortie), mais on va lui donner une note ou une évaluation de la sortie [ZEMO-03], [CHAM-07] et [AMMA-07].

II.4 Réseaux de neurones multicouches (MLP)

Le réseau de neurones multicouches est un type de réseaux non bouclé ou statique, contient une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie, comme on peut le voir sur la figure II.5. Chaque neurone de la couche inférieure est connecté à tous les neurones de la couche suivante, ainsi de suite, on peut distinguer entre la couche d'entrée qui réceptionne les variables d'entrée, la couche de sortie qui donne les variables de sortie (le résultat du réseau) et les couches qui n'ont pas de contact avec

l'extérieur, nommées couches cachées. Le nombre de neurone des couches cachées est égale au nombre des variables d'entrée, et le nombre de neurones des couches de sortie est égale au nombre de variables de sortie. Le grand problème avec ce type du réseau est comment choisir le nombre des couches et surtout le nombre de neurones d'une couche cachée [BENE-07], [CHAM-07] et [AMMA-07].

Les réseaux de neurones artificiels possèdent une propriété importante qui clarifie leur intérêt, cette propriété est l'approximation des fonctions à partir d'un ensemble entrée-sortie d'un comportement ou d'un système ou d'un processus [CHAM-07] :

- Ne pas utiliser toutes les variables qui contribuent au problème.
- L'ensemble des entrées, sorties est d'un volume défini ou d'un nombre fini.
- Capacité de modéliser malgré l'existence de bruits au niveau de la base des données.
- ***Théorème de l'approximation d'Hornik***

Toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachée en nombre fini, possédant tous la même fonction d'activation et un neurone de sortie linéaire [Hor1989], [CHAM-07] et [LE RO-08]

II.5 Algorithme rétro-propagation d'apprentissage

L'apprentissage d'un réseau de neurones statique (dont la MLP fait partie) est effectué par des algorithmes d'optimisation qui minimisent l'erreur quadratique. L'algorithme de retro-propagation est un algorithme basé sur le gradient de descente qui fait cette tâche, pour expliquer la méthode de travail de l'algorithme de retro-propagation, on va proposer l'exemple d'un réseau MLP avec une seule couche cachée comme ont peut le voir sur la figure II.5 [CHAM-07] et [ZEMO-03].

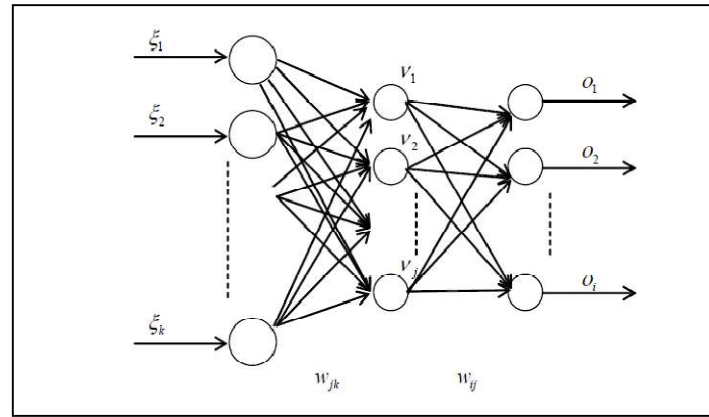


Figure II.5 Réseaux de neurones à trois couches [ZEMO-03].

Chaque itération de cet algorithme est répartie en deux phases : propagation et retro-propagation.

- **Propagation** : on va suivre la direction de transfert de l'information d'entrée vers la sortie, au début on calcule les sorties des neurones de la couche cachée (II.3), puis on utilise leurs sorties, considérées comme des entrées de la couche suivante, pour calculer les sorties de la dernière couche (couche de sortie) (II.4). On termine la phase de propagation par le calcul de la différence entre la sortie des réseaux et la sortie réelle du problème, nommée *erreur quadratique* (II.5) [ZEMO-03].

La valeur de sortie de la couche cachée j pour l'entrée m est donnée par :

$$V_j^m = f(a_j^m) = f(\sum_k w_{jk} \zeta_k^m) \quad (\text{II.3})$$

Et le neurone de sortie i par :

$$O_i^m = f(a_i^m) = f(\sum_j w_{ij} V_j^m) \quad (\text{II.4})$$

L'erreur globale (E^m) et partielle E sont données par :

$$E^m = \frac{1}{2} \sum_i (O_i^m - \zeta_k^m)^2 \quad \text{et} \quad E = \sum_i E^m \quad (\text{II.5})$$

Donc, dans cette phase on calcule les sorties des neurones de la couche cachée et la couche de sortie et l'erreur quadratique, l'information est propagée de la couche d'entrée jusqu'à la couche de sortie. Donc, cela prend le nom de propagation, qui se termine par le calcul de l'erreur quadratique selon l'équation (II.5).

- **Phase de rétro-propagation**: Pour apprendre ou régler les poids on suit une direction inverse qui démarre de la couche de sortie vers la couche d'entrée tout en se propageant, c'est pour cette raison que l'algorithme d'apprentissage est nommé ainsi.

La modification ou le réglage est effectué par un gradient de descente tout en ajoutant à chaque poids w une quantité Δw suivant l'équation ci-dessous : [ZEMO-03].

$$w_{ik}^{\text{nouveau}} = w_{ik}^{\text{ancien}} + \Delta w_{ik} \quad (\text{II.6})$$

où Δw est donnée par :

$$\Delta w = -\eta \frac{\partial E}{\partial w} = -\eta \sum_m \frac{\partial E^m}{\partial w} \quad (\text{II.7})$$

Pour le calcul de Δw on distingue deux cas :

1. Cas de neurones de la couche de sortie :

$$\Delta w_{ij} = \eta \sum_m (f'_i(a_i^m) \cdot (\xi_i^m - o_i^m) \cdot V_j^m) \quad (\text{II.8})$$

2. Cas de neurones de la couche cachée

$$\Delta w_{jk} = \eta \sum_m (f'_j(a_j^m) \cdot \zeta_k^m (\sum_i \zeta_i^m - o_i^m) f'_i(a_i^m) \cdot w_{ij}) \quad (\text{II.9})$$

où :

η : est le pas d'apprentissage qu'il faut bien choisir pour assurer une bonne convergence, ni petit entraînant une procédure lente, ni grand entraînant l'apparition des oscillations [ZEMO-03].

Pour la conception d'un réseau de neurones MLP, on va respecter les étapes suivantes [AMMA-07]:

1. Bien choisir les échantillons de la base de données, coté qualité et quantité, cette base se divise en deux bases d'apprentissage et de validation par une forme qui représente bien le problème ou le comportement de modèles ou systèmes étudiés.
2. conception du réseau MLP : exactement c'est le choix du nombre de couches cachées et le nombre de neurones dans ces couches, le choix du type de fonction d'activation dans les couches cachées et la couche de sortie et l'algorithme d'apprentissage du réseau.
3. Ces étapes se terminent par l'apprentissage du réseau de neurone MLP optimal choisi, permettant de donner un modèle qui représente bien le comportement étudié.
4. L'utilisation ou la mise en œuvre de ce modèle à la place du modèle réel.

II.6 Sous-apprentissage, généralisation et sur-apprentissage [ZEMO-03] et [AMMA-07]

Toutes les étapes précédentes sont importantes et influent sur la qualité du réseau conçu, mais la phase la plus sensible et la plus cruciale dans la conception d'un réseau de

neurones MLP est celle de l'apprentissage, qui dépend essentiellement du nombre de neurones dans la couche cachée K , ce nombre présente la complexité du réseau.

Comme on la déjà dit, l'apprentissage est le choix optimal des paramètres du réseau qui lui permette de faire la tâche pour laquelle il a été conçu. Le grand intérêt du réseau de neurones vient de la propriété de généralisation, celle-ci est la capacité du réseau de connaître les entrées qui n'existent pas dans la base d'apprentissage, donc le réseau a bien appris et ayant une capacité de généralisation [ZEMO-03].

Pour assurer la propriété de généralisation ou un bon apprentissage, on divise la base de données en deux groupes, un groupe pour l'apprentissage et l'autre pour la validation; on calcule l'erreur quadratique moyenne pour les deux bases en fonction de la complexité du réseau K ou nombre de neurones cachés. la figure II.6 illustre la bonne généralisation d'un réseau en fonction de sa complexité, on voit qu'au début d'apprentissage l'erreur de validation et d'apprentissage diminue, mais après certaines complexités, l'erreur de validation augmente, par contre l'erreur d'apprentissage diminue, après le point ($E_v > E_p$) le réseau entre dans une autre phase dite sur-apprentissage, à cette phase le réseau perd sa capacité de généralisation, et ne connaît que les entrées de base seulement :il a beaucoup appris, le point k est choisi comme meilleur nombre de neurones de la couche cachée [ZEMO-03].

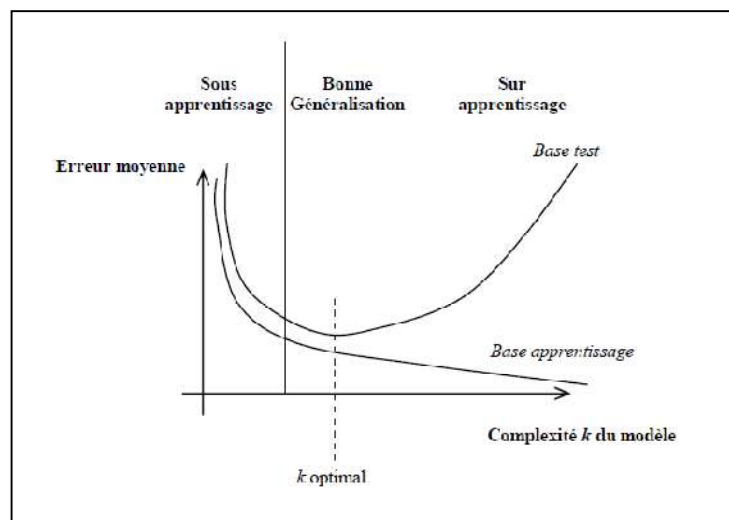


Figure II.6 Complexité d'un réseau de neurone [ZEMO-03].

II.7 Intelligence d'Essaim (collective) ou Swarm Intelligence

A la fin des années quatre-vingts, une autre porte de l'intelligence artificielle est ouverte, cette nouvelle branche est inspirée de la nature, la biologie et la science sociale et même des autres domaines comme la physique. Où dans la nature, on observe que plusieurs

petites espèces, qui font des mouvements individuels, sont capables de trouver des solutions complexes. Cependant, un seul agent ou individu de ces insectes n'est pas capable de faire cette tâche tout seul. Donc il ya une certaine intelligence, pour faire des tâches complexes comme la construction d'un nid, recherche de la nourriture, défense contre les ennemis ...etc. La caractéristique la plus importante de ces espèces est l'auto-organisation malgré l'absence du chef qui guide l'ensemble des individus[TFAI-07].

On peut dire alors que l'Intelligence d'Essaim (IE) caractérise un système composé de groupes des particules ou multi-agents ou des entités (non complexes) qui s'interagissent entre elles ou effectuent un travail collectif, d'où l'émergence d'un comportement complexe global [TFAI-07].

L'utilisation de cette stratégie chez les différents insectes sociaux permet d'acquérir les bénéfices suivants[TFAI-07] :

- Trouver des solutions dans un temps raisonnable;
- Performance collective est meilleure que celle individuelle ;
- Meilleure adaptation à l'environnement ;
- Assurer l'efficacité de systèmes et réaliser les tâches même en perdant quelques unités ou agents ;
- Réduction du coût des unités.

Par contre, elle possède aussi des inconvénients:

- Difficulté de justification et de compréhension de la solution trouvée ;
- Besoin d'un grand nombre des agents du système et le danger de conflits ;
- Incapacité de prédire la solution par intelligence collective ;
- Danger d'oscillations ou du blocage
- Absence de coopérations locales intentionnelles, c'est-à-dire de comportements volontairement coopératifs dédiés ou exigés par les individus.

On trouve ce comportement d'intelligence chez *les oiseaux volants en groupes (Figure II.7.a), les bancs de poissons (Figure II.7.b), les colonies de fourmis (Figure II.7.c),...*etc. Et même dans d'autres catégories inspirées des sciences physiques ou biologiques comme par exemple la gravité entre les planètes et le comportement des cellules de défense dans le corps humain.



a. Nuage d'oiseaux.



b. Banc de poissons.



c. Colonie de fourmis.

Figure II.7 Exemples de sociétés animales.

Les chercheurs profitent de ces comportements inspirés de la nature pour construire des algorithmes de recherche des solutions optimales aux problèmes d'optimisation. La caractéristique d'Intelligence collective ou d'Essaim la plus importante est l'auto organisation. La théorie d'auto organisation provient essentiellement des sciences physiques et chimiques et des systèmes biologiques, pour éviter des fluctuations indésirables, à partir de quelques phénomènes observés dans ces domaines [TFAI-07].

Ce terme, introduit pour l'étude des comportements collectifs complexes chez les insectes sociaux ou les systèmes multi-agents, découle de simples actions libres des individus d'un système ou société qui réalise une tâche complexe, malgré l'absence d'un chef qui guide l'ensemble des individus [TFAI-07].

Nicolas Monmarché [MONM-00] regroupe le processus d'auto organisation en 4 types :

- **Recrutement et exploitation collective des sources de nourriture :** par des stratégies permettant l'adaptation au milieu.
- **Division du travail et organisation des rôles sociaux :** dans une société, les insectes sont divisés en plusieurs groupes, où chacun d'entre eux s'occupe d'une tâche spécifique (constriction du nid, la défense, nettoyage la recherche de la nourriture).

- **Organisation de l'environnement** : La construction du nid est symbole de l'organisation distribuée des insectes, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.
- **Reconnaissance interindividuelle** : les insectes de mêmes sociétés se connaissent entre eux par quelques techniques.

On parle d'intelligence collective quand un groupe social arrive à résoudre un problème alors qu'un agent isolé n'arrivera pas à le faire (Bonabeau and Theraulaz, 1994) [MONM-00]. Ce comportement d'intelligence sociale est effectué par des interactions, ou l'auto organisation par des réactions entre les individus et l'environnement, par exemple les fourmis pour trouver l'origine de la nourriture utilisent la phéromone. La fourmi pose la phéromone sur le sol entre la colonie et l'origine de la nourriture, cette route est renforcée par les autres fourmis, donc la route la plus courte a de grande chance d'être concentrée en phéromone (lorsque une fourmi prend la route la plus courte, elle fait plusieurs vas et viens plus rapidement que les fourmis qui prennent une route plus longue), donc la colonie de fourmis a réussi à trouver la route la plus courte malgré l'absence d'un capitaine ou d'un chef qui dirige le groupe, où chaque fourmi travaille seule, pour trouver la route la plus courte, cette route change selon les variations de l'environnement [TFAI-07].

C'est la même chose chez les humains, pour trouver la route la plus courte et la plus facile ou de bonne qualité parmi plusieurs routes, malgré qu'aucun d'entre eux ne connaît la meilleure route, cette dernière sera renforcée avec le temps, où à chaque fois, elle est utilisée par d'autres humains, malgré qu'ils ne la connaissent pas, mais chacun d'entre eux utilise une stratégie ou un principe, « je fais ce que tu as fait » .

Cette route trouvée par les fourmis, ou par les humains, reste le chemin principal et ne change pas sauf s'il y a changement dans l'environnement, par exemple l'encombrement, le blocage ou une coupure dans une route principale, obligeant les individus à rechercher de nouvelles solutions, ou dans certains cas par exemple comme celui d'un individu qui a perdu sa route, et par exploration, il arrive à trouver de nouvelles solutions ou de nouvelles routes. Si un conducteur suit une route secondaire, les autres conducteurs le suivent malgré qu'ils ignorent cette route, donc cela consiste à trouver une solution au problème de blocage ou la coupure malgré l'absence du chef ou du guide. Pour un changement d'environnement, les individus changent localement pour relier cette route, on dit qu'il ya une auto organisation qui permet au système de s'adapter avec l'environnement. Cette union entre l'environnant et les individus émerge une nouvelle position ou solution, malgré la non existence d'un plan préparé auparavant [TFAI-07].

La même chose par exemple dans les stades, si un supporteur fait un geste de joie, les supporteurs dans son entourage le font aussi, et ceux de l'entourage de son entourage font le même geste, donc en quelques seconds tout le stade fait le même geste. Le même constat pour l'applaudissement, si quelqu'un applaudit, la personne la plus proche fait la même chose, puis les autres vont le suivre, et également pour l'arrêt de l'applaudissement.

Aussi les vols d'oiseaux, suivent la même méthode ou stratégie, généralement basée sur la position actuelle du vol et la suivante du vol le plus proche à la nourriture.

On observe que chaque système d'Intelligence d'Essaim, collectif ou social, possède deux propriétés, l'exploration et l'exploitation [TFAI-07].

Par exemple, comme on vient de le voir précédemment, le chemin trouvé par les fourmis ou les humains est dû à l'existence des individus qui renforcent cette route, c'est l'**exploitation**, et d'autres individus volontaires font des mouvements libres par des pas aléatoires, c'est l'**exploration**. Chez les oiseaux, il y a des mouvements locaux aléatoires (exploitation) et aussi des mouvements globaux (exploration). Il en est de même pour les différents insectes sociaux ou différents systèmes multi-agents[TFAI-07].

Les algorithmes basés sur l'Intelligence d'Essaim ou du groupe est un type d'intelligence très vaste et très différent qui se développe avec une vitesse incroyable grâce aux nombres élevés des méthodes proposées par les chercheurs ainsi que les différentes variantes et implémentations de ces méthodes, souvent inspirées de plusieurs origines comme la biologie, la sociologie, la géologie, la physique, la zoologie,...etc. Parmi ces méthodes on trouve l'Optimisation par Essaim de Particules, (Particle Swarm Optimization PSO), l'Optimisation Basée sur Enseignement-Apprentissage (Teaching-Learning Based Optimization TLBO), l'Algorithme des Chauves Souris (Bat Algorithm BA), recherche Coucou (Cuckoo Search CS), les Colonies d'Abeilles Artificielles (Artificial Bees Colony ABC), l'Algorithme d'Optimisation de l'Exploration Bactérienne (BEO), l'Algorithme de Recherche Gravitationnelle (Gravitational Search Algorithm GSA), l'Optimisation basée sur la Biogéographie...etc. Ces méthodes simulent l'intelligence des groupes d'animaux pour faire différentes tâches telles que, le mariage, le croisement, la reproduction des êtres vivants, la recherche de la nourriture, la défense contre les ennemis, et autres différents besoins pour vivre et continuer de vivre sur la terre. Dans ce chapitre, on va éclaircir les stratégies d'optimisation par certains algorithmes d'Intelligence par Essaim où le détail des algorithmes PSO, TLBO, BA et CS est présenté dans les prochaines sections.

II.7.1 Optimisation par Essaim de Particules (Particle Swarm Optimization PSO)

Le PSO est une méthode d'Intelligence d'Essaim ou du groupe, elle était proposée par Kennedy an Eberhart, [KENN-95]. Elle a été développée par simulation de certains comportements sociaux, en utilisant le principe de la recherche en groupe comme, le banc de poissons et les nuages d'oiseaux. Leur principe est simple par comparaison à celui de l'algorithme génétique, en outre, le PSO est plus rapide et ne demande pas beaucoup d'espace mémoire [TUNG-16],[RAO-17],[HASA-16] et [FU-18] .

L'algorithme démarre par une population aléatoire des solutions, où la recherche de la solution optimale est effectuée par l'adaptation des solutions initiales à chaque itération, par exemple il y 'a un group d'oiseaux dans un espace de recherche qui contient une seule pièce de nourriture, l'algorithme explore l'espace de recherche par des oiseaux qui sont distribués aléatoirement dans cet espace, aucun oiseau dans le groupe ne connaît ou peut trouver la pièce de la nourriture. La meilleure méthode ou la meilleure stratégie pour la trouver est de suivre l'oiseau le plus proche de la nourriture.

En optimisation, chaque particule distribuée ou posée dans l'espace de recherche possède deux caractéristiques, une position et une vitesse, à chaque génération les particules changent leur position dans l'espace par une vitesse qui dépende de deux facteurs, le premier est la meilleure solution de chaque particule i à l'itération précédente, P_i^{t-1} , tandis que le deuxième est la meilleure solution dans la population à l'itération précédente, P_g^{t-1} . Le mécanisme qui permet de conduire les particules vers la solution optimale est représenté par les deux équations (II.10) et (II.11). Malgré cela, l'algorithme converge des fois vers une solution locale [MOGH-16],[RAO-17]et [FU-18].

$$V_i^{t-1}(k+1) = W \cdot V_i^t(k) + C_1 \cdot r_1 (P_i^{t-1} - X_i^{t-1}) + C_2 \cdot r_2 (P_g^{t-1} - X_i^{t-1}) \quad (\text{II.10})$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (\text{II.11})$$

avec w est le poids d'inertie; c_1 et c_2 sont les coefficients d'accélération; r_1 et r_2 sont des nombres aléatoires, choisis dans l'intervalle $[0, 1]$. L'organigramme et le pseudo code de la méthode sont présentés dans les figures II.8 et II.9, respectivement.

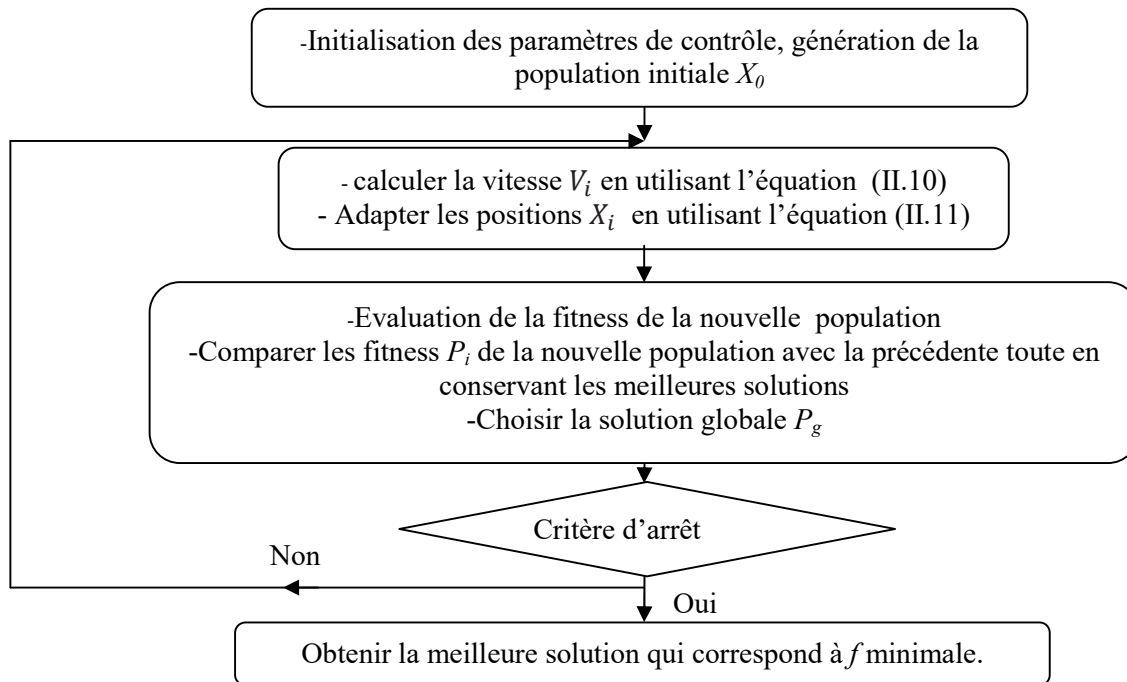


Figure II.8 Organigramme de l'algorithme PSO.

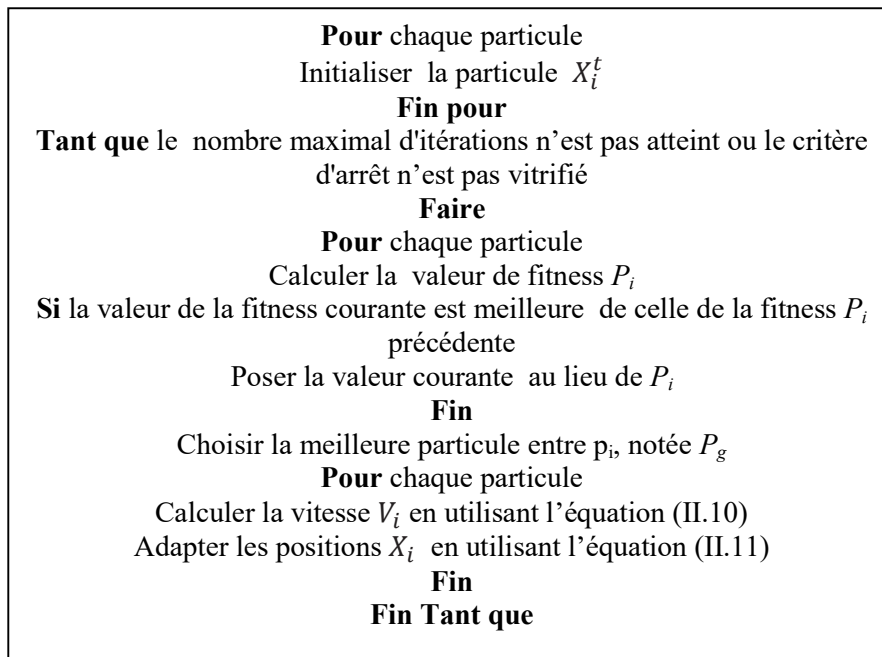


Figure II.9 Pseudo code de l'algorithme PSO.

II.7.2 Optimisation basée sur Enseignement-Apprentissage (Teaching-Learning Based Optimisation TLBO)

C'est un autre algorithme basé sur l'idée d'Intelligence d'Essaim, cet algorithme était proposé par Rao et al en 2011[RAO-13]. Il simule l'opération d'enseignement-apprentissage des étudiants ou des élèves dans une classe, où chaque étudiant étudie des modules et essaye d'évoluer son niveau, par une note, le meilleur étudiant dans la classe, ayant la meilleure note, est considéré comme un enseignant et les autres comme des élèves ou des étudiants. Par analogie, les paramètres d'optimisation présentent les modules étudiés, après génération aléatoire d'une population initiale des étudiants (solutions), à chaque itération le processus de TLBO passe par deux phases, la première s'appelle phase d'enseignement et la deuxième s'appelle phase d'apprentissage [CHEN-18a],[CHEN-18b]et[RAJI-17].

II.7.2.1 Phase d'enseignement

On évalue le niveau ou la fitness de chaque étudiant dans la population, le meilleur étudiant est considéré comme un enseignant nommé $X_{teacher}$, l'enseignant fait des efforts pour élever le niveau moyen total des étudiants dans la classe par l'équation (II.12), il faut dire que l'augmentation du niveau moyen de la classe ne dépend pas seulement de l'enseignant, mais aussi de la capacité des étudiants à capter l'enseignement, cela est traduit par le facteur Tf donné par [KANK-17]:

$$Tf = 1 + \text{round}(\text{rand})$$

donc Tf peut prendre deux valeurs : 1 et 2 [RAO-13].

$$X_{new} = X_i + \text{rand}(0,1) \cdot (X_{teacher} - Tf \cdot X_{mean}) \quad (\text{II.12})$$

où X_{mean} est la valeur moyenne des solutions de la population à l'itération courante ;

Si les nouvelles solutions X_{new} de chaque étudiant sont meilleures, alors elles remplacent X_i

Sinon on les rejette et on conserve X_i [DUAN-18]et[CHEN-18a].

II.7.2.2 Phase d'apprentissage

Après la première phase, il y a des interactions entre les étudiants (solutions obtenues à la fin de la première phase), deux à deux (solution X_i avec la solution X_j), ce qui permet d'élever le niveau de chaque étudiant selon les équations suivantes [PATE-16]:

$$X_{new} = X_i + \text{rand}(0,1) \cdot (X_i - X_j) \quad \text{if } f(X_i) < f(X_j) \quad (\text{II.13})$$

$$X_{new} = X_i + \text{rand}(0,1) \cdot (X_j - X_i) \quad \text{if } f(X_i) > f(X_j) \quad (\text{II.14})$$

Si les nouvelles solutions X_{new} sont meilleures que les solutions obtenues à la fin de la première phase, alors ces dernières sont remplacées par X_{new} , sinon elles sont conservées[CHEN-18a].

L'alternance entre les deux phases continue jusqu'à qu'un nombre maximal d'itérations est atteint ou un critère d'arrêt est vérifié. La figure II.10 résume les étapes de l'algorithme TLBO [DUAN -18] et[KANK-17].

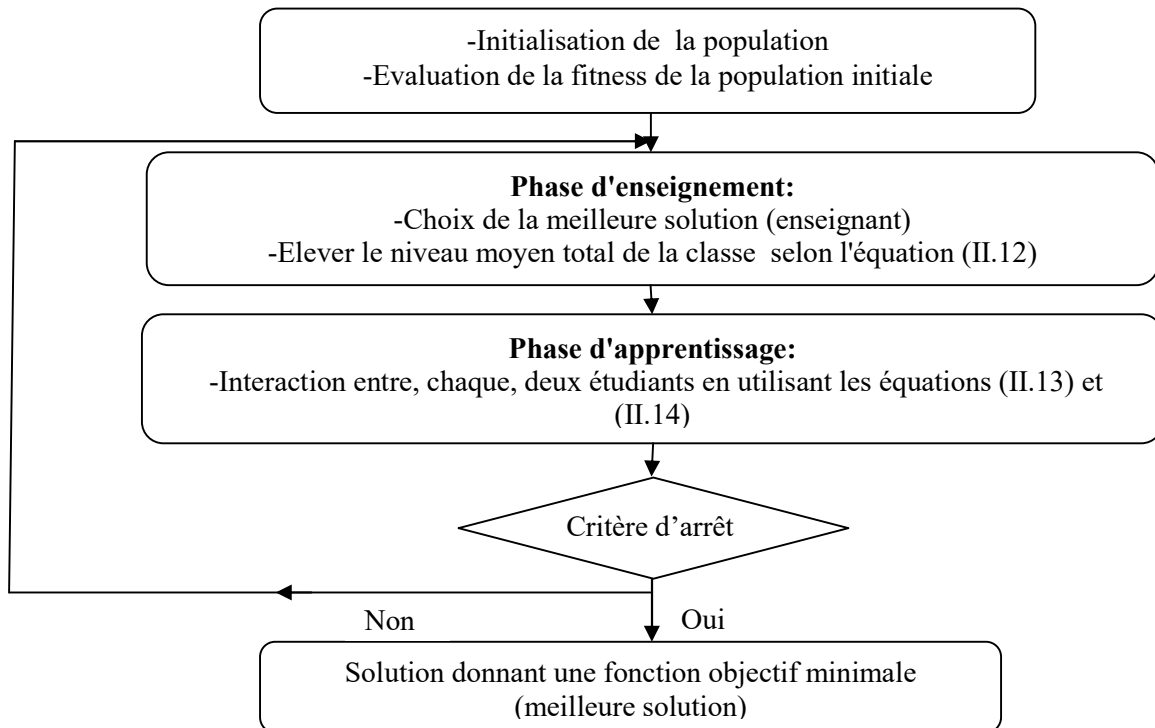


Figure II.10 Organigramme de l'algorithme TLBO.

La figure II.11, présente le pseudo-code de l'algorithme TLBO.


```

Génération aléatoires de n élèves
Calcul de la fitness de la population
TANT QUE (condition non vitrifiée)
    {Phase d'enseignement}
    Calcul de la moyenne des paramètres d'optimisation,  $X_{Mean}$ 
    Choix de la meilleure solution (enseignant)
        POUR  $i = 1 \rightarrow n$ 
            Calcul du facteur d'enseignement Tf
            Modifier les solutions selon l'équation (II.12)
            Calcul de la fitness des nouvelles solutions  $f(X_{i,new})$ 
            SI  $X_{i,new}$  est meilleure que  $X_i$ ,  $X_i = X_{i,new}$ ,
                FIN SI
            {Phase d'apprentissage}
            Sélection aléatoire d'une solution  $X_j$ , avec  $j \neq i$ 
            SI  $X_i$  est meilleure que  $X_j$ ,
                Appliquer l'équation (II.13)
                FIN SI
            Appliquer l'équation (II.14)
            FIN SI
        SI  $X_{i,new}$  est meilleure que  $X_i$ ,  $X_i = X_{i,new}$ ,
            FIN SI
        FIN POUR
    FIN TANT QUE

```

Figure II.11 Pseudo code de l'algorithme TLBO.

II.7.3 Algorithme de la Recherche Coucou (Cuckoo search CS)

Comme les deux algorithmes précédents, l'algorithme CS appartient à la classe d'Intelligence d'Essaim. Initialement proposé par Yang et Dep en 2009, l'algorithme CS est inspiré du comportement d'une espèce parasitaire dite l'oiseau coucou, et le mécanisme de vol de Lévy, cette espèce coucou utilise une stratégie parasitaire pour la reproduction et pour élever et assurer la nourriture de ses poussins, où la femelle coucou pond ses œufs dans les nids des autres oiseaux, appelés oiseaux hôtes, ces derniers et dans le cas d'exploration, s'ils trouvent des œufs du coucou ils les jettent hors du nid ou ils abandonnent carrément le nid et créent un autre dans une autre place. Certaines femelles coucou ont la capacité d'utiliser l'art mimique, pour rassembler leurs œufs, selon leurs couleurs et leurs motifs, pour augmenter la probabilité de la reproduction de leurs œufs, pour diminuer l'exploration des oiseaux hôtes, il existe aussi quelques espèces coucou qui jettent les petits des autres oiseaux directement après la sortie des œufs et plus que ça ils font la mimique de voix des parents hôtes pour augmenter la chance de trouver la nourriture[YANG-09],[FEYE-15],[ABDE-18],[CHI-17]et[CUI-17].

L'algorithme coucou utilise les deux techniques citées précédemment, le comportement parasitaire et le vol de Lévy, pour trouver l'optimum d'une fonction objectif [YANG-09],[FEYE-15],[CUI-17],[PRAk-18],[YANG-18],[ZHAN-18]et[ZHY-19].

Yang et Dep s'inspirent de ce comportement et posent trois étapes ou règles pour cet algorithme [YANG-09],[SANT-12],[CHI-17],[PRAk-18],[YANG-18]et[ZHAN-18]:

- Chaque coucou pond un seul œuf dans un nid qu'il choisit aléatoirement.
- Le meilleur nid qui possède des œufs (solution) de qualité élevée, passera à la génération suivante.
- Le nombre de hôtes du nid est fixé, l'oiseau hôte peut abandonner ou explorer les œufs étrangers par une probabilité $P_a \in [0, 1]$, dans ce cas le hôte jette les œufs du coucou ou abandonne le nid et construit un nouveau nid dans une autre place.
- La génération des nouvelles solutions est effectuée par deux méthodes : une marche aléatoire simple ou vol de Lévy.

II.7.3.1 Vol de Lévy

Le vol de Lévy est un type de *marche aléatoire*, où les pas de déplacements du coucou suivent une distribution de Lévy qui repose sur une loi de puissance donnée par l'équation (II.15) [GERB-13],[FEYE-15],[CHI-17],[CUI-17],[PRAk-18]et[ZHAN-18].

$$\text{Lévy}(\beta) \sim y = X^{-\beta}, \quad 1 < \beta \leq 3 \quad (\text{II.15})$$

où

β est une constante appartenant à l'intervalle [1, 3]

L'algorithme CS utilise le vol de Lévy pour générer les nouvelles solutions à partir de la meilleure solution courante et pour accélérer la recherche en général [FEYE-15]. La figure II.12 présente certaines méthodes utilisées pour la génération de nouvelles solutions par l'algorithme CS.

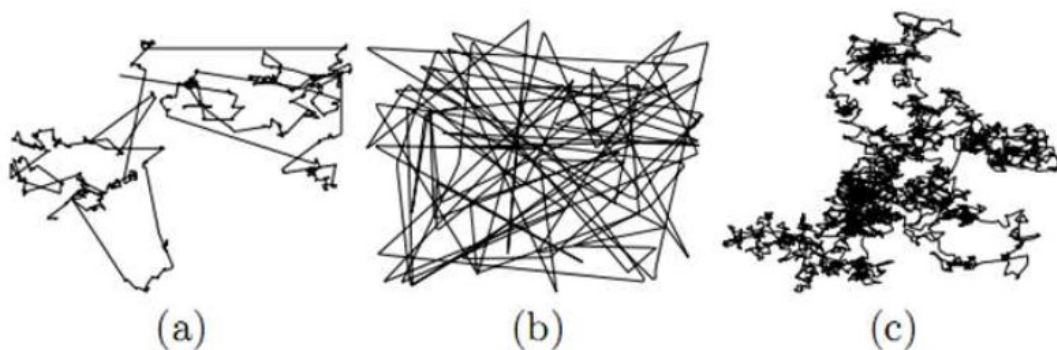


Figure II.12 Méthodes pour la génération de nouvelles solutions par l'algorithme CS
(a) Vol de Lévy, (b) Marche aléatoire simple et (c) Mouvement brownien [FEYE-15].

Il y a plusieurs méthodes pour déterminer le pas p qui doit obligatoirement suivre la distribution de Lévy, parmi ces méthodes on trouve celle de Mantegna, considérée la plus simple et la plus efficace [FEYE-15] et [ZHY-19].

$$p = \frac{u}{|v|^{1/\beta}} \quad (\text{II.16})$$

où: u et v sont des distributions gaussiennes :

$$u = N(0, \sigma_u^2) \quad , \quad v = N(0, \sigma_v^2) \quad (\text{II.17})$$

$$\sigma_u^2 = \frac{\Gamma(1 + \beta) \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{(1 + \beta)}{2}) \beta \cdot 2^{(\beta-1)/2}} \quad , \quad \sigma_v^2 = 1 \quad (\text{II.18})$$

où $\Gamma(z)$ représente la fonction Gamma:

$$\Gamma(Z) = \int_0^{+\infty} t^{Z-1} e^{-t} dt \quad (\text{II.19})$$

On peut diviser l'algorithme CS lors de la recherche de l'optimum en deux étapes:

1. Recherche globale

Cette étape est effectuée selon l'équation suivante [FEYE-15], [SANT-12], [ABDE-18] [CHI-17], [PRAK-18], [YANG-18], [ZHAN-18] et [ZHY-19]:

$$X_{new,i} = X_i + \alpha \otimes Levy(\beta) \quad (\text{II.20})$$

$\alpha > 0$ est le volume du pas de déplacement qui dépend du problème d'optimisation, \otimes est l'opérateur du produit terme à terme.

Si les nouvelles solutions sont meilleures que les précédentes, ces dernières sont donc remplacées par les nouvelles.

2. Recherche locale

On a cité précédemment que l'oiseau hôte peut explorer les œufs du coucou et les rejeter en dehors du nid ou créer un nouveau nid dans une autre place, l'algorithme CS emploie l'équation (II.21) pour générer les nouvelles solutions [FEYE-15], [ABDE-18], [CHI-17], [CUI-17], [PRAK-18], [ZHAN-18] et [ZHY-19]:

$$X'_{new,i} = X_i + rand(0,1) \cdot (X_p - X_m) \quad (\text{II.21})$$

p et m : sont les volumes de la population, choisis aléatoirement avec $p \neq m$.

Si les nouvelles solutions sont mauvaises, on conserve les solutions précédentes, dans le cas inverse on les remplace par les nouvelles solutions. Les figures II.13 et II.14 présentent l'organigramme et le pseudo-code de la méthode CS.

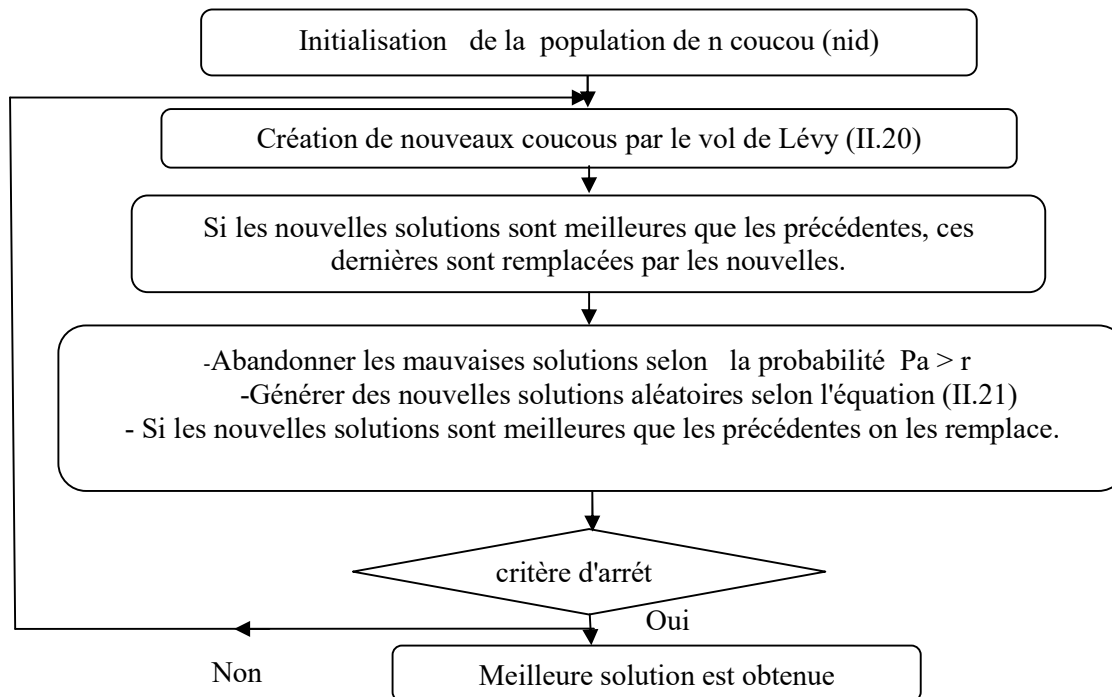


Figure II.13 Organigramme de l'algorithme CS.

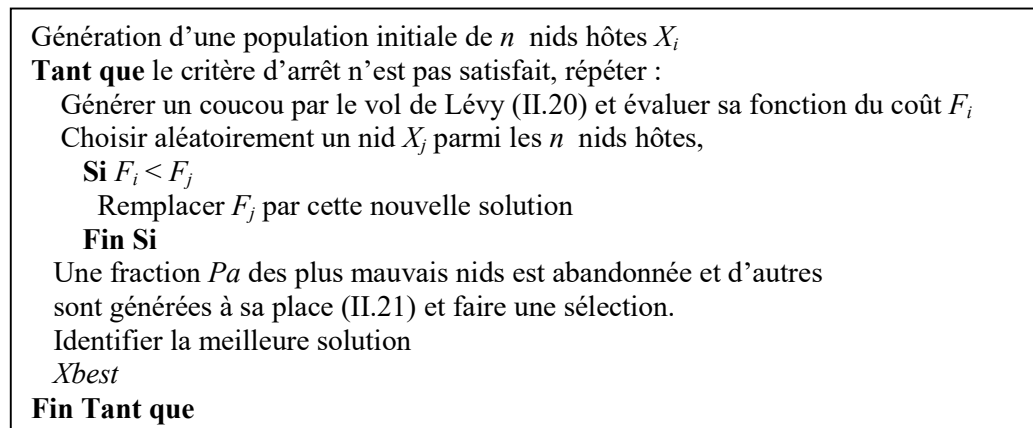


Figure II. 14 Pseudo-code de l'algorithme CS [FEYE-15]

II.7.4 Algorithme des chauves-souris (Bat Algorithm)

Un autre algorithme d'Intelligence d'Essaim à base de population, il s'agit de l'algorithme des chauves souris. Cet algorithme était proposé par yang en 2010 [YANG-10], aussi inspiré de la nature, exactement des comportements du phénomène d'écholocation de micros chauves souris [YANG-10],[TEY-18]et[ADAR-16].

La plus part des chauves souris sont des espèces insectivores. Les micros chauves souris possèdent des capacités et des caractéristiques qui les permettent de chasser efficacement et d'éviter les obstacles malgré qu'ils ne voient pas bien. Ils volent par des grandes vitesses, grâce à une technique très développée, c'est l'écholocation, les chauves souris pendant la recherche de la nourriture, et pendant leur déplacement, émettent des cris. Ces cris ont des pulsations très variables selon la position des chauves souris et leur distance par rapport à la proie. Après imitation de ces cris dans l'espace de recherche ou dans l'entourage de ces chauves souris, les écholocations tombent sur différents corps et se réfléchissent aux oreilles des chauves souris en les permettant alors, d'évaluer exactement la distance qui les sépare de ces corps, de la proie ou le but, leur volume et leur vitesse, et aussi de trouver ou déterminer la meilleure méthode pour la chasse, à partir de l'analyse des écholocations réfléchis dans leur cerveau.

Les chauves souris émettent aussi des cris ultrasons audibles utilisés pour la communication entre eux [BENM-08] et [YANG-10].

La période de chaque pulsation est très courte, elle dure de 8 à 10 ms, et les fréquences de la majorité des chauves souris sont constantes, en générale elles varient entre 25 kHz et 100 kHz, le nombre des cris émis est entre 10 et 20 par seconde, ce nombre augmente avec la chasse où il peut atteindre jusqu'à 200 pulsations par seconde si les chauves souris volent trop près de la proie [YANG-10] et [ADAR-16].

La longueur d'onde des cris ultrasons des chauves souris, λ , à une fréquence constante f est donnée par : $\lambda=v/f$, avec v est la vitesse de sons dans l'air, elle est égale à 340 m/s [YANG-13] et [ADAR-16].

La valeur des pulses émises peut atteindre 110 dB, elle appartient vraiment bien à l'intervalle des ultrasons, et leur intensité varie durant la recherche, elle est grande au début, puis elle diminue graduellement au fur et à mesure que les chauves souris s'approchent de la proie [ADAR-16].

Il est à noter aussi que la plus part des chauves souris possèdent une grande sensibilité à l'odeur, ils ont un odorat développé, et quelques chauves souris possèdent une bonne vision. Dans le monde réel les chauves souris utilisent tous ces sens en même temps pour augmenter leur performance en chasse de proie surtout dans l'obscurité. En étudiant le comportement d'écholocation, il peut être associé à un problème d'optimisation, d'où l'apparition de l'algorithme nommé algorithme de chauves souris [YANG-11].

L'algorithme des chauves souris simule le comportement d'écholocation, où les chauves souris volent aléatoirement pour trouver la nourriture (la proie), aucun des chauves

souris ne connaît l'existence ou l'endroit de la nourriture (la proie), mais ils peuvent évaluer la distance qui les sépare de la cible (nourriture) par écholocation. Dans le but de choisir la meilleure méthode de chasse, ils suivent la chauve souris la plus proche de la proie. Donc le processus de recherche de la nourriture a une grande ressemblance avec celui de la recherche de l'optimum d'une fonction objectif [YANG-11].

Le mécanisme d'optimisation par cet algorithme démarre par une population des chauves souris distribuée aléatoirement dans un espace de recherche. Pour trouver l'optimum d'une fonction objectif, chaque chauve souris représente une solution, on évalue la fitness des chauves souris à chaque itération, et on change leur position en utilisant leurs fréquences, vitesses, la meilleure position courante de chaque chauve souris, et la meilleure solution courante trouvée par toute la population. La meilleure solution est maintenue après la vérification d'un critère d'arrêt [BENM-08].

L'algorithme des chauves souris suit les trois 3 règles, ci-dessous, proposées par Yang et instaurées à partir des caractéristiques d'écholocation [YANG-10],[OSAB-18]et[TARA-19].

- Tous les chauves souris utilisent l'écholocation et connaissent la distance qui les sépare de la proie et l'emplacement des différents obstacles de façon magique;
- Les chauves souris volent aléatoirement à une position x_i , avec une vitesse v_i et une fréquence f_i , en changeant la longueur d'onde λ , et l'intensité A_0 pour chercher la proie. Ils peuvent automatiquement ajuster la longueur d'onde (ou fréquence) de leurs impulsions émises et le taux d'émission d'impulsions $r \in [0,1]$, en fonction de la proximité de leur cible;
- L'intensité varie selon plusieurs méthodes, généralement elle passe d'une grande valeur à une petite valeur.

D'après les trois règles précédentes, les déplacements ou les mouvements des chauves souris et l'adaptation des x_i et v_i de chaque chauve souris sont effectués selon les équations suivantes [YANG-10],[TEY-18],[OSAB-18],[CAI -18],[TARA- 19] et [WANG -19]:

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \cdot \beta \quad (\text{II.22})$$

$$v_i^t = v_i^t + (x_i^t + x^*) \cdot f_i \quad (\text{II.23})$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (\text{II.24})$$

avec:

f_i est la fréquence de la chauve souris i , où $f \in [f_{\min}, f_{\max}]$; f_{\min} et f_{\max} dépendent du domaine de recherche, par simplification $f \in [0, f_{\max}]$.

β est un vecteur aléatoire $\in [0, 1]$ selon une distribution uniforme.

x^* est la meilleure solution courante.

✓ Recherche locale

Choisir une solution parmi les meilleures solutions, et générer localement une nouvelle solution autour d'elle par une marche aléatoire (Random Walk) [YANG-10],[TEY-18],[ADAR-16],[OSAB-18],[CAI -18],[TARA- 19] et [WANG -19]:

$$x_{new} = x_{old} + \varepsilon \langle A_t \rangle \quad (II.26)$$

avec :

ε est un nombre aléatoire varie dans l'intervalle $[-1, 1]$.

$A_t = \langle A_{ti} \rangle$, es t la moyenne de l'intensité de tous les chauves souris à cette itération t.

L'intensité et la pulsation

L'intensité de la recherche locale est contrôlée par l'intensité A et le tau de pulsation r . Ces deux paramètres sont adaptés au cours du processus itératif, où, l'intensité décroît et le tau de pulsation croît jusqu'à que la chauve souris chasse sa proie. A et r sont actualisés (adaptés) selon l'équation suivante [YANG-10],[OSAB-18],[TARA-19]et [WANG-19]:

$$A_{t+1,i} = \alpha A_{t,i} ; r_{t+1,i} = r_{0,i} [1 - \exp(-\gamma t)] , \quad (II.26)$$

avec $0 < \alpha < 1$ et $\gamma > 0$; se sont des constantes.

si $t \rightarrow \infty$, alors $A_{t,i} \rightarrow 0$ et $r_{t,i} \rightarrow r_{i,0}$.

les valeurs initiales de A , A_0 , peuvent êtres : $A_0 \in [1, 2]$, et celles de r : $r_0 \in [0, 1]$

Les étapes de l'algorithme des chauves souris sont résumées dans l'organigramme suivant :

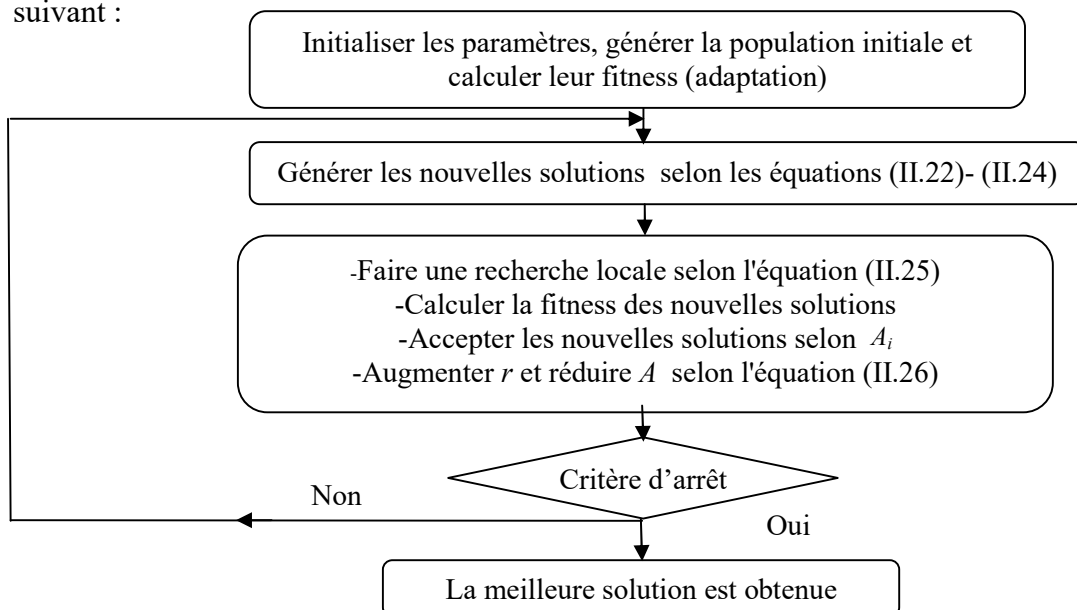


Figure II.14 Organigramme de l'algorithme BA.

La figure II.15 présente le pseudo-code de l'algorithme BA.

```

-Initialiser la population des chauves souris  $x_i$  ( $i = 1, 2, \dots, n$ ), la vitesse  $v_i$  et la
fréquence  $f_i$ .
-Initialiser le tau de pulsation  $r_i$  et l'intensité  $A_i$ 
Tant que (le critère d'arrêt n'est pas vérifié)
- Générer les nouvelles solutions en utilisant les équations (II.22) à (II.24)
  Si ( $\text{rand} > r_i$ )
    Sélectionner une solution parmi les meilleures solutions
    Générer des solutions locales par le vol aléatoire autour de la meilleure
solution selon l'équation (II.25)
  Fin Si

  Si ( $\text{rand} < A_i$  et  $f(x_i) < f(x^*)$ )
    Accepter les nouvelles solutions
    Augmenter  $r_i$  et réduire  $A_i$ 
  Fin Si
-Ordonner les chauves souris et trouver la meilleur solution courante  $x^*$ 
Fin Tant que

```

Figure II. 15 Pseudo-code de l'algorithme BA.

II.8 Choix du critère d'arrêt

Plusieurs critères pour arrêter le processus itératif des algorithmes basés sur l'Intelligence d'Essaim sont utilisés [FEYE-15] et [HAJJ-03], parmi eux on cite:

- La différence entre la meilleure solution de l'itération actuelle et celle de la précédente $\|x_k - x_{k-1}\| < \epsilon$;
- La différence entre la valeur de la fonction objectif actuelle et celle de la précédente $\|f_k - f_{k-1}\| < \epsilon$;
- Nombre maximal des itérations;
- Temps défini à l'exécution;
- Valeur de la fonction objectif arrive à une valeur définie;
- Nombre maximal d'évaluations de la fonction objectif;
- Hybridation entre deux ou trois critères précédents;

Il existe aussi un critère d'arrêt très utilisé dans le cas des algorithmes basés sur l'Intelligence d'Essaim :

Une convergence prématurée de l'algorithme peut être détectée si toutes les solutions tendent à être identiques, c'est-à-dire lorsque la relation suivante est satisfaite [FEYE-15]:

$$\exists X_{best} : \frac{\max_{i=1, \dots, P} \|X^{(k)} - X_{best}\|}{\|X_{max} - X_{min}\|} < \varepsilon \quad (\text{II.27})$$

On choisit dans la plupart des cas la tolérance $\varepsilon = 10^{-5}$.

avec :

p : taille de la population.

k : Indice de l'itération.

X_{best} : Meilleure solution jusqu'à l'itération k .

X_{min} et X_{max} : valeurs maximale et minimale des paramètres d'optimisation.

II.9 Conclusion

Dans ce chapitre, des techniques d'intelligence artificielle ont été présentées. Les réseaux de neurones artificiels simulent le travail du cerveau humain pour la résolution des problèmes. Plusieurs types existent, parmi eux on trouve le réseau MLP, capable d'approximer n'importe quelle fonction non linéaire, si son apprentissage est bien fait. La deuxième branche de l'intelligence artificielle est l'Intelligence d'Essaim ou collective, elle est inspirée de la nature, où les chercheurs ont observé que plusieurs insectes peuvent trouver des solutions à des problèmes complexes, par des comportements individuels, malgré l'absence du chef pour les guider. Les chercheurs ont simulé ces comportements et développé plusieurs algorithmes d'optimisation, comme l'optimisation par essaim de particules PSO, l'optimisation basée sur enseignement-apprentissage TLBO, la recherche coucou CS, et l'algorithme des chauves souris BA. A la fin du chapitre, quelques critères d'arrêt qui peuvent être appliqués à ces algorithmes ont été présentés.

On s'intéresse à l'application de cette branche des méthodes d'optimisation pour la résolution des problèmes d'optimisation relatifs aux machines électriques, à cet effet la démarche de la conception optimale des machines électriques est discutée dans le troisième chapitre.

***Chapitre III : Conception
optimale des machines
électriques***

III.1 Introduction

La conception optimale des machines électriques est un champ de recherche très important qui a connu beaucoup de développements, notamment ces dernières décennies, généralement liés au développement des matériaux de construction des dispositifs électrotechniques et de l'électronique de puissance, Il est à noter aussi que la grande concurrence entre les entreprises industrielles et l'utilisation croissante des machines dans les différentes activités humaines, que se soient industrielles ou domestiques, ont incité les chercheurs à améliorer ou optimiser les performances du produit existant et réduire le temps et le coût de sa fabrication [HAJJ-03], [FONT-07], [HAMO-12] et [BRIS-07].

Dans ce chapitre, on s'intéresse à la discussion des différentes étapes à suivre lors d'un processus de conception optimale des dispositifs électromagnétiques. Un intérêt particulier est attribué à la modélisation des machines électriques par le calcul du champ électromagnétique, qui est ensuite intégrée ou associée à une stratégie d'optimisation, permettant ainsi d'automatiser le processus de conception et le rendre plus fiable que possible. Cette démarche est appliquée à trois machines électriques dont les cahiers des charges relatifs à chacune d'elles sont détaillés également dans le présent chapitre. Ces applications concernent une presse à aimants le TEAM 25, considérée comme un problème test international, un moteur synchrone à reluctance variable avec barrières du flux magnétique et une machine synchrone à double excitation.

III.2 Etapes de la conception optimale des machines électriques

Avant d'aborder les étapes de la conception optimale des dispositifs électrotechniques, machines électriques, on va tout d'abord commencer par une simple définition de cette dernière.

III.2.1 Définition de la conception optimale

La conception optimale est la réalisation d'un dispositif en assurant les spécifications et les performances demandées par le consommateur et en respectant les contraintes citées dans le cahier des charges et les possibilités technologiques.

La conception consiste en la définition d'un dispositif ou d'un système (constitué de plusieurs objets) répondant à un critère ou un besoin défini en tant qu'objectif dans le cadre d'un cahier des charges bien déterminé. Celui-ci doit comprendre l'ensemble des exigences à réaliser sous forme de performances, de spécifications ou de contraintes techniques, économiques et d'encombrement,... etc. [REDJ-06], [ESPA-99], [MOKH-05] et [JANN-10].

A la fin de conception on obtient une définition précise de ce dispositif:

- La structure ;
- L'alimentation ;
- Les types des matériaux qui le constituent ;
- Les dimensions des différents composants de ce dispositif [FONT-07] ;

Généralement la conception optimale des machines électriques ou de dispositifs électrotechniques passe par deux étapes essentielles et importantes, la première est le choix de la structure de la machine ou du dispositif électrique à concevoir et la deuxième c'est le dimensionnement de cette structure. La figure III.1 illustre les axes principaux d'une démarche de conception.

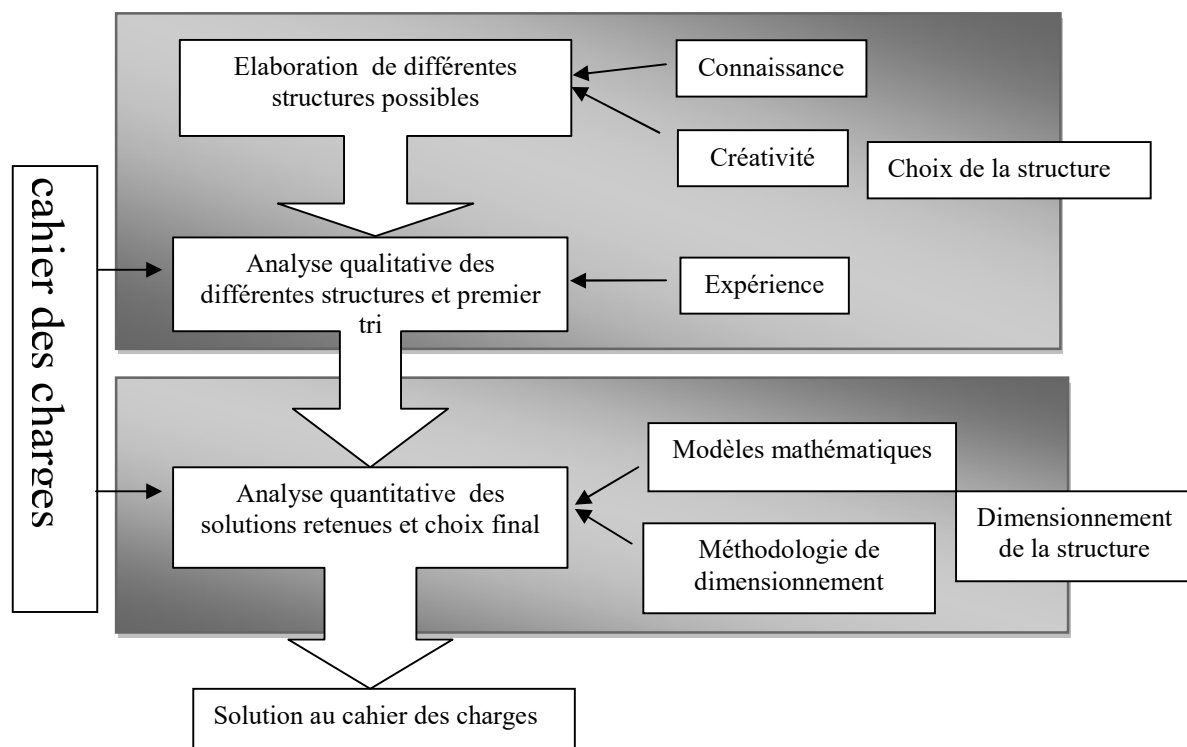


Figure III.1 Organigramme général de la conception [ESPA-99].

III.2.2 Choix de la structure de la machine à concevoir

Dans cette étape le concepteur essaye de définir la structure qui correspond aux spécifications ou aux besoins exprimés dans le cahier des charges. Pour cela, on se base sur la connaissance des structures déjà existantes est proches à quelques degrés à celles demandées, on exploite aussi le savoir faire et même la créativité pour trouver toutes les solutions possibles.

Après le choix et la définition de la structure initiale, elle vient la deuxième étape, de grands intérêts pour le concepteur, c'est l'étape de dimensionnement.

III.2.3 Dimensionnement des machines électriques

Le dimensionnement est la définition des dimensions et des paramètres géométriques et physiques des matériaux utilisés dans la machine et qui permettent d'obtenir les performances ou les spécifications données ou exprimées dans le cahier des charges, toute en respectant les contraintes qui y sont associées.

On parle aussi de la modélisation, qui est l'opération qui se sert des dimensions géométriques et caractéristiques physiques de matériaux pour trouver ou obtenir les performances du système. Il se trouve aussi, qu'ils utilisent le nom du problème direct pour désigner la modélisation et inverse pour désigner le dimensionnement. On donne ci-après les définitions relatives à chacun des deux problèmes selon [FONT-07].

Définition 1 (Problème Direct) *A partir d'un actionneur électromagnétique où la structure, les dimensions et les compositions sont connues, calculons quelques valeurs caractéristiques ; par exemple, le champ électromagnétique, le couple, ...etc.* Tel est montré dans la figure III.2, [FONT-07].

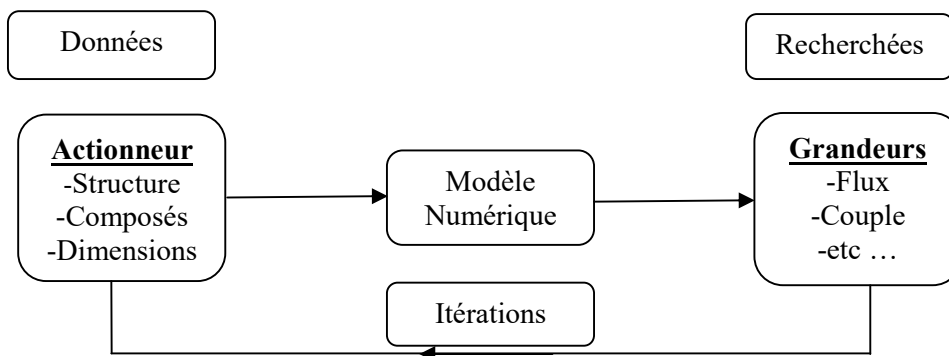


Figure III.2 Problème direct [FONT-07].

Définition 2 (Problème Inverse) *A partir des valeurs caractéristiques fixées dans le cahier des charges (par exemple le couple), retrouvons la structure, les dimensions ainsi que les compositions de l'actionneur* [FONT-07]. La démarche du problème inverse est illustrée dans la figure III.3.

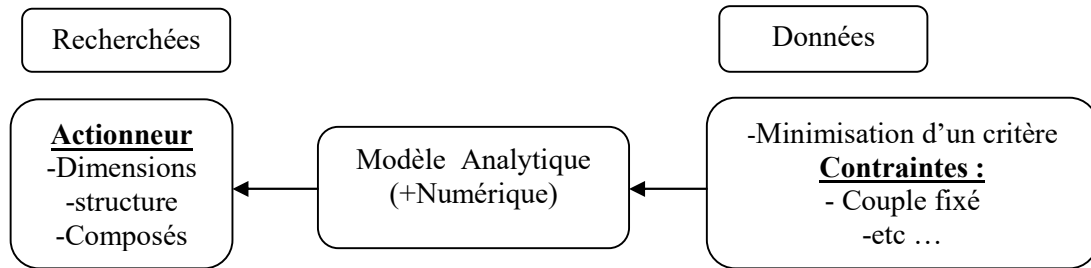


Figure III.3 Problème inverse [FONT-07].

Malgré que le deuxième problème exprime bien le concept de dimensionnement, il n'est pas toujours utilisé pour faire le dimensionnement d'un dispositif. On peut aussi utiliser le modèle directe pour le dimensionnement, il est utilisé pour trouver une solution au problème inverse.

Le dimensionnement peut être fait par trois ou quatre stratégies différentes, qui peuvent être directes ou inverses. Elles sont présentées ci-dessous.

➤ **Par essai et erreur**

C'est la méthode la plus ancienne, faite manuellement et aléatoirement par une méthode itérative sans utilisation d'algorithmes, cette méthode utilise le modèle directe, à chaque fois le concepteur propose des paramètres d'entrés et évalue les performances obtenues. Malgré que cette technique donne une solution qui respecte le cahier des charges, solution réalisable, mais elle n'est pas optimale (n'est pas la meilleure). Au début, cette méthode était très coûteuse, en temps et en coût, par ce qu'elle nécessite la réalisation de prototypes réels, mais actuellement et grâce à l'utilisation de prototypes virtuels son coût économique est baissé, mais elle reste toujours très coûteuse en temps de calcul [JANN-10].

➤ **Méthode procédurale**

C'est la méthode la plus directe et la plus immédiate, démarre par l'écriture des performances sous forme d'une série d'équations mathématiques simples, où par fois on fixe quelques paramètres pour simplifier le modèle mathématique, puis on résout une équation, ce qui permet au moins d'obtenir certains paramètres, qui peuvent être utilisés pour déterminer les autres paramètres. Avec la nécessité d'effectuer des retours en arrière pour corriger ou améliorer la solution, les équations sont simples avec une ou deux inconnues. Cette méthode est simple, et courte mais elle n'est pas générique, où elle nécessite une procédure spécifique à chaque dispositif [ESPA-99] et [MOKH-05].

➤ **Systèmes experts**

Cette approche, utilisée dans la phase du choix de la structure, et aussi utilisée dans le dimensionnement, elle est basée sur l'intelligence artificielle, après ramassage des expériences et des connaissances de l'expert concernant le dispositif ou la structure choisie, vient l'étape de la modélisation et l'organisation de ces informations sous forme des règles. Le système joue le rôle d'un expert vivant, où et à partir des questions traduites en langage informatique, il donne la solution ou la structure demandée qui répond aux besoins exprimés dans le cahier des charges. Le système est plus efficace dans le cas où le nombre des règles est grand, mais son inconvénient réside dans l'utilisation d'un langage informatique de haut niveau, donc l'obtention de solution prend beaucoup du temps [ESPA-99] et [MOKH-05].

➤ **Par optimisation**

Le dimensionnement par optimisation est considéré comme la meilleure méthode par comparaison avec celles citées précédemment. L'utilisation de cette méthode permet d'obtenir la solution optimale. Il y a une grande relation entre un problème de dimensionnement et un modèle d'optimisation, où chaque problème de dimensionnement peut être transformé facilement à un autre d'optimisation, où le but recherché est exprimé sous forme d'une fonction objectif, minimisation d'un coût ou maximisation d'une performance, en prenant en considération les équations contraintes et les variables de conception. La solution optimale est trouvée par des algorithmes itératifs en utilisant un modèle direct, où l'algorithme génère une solution initiale, et à chaque itération il essaye d'améliorer cette solution pour donner la valeur souhaitée de la fonction objectif, la démarche est illustrée dans la figure suivante [JANN-10], [ESPA-99] et [MOKH-05]:

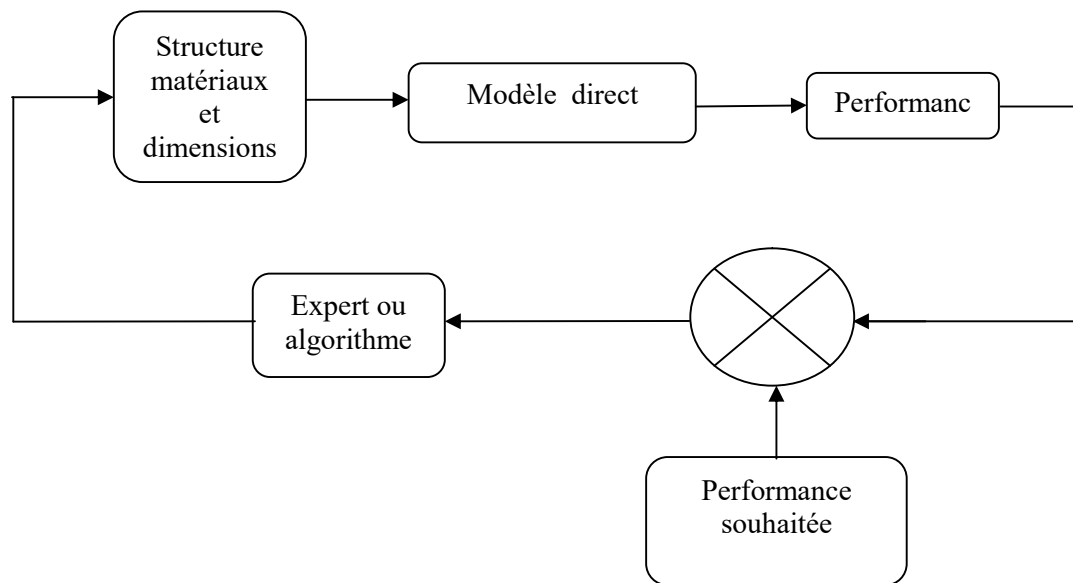


Figure III.4 Résolution d'un problème inverse avec un modèle direct [SERH-09].

Les méthodes par essai et erreur et par optimisation sont considérées comme des techniques basées sur des modèles directs. La méthode procédurale est considérée comme une technique du modèle inverse. Le système expert peut être basé sur les modèles inverse et direct.

Le dimensionnement par optimisation, dit aussi la conception optimale, est la démarche adoptée dans ce travail de thèse, elle est expliquée en détail dans la section ci-dessous.

III.3 Démarche de la conception optimale

La conception par optimisation ou la conception optimale passe par quatre ou cinq étapes essentielles et liées entre elles. Ces étapes sont illustrées dans la figure III.5 [TRAN-09] et [HAJJ-03].

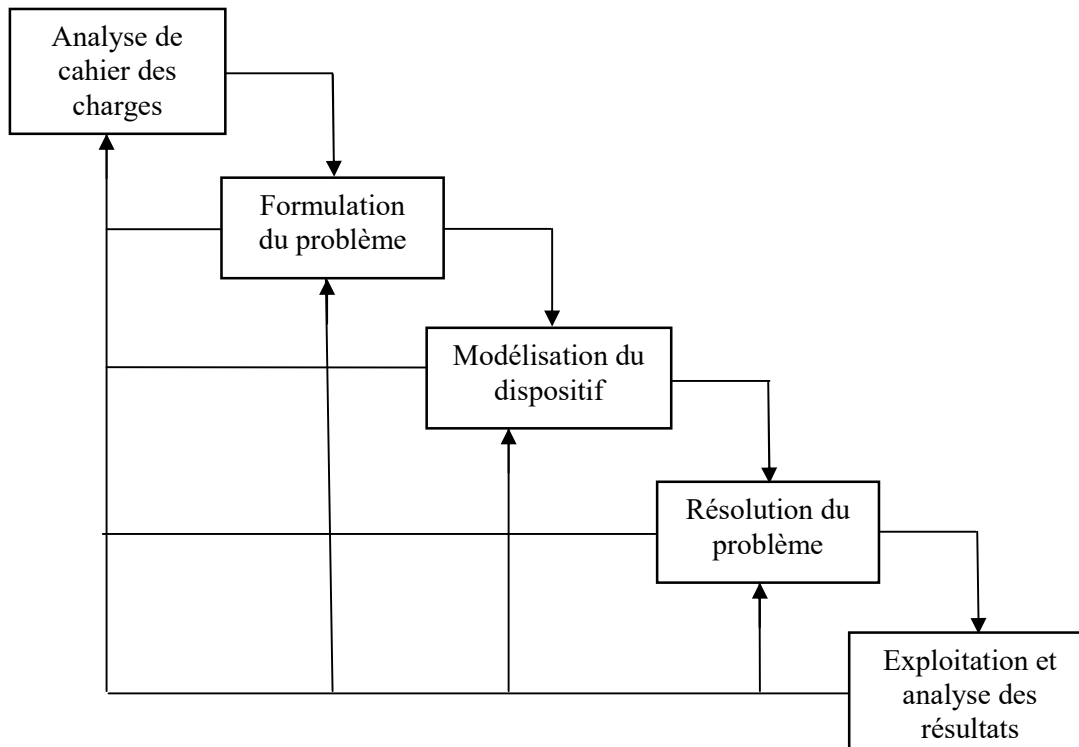


Figure III.5 Démarche de la conception optimale des machines électriques
[NGUY-11].

➤ Analyse de cahier des charges

Le cahier des charges contient les renseignements de dispositif à concevoir, il exprime les besoins de l'utilisateur comme le couple, la vitesse, la force, le rendement, l'encombrement, la puissance...etc. L'analyse du cahier des charges est une étape essentielle, permettant de bien comprendre la fonction du dispositif à concevoir, la bonne connaissance des performances demandées ou désirées par le client, aussi les contraintes qu'il faut respecter, choses qui permettent d'extraire les paramètres de conception et définir avec précision l'objectif à minimiser (le coût, le volume, ondulations du couple,...etc.) ou maximiser une performance (le couple, la force, facteur de puissance...etc.).

➤ Formulation du problème d'optimisation

Après la connaissance du but visé à partir de l'analyse de cahier des charges, on le traduit sous forme mathématique ou à un problème d'optimisation. Où, la fonction objectif exprime le critère à maximiser, par exemple la force ou le couple, le facteur de puissance, le rendement, ou minimiser le coût de fabrication, de consommation électrique, le volume,... etc. , celle-ci peut être écrite sous plusieurs formes selon le problème traité et les performances demandées dans le cahier des charges. Généralement, elle est écrite

sous forme d'une minimisation des écarts ou des erreurs entre la structure actuelle et celle désirée ou spécifiée dans le cahier des charges si celle-ci est connue. Dans cette étape, le concepteur doit définir avec précision:

1. La forme mathématique de la fonction objectif;
2. Les formes mathématiques des contraintes d'égalité et d'inégalité, extraites du cahier des charges et même, il est toute à fait possible d'ajouter des contraintes par le concepteur, si cela est nécessaire;
3. Les variables de conception et leur intervalle de variation. Les variables de conception peuvent être des paramètres géométriques, ou des propriétés physiques des matériaux.

➤ **Modélisations du dispositif**

Après la formulation mathématique du problème d'optimisation, vient le calcul des sorties ou des réponses (les valeurs de la fonction objectif et même des contraintes) du problème à partir des variables d'entrées. Cela passe par la modélisation du dispositif à concevoir toute en tenant compte des différents phénomènes physiques qui interviennent au cœur de ce dispositif, thermique, magnétique, mécaniques ...etc. Cette modélisation peut être analytique, numérique ou semi-analytique, on explique ces méthodes dans la section III.4.

La formulation et la modélisation sont des étapes très importantes et délicates et elles forment la base des étapes qui viennent après.

➤ **Résolution du problème**

La résolution du problème d'optimisation déjà formulé peut être effectuée par plusieurs méthodes, voir les deux premiers chapitres. Le choix d'une méthode dépend du problème d'optimisation, mono ou multi-objectif, contraint ou non contraint, de la taille du problème, du type de la modélisation, analytique, semi-analytique ou numérique ,...etc.

➤ **Exploitation et analyse des résultats**

Après la résolution, vient l'étape d'exploitation des résultats, où une analyse de leur pertinence s'avère nécessaire pour extraire les fautes si celles ci existent, et évaluer la qualité des résultats et de la solution trouvée.

En cas d'échecs, ou aboutir à de mauvais résultats, il faut réviser toutes les étapes précédentes et essayer de déterminer l'origine de l'échec. On cite ci-après quelques exemples.

1. **Etape de la résolution du problème** par exemple la méthode choisie pour résoudre le problème ne converge pas, dans ce cas il faut régler les paramètres de contrôle de l'algorithme, ou changer le contexte initial de l'algorithme exact ou déterministe, comme c'est possible que le critère d'arrêt n'est pas bien choisi ...etc.
2. **Etape de la modélisation ou de la formulation du problème** par exemple la formulation de la fonction objectif n'est pas adéquate, il faut donc la reformuler, ou par exemple la modélisation n'a pas bien cerné le comportement du dispositif modélisé, c'est le cas de certains modèles analytiques basés sur des hypothèses simplificatrices, ou peut être le mauvais choix des variables de conception, ou la méthode d'introduction des contraintes...etc.
3. **Etape d'analyse du cahier des charges** qui est mal défini, voir peut être irréalisable, c'est le cas par exemple où beaucoup de contraintes sont imposées, ou le cas inverse qui nécessite l'ajout de quelques contraintes pour améliorer la solution et supprimer la recherche dans des zones sans importance. Dans tels cas, la révision du cahier des charges est obligatoire.

III.4 Modélisation des dispositifs électromagnétiques

La modélisation des dispositifs électromagnétiques est une étape essentielle dans le processus de leur conception. Grâce à la modélisation des phénomènes physiques qui règnent dans ces dispositifs, on arrive à établir la relation qui existe entre les performances du dispositif (comme le rendement, le couple, la force,...etc.) et ses caractéristiques géométriques (comme le rayon du rotor ou du stator, la taille des bobines, le volume de l'aimant) et physiques des matériaux (comme la perméabilité, la conductivité ...etc.), et même le courant ou la tension d'alimentation. On peut aussi obtenir les valeurs de la fonction objectif et des contraintes.

Dans ce travail, on s'intéresse à la modélisation par le calcul du champ électromagnétique.

III.4.1 Calcul du champ électromagnétique

Tous les phénomènes électromagnétiques dans un dispositif électrotechnique sont régis par les quatre équations de Maxwell. Sous leur forme la plus générale, ces équations sont données par [BRIS-07]:

$$\text{rot}(\vec{H}) = \vec{J} \quad (\text{III.1})$$

$$\text{rot}(\vec{E}) = -\partial\vec{B}/\partial t \quad (\text{III.2})$$

$$\text{div}(\vec{B}) = 0 \quad (\text{III.3})$$

$$\text{div}(\vec{D}) = \rho \quad (\text{III.4})$$

avec :

\vec{E} , \vec{H} , \vec{D} et \vec{B} sont les vecteurs des champs électrique, magnétique, induction électrique et induction magnétique, respectivement.

\vec{J} : La densité du courant et ρ la densité de charge surfacique

Les grandeurs précédentes sont liées entre elles par les caractéristiques physiques des milieux constitutifs. Ces relations sont données par la loi d'Ohm et les lois des milieux magnétique et électrique.

$$\vec{J} = \sigma\vec{E} \quad (\text{Loi d'Ohm}) \quad (\text{III.5})$$

$$\vec{B} = \mu\vec{H} \quad (\text{Relation magnétique}) \quad (\text{III.6})$$

$$\vec{D} = \varepsilon\vec{E} \quad (\text{Relation diélectrique}) \quad (\text{III.7})$$

Où σ , μ et ε sont les caractéristiques du milieu considéré et représentent respectivement: la conductivité électrique [Ω/m]⁻¹, la perméabilité magnétique [H/m] et la permittivité électrique [F/m] [BELL-08].

$$\mu = \mu_0 \cdot \mu_r \quad \varepsilon = \varepsilon_0 \cdot \varepsilon_r$$

Avec :

μ_0 Perméabilité du vide, égale à $4\pi \cdot 10^{-7}$ [H/m]; μ_r perméabilité relative.

ε_0 permittivité du vide elle vaut $10^{-9}/36$ [F/m]; ε_r permittivité relative.

Il est connu que le calcul du champ électromagnétique permet d'extraire les différentes grandeurs comme la force, le couple, la puissance...etc. Dans la majorité des cas, les équations précédentes sont combinées ensemble pour simplifier le problème mathématique et le ramener à une seule équation différentielle aux dérivées partielles, formulée en terme d'une

ou deux variables d'état. Le potentiel vecteur magnétique est considéré comme la variable la plus utilisée, et la connaissance de cette dernière permet le calcul des autres grandeurs.

III.4.2 Formulation avec le potentiel vecteur magnétique

La divergence d'un rotationnel est toujours nulle, donc il est possible d'exprimer l'induction en utilisant le rotationnel d'une grandeur vectorielle \vec{A} [T.m] comme suit [HACI-08]:

$$\text{div}(\text{rot}(\vec{A})) = 0 \quad (\text{III.8})$$

D'après les équations (III.3), et (III.8), on peut définir le potentiel vecteur magnétique \vec{A} tel que :

$$\vec{B} = \text{rot}(\vec{A}) \quad (\text{III.9})$$

L'induction magnétique \vec{B} est la dérivée de \vec{A} . L'emploi du potentiel vecteur magnétique \vec{A} permet de simplifier la résolution du problème électromagnétique.

III.4.2.1 Cas magnétostatique

Dans le cas du problème magnétostatique, les équations à résoudre s'écrivent:

$$\text{div}\vec{B} = 0 \quad (\text{III.10})$$

$$\text{rot}(\vec{H}) = \vec{J} \quad (\text{III.11})$$

Tenant compte de l'équation (III.9) et remplaçant \mathbf{B} dans l'équation (III.6), on obtient:

$$\vec{H} = \frac{1}{\mu} \text{rot}(\vec{A}) \quad (\text{III.12})$$

$$\vec{H} = \nu \cdot \text{rot}(\vec{A}) \quad (\text{III.13})$$

où ν représente la reluctivité magnétique $\nu = \frac{1}{\mu}$

En remplace \vec{H} dans l'équation (III.11), on trouve:

$$\text{rot}(\nu \cdot \text{rot}(\vec{A})) = \vec{J} \quad (\text{III.14})$$

L'équation (III.14) est appelée équation de diffusion électromagnétique de la grandeur \vec{A} .

Pour assurer l'unicité de la solution généralement on ajoute une contrainte supplémentaire nommée la jauge de Coulomb. Elle est naturellement vérifiée en 2D [BRIS-07]:

$$\text{div}(\vec{A}) = 0 \quad (\text{III.15})$$

❖ Systèmes 2D cartésiens

En coordonnées cartésiennes, \vec{A} ne possède qu'une seule composante (dans la direction Oz) non nulle et ne dépend que de x et y, il en est de même pour la densité de courant [BRIS-07].

$$\vec{J} = \begin{bmatrix} 0 \\ 0 \\ J_Z(x,y) \end{bmatrix}; \vec{A} = \begin{bmatrix} 0 \\ 0 \\ A_Z(x,y) \end{bmatrix}; \vec{B} = \begin{bmatrix} B_x(x,y) \\ B_y(x,y) \\ 0 \end{bmatrix} \quad (\text{III.16})$$

On voit bien que le potentiel vecteur magnétique \vec{A} ne possède qu'une seule composante A_Z dépendant de x et de y et qui est dans la direction Oz, sa divergence est naturellement nulle (la jauge de Coulomb).

L'équation de diffusion électromagnétique dans le plan (Oxy) est écrite sous la forme suivante:

$$-\frac{\partial}{\partial y} \left(\nu \frac{d A_Z(x,y)}{dy} \right) - \frac{\partial}{\partial x} \left(\nu \frac{d A_Z(x,y)}{dx} \right) = J_z(x,y) \quad (\text{III.17})$$

III.4.3 Types de modélisation

Il existe trois types de modélisation, analytique, numérique et semi-numérique.

- **Modélisation Analytique** : le modèle analytique donne ou éclaire bien les relations entre les paramètres d'entrée et leur influence sur les performances globales du dispositif modélisé.
- **Modélisation semi-numérique**: on trouve dans cette catégorie plusieurs méthodes, comme celle des réseaux de reluctances, les circuits électriques équivalents,...etc. L'utilisation des réseaux de reluctances permet de prendre en compte la saturation et la non linéarité des matériaux, cette méthode utilise une représentation des circuits magnétiques par des petits éléments nommés la reluctance et les sources de potentiel, elle utilise les lois de Kirchhoff pour extraire le flux dans les différentes branches [DOGA-13]. Cette méthode est plus exacte que celles de modèle analytique puisqu'elle prend en compte la saturation magnétique.
- **Modélisation numérique**: consiste à résoudre les équations du champ électromagnétique par des méthodes numériques. Elle est basée sur la discrétisation du domaine en sous-domaines élémentaires, où l'inconnue ou ses dérivées sont approximées localement, conduisant alors à la construction d'un système algébrique qui est entièrement défini par

la considération des conditions aux limites. Ce modèle permet de prendre en compte la non-linéarité physique des matériaux et les structures géométriques complexes. Plusieurs méthodes numériques existent, parmi elles on trouve, les différences finies, les volumes finis, et les éléments finis, cette dernière est la plus utilisée. C'est cette méthode qu'on a utilisé dans ce travail pour modéliser les machines étudiées. Les principales étapes de son application sont les suivantes :

1. Maillage en éléments finis

En deux dimensions, cas des dispositifs considérés dans ce travail, le domaine d'étude est découpé en éléments triangulaires de premier ordre, ils s'adaptent aux différentes configurations géométriques.

2. Approximation et construction des fonctions d'interpolation

On peut écrire l'équation différentielle (III.17), sous une forme plus générale donnée par l'équation (III.18). Considérons un certain domaine Ω dans lequel est défini un champ U , dans notre cas, vectoriel, et régit par l'équation différentielle aux dérivées partielles suivante [BELL-08]:

$$L(U) = F \quad (\text{III.18})$$

où :

L est un opérateur différentiel

F est une fonction connue, appelée terme de source.

Après discrétisation, on approxime la variable inconnue U par une fonction polynomiale sur chacun de ces éléments. Cette approximation est de la forme suivante :

$$U^e(x, y) = \sum_{i=1}^n \alpha_i(x, y) U_i^e \quad (\text{III.19})$$

où:

U_i^e : Valeur de U au nœud i de l'élément e

n : Nombre de nœuds de l'élément e

α_i : Fonction d'interpolation.

Afin d'avoir la convergence de la solution, l'approximation nodale doit satisfaire les propriétés, ci-dessous:

- a- la fonction approchée $U_i^e(x, y)$ coïncide avec la fonction exacte $U_{ex}(x, y)$ en tous les nœuds d'interpolation de l'élément, de coordonnées x_i, y_i :

$$U_{ex}(x, y) = U_i^e(x, y) = U_i(x, y) \quad (\text{III.20})$$

D'où la fonction d'interpolation doit vérifiée:

$$\alpha_i(x_j, y_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad (\text{III.21})$$

où i et j sont les numéros des nœuds de l'élément fini considéré [BELL-8].

3. Formulation intégrale de l'équation différentielle aux dérivées partielles

La formulation intégrale d'une équation différentielle aux dérivées partielles, EDP, par la méthode des éléments finis, peut être obtenue par plusieurs méthodes, la plus utilisée est celle des résidus pondérés.

Les résidus pondérés sont des méthodes projectives où on cherche à projeter l'équation locale sur des fonctions de base d'un espace de fonctions de pondération, et on essaie de minimiser le résidu par l'approximation de la fonction de l'inconnue dont l'expression est :

$$\int_{\Omega} R\phi(x, y)d\Omega = 0 \quad (\text{III.22})$$

avec :

R est le résidu de l'approximation défini comme suit :

$$R = L(U) - F \quad (\text{III.23})$$

ϕ est la fonction de pondération.

Les valeurs de U qui permettent d'annuler l'intégrale (III.22) sont solutions du système algébrique obtenu [BELL-08].

4. Choix de la fonction de pondération

Il y'a aussi plusieurs méthodes pour le choix des fonctions de pondération ϕ , la méthode la plus utilisée est celle de Gallerkine qui consiste à mettre la fonction de pondération égale à la fonction d'interpolation $\alpha_i(x, y)$ [BELL-08].

$$\phi_k = \alpha_k \quad k=1, \dots, n \quad (\text{III.24})$$

La forme intégrale de l'équation différentielle (III .18) est :

$$\int_{\Omega} \alpha_i(x, y) [L(\sum_j \alpha_j(x, y) U_j) + F] d\Omega = 0 \quad (\text{III.25})$$

Cette intégrale sera calculée sur chaque élément fini e constituant le domaine d'étude Ω , d'où la forme matricielle élémentaire suivante:

$$[K]^e \cdot [U]^e = [F]^e \quad (\text{III. 26})$$

Ensuite, ces matrices élémentaires sont assemblées pour former le système global.

La matrice globale étant $[K]$ et le vecteur global étant $[F]$, on obtient ainsi un système de la forme [HACI-08] :

$$[K] \cdot [U] = [F] \quad (\text{III.27})$$

La résolution du système d'équations (III .27) est la dernière étape, après l'introduction des conditions aux limites, qui mène à la détermination des valeurs U_i de la fonction $U(x,y)$ cherchées en chaque nœud du domaine.

Les conditions aux limites les plus communément utilisées en électrotechnique sont celles de Dirichlet et Newman [BRIS-07]:

- **Condition de Dirichlet:** on impose une valeur constante de l'inconnue sur les frontières du domaine d'étude Ω .
- **Condition de Newman:** on impose des valeurs constantes sur la dérivée normale de l'inconnue ($\partial U / \partial n$) sur des frontières perpendiculaires au domaine d'étude Ω .

III.5 Problèmes de conception optimale étudiés

La démarche de conception optimale adoptée dans ce travail de thèse, basée sur l'association des méthodes d'optimisation à une modélisation numérique par éléments finis, a été appliquée pour la résolution de trois problèmes électrotechniques, versés dans le domaine de la conception optimale des machines électriques, il s'agit d'une presse à aimant, TEAM workshop problème 25, considérée comme l'un des problèmes test internationaux pour la validation des nouvelles techniques d'optimisation, un moteur à reluctance variable à barrières du flux et une machine synchrone à double excitation. Ces deux dernières applications s'inscrivent dans le cadre de la conception des machines électriques à structures innovantes dont les applications sont diverses et nombreuses.

III.5.1 TEAM workshop problem 25 [w1][Taka-96]

La configuration au quelle on s'intéresse est présentée dans la figure III.6, c'est une presse avec électro-aimant pour l'orientation de la poudre magnétique. Elle est employée pour produire des aimants permanents anisotropes. La presse et l'électro-aimant sont faits en acier. Les moules de la matrice sont placés pour former une distribution radiale du flux. La poudre magnétique est insérée dans la cavité. Le modèle peut être considéré comme bidimensionnel.

Le dispositif est alimenté par deux bobines inductrices, dont chaque enroulement porte 4253 ampères-tours, tandis que son armature est constituée par un matériau magnétique non linéaire. L'analyse préliminaire montre qu'une distribution radiale du flux peut être presque obtenue en paramétrant la forme de la matrice extérieure par la combinaison d'une ligne droite, d'un cercle et d'une ellipse (figure III.7). La forme du moule intérieur de la matrice est un cercle.

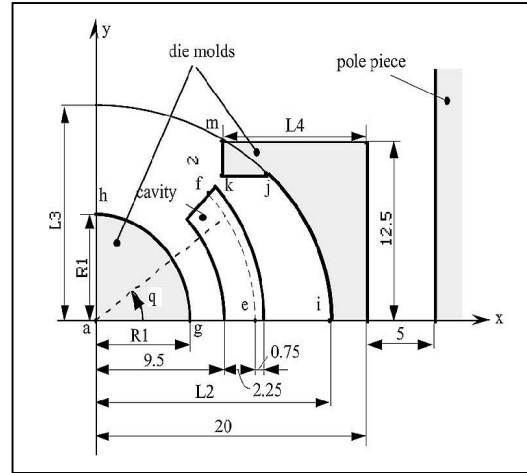
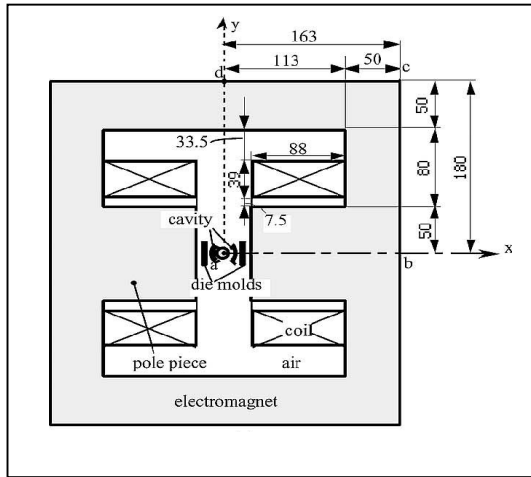


Figure III.6 La presse électro-aimant ;
Vue générale.

Figure III.7 La presse électro-aimant ;
Vue zoomée.

La courbe de la première aimantation, caractérisant le matériau ferromagnétique doux dans le dispositif, est donnée dans la figure III.8.

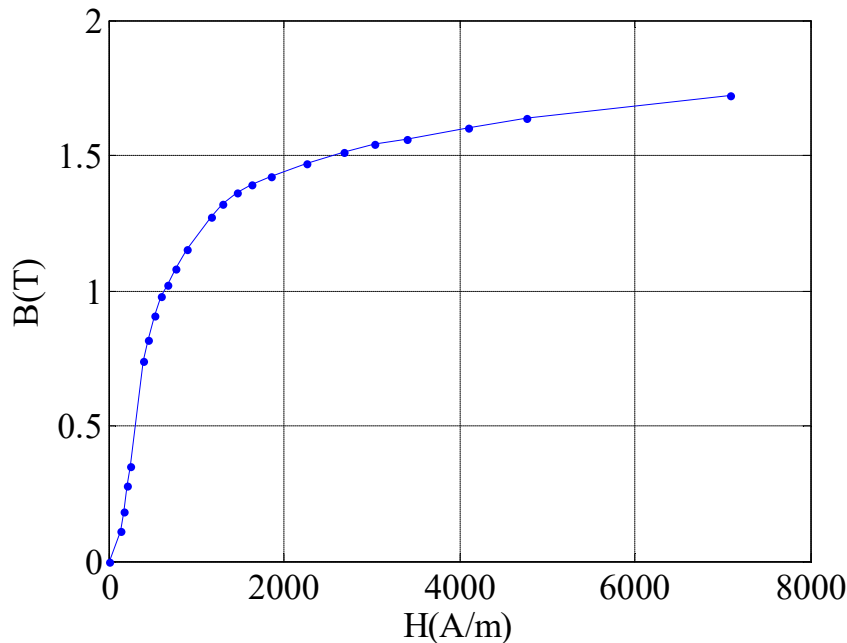


Figure III.8 Courbe caractéristique B(H) du matériau utilisé.

III.5.1.1 Formulation du problème d'optimisation

La forme du moule est définie par deux pièces magnétiques dont la géométrie dépend des paramètres suivants : le rayon R_1 de la matrice intérieure, les axes long et court L_2 et L_3 de l'ellipse et la dimension L_4 . Ces variables de conception sont visibles dans la figure III.7. Les contraintes concernant les limites de variation des paramètres du problème sont données dans le tableau III.1.

paramètres	Valeur minimale	Valeur maximale
R_1 [mm]	5	9.4
L_1 [mm]	12.6	18
L_2 [mm]	14	45
L_4 [mm]	4	19

Tableau II.1 Limites de variation des paramètres de conception.

L'objectif recherché est de disposer le long de la courbe ef (voir figure III.7), d'une induction magnétique \mathbf{B} radiale et valant 0.35 T comme est précisé par la relation (III.28):

$$\begin{cases} B_{xc} = 0.35 \times \cos(\theta(i)) \\ B_{yc} = 0.35 \times \sin(\theta(i)) \end{cases} \quad (III.28)$$

La fonction objectif est donnée par :

$$OF = \sum_{i=1}^{10} \left\{ (B_{xci} - B_{xi})^2 + (B_{yci} - B_{yi})^2 \right\} \quad (III.29)$$

Dans le cas magnétostatique non-linéaire, la méthode des éléments finis est appliquée pour la résolution de l'équation différentielle aux dérivées partielles donnée par l'équation (III.17). Une fois le potentiel vecteur magnétique est connu, on calcule les valeurs de l'induction magnétique.

III.5.2 Moteur synchrone à reluctance variable avec barrières du flux (MSRV)

On s'intéresse ici à l'optimisation de forme du rotor d'une machine synchrone à reluctance variable avec barrières du flux, de manière à maximiser son couple électromagnétique [SERH-09]. La géométrie du MSRV est illustrée dans la figure III.9. Le rotor du MSRV est composé de huit barrières du flux en aluminium. Dans le stator, les enroulements statoriques sont à simple couche sans raccourcissement. Le nombre d'encoches statoriques est douze, avec une seule encoche par pôle et par phase. Les paramètres

géométriques de la machine étudiée sont donnés dans le tableau III.2. Cependant, les paramètres de conception du rotor sont illustrés dans figure III.10.

Le stator est formé d'un empilement de tôles ferromagnétiques M 19. Les dents et les encoches ont une section droite ou uniforme. La construction du stator et du rotor est similaire. La courbe $B(H)$ de la première aimantation du matériau utilisé est donnée dans la figure III.11. L'enroulement du rotor est réalisé avec un fil rond. Sa distribution correspond à un emplacement par pôle et par phase.

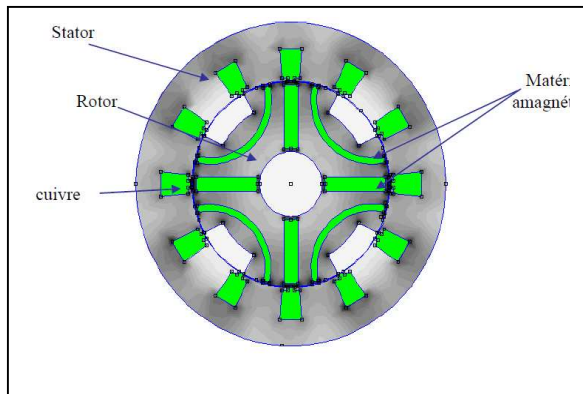


Figure III.9 Géométrie de la MSRVM étudiée.

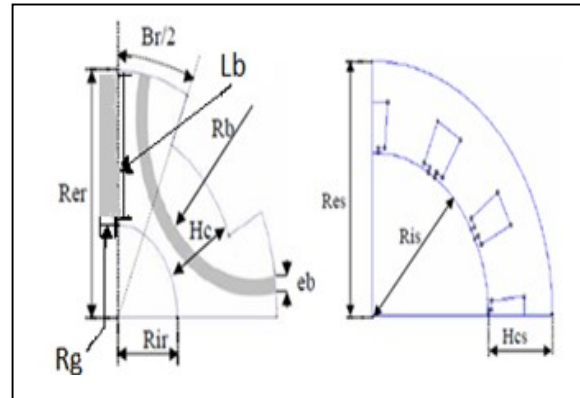


Figure III.10 Paramètres de conception du rotor.

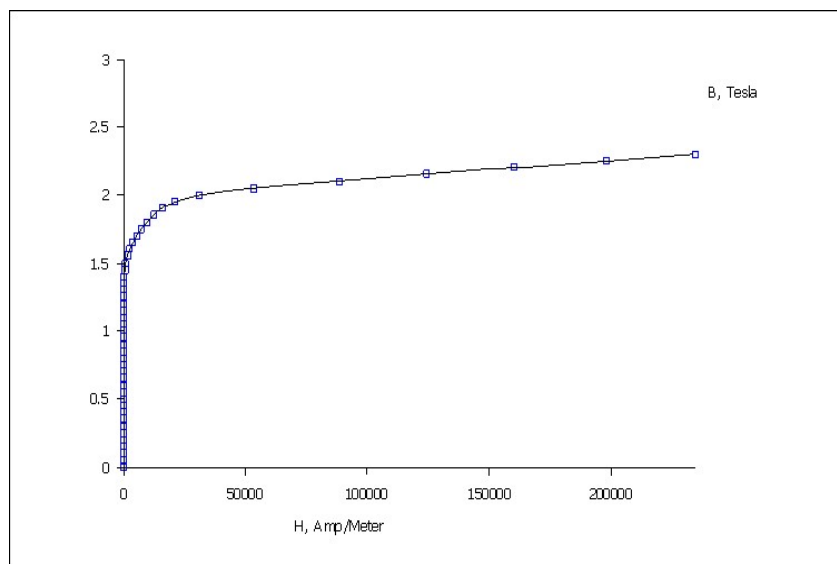


Figure III.11 Courbe $B(H)$ du matériau utilisé.

Paramètres	Symbole	Dimensions
Rayon externe du stator	Res	50 [mm]
Rayon interne du stator	Ris	32 [mm]
Rayon extérieur du rotor	Rer	31.7 [mm]
Rayon interne du rotor	Rir	9.8 [mm]
Hauteur de la culasse du stator	Hes	18 [mm]
Hauteur de la culasse du rotor	Hcr	21.9 [mm]
Epaisseur des barrières du flux	e_b	x_1
Ouverture angulaire du rotor	β_r	x_2
Longueur des barrières du flux	Lb	20.5 [mm]
Largeur des barrières du flux	Rg	4.2 [mm]

Tableau III.2 Paramètres de conception de la MSRV étudiée.

III.5.2.1 Analyse du champ électromagnétique dans la MSRV

La méthode des éléments finis est appliquée pour résoudre l'équation différentielle aux dérivées partielles du champ électromagnétique dans la machine considérée. En régime magnétostatique, elle est donnée par:

$$\text{rot}(\nu \cdot \text{rot}(\vec{A})) = \mathbf{J} \quad (\text{III.30})$$

avec :

ν est la reluctivité magnétique ; \vec{A} est le potentiel vecteur magnétique et \mathbf{J} la densité du courant dans les encoches statoriques, son amplitude est $1.9456 \cdot 10^4 [\text{A/m}^2]$.

La solution éléments finis est obtenue par l'utilisation du logiciel FEMM [w2]. La figure III.12 montre les conditions aux limites imposées sur les frontières de la machine ainsi que la répartition des courants triphasés dans les encoches.

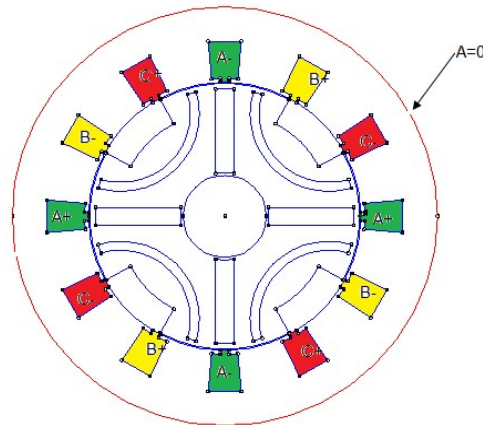


Figure III.12 Conditions aux limites et distribution des courants dans la MSRVM étudiée.

A, B et C : les trois phases du système triphasé.

III.5.2.2 Formulation du problème d'optimisation

L'objectif consiste à maximiser le couple électromagnétique. Mathématiquement, cet objectif peut être tout simplement traduit par cette équation :

$$f = \max(Ce(x_i)) \quad i = 1, 2 \quad (\text{III.31})$$

où :

x_i est le vecteur des paramètres de conception.

Le problème d'optimisation considéré peut être formulé de deux manières différentes, c'est-à-dire donné par deux fonctions objectifs.

❖ Première fonction objectif

Elle traduit directement l'objectif souhaité, elle est donnée par :

$$f_1 = \frac{1}{\sqrt{\Gamma/n}} \quad (\text{III.32})$$

Où :

f_1 est la fonction objectif;

n est le nombre de positions angulaires du rotor, θ , dans lesquelles le couple électromagnétique est calculé. Il est choisi égal à 7.

Les sept positions, angulaires du rotor, choisies sont les suivantes :

$$\theta_k = [6^\circ \ 10^\circ \ 18^\circ \ 21^\circ \ 25^\circ \ 34^\circ \ 37^\circ];$$

$$\Gamma = \sum_k ce^2(\theta_k) \quad , \quad k = 1, \dots, n. \quad (\text{III.33})$$

ce est le couple électromagnétique à chaque position angulaire du rotor [N.m] ;

Γ exprime la somme des valeurs du couple électromagnétique aux différentes positions angulaires du rotor θ_k .

❖ Deuxième fonction objectif

Une deuxième formulation du problème d'optimisation étudié a été proposée. Elle consiste à mettre la fonction objectif sous la forme d'une somme des écarts types entre les valeurs du couple électromagnétique obtenu avec les valeurs initiales des paramètres géométriques du rotor et celui obtenu avec les nouvelles valeurs de ces paramètres à chaque itération de l'algorithme d'optimisation appliqué. Dans les deux cas, le couple est calculé aux sept positions choisies θ_k , où le vecteur des positions choisies est le suivant :

$$\theta_k = [6^\circ \ 12^\circ \ 16^\circ \ 19^\circ \ 27^\circ \ 35^\circ \ 43^\circ] ;$$

La deuxième fonction objectif est donnée par l'équation suivante :

$$f_2 = \sum_k (7 \times ce_{k_{nit}}^2 - ce_{k_{cal}}^2) \quad k = 1, \dots, 7 \quad (\text{III.34})$$

avec :

f_2 est la fonction objectif;

$ce_{k_{nit}}$ est la valeur du couple calculé au point k correspondant à la position rotorique θ_k pour les paramètres géométriques initiaux du rotor.

$ce_{k_{cal}}$ est la valeur du couple calculé au point k correspondant à la position rotorique θ_k pour les paramètres géométriques du rotor obtenus à chaque itération de l'algorithme d'optimisation utilisé.

❖ Paramètres d'optimisation

Sur la base de l'étude réalisée par Serhoud [SERH-09] visant à identifier les paramètres géométriques les plus influents sur la valeur du couple électromagnétique, où il a considéré plusieurs paramètres, dont les plus importants sont l'ouverture angulaire des pôles rotoriques β_r et l'épaisseur des barrières du flux e_b . Dans ce cas, l'épaisseur de l'entrefer et le diamètre extérieur du moteur restent inchangés.

Les paramètres d'optimisation sont [SERH-09]:

- L'ouverture angulaire du pôle rotorique β_r .
- L'épaisseur des barrières du flux e_b .

$$x_i = [e_b, \beta_r] \quad i = 1, 2 \quad (\text{III.35})$$

Généralement l'ouverture du pôle est choisie selon la relation suivante:

$$0.65 \tau_p < \beta_r < 0.75 \tau_p \quad (\text{III.36})$$

Avec :

$\tau_p = \pi/2P$ [Radian] désigne le pas polaire et P le nombre de paires de pôles.

Les limites de variation des deux paramètres d'optimisation sont indiquées dans le tableau III.3.

Paramètres	Valeur minimale	Valeur maximale
e_b [mm]	0.4500	1.8804
$\beta_r/2$ [°]	14.6250	16.8750

Tableau III.3 Limites de variation des paramètres d'optimisation.

Une contrainte est considérée dans ce problème. Il s'agit de la valeur maximale de la densité du flux qui doit être inférieure à 2 Tesla (induction de saturation des circuits magnétiques) [20].

III.5.3 Machine synchrone à double excitation

Cette application concerne une machine synchrone à double excitation, MSDE. La structure de la machine considérée ainsi que ses différents paramètres géométriques sont illustrés dans la figure III.13 [DIRI-15], [DIRI-18] et [OUAG-15].

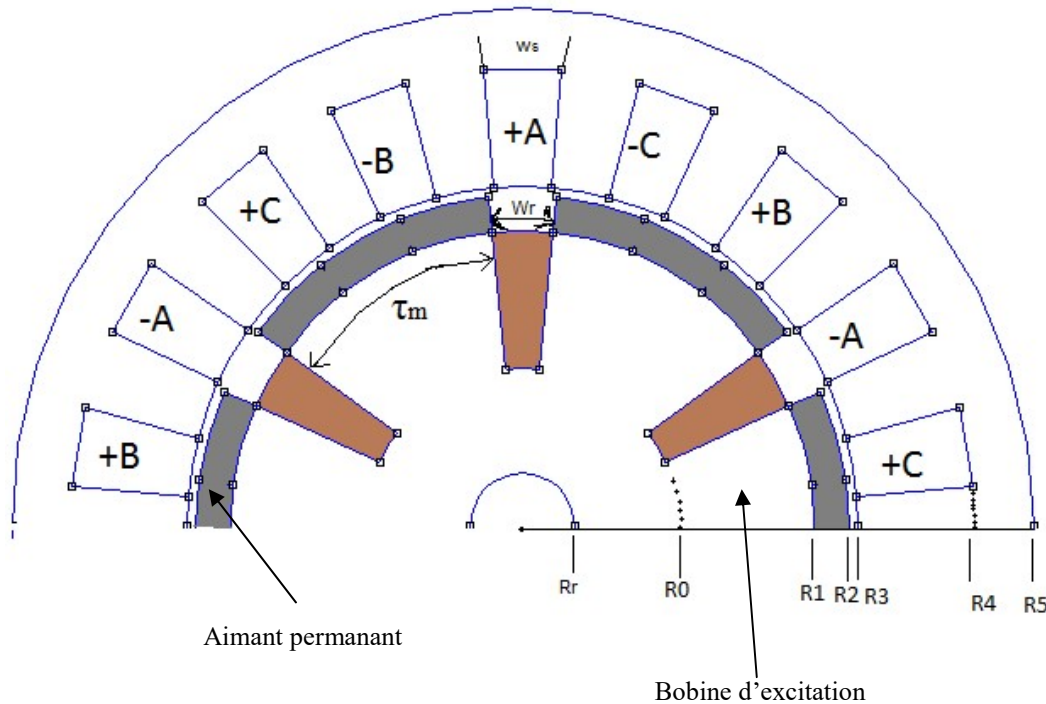


Figure III.13 Géométrie de la machine synchrone à double excitation étudiée. A, B et C sont les trois phases de l'enroulement statorique.

Les principales caractéristiques géométriques de la machine sont regroupées dans le tableau III.4. Cependant, les caractéristiques physiques sont données dans le tableau III.5.

Paramètres	Valeurs
Entrefer mécanique [mm]	1.5 [mm]
Nombre de paires de pôles P	3
l'arc polaire τ_p [rad]	$\pi/3$
Rayon minimal de la bobine d'excitation R0	26.7 [mm]
Rayon de l'arbre rotorique Rr	$R_0/3$ [mm]
Rayon interne de l'aimant R1	50 [mm]
Rayon externe de l'aimant R2	56 [mm]
Rayon interne du stator R3	57.5 [mm]
Rayon des dents statoriques R4	77.5 [mm]
Rayon externe du stator R5	87.5 [mm]
Ouverture angulaire de l'aimant τ_m [rad]	$0.8 \cdot \tau_p$
Largeur de l'encoche statorique W_s [rad]	$\pi/18$
Largeur des bobines d'excitation rotoriques w_r [rad]	$\pi/15$
Longueur de la machine L [m]	0.675

Tableau III.4 Caractéristiques géométriques de la machine considérée.

Paramètres	Valeurs
La perméabilité relative des circuits magnétiques μ_r	7500
L'aimantation rémanente de l'aimant (terre rare de type Nd-Fe-B) B_r [T]	1
Type d'aimantation	Radiale
Densité du courant d'excitation dans les bobines rotoriques	$5[A/mm^2]$
Densité du courant triphasé dans les encoches statoriques	$10[A/mm^2]$

Tableau III.5 Caractéristiques physiques de la machine considérée.

III.5.3.1 Analyse du champ électromagnétique dans la MSDE

Le problème électromagnétique considéré dans cette application est magnétostatique linéaire, il est régi par l'équation différentielle aux dérivées partielles suivante :

$$\text{rot}(\nu \cdot \text{rot}(\vec{A})) = \mathbf{J} + \text{rot} \nu B_r \quad (\text{III.37})$$

avec :

ν est la reluctivité magnétique ; \vec{A} est le potentiel vecteur magnétique; \mathbf{J} la densité du courant dans les bobines rotoriques et B_r l'induction rémanente de l'aimant.

L'EDP est résolue par la méthode des éléments finis en utilisant le logiciel FEMM [w2].

III.5. 3.2 Formulation du problème d'optimisation

Dans cette application on vise à maximiser le couple électromagnétique de la MSDE tout en minimisant le volume de l'aimant. Donc on a deux objectifs ou deux fonctions objectifs à considérer simultanément ; soit f_1 la fonction objectif qui représente la maximisation du couple et f_2 celle qui représente la minimisation du volume de l'aimant. Dans ce cas le problème d'optimisation n'est plus mono-objectif mais il est multi-objectif. Plusieurs méthodes existent pour formuler ce type de problèmes, parmi elles on trouve les méthodes d'agrégation [W3]. On a choisi la méthode d'agrégation par pondération car elle est plus simple à mettre en œuvre et ne nécessite pas la modification de l'algorithme de résolution, originalement conçu pour la résolution des problèmes mono-objectifs. Ces algorithmes, tels que le PSO et le TLBO par exemple, sont donc utilisés directement sans aucune modification ou adaptation au problème multi-objectif traité.

Le problème d'optimisation posé est bi-objectif ou multi-objectif, mais il est traité comme un problème mono objectif, dont la formulation est la suivante :

❖ Fonction objectif

On transforme le problème original, multi-objectif, à un autre à objectif unique ou mono-objectif dont la fonction objectif est donnée par l'équation III.38.

$$F = w_1 \cdot f_1 + w_2 \cdot f_2 \quad (\text{III.38})$$

avec :

w_1 et w_2 sont les poids de pondération, cette pondération est liée au critère à minimiser.

$$f_1 = \frac{1}{\sum_k ce^2(\theta_k)} \quad k=1, \dots, 10 \quad (\text{III.39})$$

ce : Le couple électromagnétique à la position angulaire θ_k .

L'indice k réfère à la position angulaire du rotor θ_k dans laquelle le couple électromagnétique est calculé. Dix positions distantes de 2° sont considérées sur l'intervalle allant de 10° à 350° .

$$f_2 = L \cdot \pi \cdot \frac{\tau_m}{360} \cdot (R_2^2 - R_1^2) \quad (\text{III.40})$$

où :

L est la longueur de la machine.

τ_m est l'ouverture angulaire de l'aimant.

❖ Paramètres d'optimisation

Pour choisir les paramètres d'optimisation les plus influents sur l'objectif considéré on a effectué une étude paramétrique dans laquelle tous les paramètres géométriques de la

machine, voir la figure III.13, sont considérés, où chaque fois on varie la valeur d'un paramètre sur un certain intervalle tout en gardant les autres paramètres fixes.

A partir des résultats de l'étude paramétrique effectuée on a retenu les paramètres d'optimisation suivants :

Le rayon rotorique R_1 ;

L'épaisseur de l'entrefer e ;

L'ouverture angulaire de l'aimant τ_m .

Le rayon de l'aimant R_2 , dépend de l'épaisseur de l'entrefer et du rayon d'alésage R_3 . Il est considéré fixe et donné par :

$$R_2 = R_3 - e \quad (\text{III.41})$$

Les limites de variation des trois paramètres de conception sont indiquées dans le tableau III.6.

Paramètre	Valeur minimale	Valeur maximale
$R_1[\text{mm}]$	50	54
$e[\text{mm}]$	0.7	1.7
$\tau_m[^\circ]$	$0.55 \cdot \tau_p$	$0.95 \cdot \tau_p$

Tableau III.6 Limites de variation des paramètres d'optimisation de la MSDE étudiée.

Finalement le problème d'optimisation est donné par la formulation suivante:

$$\begin{cases} \text{Minimiser } (F(X)) \\ X = [R_1, e, \tau_m] \end{cases} \quad (\text{III.42})$$

où :

F est la fonction objectif ;

X est le vecteur des variables d'optimisation.

III.6 Conclusion

Ce chapitre a été consacré à la présentation de la démarche de conception optimale des machines électriques, où les différentes étapes nécessaires ont été discutées. La démarche de la conception optimale basée sur l'utilisation conjointe d'une modélisation numérique de la machine, pour le calcul du champ électromagnétique, et une stratégie d'optimisation semble la plus intéressante parmi toutes celles existantes. Elle permet l'automatisation du processus de conception et offre une grande souplesse d'utilisation aux concepteurs.

La modélisation numérique par la méthode des éléments finis reste la plus fiable, la plus rigoureuse et la plus générale pour analyser le champ électromagnétique dans les machines électriques. Une présentation de la méthode a été également faite dans ce chapitre.

Les cahiers des charges de trois applications différentes concernant une presse à aimant, considérée comme l'un des problèmes test internationaux, une machine à reluctance variable à barrières du flux et une machine synchrone à double excitation ont été présentés.

Le prochain chapitre fait l'objet de la mise en œuvre de quelques algorithmes d'optimisation appartenant à la classe des méthodes d'Intelligence Artificielle et des améliorations apportées à leurs versions originales au cours de ce travail

*Chapitre IV : Stratégies
d'hybridation appliquées aux
algorithmes d'Intelligence
d'Essaim*

IV.1 Introduction

Ce chapitre est consacré à la présentation et la mise en œuvre des différentes améliorations introduites sur les versions originales de certains algorithmes d'Intelligence d'Essaim, souvent connus sous le nom des méta-heuristiques. Ces améliorations sont le résultat de plusieurs hybridations appliquées aux algorithmes suivants : le PSO, le TLBO et le BA. Les pseudo-codes de ces algorithmes, déjà présentés dans le chapitre II, sont implémentés sous MATLAB et appliqués à des fonctions Benchmark, fonctions test, dont le but est de bien comprendre leur mécanisme d'optimisation et connaître les avantages et les inconvénients de chacun d'eux. En effet, cela nous a permis, au premier lieu de maîtriser le réglage de leurs paramètres de contrôle et ensuite d'acquérir les informations nécessaires pour agir dans le sens d'améliorer leurs performances.

Les nouvelles versions des algorithmes classiques, autrement dit les versions hybrides développées au cours de ce travail, sont appliquées aux mêmes fonctions test que les versions originales afin de les valider et de faire une étude comparative entre elles.

IV.2 Types d'hybridation

L'objectif de l'hybridation est de réduire les inconvénients des méthodes d'optimisation, soient stochastiques ou déterministes, de sorte à établir un certain équilibre entre les deux phases principales dans un processus d'optimisation, l'exploration et l'exploitation, et assurer une bonne convergence vers la meilleure solution optimale, au lieu d'être piégées dans un optimum local.

Dans la littérature, plusieurs techniques d'hybridation ou de réassemblage entre deux ou plus des méthodes pour construire une nouvelle plus robuste et plus forte existent. Elles peuvent être classées en trois techniques principales comme est illustré dans la figure IV.1 [BOUA-11]

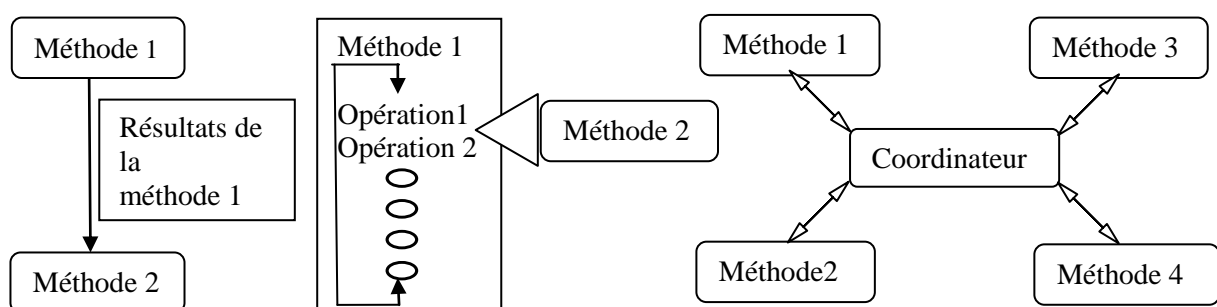


Figure IV.1:Types d'hybridation.

➤ **Hybridation séquentielle**

Ce type d'hybridation n'introduit pas des modifications dans les méthodes à hybrider. C'est une hybridation simple, où les solutions trouvées par la première méthode sont utilisées comme des solutions initiales de la deuxième méthode.

➤ **Hybridation parallèle synchrone**

Dans ce type, on intègre une méthode dans le processus d'une autre méthode, ce qui donne lieu à de grandes interactions entre les méthodes. Ce type est considéré plus fin que le premier.

➤ **Hybridation parallèle asynchrone**

Dans ce type, les méthodes sont exécutées au même temps, simultanément ou parallèlement. Avec l'existence d'un coordinateur qui assure la distribution et la transformation des informations entre ces méthodes, ce type d'hybridation introduit des modifications dans les méthodes originales pour assurer la transformation des résultats entre les méthodes.

Dans ce travail, les deux types d'hybridation, séquentielle et parallèle synchrone, sont adoptés. Dans les sections ci-dessous, on va discuter les différentes hybridations développées au cours de ce travail.

IV.3 Hybridation des réseaux de neurones artificiels avec des algorithmes d'Intelligence d'Essaim [BOUR-18]

L'objectif fondamental de cette utilisation conjointe des deux branches de l'intelligence artificielle est d'améliorer les performances des algorithmes d'Intelligence d'Essaim, comme le PSO et le TLBO, et réduire le temps de calcul très important que nécessitent ces algorithmes pour atteindre la solution optimale, à cause des multiples appels au calcul des fonctions objectifs et des fonctions contraintes, surtout lorsque une modélisation éléments finis est utilisée, et c'est le cas d'ailleurs de ce travail.

Pour ce faire, on a proposé un modèle approximé de celui original, en utilisant la propriété d'approximation du réseau multicouche (MLP) (exprimée par le théorème d'Hornik).

A partir d'un nombre d'exemples (chaque exemple contient les entrées du système à optimiser et les sorties correspondantes à ces entrées, qui sont généralement les valeurs de la fonction objectif et les contraintes si elles existent) on forme une base de données utilisée pour entraîner et apprendre le modèle approximatif (le réseau MLP).

Après l'étape de l'apprentissage, le modèle est prêt à l'utilisation, c'est à dire le MLP a appris, on applique l'algorithme d'optimisation à ce modèle (approximatif) au lieu de son application directe au modèle original avec les éléments finis). Le but de cette utilisation est l'accélération du processus d'optimisation toute en éliminant les multiples appels aux calculs par éléments finis.

Pour cette hybridation avec le réseau de neurones MLP on a choisi deux algorithmes le PSO et le TLBO. Il en résulte les deux algorithmes hybrides NN-PSO et NN-TLBO. L'organigramme qui présente cette hybridation est illustré dans la figure IV.2.

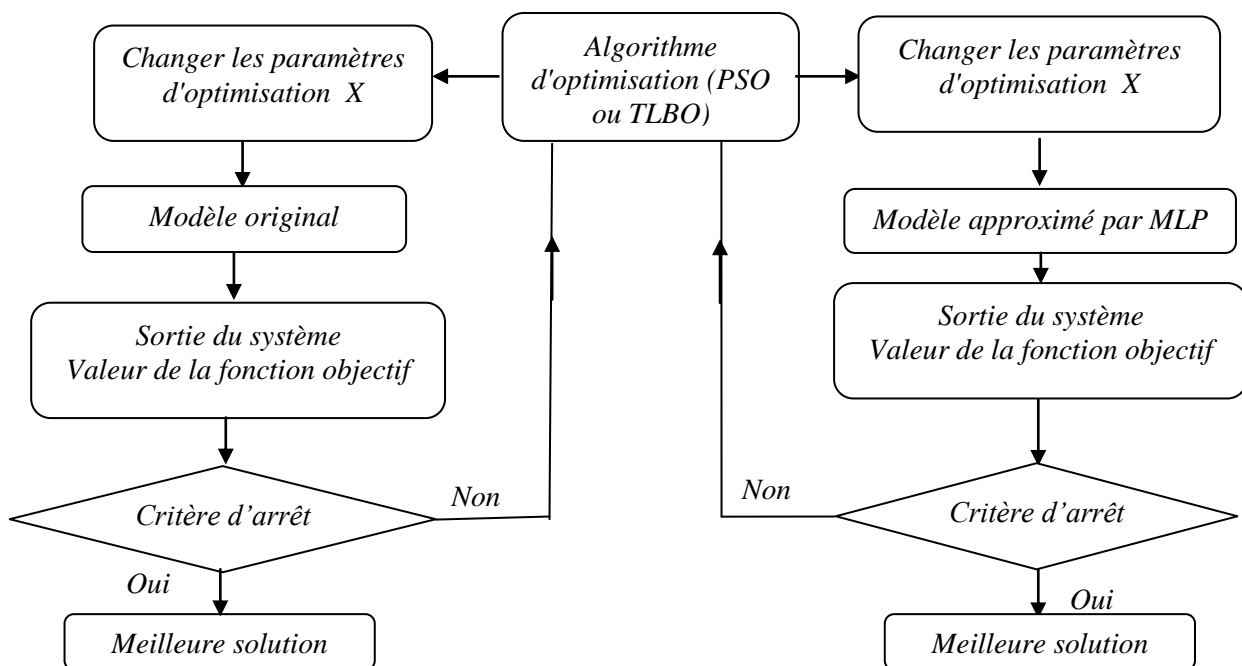


Figure IV.2: Modèles original (à gauche) et approché : hybridation entre le MLP et les méthodes d'optimisation (à droite).

Au début, cette technique d'hybridation a été appliquée pour résoudre trois fonctions test connues, les résultats sont présentés à la fin du chapitre.

IV.4 Hybridation entre méthodes méta-heuristiques et méthodes exactes

Comme on a déjà vu précédemment, l'inconvénient majeur des algorithmes basés sur l'intelligence artificielle ou l'Intelligence d'Essaim est le temps de calcul très élevé. Ce problème est dû en fait au grand nombre d'appels à la fonction objectif, voir donc multiples appels au modèle éléments finis. Par contre, la qualité de la solution est acceptable, mais elle reste inférieure à celle obtenue par une méthode exacte lorsque son contexte initial est bien

choisi (solution initiale proche de celle optimale). Ce type d'hybridation est classé dans l'hybridation séquentielle.

Cette hybridation est étudiée à travers le couplage de l'algorithme BA avec la méthode du simplexe et celui de TLBO avec les méthodes du simplexe et de Quasi-Newton. La validité de ce modèle hybride est vérifiée à travers son application aux trois fonctions test différentes.

IV.4.1 Hybridation entre le BA et la méthode de Nelder-Mead simplexe

L'algorithme des chauves souris, le BA, est un algorithme très compliqué, où il possède plusieurs paramètres de contrôle, il converge vers la solution optimale lorsque, ses paramètres de contrôle sont bien réglés ou bien adaptés au problème d'optimisation posé. Ce réglage est très difficile à cause des grandes interactions entre ces paramètres. Un mauvais réglage de ces derniers conduit à des temps de calcul prohibitifs et une convergence assez lente.

La tâche principale des paramètres de contrôle de l'algorithme BA est le balancement entre la recherche locale et la recherche globale. Pour trouver l'équilibre entre ces deux phases de recherche de l'optimum, sachant que le déséquilibre entre elles est souvent la cause de la divergence de l'algorithme BA, on a proposé une technique indirecte pour réduire l'influence du choix non approprié des paramètres de contrôle, cette technique consiste à coupler le BA avec la méthode exacte de Nelder-Mead.

Le mécanisme d'hybridation entre le BA et NM est présenté dans l'organigramme de la figure IV.3.

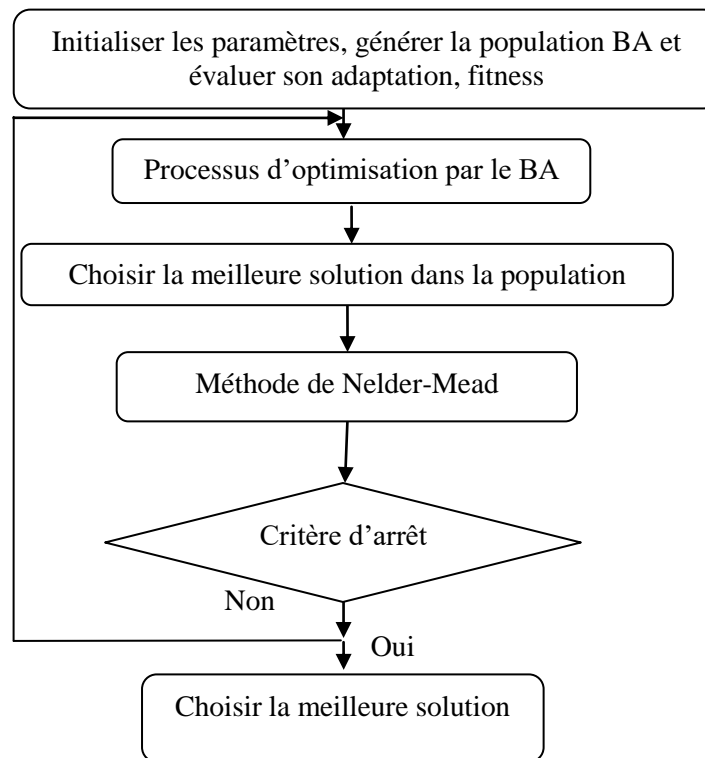


Figure IV.3 Hybridation entre le BA et le NM

IV.4.2 Hybridation entre le TLBO et les méthodes exactes: Nelder-Mead simplexe et Quasi-Newton

L'algorithme TLBO est simple, ne contient aucun paramètre de contrôle, sauf le nombre des itérations. Cependant, son mécanisme ne permet pas une bonne exploitation de l'entourage de la solution trouvée.

Pour renforcer la recherche locale autour de la solution trouvée (exploitation de l'entourage), augmenter la vitesse de convergence, et diminuer le nombre d'appels à la fonction objectif, on a procédé à l'hybridation de TLBO avec deux méthodes exactes. La première est la méthode de Nelder-Mead simplexe et la deuxième celle de Quasi-Newton.

L'hybridation avec les méthodes exactes n'est pas la seule possibilité pour améliorer la procédure d'optimisation par les méthodes de l'intelligence artificielle. Un autre type d'hybridation a été aussi investigué dans ce travail, il est basé sur la combinaison et l'intégration de différentes phases de plusieurs algorithmes pour créer un nouvel algorithme plus robuste que les algorithmes originaux. Ce type d'hybridation est discuté dans la prochaine section.

IV.5 Hybridation entre méthodes méta-heuristiques (méta-heuristique-méta-heuristique)

Les algorithmes TLBO et BA sont des algorithmes stochastiques récents, leur apparition se date à partir de l'année 2010. Malgré que tous les deux appartiennent à la même classe, leurs propriétés pour la recherche de l'optimum sont différentes. Sur la base des différences existantes entre les algorithmes méta-heuristiques qu'on s'est investi dans leur hybridation. Il s'agit d'une hybridation synchrone parallèle.

Deux techniques ont été développées dans ce travail. La première consiste à intégrer une phase de croisement, similaire à celle utilisée dans l'algorithme génétique, dans l'algorithme BA. Cependant, la deuxième combine les propriétés des algorithmes TLBO et BA, où l'équation de la recherche locale de l'algorithme BA est intégrée dans le processus d'optimisation par l'algorithme TLBO.

IV.5.1 BA avec la technique de croisement (Crossover-BA ou C-BA) [BOUR-19a]

Pour éviter les problèmes de convergence résultants du mauvais réglage des paramètres de contrôle de l'algorithme BA, la propriété du croisement, utilisée dans l'algorithme génétique, est alors utilisée dans le BA, où une nouvelle phase de croisement est intégrée dans son algorithme original ou classique. Cette phase additionnelle permet au BA de bien exploiter l'entourage de la solution optimale à chaque itération de son processus d'optimisation [BOUR-19a].

Avec cette nouvelle phase supplémentaire, à chaque itération du processus de l'algorithme BA, une population secondaire est ainsi générée. Trois types de croisement ont été développés.

- **Type 1:** Le croisement a été effectué entre les deux ou les trois meilleures solutions de la population originale; il en résulte une population secondaire. La meilleure solution dans la population secondaire a été choisie et comparée avec celle de la population principale. Le meilleur résultat a été maintenu.
- **Type 2:** Similaire au premier type, sauf que dans ce cas, une comparaison a été faite entre les mauvaises solutions de la population originale et les solutions de la nouvelle population. Les meilleures solutions remplacent les mauvaises.
- **Type 3:** La moyenne de n meilleures solutions a été calculée afin de générer une nouvelle solution. Ensuite, un croisement a été fait entre la meilleure solution de la population initiale et la moyenne calculée.

Pour mieux expliquer le processus de l'algorithme hybride C-BA, on considère le problème tridimensionnel avec un volume de la population égale à n , il est donné par l'équation (IV.1). Pour résoudre ce problème avec la méthode C-BA, à chaque itération du processus, deux étapes doivent être considérées.

Première étape

La première étape est similaire à celle de BA classique; après, les solutions x_i sont ordonnées en fonction des valeurs de la fonction objectif, f_i , par ordre décroissant; la première ligne correspond aux meilleures solutions (valeurs minimales de la fonction objectif), cependant la dernière ligne correspond aux plus mauvaises valeurs (valeurs maximales de la fonction objectif).

Seconde étape

C'est l'étape de croisement. Il est fait entre la matrice ordonnée, qui contient les solutions ordonnées, et leurs adaptations ou fitness correspondantes.

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ A_{n-1} \\ A_n \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & f_1 \\ x_{2,1} & x_{2,2} & x_{2,3} & f_2 \\ x_{3,1} & x_{3,2} & x_{3,3} & f_3 \\ x_{4,1} & x_{4,2} & x_{4,3} & f_4 \\ x_{5,2} & x_{5,2} & x_{5,3} & f_5 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n-1,1} & x_{n-1,2} & x_{n-1,3} & f_{n-1} \\ x_{n,1} & x_{n,2} & x_{n,3} & f_n \end{bmatrix} \quad (IV.1)$$

Avec

$A_{i=1,..,n}$ sont les intensités des solutions x_i

\mathbf{X} : représente les solutions de la population originale, ordonnée selon la valeur de leur fitness (f_1 est la meilleure solution, f_n est la mauvaise solution).

f_2, \dots, f_n : représente les fitness des solutions ordonnées

Comme on a déjà mentionné précédemment, le croisement peut être fait suivant trois manières différentes:

Pour le type 1

1. Le croisement est fait entre la première et la deuxième solution, lignes 1 et 2 de l'équation IV.1, une nouvelle matrice est générée, pour l'exemple proposé, elle donne (2^3) solutions; après, les deux premières lignes de la population originale sont incluses dans la nouvelle

population, une nouvelle matrice S est générée où l'adaptation de chaque solution est calculée.

2. La meilleure solution en S est comparée avec celle de la population originale ($X_{Ordonnée}$). La mauvaise solution est supprimée et remplacée par la meilleure.

Pour le type 2

Pour le faire, on suit les étapes ci-dessous :

1. Les deux premières étapes sont identiques à celles du type 1.

2. Les solutions de la nouvelle population S sont comparées avec celles de la population principale. Les meilleures solutions sont conservées et les mauvaises sont supprimées. L'intensité A pour les nouvelles solutions est calculée par:

$$A_{si} = \text{rand}.\text{Mean}(A_{mi}) \quad (\text{IV.2})$$

où:

A_{mi} : présente l'intensité de l'ancienne chauve souris i , supprimée de la population principale.

A_{si} : présente l'intensité de la nouvelle chauve souris i , intégrée dans la population principale.

Mean : désigne la moyenne.

Les nouvelles solutions sont acceptées ou refusées en fonction de l'intensité A , selon la condition suivante :

Si ($\text{rand} < A_i \ \& \ f(x_i) < f(x^*)$)
Accepter les nouvelles solutions
Fin si

Pour le type 3

Au lieu de croiser entre deux ou trois meilleures solutions, une moyenne de n meilleures solutions est calculée afin de générer une nouvelle solution; ensuite, le croisement est fait entre la nouvelle solution générée et la meilleure solution de la population originale.

La nouvelle solution est obtenue par:

$$x_{mean} = \text{Mean}(x_i) \quad (\text{IV.3})$$

Avec: $i=1, \dots, k$;

x_{mean} : est la moyenne de k meilleures solutions.

k est choisi au hasard; il est préférable de le choisir égale à la moitié du volume de la population n .

1. Le croisement est fait entre la meilleure solution de la population originale et les nouvelles solutions x_{mean} . Pour l'exemple proposé, on croise la ligne 1 (solution x_1) de la population

originale et x_{mean} qui correspond à la moyenne des 5 meilleures solutions. La nouvelle matrice est alors obtenue et contient (2^3) solutions. La ligne 1 de la population originale, incluse dans la nouvelle population, est supprimée. La nouvelle matrice S ne contient donc que 7 solutions. L'adaptation de la nouvelle population S est également calculée.

2. On compare la nouvelle population S avec la population originale $X_{ordonnée}$, les meilleures solutions sont maintenues et les mauvaises sont supprimées.

L'intensité A pour les nouvelles solutions, incluses dans la population originale, est calculée par l'équation (IV.2).

Les nouvelles solutions sont acceptées ou refusées selon l'intensité A , suivant la condition ci-dessous :

Si ($rand < A_i \ \& \ f(x_i) < f(x^*)$)

Accepter les nouvelles solutions

Fin Si

Les étapes de l'algorithme développé sont résumées dans l'organigramme de la figure IV.4.

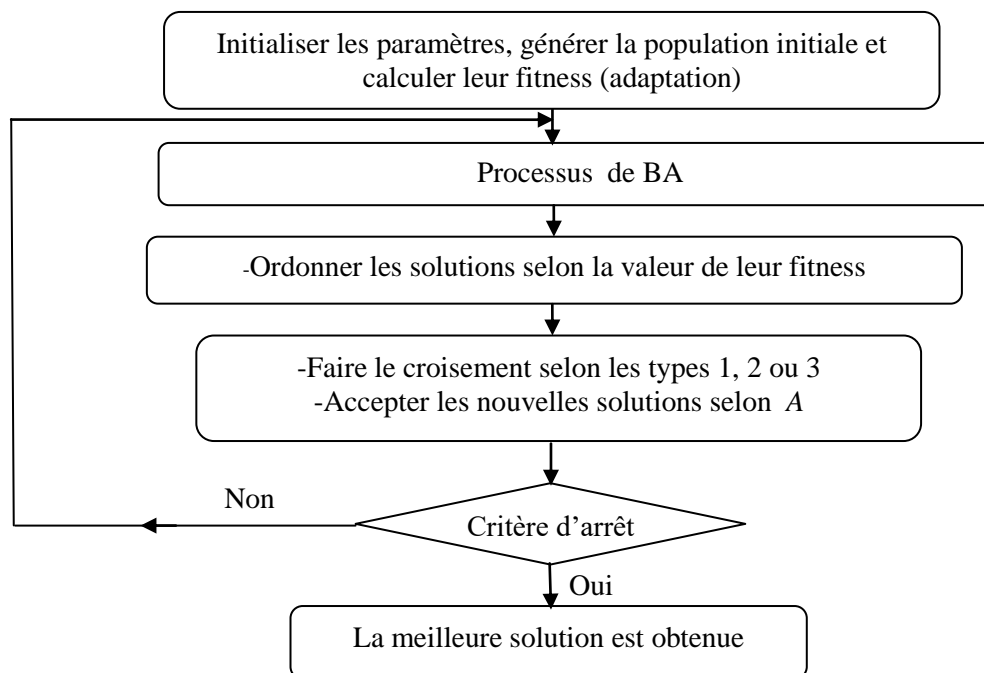


Figure IV.4 Organigramme de l'algorithme C-BA.

Le pseudo-code de l'algorithme C-BA est présenté dans la figure IV.5. Où la phase du croisement, la nouvelle phase intégrée dans l'algorithme BA original ou classique, est soulignée.

```

-Initialiser la population des chauves souris  $x_i$  ( $i = 1, 2, \dots, n$ ), la vitesse  $v_i$  et la
fréquence  $f_i$ .
-Initialiser le tau de pulsation  $r_i$  et l'intensité  $A_i$ 
Tant que (le critère d'arrêt n'est pas vérifié)
- Générer les nouvelles solutions en utilisant les équations (II.22) à (II.24)

    Si ( $\text{rand} > r_i$ )
        Sélectionner une solution parmi les meilleures solutions
        Générer des solutions locales par le vol aléatoire autour de la meilleure
solution selon l'équation (II.25)
    Fin Si

    Si ( $\text{rand} < A_i$  et  $f(x_i) < f(x^*)$ )
        Accepter les nouvelles solutions
        Augmenter  $r_i$  et réduire  $A_i$ 
    Fin Si
-Ordonner les chauves souris et trouver la meilleure solution courante  $x^*$ 
-Choisir des solutions parmi les meilleures solutions
-Générer une population secondaire par le croisement entre les
meilleures solutions
-Remplacer la mauvaise solution par la meilleure solution (type1) ou la
population secondaire dans la population originale type 2 ou 3.
    Si ( $\text{rand} < A_{si}$  &  $f(x_{si}) < f(x^*)$ )
        Accepter les nouvelles solutions
    Fin Si
Fin Tant que

```

Figure IV. 5 Pseudo-code de l'algorithme C-BA.

IV.5.2 TLBO avec la technique de recherche locale aléatoire (Radom Local Search) TLBO-RLS [BOUR-19b]

Le nouvel algorithme, TLBO-RLS, résulte de l'ajout de l'équation de la recherche locale aléatoire (Radom Local Search) de l'algorithme BA, dans le processus d'optimisation par l'algorithme TLBO.

Cette idée a été inspirée d'un type d'apprentissage connu sous le nom de l'apprentissage personnel, où un pourcentage des étudiants essaye d'élever son niveau en se basant sur des efforts personnels. Dans ce travail, on a proposé une approche basée sur la recherche locale aléatoire, elle est effectuée selon l'équation suivante :

$$x_{new} = x_{old} + \varepsilon \langle A_t \rangle \quad (IV.3)$$

avec :

ε est un nombre aléatoire qui varie dans l'intervalle $[-1,1]$.

$A_t = \langle A_{ti} \rangle$, est la moyenne de l'intensité de toutes les solutions à l'itération t .

L'intensité de la recherche locale est contrôlée par l'intensité A et le tau de pulsation r . Ces deux paramètres sont adaptés au cours du processus itératif, où, l'intensité décroît et le

tau de pulsation croit jusqu'à la fin de la recherche. A et r sont actualisés (adaptés) selon l'équation suivante [YANG-10] :

$$A_{t+1,i} = \alpha A_{t,i}; r_{t+1,i} = r_{0,i}[1 - \exp(-\gamma t)] \quad (\text{IV.4})$$

avec $0 < \alpha < 1$ et $\gamma > 0$; sont des constantes. si $t \rightarrow \infty$, alors $A_{t,i} \rightarrow 0$ et $r_{t,i} \rightarrow r_{t,0}$.

Les valeurs initiales de A : A_0 et de r : r_0 , peuvent être définies par : $A_0 \in [1, 2]$ et $r_0 \in [0, 1]$.

L'ajout de cette équation dans le processus classique de TLBO permet, de renforcer la recherche locale, c'est à dire renforcer l'exploitation de l'entourage de la solution trouvée à chaque itération de l'algorithme, réduire le temps de calcul nécessaire pour la convergence et améliorer la qualité de la solution obtenue. La mise en œuvre de l'algorithme développé, TLBO-RLS, est compliquée par rapport à celle de TLBO classique à cause des deux nouveaux paramètres de contrôle ajoutés. Il est à noter que le réglage de ces deux paramètres avec le TLBO reste plus facile qu'avec le BA.

Le pseudo-code de l'algorithme TLBO-RLS est donné dans la figure IV.6. La nouvelle phase intégrée dans le TLBO classique est soulignée.

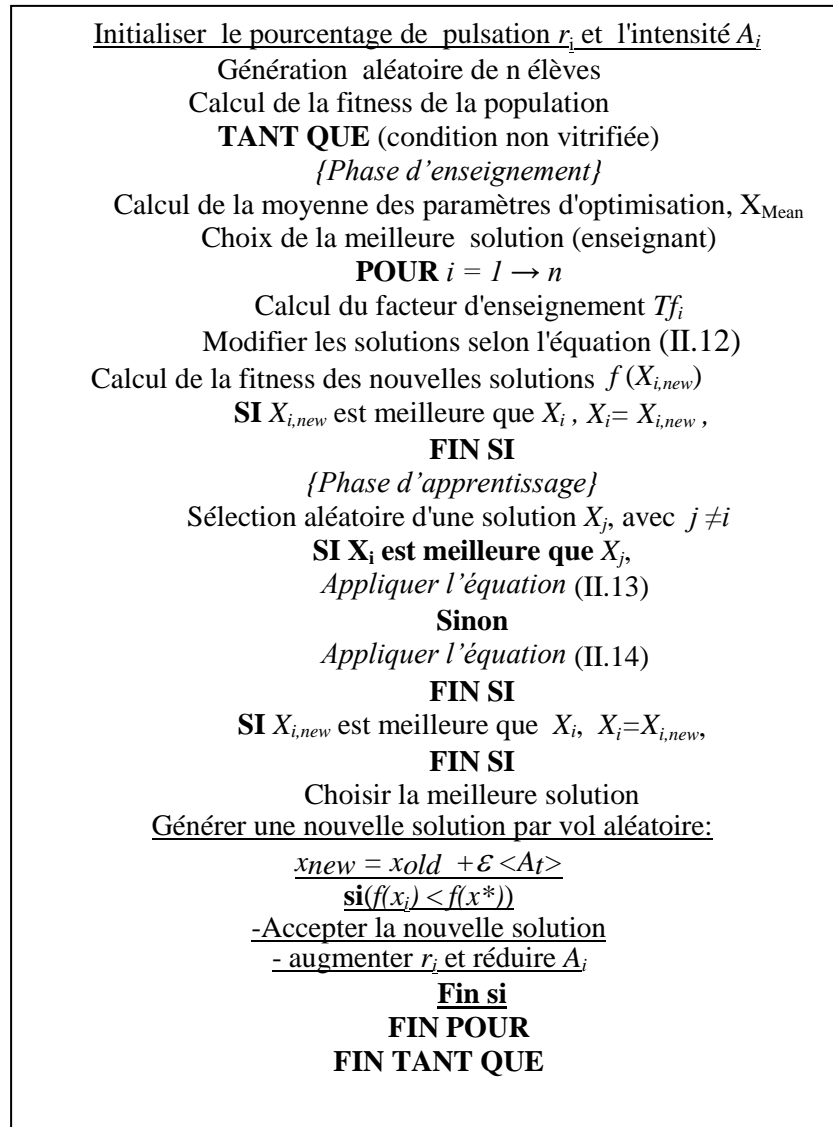


Figure IV.6 pseudo-code de TLBO-RLS.

Tous les algorithmes, versions originales et classiques, ont été appliqués à trois fonctions test. Les résultats de ces applications sont présentés dans les sections suivantes.

IV.6 Résultats et discussions

Pour étudier l'efficacité et la validité des nouvelles versions développées, on les a appliquées ainsi que les versions classiques à trois fonctions test dont la solution est connue.

➤ **Première fonction F1:** écrite sous la forme suivante :

$$F1(x) = 10n + \sum_{i=1}^{10} (x_i - 2.5)^2 - 10 \cos(2\pi(x_i - 2.5))^2 \quad (IV.5)$$

avec

$$0 \leq x_i \leq 10$$

Cette fonction est dite la fonction de Rastrigin, elle contient 100 minimums locaux et un minimum global. Les valeurs de la solution optimale x^* et la fonction objectif correspondante $f(x^*)$ sont :

$$x^*=(2.5,2.5) \text{ et } f(x^*)=0.$$

La Figure IV.7 est une présentation tridimensionnelle de la fonction.

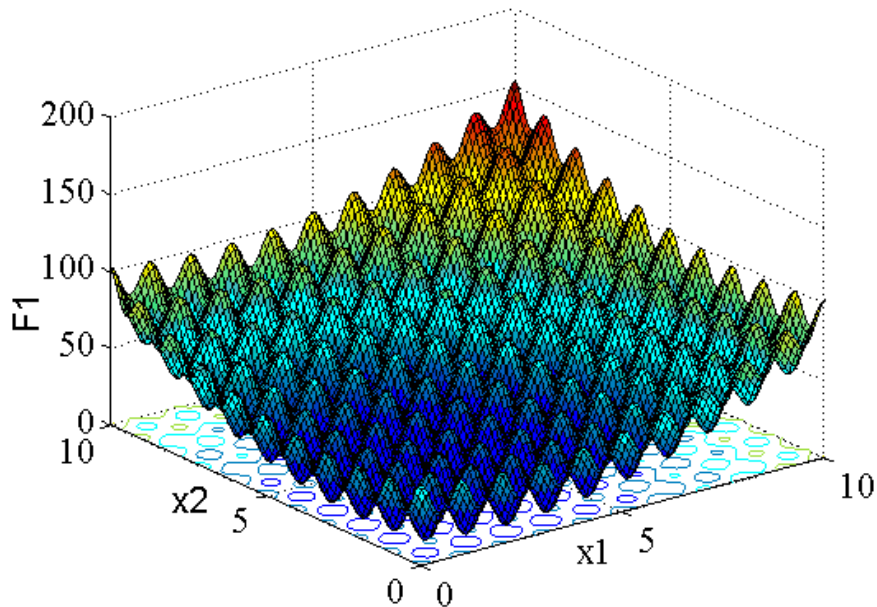


Figure IV.7: Fonction de Rastrigin, F1, en 3D.

➤ **Deuxième fonction F2:** elle est écrite sous la forme suivante :

$$F2(x_1, x_2) = 20 * (1 - \exp(-0.2 * \sqrt{0.5 * (x_1^2 + x_2^2)})) - \exp(0.5 * \cos 2\pi x_1 + \cos 2\pi x_2) + \exp(1) \quad (\text{IV.6})$$

avec : $-32.768 \leq x_1, x_2 \leq 32.768$

Cette fonction est nommée la fonction d'Ackly. Les valeurs de la solution optimale x^* et la fonction objectif correspondante $f(x^*)$ sont :

$$f(x^*)=0, x^*=(0,0)$$

La figure IV.8 est une présentation tridimensionnelle de cette fonction.

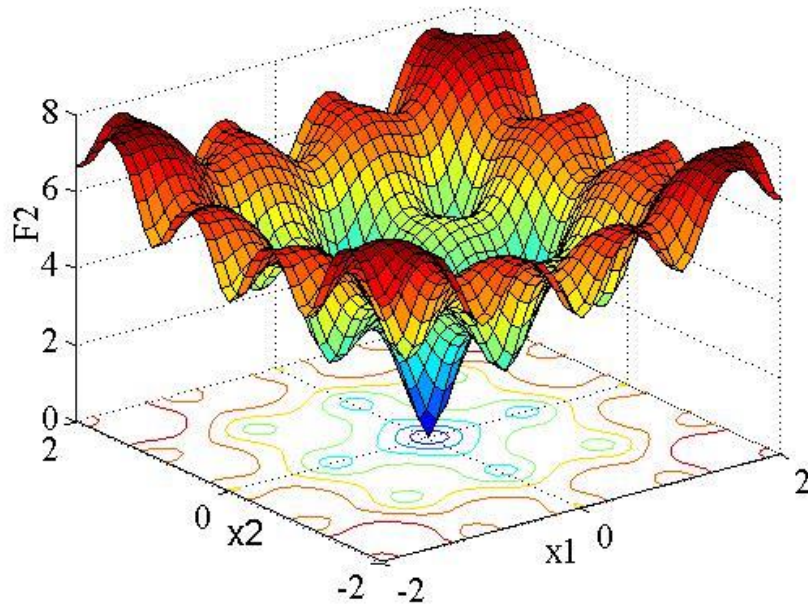


Figure IV.8 Fonction d'Ackly, F2, en 3D.

➤ **Troisième fonction F3:** donnée sous la forme suivante :

$$F_3(x_1, x_2) = 0.01 \cdot ((x_1 + 0.5))^4 - x_1^2 - 20x_1 + ((x_2 + 0.5))^4 - 30x_2^2 - 20x_2 \quad (\text{IV.7})$$

avec : $-7 \leq x_1, x_2 \leq 6$

Sa présentation 3D est donnée dans la figure IV.9.

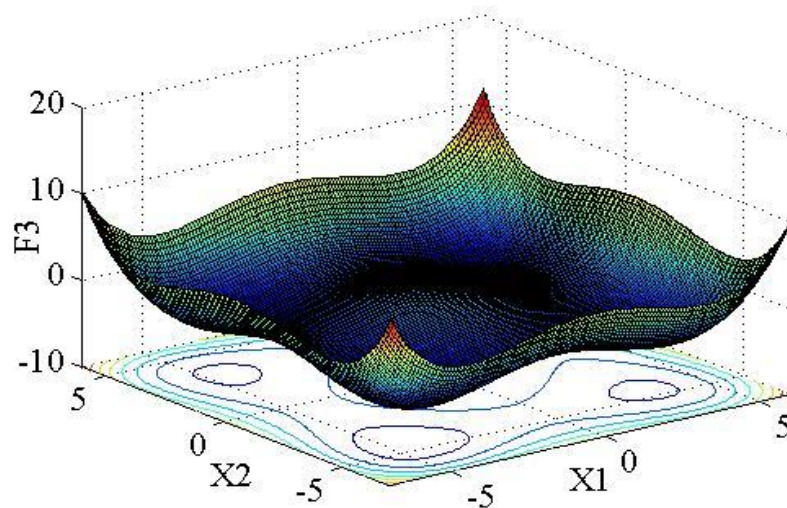


Figure IV.9 Fonction F3 en 3D.

IV.6.1 Algorithmes hybrides réseaux de neurones-Intelligence d'Essaim : NN-PSO et NN-TLBO

Les deux algorithmes hybrides, présentés dans la section IV.3, le NN-PSO et le NN-TLBO ont été appliqués pour la résolution des trois fonctions test considérées. Une comparaison entre les résultats obtenus avec ces deux algorithmes et ceux obtenus avec leurs versions classiques, PSO et TLBO, a été effectuée.

Le temps de calcul (CPU), les valeurs moyenne (MEAN) et maximale de la fonction objectif, le nombre d'appels à la fonction objectif, ainsi que l'écart type, dont l'expression est donnée par l'équation IV.8, sont évalués pour chaque algorithme. Le nombre d'itérations est fixé à 60 pour tous les algorithmes utilisés.

$$Ecart-type = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - \bar{f})^2} \quad (IV.8)$$

Avec :

n : Taille de la population ;

f_i Valeur de la fonction objectif pour la solution i de la population finale;

\bar{f} Moyenne des valeurs des fonctions objectifs de la population final , $\bar{f} = \frac{1}{n} \sum_{i=1}^n f_i$.

➤ Fonction F1

Les résultats d'optimisation concernant la fonction F1 sont résumés dans le tableau IV.1.

Algorithmes	F1	PSO	TLBO	NN-PSO	NN-TLBO
F1	0	1.5649e-12	2.2346e-12	3.6482e-2	3.6482e-2
CPU [S]	-	0.1459	0.2594	28.2911	72.7924
Ecart-type	-	1.6785e-8	2.2388e-4	5.3723e-7	8.2750e-11
Valeur maximale	-	5.8418e-8	0.0010	0.0364	0.0364
Valeur moyenne (Mean)	-	1.008e-8	0.5007e-4	0.0364	0.0364
X1	2.5	2.500	2.5000	2.5001	2.5001
X2	2.5	2.500	2.5000	2.4996	2.4996
Nombre d'appels à F1	-	1220	2220	1220	2420

Tableau IV.1 Résultats d'optimisation pour F1

La figure IV.10 présente l'évaluation de la fonction objectif, F1, avec les itérations des algorithmes appliqués.

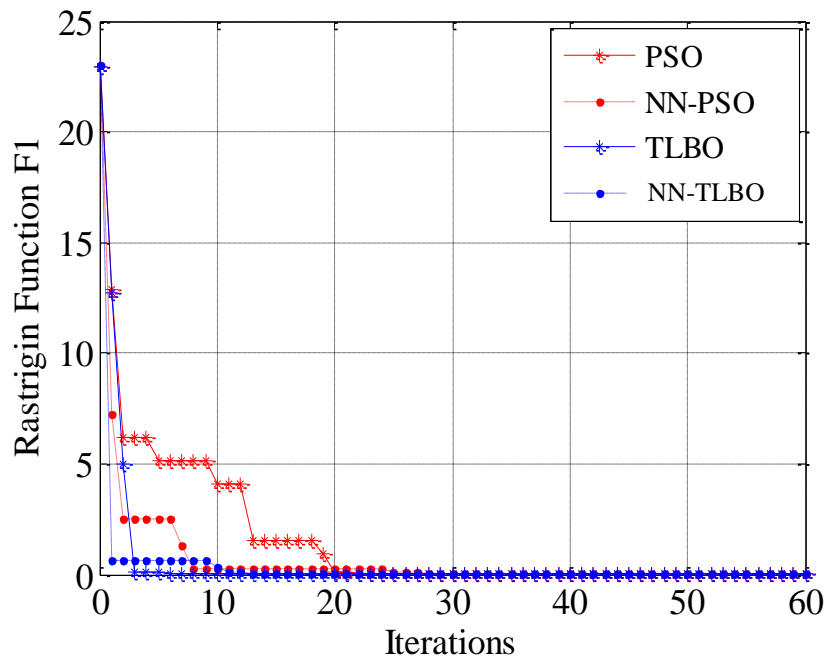


Figure IV. 10 Variation de la fonction F1 avec les itérations des algorithmes utilisés.

➤ **Fonction F2**

Les résultats issus de l'application des différents algorithmes à la fonction F2 sont regroupés dans le tableau IV.2. Tandis que sa variation avec les itérations des algorithmes utilisés est présentée dans la figure IV.11.

Algorithmes	F2	PSO	TLBO	NN-PSO	NN-TLBO
F2	0	5.5351e-7	1.1362e-1	8.4421e-1	8.4420e-1
CPU [S]	-	0.1820	0.24132	32.583	47.030
Ecart type	-	1.05723	1.9782	4.9253e-9	2.5347e-14
Valeur maximale	-	3.5368e-5	6.9988e-10	0.8442	0.8442
Valeur moyenne	-	1.6577e-5	6.9988e-10	0.84420	0.8442
X1	0	0.19164e-6	0.35849e-11	-2.0825e-1	-2.0821e-3
X2	0	-0.0396e-6	-0.18129e-11	2.3587e-1	-2.35888e-3
Nombre d'appels à F2	-	1200	2420	1220	2140

Tableau IV.2 Résultats d'optimisation pour F2.

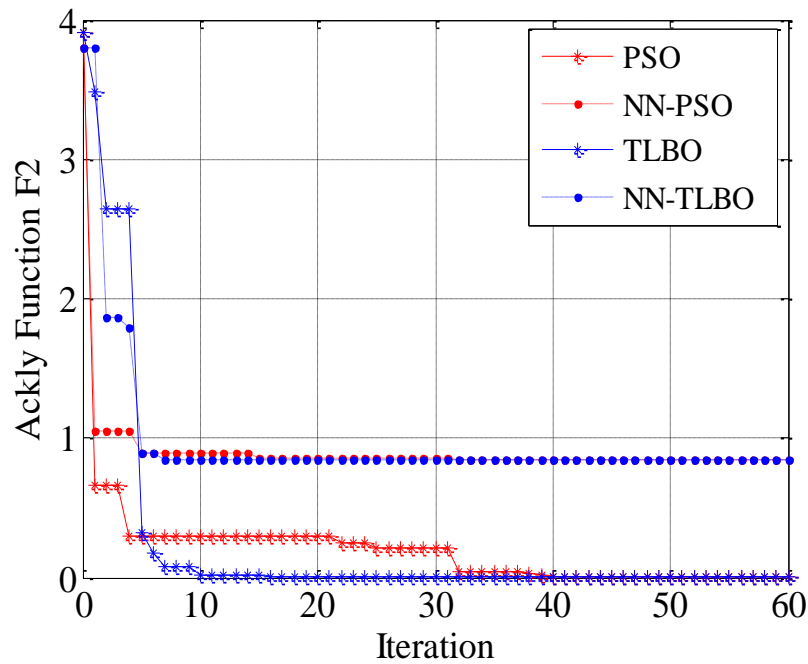


Figure IV.11 Variation de la fonction F2 avec les itérations des algorithmes utilisés.

➤ **Fonction F₃**

Les résultats d'optimisation concernant la fonction F3 sont donnés dans le tableau IV.3. La figure IV.12 présente sa variation avec les itérations des algorithmes utilisés.

Algorithmes	<i>F3</i>	<i>PSO</i>	<i>TLBO</i>	<i>NN-PSO</i>	<i>NN-TLBO</i>
F3	-5.2327	-5.2327	-5.2327	-5.2327	-5.2327
CPU [S]	-	0.031518	0.097522	23.632406	44.132259
Ecart-type	-	1.0221e-10	1.6427e-15	1.022e-11	1.9961e-14
Valeur maximale	-	-5.2327	-5.2327	-5.2327	-5.2327
Valeur moyenne	-	-5.2327	-5.2327	-5.2327	-5.2327
X1	-4.4537	-4.4537	-4.4537	-4.4538	-4.4537
X2	-4.4537	-4.4537	-4.4537	-4.4537	-4.4537
Nombre d'appels à F3	-	1220	2180	1220	2340

Tableau IV.3 Résultats d'optimisation pour la fonction F3.

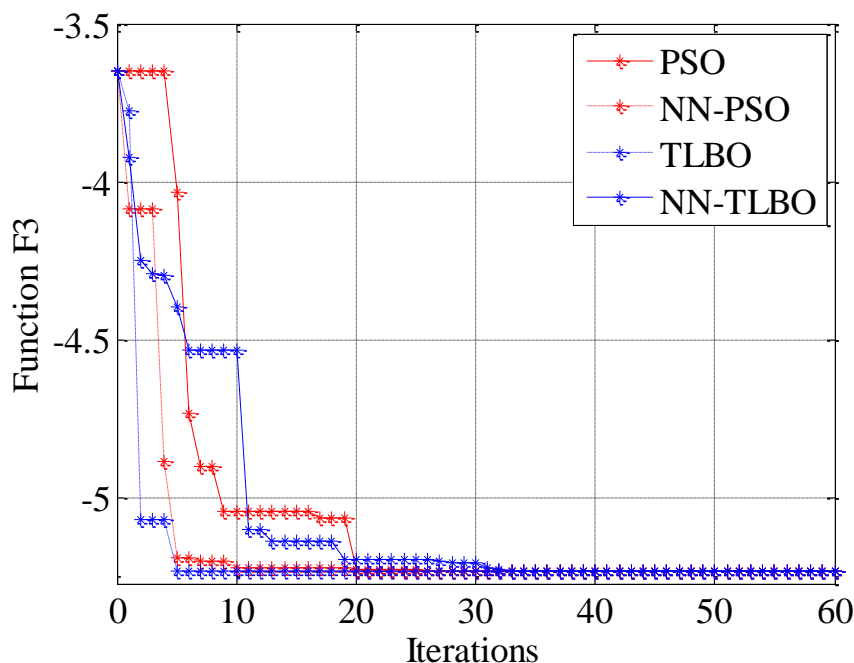


Figure IV. 12 Variation de la fonction F3 avec les itérations des algorithmes utilisés.

A partir des résultats donnés dans les tableaux IV.1, IV.2 et IV.3 nous remarquons que les solutions optimales pour les fonctions, F1, F2 et F3, obtenues avec les différents algorithmes utilisés sont proches. Une légère différence dans les valeurs des résultats, plus nette pour F1 et F2, en faveur de celles obtenues avec le PSO et le TLBO par rapport à celles obtenues avec le NN-PSO et le NN-TLBO est observée. En outre, le temps de calcul pour les deux algorithmes hybrides NN-PSO et NN-TLBO est nettement plus élevé que celui des algorithmes classiques.

La différence des résultats entre les versions hybrides et celles originales peut être imputée à la complexité des fonctions, F1 et F2, qui contiennent plusieurs minima, fonctions multimodales, et les larges intervalles de variation de leurs variables d'optimisation. Ces deux caractéristiques perturbent le réseau de neurone et l'empêchent d'émettre ou de donner la bonne approximation de la fonction objectif. Sachant qu'une bonne approximation nécessite un grand nombre d'exemples dans la base de données. Cela conduit à un réseau avec un grand nombre de neurones dans la couche cachée, impliquant donc un temps de calcul plus important pour évaluer la sortie du réseau, dû au nombre élevé de connexions entre les neurones du réseau.

Pour la fonction F3, les valeurs optimales obtenues avec l'ensemble des algorithmes appliqués sont identiques. Cette fonction est plus simple que les deux précédentes. Par contre, le temps de calcul des algorithmes hybrides reste plus élevé que celui des algorithmes

classiques, car l'intervalle de variation des variables d'optimisation de la fonction F3 est aussi large.

Finalement, on peut dire que les résultats obtenus confirment clairement le théorème d'Hornik, qui stipule qu'un réseau de neurone avec une couche cachée est capable d'approximer n'importe quelle fonction avec une précision arbitraire. On a investi cette capacité pour l'approximation des modèles éléments finis des dispositifs électromagnétiques.

IV.6.2 Algorithmes hybrides méta-heuristiques avec les méthodes exactes : NM-BA, TLBO-NM et TLBO-QN

L'ensemble de ces algorithmes est appliqué aux trois fonctions test choisies. Les résultats obtenus pour chaque fonction sont présentés ci-dessous, où le nombre des itérations est fixé à 60 pour tous les algorithmes.

IV.6.2.1 Nelder Mead-Chaue Souris: NM-BA

Les algorithmes NM-BA et BA sont appliqués aux trois fonctions test considérées.

➤ Fonction F1

Les résultats obtenus sont illustrés dans le tableau IV.4.

Algorithmes	<i>F1</i>	<i>BA</i>	<i>NM-BA</i>
F1	0	1.5710e-08	5.0813e-09
CPU [S]	-	0.2670	1.1316
Ecart-type	-	0.2471	0.0518
Valeur maximale	-	1.1067	0.1605
Valeur moyenne	-	0.0572	0.0567
X1	2.5	2.5000	2.5000
X2	2.5	2.5000	2.5000
Nombre d'appels à F1	-	1120	99

Tableau IV.4 Résultats obtenus avec le NM-BA et le BA, appliqués à F1.

La variation de la fonction F1 avec les itérations des deux algorithmes, NM-BA et BA, est présentée dans la figure IV.13.

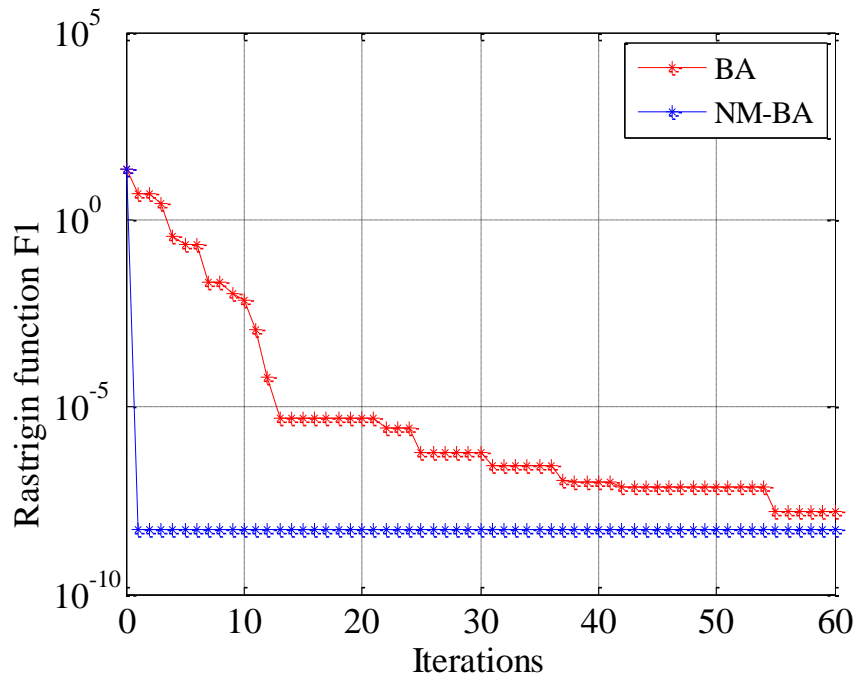


Figure IV. 13 Variation de la fonction F1 avec les itérations des algorithmes NM-BA et BA.

➤ Fonction F2

Les résultats obtenus sont récapitulés dans le tableau IV.5. Tandis que la figure IV.14, présente la variation de la fonction F2 avec les itérations des algorithmes utilisés.

Algorithmes	F2	BA	NM-BA
F2	0	1.2505e-5	5.9530e-11
CPU [S]	-	0.2210	0.6920
Ecart-type	-	3.3970	0.1690
Valeur maximale	-	10.5761	0.5900
Valeur moyenne	-	1.3679	0.1931
X1	0	-0.41952e-6	-0.0805e-10
X2	0	0.1818e-6	-0.1944e-10
Nombre d'appels à F2	-	660	476

Tableau IV.5 Résultats obtenus avec le NM-BA et le BA appliqués à F2.

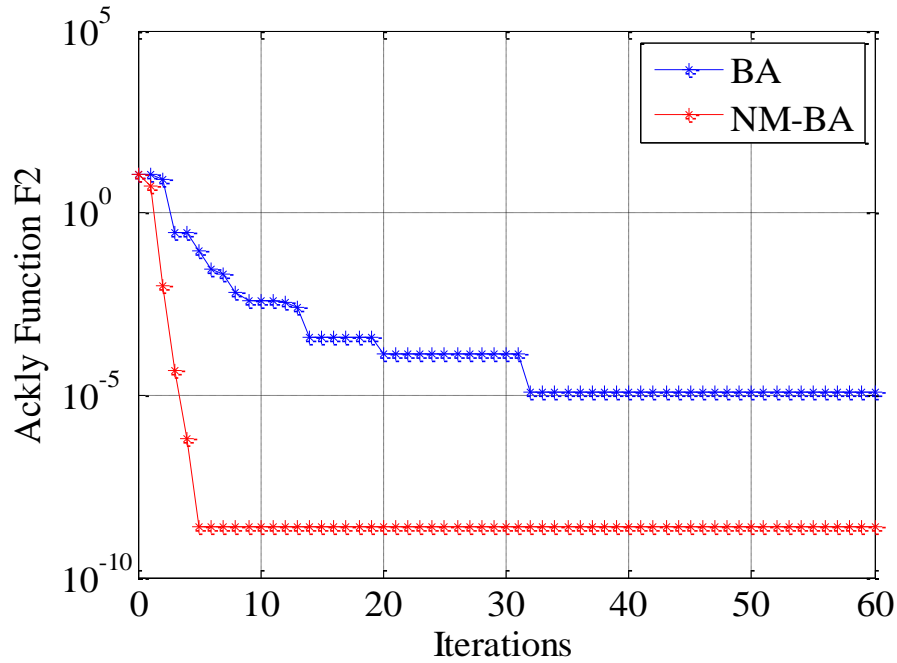


Figure IV. 14 Variation de la fonction F2 avec les itérations des algorithmes NM-BA et BA.

➤ Fonction F3

L'algorithme hybride NM-BA ainsi que celui classique, BA, sont appliqués à la fonction F3, les résultats obtenus sont résumés dans le tableau IV.6. La figure IV.15 illustre la variation de F3 avec les itérations des deux algorithmes.

Algorithmes	<i>F3</i>	<i>BA</i>	<i>NM-BA</i>
F3	-5.2327	-5.2327	-5.2327
CPU [S]	-	0.2961	0.9487
Ecart-type	-	0.0112	0.0081
Valeur maximale	-	-5.1827	-5.2033
Valeur moyenne	-	-5.2296	-5.2268
X1	-4.4537	-4.4537	-4.4538
X2	-4.4537	-4.4537	-4.4537
Nombre d'appels à F3	-	1060	156

Tableau IV.6 Résultats obtenus avec NM-BA BA appliqués à F3.

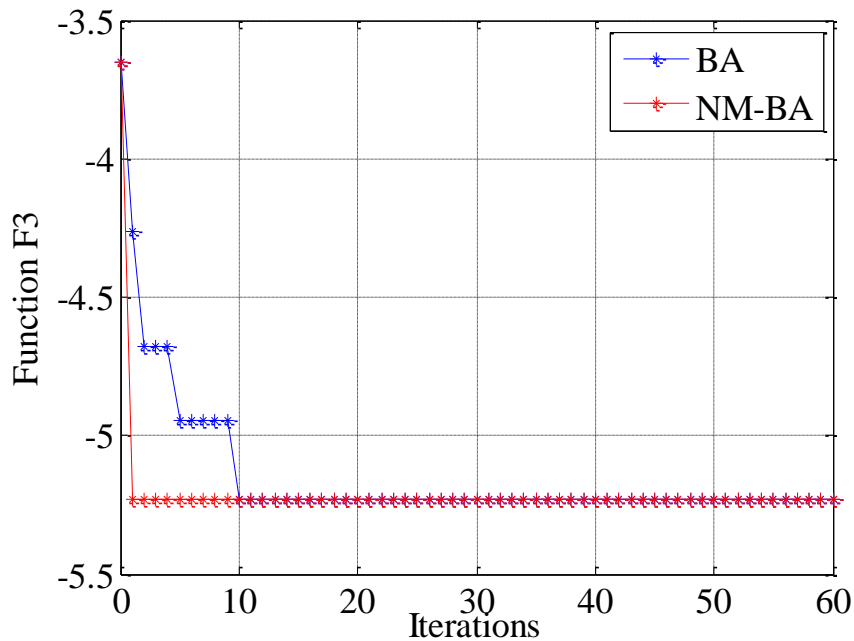


Figure IV. 15 Variation de la fonction F3 avec les itérations des algorithmes NM-BA et BA.

Les valeurs optimales trouvées par les deux algorithmes BA et NM-BA sont presque identiques. Cela signifie que les deux algorithmes ont la capacité de localiser la bonne solution. L'algorithme hybride NM-BA a permis l'obtention de la meilleure valeur optimale avec, un nombre d'appels à la fonction objectif réduit, un écart type et valeur moyenne inférieurs à ceux obtenus avec le BA, où ce dernier nécessite un grand nombre d'évaluations de la fonction objectif pour atteindre la valeur optimale.

IV.6.2.2 Algorithmes hybrides TLBO-NM et TLBO-QN

Les deux versions hybrides de l'algorithme TLBO, TLBO-NM et TLBO-QN, sont appliquées pour la résolution des trois fonctions test considérées. Afin d'évaluer l'efficacité de ces hybridations leurs résultats sont comparés avec ceux de TLBO classique.

➤ *Fonction F1*

Le tableau IV.7 présente l'ensemble des résultats obtenus. La variation de la fonction objectif avec les itérations des algorithmes utilisés est montrée dans la figure IV.16.

Algorithmes	TLBO	TLBO-NM	TLBO-QN
F1	2.2346e-12	1.0765 e-12	7.1054e-15
CPU [S]	0.2594	0.9880	4.4221
Ecart-type	2.2388e-4	0.47958	8.7219e-5
Valeur maximale	0.0010	2.1447	3.7674e-4
Valeur moyenne	0.5007e-4	0.1072	2.5116e-5
X1	2.5000	2.5000	2.5000
X2	2.5000	2.5000	2.5000
Nombre d'appels à F1	2220	5431	607

Tableau IV.7 Résultats obtenus avec les algorithmes TLBO, TLBO-NM et TLBO-QN appliqués à la fonction F1.

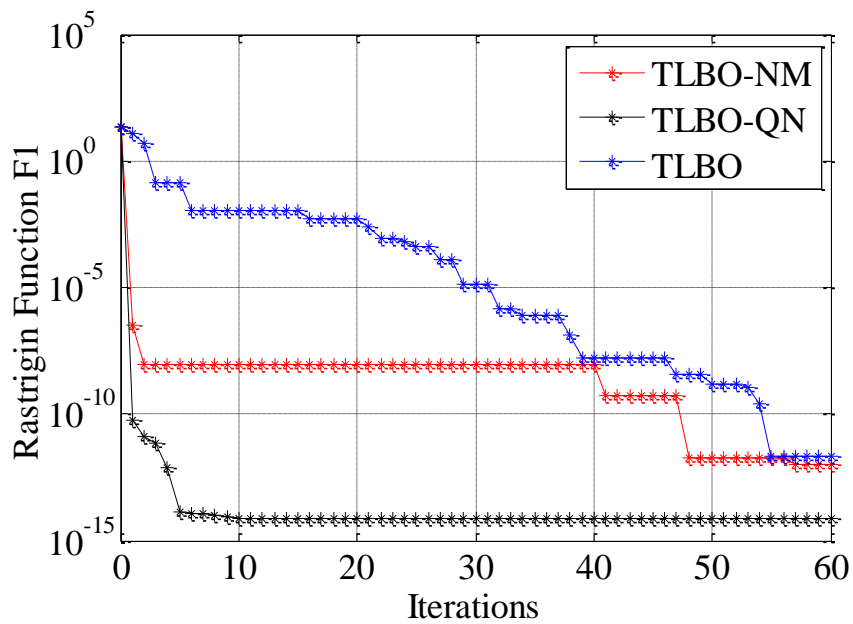


Figure IV. 16 Variation de la fonction F1 avec les itérations des algorithmes TLBO-NM, TLBO-QN et TLBO.

➤ Fonction F2

Les résultats obtenus sont regroupés dans le tableau IV.8. La figure IV.17 présente l'évaluation de la fonction F2 avec les itérations des algorithmes utilisés.

Algorithmes	<i>TLBO</i>	<i>TLBO-NM</i>	<i>TLBO-QN</i>
F2	1.7451e-10	4.3007e-11	0
CPU [S]	0.0660	0.4860	2.3817
Ecart-type	1.5220e-9	7.6911e-10	3.1920e-10
Valeur maximale	6.1162e-9	1.9529e-10	1.3375e-8
Valeur moyenne	1.2634e-9	1.6456e-9	4.6721e-10
X1	0.61500e-10	0.1162e-10	0.1360e-15
X2	0.0497e-10	0.0979e-10	-0.0585e-15
Nombre d'Appels à F2	2420	2766	2642

Tableau IV.8 Résultats obtenus avec les algorithmes TLBO, TLBO-NM et TLBO-QN appliqués à F2.

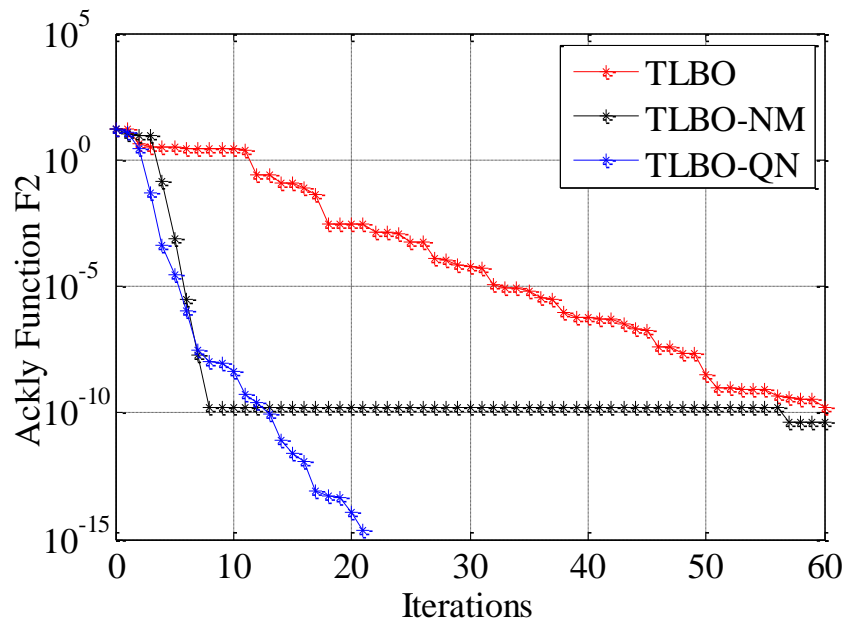


Figure IV.17 Variation de la fonction F2 avec les itérations des algorithmes TLBO-NM, TLBO-QN et TLBO.

➤ Fonction F3

Les résultats obtenus sont résumés dans le tableau IV.9. La figure IV.18 présente l'évaluation de la fonction F3 avec les itérations des algorithmes appliqués.

Algorithmes	TLBO	TLBO-NM	TLBO-QN
F3	-5.2327	-5.2327	-5.2327
CPU [S]	0.1237	1.0008	4.7070
Ecart-type	1.8338e-15	1.3879e-13	1.1881e-15
Valeur maximale	-5.2327	-5.2327	-5.2327
Valeur moyenne	-5.2327	-5.2327	-5.2327
X1	-4.4537	-4.4537	-4.4537
X2	-4.4537	-4.4537	-4.4537
Nombre d'appels à F3	2180	4761	2579

Tableau IV.9 Résultats obtenus avec les algorithmes TLBO, TLBO-NM et TLBO-QN appliqués à F3.

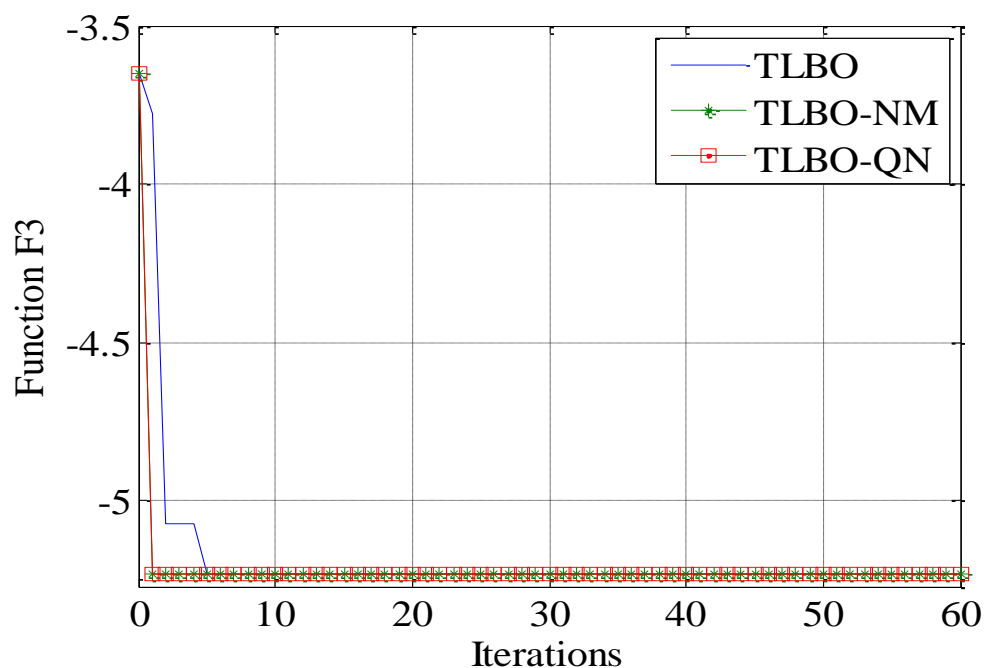


Figure IV. 18 Variation de la fonction F3 avec les itérations des algorithmes utilisés.

Les deux versions hybrides ainsi que celle classique de l'algorithme TLBO, ont pu localiser la solution optimale. Les deux algorithmes hybrides, TLBO-NM et TLBO-QN, ont fourni la meilleure valeur optimale de la fonction objectif avec un temps de calcul légèrement plus important que celui de TLBO classique. Cela est dû au nombre d'appels à la fonction objectif pour atteindre cette valeur, où le TLBO-NM trouve beaucoup de difficultés pour s'échapper de la solution locale. Par contre, le handicap majeur de l'algorithme TLBO-QN étant le calcul du gradient et du hessien de la fonction objectif.

IV.6.3 Algorithmes C-BA et TLBO-RLS

IV.6.3.1 Algorithme C-BA

La validité des trois types de croisement appliqués à l'algorithme BA est évaluée à travers leur application aux fonctions test considérées. Les résultats obtenus avec le C-BA sont comparés à ceux de BA classique.

➤ **Fonction F1**

Les résultats obtenus sont regroupés dans le tableau IV.10. La figure IV.19 donne la variation de la fonction objectif avec les itérations des algorithmes utilisés.

Algorithmes	BA	C-BA Type 1	C-BA Type 2	C-BA Type 3
F1	1.5710e-08	7.0709e-10	0	0
CPU [s]	0.2670	0.2059	0.3073	0.4143
Ecart-type	0.2471	0.0033	0	0
Valeur maximale	1.1067	0.0147	0	0
Valeur moyenne	0.0572	0.0008	0	0
X1	2.5000	2.5000	2.5000	2.5000
X2	2.5000	2.5000	2.5000	2.5000
Nombre d'appels à F1	1120	1476	514	644

Tableau IV.10 Résultats obtenus avec les algorithmes BA et C-BA de types 1, 2 et 3 appliqués à F1.

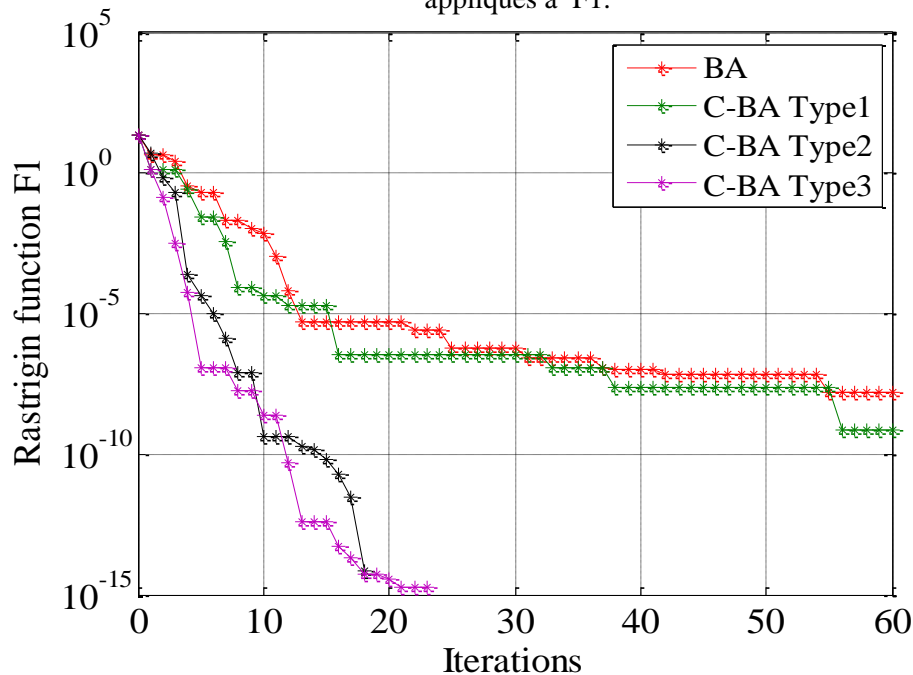


Figure IV. 19 Variation de la fonction F1 avec les itérations des algorithmes utilisés.

➤ **Fonction F2**

Les résultats d'optimisation ainsi que la variation de la fonction objectif avec les itérations des algorithmes utilisés sont présentés dans le tableau IV.11 et la figure IV.20, respectivement.

Algorithmes	BA	C-BA Type 1	C-BA Type 2	C-BA Type3
F2	1.2505e-5	1.2414e-5	0	0
CPU [S]	0.2210	0.3266	0.3000	0.3487
Ecart-type	3.3970	0.2164	0	0
Valeur maximale	10.5761	0.9472	0	0
Valeur moyenne	1.3679	0.0756	0	0
X1	-0.41952e-6	3.9143e-6	-0.1632e-15	-0.1901e-15
X2	0.1818e-6	-1.9852e-6	0.2171e-15	0.0144e-15
Nombre d'appels à F2	660	1514	1164	1034

Tableau IV.11 Résultats obtenus avec les algorithmes BA et C-BA de types 1, 2 et 3 appliqués à F2.

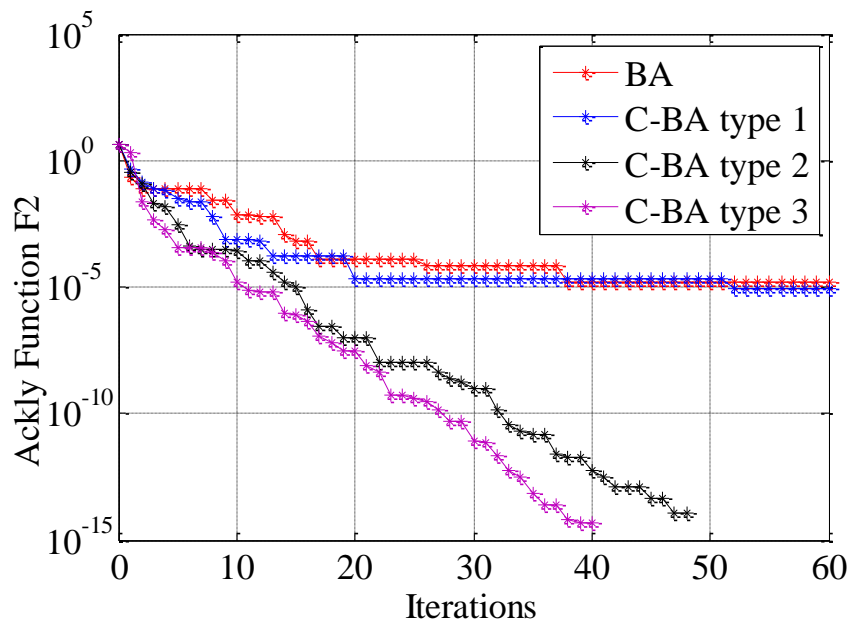


Figure IV. 20 Variation de la fonction F2 avec les itérations des algorithmes utilisés.

➤ **Fonction F3**

Les résultats issus de l'application des algorithmes BA et ses trois versions hybrides, C-BA type 1, C-BA type 2 et C-BA type 3 pour la résolution de la fonction F3 sont

présentés dans le tableau IV.12. L'évolution de la fonction F3 avec les itérations des algorithmes utilisés est présentée dans la figure IV.21.

Algorithmes	BA	C-BA Type1	C-BA Type2	C-BA Type3
F3	-5.2327	-5.2327	-5.2327	-5.2327
CPU [S]	0.2961	0.2626	0.4280	0.3485
Ecart-type	0.0112	0.0023	0.0177	0.0034
Valeur maximale	-5.1827	-5.2226	-5.1560	-5.2177
Valeur moyenne	-5.2296	-5.2319	-5.2265	-5.2312
X1	-4.4537	-4.4539	-4.4537	-4.4537
X2	-4.45379	-4.4539	-4.4537	-4.4537
Nombre d'appels à F3	1060	922	1274	1170

Tableau IV.12 Résultats obtenus avec les algorithmes BA et C-BA de types 1, 2 et 3 appliqués à F3.

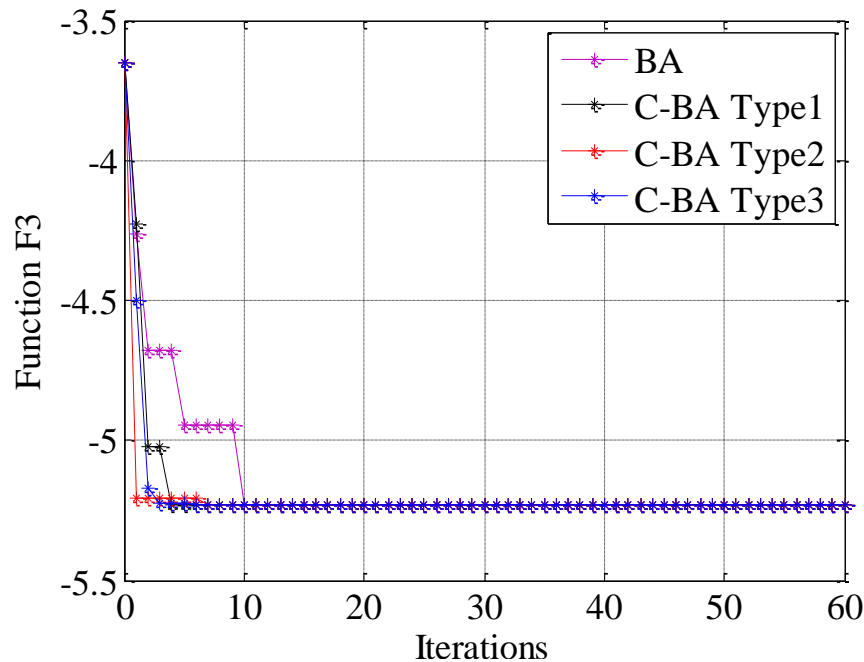


Figure IV. 21 Variation de la fonction F3 avec les itérations des algorithmes utilisés.

L'objectif fondamental de cette application est de tester la validité de la technique de croisement et démontrer son efficacité à améliorer les performances de l'algorithme BA. En outre, et sur la base des résultats obtenus on peut déterminer le meilleur type de croisement parmi les trois types développés.

Nous remarquons que les trois versions de l'algorithme BA avec croisement, C-BA types 1, 2 et 3 ainsi que la version classique de l'algorithme, sont capables de localiser ou de s'approcher de la solution optimale. Sauf que, les algorithmes C-BA types 1, 2 et 3 ont donné la meilleure valeur optimale de la fonction objectif par rapport à celle obtenue avec l'algorithme original BA. En plus, leur convergence est plus rapide que celle de l'algorithme BA. Les trois versions de l'algorithme C-BA nécessitent généralement un nombre d'appels à la fonction objectif réduit pour atteindre la solution optimale avec un écart type et une valeur moyenne de la fonction objectif meilleurs que ceux de l'algorithme original BA.

On constate également que l'algorithme hybride C-BA type 3 possède des caractéristiques plus supérieures que celles des deux algorithmes hybrides C-BA types 1 et 2, où il converge plus rapidement vers la meilleure solution optimale.

Malgré que l'algorithme C-BA type 1 est meilleur que le BA original, il reste moins attractif que les deux autres types, 2 et 3, surtout lorsque les valeurs des deux premières meilleures solutions sont choisies très proches l'une de l'autre, après leur croisement, il en résulte une solution très proche de la meilleure solution courante, voir donc une convergence vers une solution locale.

Les deux algorithmes C-BA type 2 et C-BA type 3 ont presque les mêmes caractéristiques, avec une légère supériorité de C-BA type3 où, la nouvelle population assure une diversité de solutions, à son tour, elle assure à l'algorithme de ne pas être piégé dans une solution locale.

Tenant compte de ces constatations, seulement l'algorithme C-BA type3 est appliqué pour la résolution des problèmes d'optimisation des dispositifs électrotechniques considérés dans ce travail de thèse.

IV.6.3.2 Algorithme TLBO-RLS

Les résultats issus de l'application de l'algorithme TLBO-RLS aux trois fonctions test considérées sont regroupés dans les tableaux IV.13, IV.14 et IV.15 pour les fonctions F1, F2 et F3, respectivement. Ils sont comparés avec ceux issus de l'application des algorithmes TLBO, TLBO-NM et TLBO-QN.

Les figures IV.22, IV.23 et IV.24 montrent la variation des fonctions F1, F2 et F3 avec les itérations des algorithmes TLBO, TLBO-RLS, TLBO-NM et TLBO-QN.

➤ **Fonction F1**

Algorithmes	<i>TLBO</i>	<i>TLBO-RLS</i>	<i>TLBO-NM</i>	<i>TLBO-QN</i>
F1	2.2346e-12	0	1.0765 e-12	7.1054e-15
CPU [S]	0.2594	0.18740	0.9880	4.4221
Ecart-type	2.2388e-4	0	0.47958	8.7219e-5
Valeur maximale	0.0010	0	2.1447	3.7674e-4
Valeur moyenne	0.5007e-4	0	0.1072	2.5116e-5
X1	2.5000	2.5000	2.5000	2.5000
X2	2.5000	2.5000	2.5000	2.5000
Nombre d'appels à F1	2220	740	5431	607

Tableau IV.13 Résultats obtenus avec l’algorithme TLBO-RLS comparés à ceux des autres versions de TLBO, appliqués à F1.

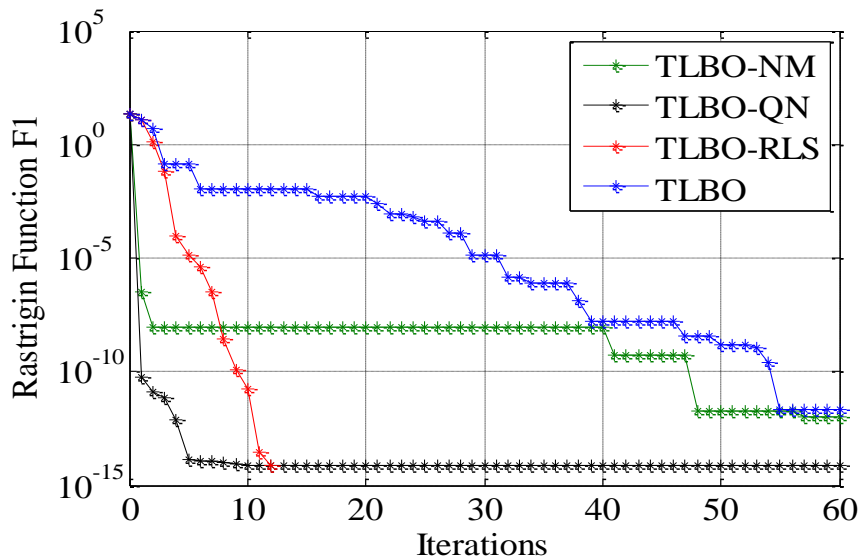


Figure IV. 22 Variation de la fonction F1 avec les itérations des algorithmes TLBO-RLS, TLBO, TLBO-NM et TLBO-QN.

➤ **Fonction F2**

Algorithmes	<i>TLBO</i>	<i>TLBO-RLS</i>	<i>TLBO-NM</i>	<i>TLBO-QN</i>
F2	1.7451e-10	0	4.3007e-11	0
CPU[S]	0.0660	0.1051	0.4860	2.3817
Ecart-type	1.5220e-9	7.7396e-9	7.6911e-10	3.1920e-10
Valeur maximale	6.1162e-9	2.7555e-8	1.9529e-10	1.3375e-8
Valeur moyenne	1.2634e-9	7.3119e-9	1.6456e-9	4.6721e-10
X1	0.6150e-10	0.0843e-15	0.1162e-10	0.1360e-15
X2	0.0497e-10	0.2516e-15	0.0979e-10	-0.0585e-15
Nombre d'appels à F2	2420	1880	2766	2642

Tableau IV.14 Résultats obtenus avec l'algorithme TLBO-RLS comparés à ceux des autres versions de TLBO, appliqués à F2.

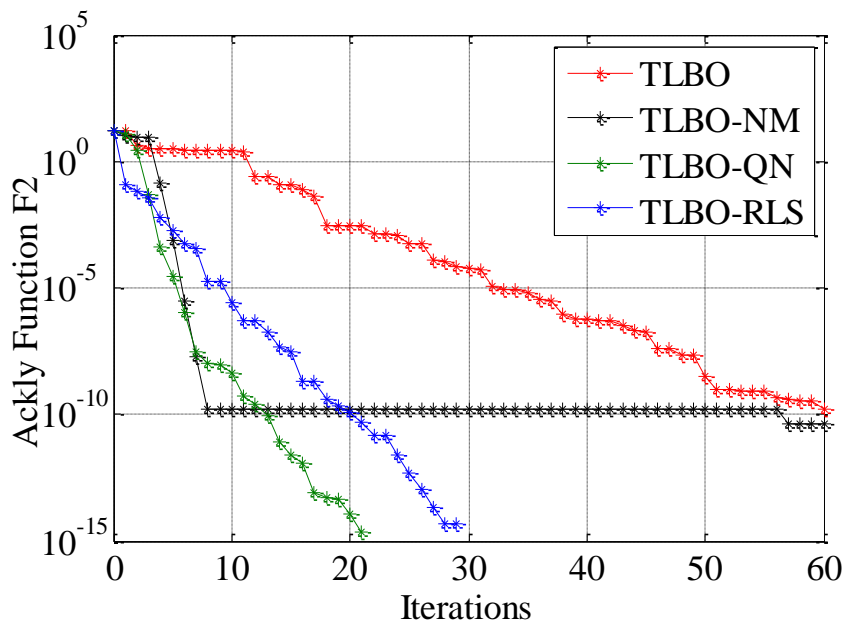


Figure IV. 23 Variation de la fonction F2 avec les itérations des algorithmes utilisés.

➤ **Fonction F3**

Algorithmes	<i>TLBO</i>	<i>TLBO-RLS</i>	<i>TLBO-NM</i>	<i>TLBO-QN</i>
F 3	-5.2327	-5.2327	-5.2327	-5.2327
CPU [S]	0.1237	0.1942	1.0008	4.7070
Ecart-type	1.8338e-15	1.6807e-14	1.3879e-13	1.1881e-15
Valeur maximale	-5.2327	-5.2327	-5.2327	-5.2327
Valeur minimale	-5.2327	-5.2327	-5.2327	-5.2327
X1	-4.4537	-4.4537	-4.4537	-4.4537
X2	-4.4537	-4.4537	-4.4537	-4.4537
Nombre d'appels à F3	2180	3620	4761	2579

Tableau IV.15 Résultats des algorithmes TLBO-RLS, TLBO-NM, TLBO-QN et TLBO appliqués à F3.

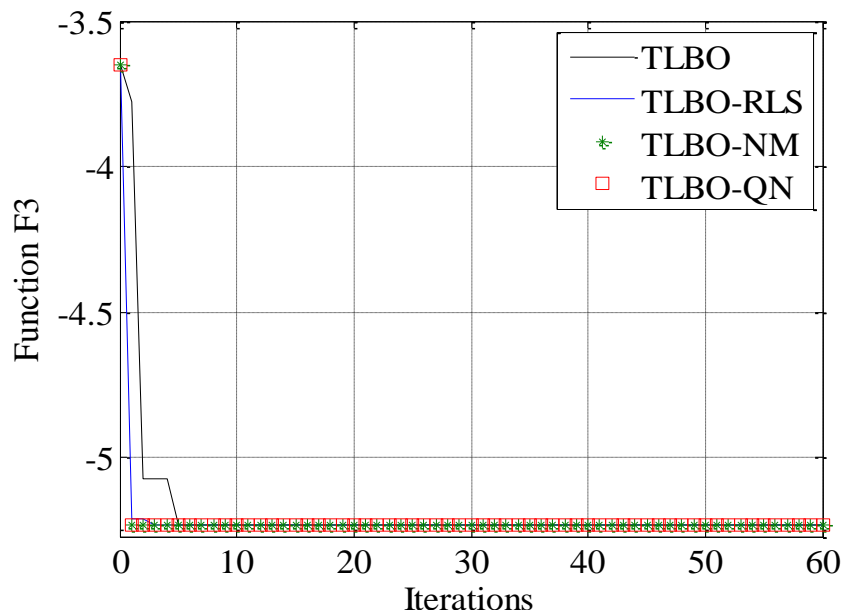


Figure IV. 24 Variation de la fonction F3 avec les itérations des algorithmes utilisés.

On remarque bien que l'algorithme TLBO-RLS ainsi que les deux autres versions hybrides de l'algorithme TLBO développées au cours de ce travail, le TLBO-NM et le TLBO-QN, ont une grande capacité pour localiser ou de se rapprocher de la solution exacte.

Le nouvel algorithme TLBO-RLS a donné la meilleure valeur de la fonction objectif avec une convergence plus rapide et un nombre d'appels à la fonction objectif plus réduit que celui que nécessite l'algorithme TLBO original.

Les deux algorithmes hybrides, TLBO-QN et TLBO-RLS, ont donné des résultats très proches. Sauf que, le TLBO-QN nécessite un temps de calcul plus important que le TLBO-RLS. Cela est dû en réalité au nombre très important d'appels à la fonction objectif pour évaluer le gradient et le hessien.

IV.7 Conclusion

Dans le but d'améliorer les stratégies d'optimisation basées sur les algorithmes d'Intelligence d'Essaim, différentes techniques d'hybridation ont été appliquées aux versions originales de ces algorithmes, d'où l'émergence de nouvelles méthodes ou algorithmes hybrides.

Ce chapitre a fait l'objet de l'implémentation de ces nouvelles méthodes dont la validation est testée à travers leur application à la résolution de trois fonctions Benchmark connues.

Le prochain chapitre est consacré à la présentation des résultats de l'application des différentes stratégies d'optimisation adoptées dans ce travail, classiques et améliorées, aux problèmes de conception optimale dont les cahiers des charges ont été présentés dans le chapitre 3.

***Chapitre V : Résultats
d'applications des techniques
d'Intelligence Artificielle
pour la conception optimale
des machines électriques***

V.1 Introduction

Dans le présent chapitre, les résultats issus d'applications des différentes stratégies d'optimisation basées sur l'intelligence artificielle, versions classiques ainsi que celles améliorées, développées au cours de ce travail de thèse, pour la résolution de trois problèmes de conception optimale des dispositifs électrotechniques, sont présentés et discutés.

Le premier problème est une presse à électro-aimant, considéré comme un problème test international pour valider les méthodes d'optimisation, il est connu sous le nom de TEAM workshop problème 25 ou TEAM 25 tout court. Le deuxième est une machine synchrone à reluctance variable avec barrières du flux. Tandis que le troisième concerne une machine synchrone à double excitation.

Pour la deuxième application, l'objectif consiste à trouver la structure optimale du rotor de la machine pour maximiser le couple électromagnétique. Dans la troisième application, un problème multi-objectif est posé, où, l'objectif revient en réalité à établir un certain compromis entre le couple électromagnétique et le volume de l'aimant, un couple électromagnétique maximal avec un volume de l'aimant minimal.

Au début, Nous commençons par la présentation des résultats du modèle approximé, issu du couplage entre les réseaux de neurones et les deux algorithmes d'Intelligence d'Essaim, le TLBO et le PSO, tous deux ainsi que leurs versions originales, sont appliqués au TEAM 25 et la MSRVM avec barrières du flux. Ensuite, les résultats d'applications des algorithmes PSO, TLBO, CS, BA, NM-BA, TLBO-NM, TLBO-RLS, TLBO-QN, NM-BA et C-BA, aux dispositifs considérés sont présentés et discutés.

V.2 Modèle hybride réseau de neurones avec les méta-heuristiques : NN-PSO et NN-TLBO

Avec ce modèle, un couplage entre les éléments finis et le réseau de neurones type MLP a été effectué. Il sert à construire une base de données correspondante au dispositif étudié.

V.2.1 TEAM workshop problème 25

La base de données, construite, permet de simuler les réponses magnétiques, dans ce c'est la densité du flux, pour chaque variable de conception ou d'optimisation. On a effectué 2100 exemples où la densité du flux dans la cavité est calculée pour chaque exemple.

Nous avons utilisé l'architecture d'un réseau multicouche (MLP) de type: 4-n-22, comprenant quatre entrées (R1, L1, L2 et L4), une couche cachée avec n égale à 34 neurones et une sortie de 22 neurones.

Les algorithmes " PSO " et " TLBO " et les algorithmes hybrides " NN-PSO " et " NN-TLBO " sont appliqués pour résoudre le problème d'optimisation TEAM 25. Ces algorithmes Approchés permettent une exploration efficace du domaine de recherche, ce qui leur attribue une grande opportunité d'atteindre l'optimum global.

Pour tous les algorithmes utilisés, la taille de la population initiale est fixée à 20, le nombre maximal d'itérations est pris égal à 60. Les résultats sont résumés et comparés à ceux obtenus avec l'algorithme génétique GA, dans les tableaux V.1 et V.2.

L'évolution de la fonction objectif avec les itérations des algorithmes PSO et TLBO est montrée dans les figure V.1, et celle des algorithmes NN-PSO et NN-TLBO est montrée dans la figure V.2. Les paramètres optimaux obtenus avec ces algorithmes sont donnés dans le tableau V.2.

Algorithme	Fonction Objectif	CPU
PSO	1.8669e-05	7.6779 [H]
TLBO	1.8085e-05	5.6388 [H]
GA	1.4557e-4	1.8016 [H]
NN-PSO	1.9821e-05	40.4569 [S]
NN-TLBO	1.9820e-05	79.314737 [S]

Tableau V.1 Résultats d'optimisation obtenus avec les algorithmes PSO, TLBO, GA, NN-PSO et NN-TLBO appliqués au TEAM 25.

Paramètre Méthode	R1[mm]	L1[mm]	L2[mm]	L4[mm]
PSO	7.2498	13.9829	14.0016	13.6999
TLBO	7.2641	14.0879	14.0720	13.5296
GA	7.5846	16.3310	15.4924	10.3522
NN-PSO	7.2340	13.9907	14.0636	13.5886
NN-TLBO	7.2347	13.9929	14.0645	13.5860

Tableau V.2 Valeurs des paramètres optimaux de TEAM 25.

Comme on peut le constater à partir des résultats obtenus, les algorithmes TLBO et PSO donnent de meilleurs résultats que le GA. Cependant, leur inconvénient majeur réside dans le temps de calcul qui est nettement supérieur à celui du GA.

On remarque aussi que, les résultats des algorithmes hybrides "NN-TLBO" et "NN-PSO" sont très proches de ceux obtenus avec les algorithmes PSO et TLBO. Cela, confirme que le réseau de neurones MLP a permis une bonne approximation du modèle éléments finis du problème TEAM 25. Concernant le temps de calcul, les algorithmes hybrides "NN-PSO" et "NN-TLBO" sont beaucoup plus rapides que ceux classiques, le PSO et le TLBO. Les deux algorithmes d'Intelligence d'Essaim couplés avec les réseaux de neurones artificiels fournissent un outil d'optimisation très puissant avec un coût de résolution modeste.

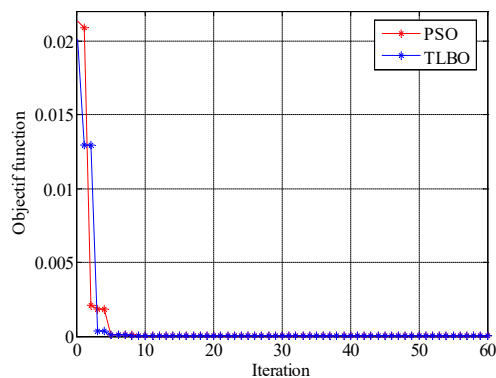


Figure V.1 Variation de la fonction objectif en fonction des itérations des algorithmes PSO et TLBO appliqués au TEAM 25.

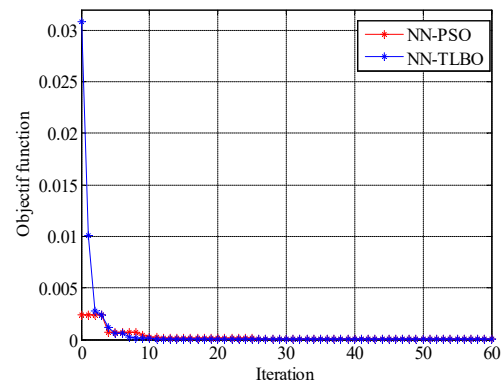


Figure V.2. Variation de la fonction objectif en fonction des itérations des algorithmes NN-PSO et NN-TLBO appliqués au TEAM 25.

V.2.2 Machine synchrone à reluctance variable avec barrières du flux

La MSRVS est modélisée avec la méthode du réseau de neurones (MLP), qui utilise une base de données de 2000 exemples, permettant de simuler la réponse du couple électromagnétique dans chaque cas.

Nous avons utilisé l'architecture d'un réseau multicouche 2-57-7 comprenant 2 entrées (les deux paramètres d'optimisation β_r et e_b), une couche cachée de 57 neurones et une sortie de 7 neurones.

Pour tous les algorithmes, la taille de la population initiale est choisie égale à 20, le nombre maximal d'itérations est fixé à 50. Dans cette application, uniquement la première formulation du problème d'optimisation f_1 , est considérée.

Les résultats d'optimisation obtenus avec les algorithmes PSO et TLBO sont comparés avec ceux obtenus avec le GA; tous sont présentés dans le tableau V.3. La figure V.3 montre les évolutions de la fonction objectif avec les itérations des algorithmes utilisés.

Résultats d'optimisation		Valeur initiale	PSO	TLBO	GA
fonction Objectif $f1$		4.1576	1.3495	1.3506	1.3500
Paramètres	$\beta_r/2$	14.80°	16.5389°	16.4300°	16.4303°
	e_b [mm]	1.2857	0.8952	0.7006	0.7158
Valeur moyenne quadratique du couple [N.m]		0.0639	0.4012	0.4060	0.4064
Valeur Maximale du couple [N.m]		0.9493	1.8076	1.8198	1.8135
CPU [H]		-	42.013	89.785	59.3688

Tableau V.3 Résultats d'optimisation obtenus avec les algorithmes PSO, TLBO et GA, appliqués à la MSRV.

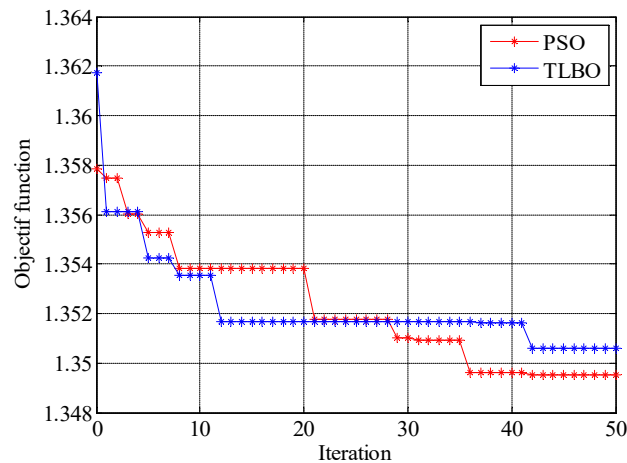


Figure V.3 Variation de la fonction objectif avec les itérations des algorithmes PSO et TLBO appliqués à la MSRV.

Les figures V.4 et V.5 montrent une comparaison entre l'évolution du couple électromagnétique en fonction de l'angle de rotation pour les valeurs initiales et optimales obtenues avec les deux algorithmes stochastiques : PSO et TLBO.

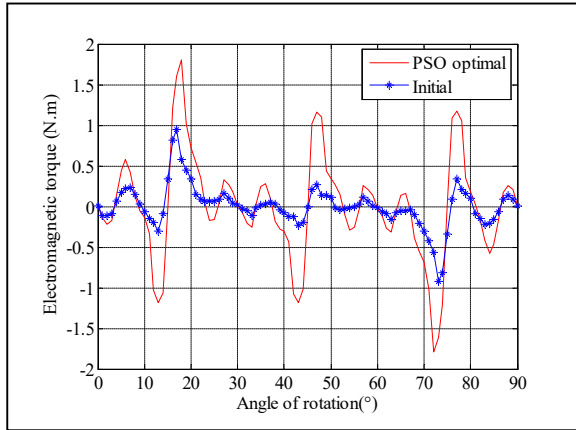


Figure V.4. Variation du couple électromagnétique en fonction de l'angle de rotation obtenu par le PSO.

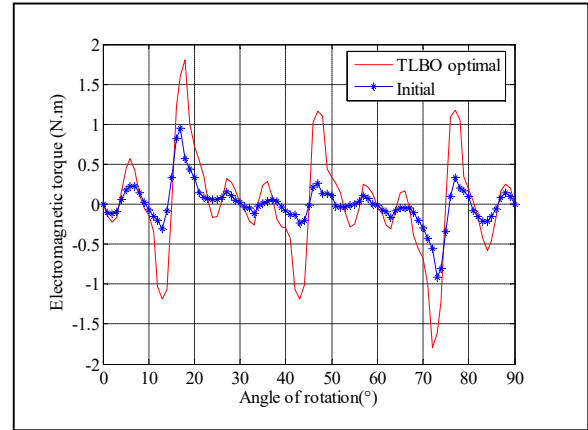


Figure V.5. Variation du couple électromagnétique en fonction de l'angle de rotation obtenu par le TLBO.

Les paramètres optimaux obtenus avec les algorithmes NN-PSO et NN-TLBO sont présentés dans le tableau V.5. Par contre, la figure V.6 montre l'évolution de la fonction objectif avec les itérations des deux algorithmes.

		<i>Valeur initiale</i>		
			<i>NN-PSO</i>	<i>NN-TLBO</i>
Fonction Objectif $f1$		-	1.3491	1.3491
paramètres	$\beta_r / 2 [^\circ]$	14.80	16.5459	16.5459
	$e_b [mm]$	1.2857	1.0965	1.0965
Valeur moyenne quadrature du couple [N.m]		0.0639	0.4040	0.4040
Valeur maximale du couple [N.m]		0.949	1.8032	1.8032
CPU [S]		-	13.0617	26.1145

Tableau V.5 Valeurs optimales des paramètres de conception de la MSRV obtenus avec les algorithmes NN-PSO et NN-TLBO.

Les figures V.7 et V.8 montrent une comparaison entre les évolutions du couple électromagnétique en fonction de l'angle de rotation pour les valeurs initiales et optimales obtenues avec les algorithmes PSO, NN-PSO et TLBO, NN-TLBO, respectivement.

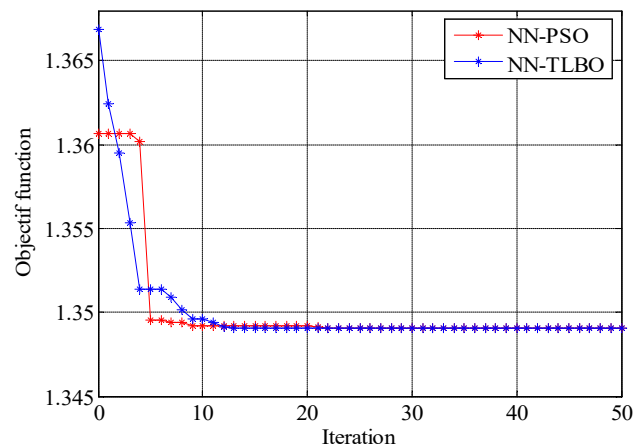


Figure V.6 Variation de la fonction objectif avec les itérations des algorithmes NN-PSO et NN-TLBO appliqués à la MRV.

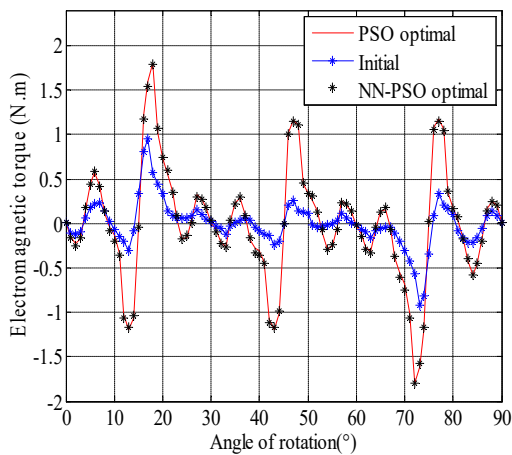


Figure V.7 Variation du couple électromagnétique en fonction de l'angle de rotation pour les paramètres initiaux et ceux optimaux obtenus avec le PSO et le NN-PSO.

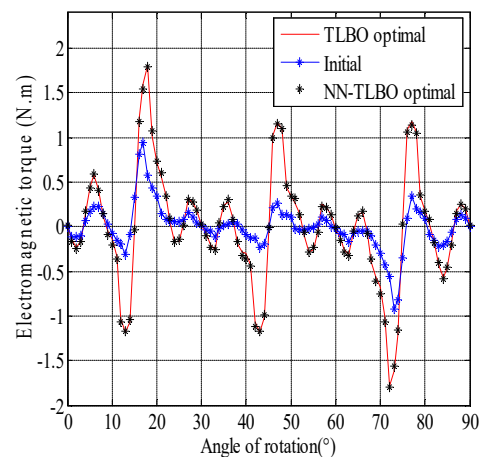


Figure V.8 Variation du couple électromagnétique en fonction de l'angle de rotation pour les paramètres initiaux et ceux optimaux obtenus avec le TLBO et le NN-TLBO.

A partir des résultats obtenus, une augmentation significative de la valeur maximale du couple électromagnétique pour les configurations optimales, issues de tous les algorithmes appliqués, est notée par rapport à celle obtenue dans le cas initial. Cela justifie l'intérêt de choisir la forme du rotor de la MSR_V pour maximiser la valeur du couple électromagnétique. La figure V.9 présente la densité du flux et les lignes équipotentielles du potentiel vecteur magnétique dans la configuration optimale de la MSR_V.

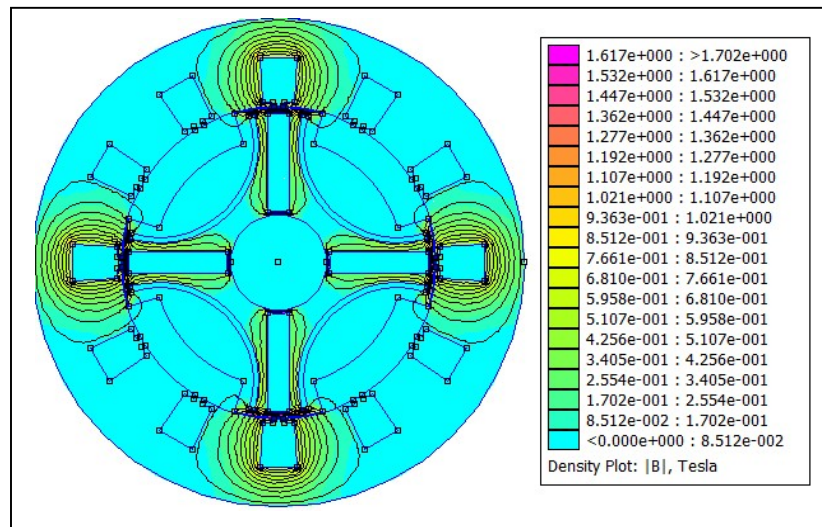


Figure V.9 Densité du flux et lignes équipotentielles dans la configuration optimale.

Dans cette partie du travail, nous avons démontré l'efficacité des algorithmes PSO et TLBO pour la conception optimale des machines électriques. Où, il a été observé à travers leur application aux deux problèmes électromagnétiques, TEAM problème 25 et un moteur synchrone à réluctance variable avec barrières du flux, qu'ils sont aussi efficaces que d'autres algorithmes souvent utilisés pour résoudre ce genre de problèmes, tels que l'algorithme génétique.

Le processus d'optimisation repose sur l'utilisation conjointe des algorithmes stochastiques et la résolution numérique par éléments finis du problème électromagnétique. Cela s'est traduit par un temps de calcul très important, principalement en raison du nombre élevé d'itérations nécessaire pour atteindre la solution optimale et de multiples appels au calcul par éléments finis pour évaluer la fonction objectif.

Pour remédier à l'inconvénient du coût de calcul, nous avons couplé ces algorithmes d'Intelligence d'Essaim avec la méthode des réseaux de neurones artificiels. La technique d'hybridation a permis d'obtenir des résultats d'une qualité similaire à celle du cas sans hybridation, mais avec un temps de calcul nettement très court.

L'utilisation des algorithmes PSO et TLBO avec le réseau de neurones pour résoudre des problèmes d'optimisation relatifs aux machines électriques est très prometteuse en raison de leur temps de simulation très court et de la bonne qualité des résultats obtenus.

V.3 Résultats des algorithmes hybrides TLBO-NM, TLBO-RLS, TLBO-QN, NM-BA et C-BA

Ces nouvelles approches sont appliquées pour la résolution des problèmes de la conception optimale relatifs aux trois dispositifs électrotechniques, dont les cahiers des charges ont été présentés dans le chapitre 3 de la thèse. Les résultats obtenus avec ces algorithmes hybrides sont comparés avec ceux issus de l'application des algorithmes méta-heuristiques originaux, TLBO, PSO, BA et CS.

V.3.1 TEAM workshop problème 25

Les résultats d'applications des algorithmes considérés sont donnés dans les tableaux V.7 et V.8. L'ensemble des résultats obtenus est comparé avec ceux obtenus par [CREV-10], où les algorithmes 1-LGA et 2-LGA (one and two Level Genetic Algorithm) ont été utilisés. Pour ce faire, on a considéré la même taille de la population, égale à 10, et le même critère d'arrêt, qui revient à fixer la valeur de la fonction objectif à une valeur égale à 0.001. Les valeurs optimales des fonctions objectifs obtenues par [CREV-10] sont : 7.56 pour l'algorithme 1-LGA et 6.54 pour l'algorithme 2-LGA.

Algorithmes	Fonction objectif OF	CPU [Min]	Ecart-type	Valeur maximale	Valeur Moyenne	Nombre d'appels à OF
PSO	9.874	918.44	5.4905e-5	0.0011	0.0010	390
CS	7.486	199.29	0.0055	0.0166	0.0067	610
BA	7.768	445.15	9.8108e-4	0.0037	0.0018	490
NM-BA	6.172	23.02	0.0618	0.1827	0.0535	121
C-BA	5.722	198.23	0.00149	0.00435	0.0021	1080
TLBO	6.183	234.84	6.1873e-4	0.0024	0.0016	580
TLBO-QN	4.545	108.04	0.0207	0.0647	0.0287	211
TLBO-NM	8.329	30.63	0.03561	0.1133	0.0585	112
TLBO-RLS	6.034	130.59	0.0114	0.0338	0.0169	95
1-LGA [CREV-10]	7.56	/	/	/	/	/
2-LGA [CREV-10]	6.54	/	/	/	/	/

Tableau V.7 Résultats d'optimisation pour le TEAM 25

Algorithmes	<i>R1</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>
PSO	6.943	13.211	14.000	14.117
CS	7.103	13.518	14.000	13.940
BA	6.689	13.306	14.00	14.085
NM-BA	6.878	13.648	14.000	14.286
C-BA	6.9436	13.701	14.000	13.522
TLBO	6.946	13.69	14.000	13.606
TLBO-QN	6.993	13.607	14.000	13.574
TLBO-NM	6.888	13.629	14.000	13.634
TLBO-RLS	6.946	13.690	14.000	13.606
1-LGA [CREV-10]	6.80	13.6	14.3	13.5
2-LGA [CREV-10]	6.82	13.59	14.3	13.5

Tableau V.8 Valeurs optimales des paramètres de conception de TEAM 25.

La figure V.10 nous renseigne sur l'évolution de la fonction objectif avec les itérations des algorithmes utilisés.

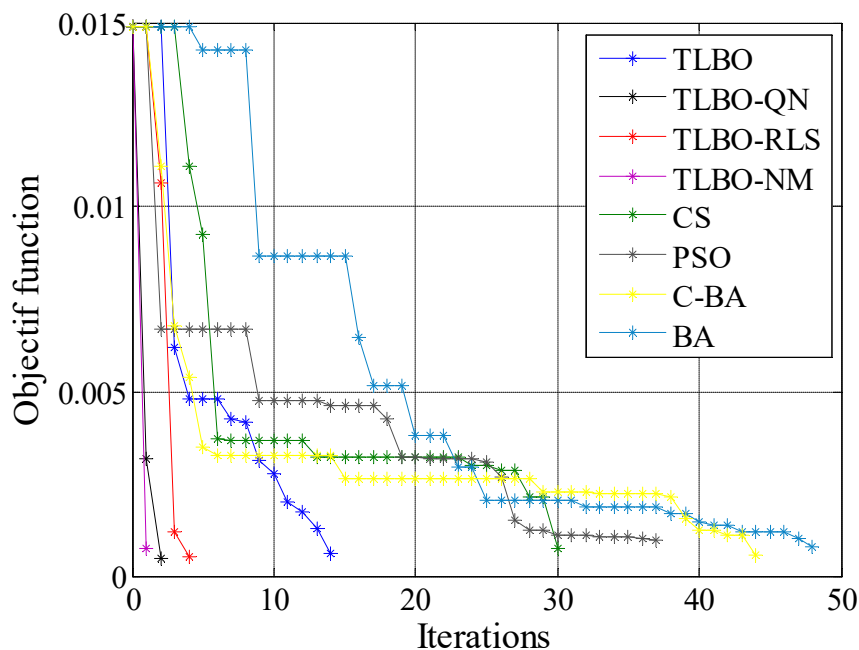


Figure V.10 Evolution de la fonction objectif avec les itérations des algorithmes appliqués au TEAM 25.

D'après ces résultats, nous observons que les méthodes hybrides ont permis d'obtenir des solutions optimales avec un nombre réduit d'appels à la fonction objectif, impliquant ainsi un temps de calcul aussi réduit par rapport à celui des algorithmes originaux, BA, TLBO, PSO et CS.

L'algorithme hybride TLBO-QN a donné la meilleure solution optimale, mais avec un nombre d'appels à la fonction objectif élevé par rapport à ceux que nécessitent les algorithmes

hybrides : TLBO-NM, NM-BA et TLBO-RLS. Après, c'est l'algorithme C-BA qui a fourni le deuxième meilleur résultat avec un nombre d'appels au calcul de la fonction objectif plus important que celui de l'algorithme original BA. Malgré ça, le temps de calcul de l'algorithme C-BA est inférieur à celui de BA. En troisième position vient le TLBO-RLS, avec un nombre d'appels à la fonction objectif et un temps de calcul moins importants que ceux de TLBO classique.

Les algorithmes hybrides, TLBO-QN, TLBO-RLS, NM-BA et C-BA, tous ont donné des meilleurs résultats par rapport à leurs versions originales et par rapport aux deux algorithmes utilisés dans [CREV-10], 1-LGA et 2-LGA.

Les hybridations développées et appliquées aux versions originales de ces algorithmes ont permis d'améliorer leur mécanisme d'optimisation tout en assurant un meilleur balancement entre l'exploitation (recherche locale) et l'exploration (recherche globale), équilibre souvent difficile à assurer avec les algorithmes classiques.

Ces constats sont confirmés par les résultats de l'analyse statistique de chaque algorithme utilisé, voir : l'écart type, la valeur maximale et la valeur moyenne de toutes les solutions de la dernière population. Sachant qu'un algorithme ayant les valeurs moyenne et maximale de la fonction objectif de la population finale proches de la valeur optimale avec un écart type minimal, signifie que toutes les solutions de la population, convergent vers la solution optimale.

Le cas où l'écart type est minimal mais la solution optimale est mauvaise, exprime que toutes les solutions de la population divergent vers une solution locale, c'est le cas du PSO par exemple.

La figure V.11 présente les lignes équipotentielles du potentiel vecteur magnétique et la densité du flux dans la configuration optimale de TEAM 25 obtenue avec le TLBO-RLS.

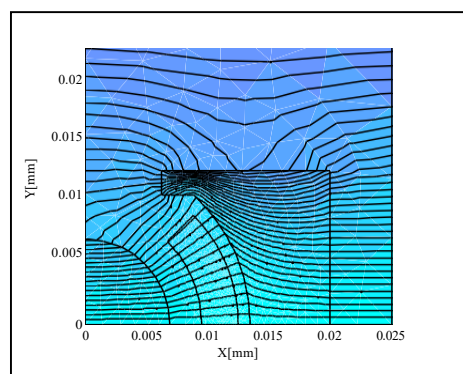


Figure V.11 Configuration optimale de TEAM 25

V.3.1.1 Etude de la robustesse des algorithmes TLBO-RLS, TLBO et NM-BA

La robustesse d'un algorithme d'optimisation est évaluée à travers la considération de plusieurs contextes initiaux et le calcul de l'écart type correspondant. Dans ce travail on s'est limité à l'étude des algorithmes TLBO, TLBO-RLS et NM-BA. Pour tenir compte des caractéristiques probabilistes et stochastiques des algorithmes considérés, plusieurs essais ont été effectués, entre 4 et 5, pour la même population initiale où le meilleur résultat a été maintenu.

➤ Robustesse des algorithmes TLBO et TLBO-RLS

Le tableau V.9 présente les fonctions objectifs obtenues avec les algorithmes TLBO et TLBO-RLS, correspondantes à 5 populations initiales différentes.

		<i>Premier test</i>	<i>Deuxième test</i>	<i>Troisième test</i>	<i>Quatrième test</i>	<i>Cinquième test</i>
TLBO	Fonction objectif OF	6.940e-3	2.681e-3	9.725e-4	6.1873e-4	2.681e-3
	N° d'appels à OF	2100	2100	150	580	2100
TLBO-RLS	Fonction objectif OF	8.788e-4	7.680e-4	5.432e-4	6.034e-4	7.2593 e-4
	N° d'appels à OF	127	99	183	95	155

Tableau V.9 Résultats obtenus avec les algorithmes TLBO et TLBO-RLS pour différents contextes initiaux.

Les valeurs de l'écart type des deux algorithmes, TLBO et TLBO-RLS, sont présentées dans le tableau V.10; la valeur la plus minimale correspond à la méthode la plus robuste. Sachant qu'on a utilisé l'expression suivante pour le calcul de l'écart type :

$$Ecart - type = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - \bar{f})^2} \quad (V.1)$$

n : Nombre des tests effectués ;

f_i La meilleure valeur de la fonction objectif pour le test correspondant, i ;

\bar{f} la moyenne des meilleures valeurs des fonctions objectifs, $\bar{f} = \frac{1}{n} \sum_{i=1}^n f_i$.

Méthodes	Ecart type
TLBO	29.012e-4
TLBO-RLS	1.532e-4

Tableau V.10 Valeurs de l'écart type des algorithmes TLBO et TLBO-RLS.

A partir de ces résultats, il est clair que l'algorithme TLBO-RLS est plus robuste que l'algorithme TLBO classique.

Les valeurs optimales des paramètres de conception obtenues pour chaque essai initial sont présentées dans les tableaux V.11 et V.12, avec le TLBO et le TLBO-RLS, respectivement.

	Premier test	Deuxième test	Troisième test	Quatrième test	cinquième test
R1[mm]	7.363	7.262	6.722	6.946	7.262
L2[mm]	15.001	14.256	13.477	13.690	14.256
L3[mm]	43.574	16.500	14.000	14.000	16.500
L4[mm]	11.718	12.868	13.271	13.606	12.868

Tableau V.11 Paramètres optimaux de Die Molds obtenus avec le TLBO.

	Premier test	Deuxième test	Troisième test	Quatrième test	cinquième test
R1[mm]	7.017	7.058	7.068	7.093	6.918
L2[mm]	13.813	13.739	13.724	13.734	13.692
L3[mm]	14.211	14.065	14.060	14.000	14.063
L4[mm]	13.323	13.368	13.587	13.635	13.550

Tableau V.11 Paramètres optimaux de Die Molds obtenus avec le TLBO-RLS.

➤ Robustesse de l'algorithme NM-BA

Les résultats de l'étude statistique issus des différents essais initiaux appliqués à l'algorithme NM-BA sont regroupés dans le tableau V.13.

Résultats	Premier test	Deuxième test	Troisième test	Quatrième test
Fonction objectif OF[e-4]	6.172	4.39916	7.1573	4.5851
CPU [Min]	23.021	16.197	8.977	31.384
écart type	0.0618	0.0987	0.0477	0.0385
Valeur maximale	0.1827	0.2739	0.1596	0.1439
Valeur moyenne	0.0535	0.1078	0.0536	0.0457
N° d'appels à OF	121	65	62	123

Tableau V.13 Résultat de l'étude statistique de l'algorithme NM-BA.

On remarque bien que la solution change avec le changement des paramètres initiaux, cela reflète parfaitement la caractéristique des algorithmes déterministes, dont le NM simplexe est l'un d'entre eux, qui stipule que la solution trouvée dépend étroitement du contexte initial choisi.

V.3.2 Machine à reluctance variable avec barrières du flux

Avant d'entamer la présentation des résultats des différents algorithmes appliqués pour optimiser la forme de la structure rotorique de la MSR.V afin d'avoir un couple électromagnétique maximal, on présente tout d'abord les résultats d'une étude préliminaire qu'on a effectuée pour mieux choisir les positions angulaires du rotor. Dans cette partie et en plus des positions rotoriques choisies ultérieurement (dans la section V.2.2), de nouvelles positions du rotor ont été considérées. Les deux vecteurs de positions rotoriques sont pris en compte avec les deux formulations, les deux fonctions objectifs, du problème posé (voir chapitre 3).

V.3.2.1 Problème 1

La première formulation, f_1 , du problème d'optimisation est résolue avec la considération des nouvelles positions angulaires du rotor. Rappelons ci-dessous la fonction objectif f_1 et les positions considérées.

$$f_1 = \frac{1}{\sqrt{\Gamma/n}}$$

Où :

n est le nombre positions angulaires du rotor, θ , dans lesquelles le couple électromagnétique est calculé. Il est choisi égal à 7.

Les sept positions rotoriques initiales sont: $\theta_{k\text{initial}} = [6^\circ 10^\circ 18^\circ 21^\circ 25^\circ 34^\circ 37^\circ]$;

$$\Gamma = \sum_k ce^2(\theta_k) \quad , \quad k = 1, \dots, n .$$

ce est le couple électromagnétique à chaque position angulaire du rotor [N.m] ;

Γ est la somme des valeurs du couple électromagnétique aux différentes positions angulaires du rotor.

Les nouvelles positions sont $\theta_{k\text{nouvelle}} = [6^\circ 12^\circ 16^\circ 19^\circ 27^\circ 35^\circ 43^\circ]$;

Dans cette étude, seulement les algorithmes PSO et TLBO ont été appliqués. Les résultats obtenus sont présentés dans le tableau V.14.

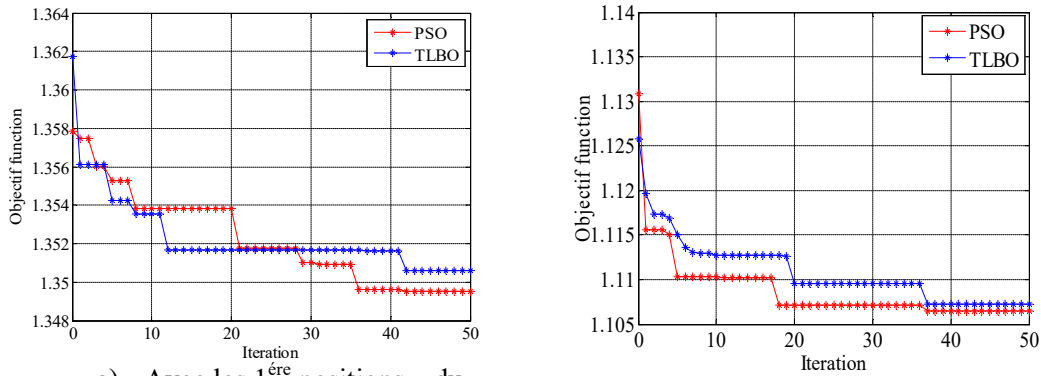
La figure V.12 illustre la différence entre les valeurs de la fonction objectif, f_l , obtenues avec les premières positions, choisies aléatoirement $\theta_{k\text{initial}}$, et les nouvelles positions $\theta_{k\text{nouvelle}}$, choisies après une étude paramétriques qu'on a menée sur la MSR.V considérée.

Les valeurs optimales des paramètres d'optimisation sont données dans le tableau V.15.

On observe bien que les paramètres de conception optimaux de la MSR.V, obtenus avec les algorithmes PSO et TLBO, ont donné une valeur moyenne du couple dans le cas des nouvelles positions rotoriques plus grande que celle obtenue dans le cas des premières positions, choisies aléatoirement. Où, elle est autour de 0.41[N.m] pour le premier cas contre 0.40 [N.m] dans le deuxième cas .

Résultats d'optimisation	Valeur initiale	PSO avec les 1 ^{ère} positions (V.2.2)	PSO avec les nouvelles positions	TLBO avec les 1 ^{ère} positions (V.2.2)	TLBO avec les nouvelles positions
Fonction objectif f_l	4.1576	1.3495	1.1065	1.3506	1.1073
Valeur quadratique moyenne du couple [N.m]	0.0639	0.4012	0.4104	0.4060	0.4101
Valeur Maximale du couple [N.m]	0.9493	1.8076	1.7857	1.8198	1.8017
CPU [H]	/	42.013	49.365	89.785	82.1914

Tableau V.14 Résultats d'optimisation de la MSR.V.



a) Avec les 1^{ère} positions du rotor, choisies arbitrairement (voir la section.2.2).

(b) Avec les nouvelles positions du rotor.

Figure V.12. Evolution de la fonction objectif f_1 avec les itérations des algorithmes PSO et TLBO pour les deux positions angulaires du rotor.

Résultats d'optimisation		Valeur initiale	PSO 1 ^{ère} positions (V.2.2)	PSO nouvelles positions	TLBO 1 ^{ère} positions (V.2.2)	TLBO nouvelles positions
Paramètres	$\beta_r/2$ [°]	14.80°	16.5389	16.4523	16.4300	16.4523
	e_b [mm]	1.2857	0.8952	1.35	0.7006	1.3498

Tableau V.15 Paramètres optimaux de la MRV pour les deux positions angulaires du rotor. Cas de la première formulation du problème d'optimisation f_1 .

Pour ce type de machines, la MSR_V, le couple est fortement ondulé, chose qui perturbe le mécanisme de l'algorithme d'optimisation pour atteindre la solution optimale globale, où le choix des meilleures positions angulaires du rotor contribue significativement à éviter la convergence prématurée de l'algorithme d'optimisation et de s'échapper du piège d'une solution locale. Ces constats sont nettement visibles dans la figure V.12, là où les premières positions aléatoires du rotor sont considérées, figure V.12.a, les variations de la fonction objectif, f_1 , sont très petites et la convergence de l'algorithme est très lente par rapport au deuxième cas, figure V.12.b.

V.3.2.2 Problème 2

Dans cette partie est à la différence avec la précédente, c'est la deuxième formulation du problème d'optimisation f_2 qui est considérée, seulement pour le deuxième vecteur des positions angulaires du rotor $\theta_{nouvelle}$.

Rappelons que la deuxième fonction objectif proposée est donnée par :

$$f_2 = \sum_k (7 \times ce_{k_{nit}}^2 - ce_{k_{cal}}^2) \quad k = 1, \dots, 7$$

avec :

$ce_{k_{nit}}$ est la valeur du couple calculé au point k correspondant à la position rotorique $\theta_{k_{nouvelle}}$, pour les paramètres géométriques initiaux du rotor.

$ce_{k_{cal}}$ est la valeur du couple calculé au point k correspondant à la position rotorique $\theta_{k_{nouvelle}}$, pour les paramètres géométriques du rotor obtenus à chaque itération de l'algorithme d'optimisation appliqué.

Les résultats obtenus sont présentés dans les tableaux V.16 et V.17.

Résultats d'optimisation	Valeur initiale	PSO	TLBO
Fonction objectif f_2	0.3469	0.2120	0.2131
Valeur quadratique moyenne du couple [N.m]	0.0639	0.4104	0.4101
Valeur maximale du couple [N.m]	0.9493	1.7857	1.8017
CPU [H]	-	46.8857	81.4019

Tableau V.16 Résultats d'optimisation du problème 2.

Résultats d'optimisation	valeur initiale	PSO	TLBO	
Parameters	$\beta_r/2$ [°]	14.80	16.4524	16.4523
	e_b [mm]	1.2857	1.35	1.3498

Tableau V.17. Paramètres optimaux pour le problème 2.

Pour les deux problèmes posés, les algorithmes d'optimisation utilisés, ont abouti à un résultat identique (formulations f_1 et f_2 pour les nouvelles positions). On remarque que pour les deux problèmes la valeur optimale de la variable e_b coïncide avec la limite supérieure de la contrainte du domaine imposée sur sa valeur. On peut dire alors que le problème est mal posé. Sur la base des résultats obtenus, on a proposé un nouvel intervalle de variation de la variable e_b .

➤ **Nouvel intervalle de variation des paramètres de conception**

Le nouvel intervalle de variation des paramètres de conception proposé est donné dans le tableau ci-dessous.

Paramètre	Valeur minimale	Valeur maximale
e_b	0.45 [mm]	1.8804 [mm]
$\beta_r/2$	14.625°	16.875°

Tableau V.18 Nouvelles limites de variation des paramètres de conception de la MSRV.

➤ **Résultats d'optimisation pour le nouvel intervalle**

On présente dans le tableau V.19 une comparaison entre les résultats d'optimisation obtenus avec la considération du nouvel intervalle de variation des deux paramètres de conception et ceux de l'intervalle initial.

Résultats d'optimisation	Valeur initiale	PSO	PSO Nouvel intervalle	TLBO	TLBO Nouvel intervalle
Fonction objectif f_2	0.3469	0.2120	0.2110	0.2131	0.2094
Valeur Quadratique moyenne du couple [N.m]	0.0639	0.4104	0.4103	0.4101	0.4103
Valeur Maximale du couple [N.m]	0.9493	1.7857	1.8050	1.8017	1.7992
CPU [H]	/	46.8857	45.38088	81.4019	106.8432

Tableau V.19 Résultats d'optimisation obtenus pour les deux intervalles de variation des paramètres de conception.

Les paramètres de conception optimaux sont donnés dans le tableau ci-dessous.

Résultats d'Optimisation	Valeur initiale	PSO	PSO Nouvel intervalle	TLBO	TLBO Nouvel intervalle
Paramètres	$\beta_r/2[^\circ]$	14.80	16.4524	16.4523	16.4523
	e_b [mm]	1.2857	1.35	1.3853	1.3498

Tableau V.20 Paramètres optimaux.

La figure V.13 (a et b) illustre l'évolution de la fonction objectif avec les itérations des algorithmes utilisés dans les deux cas des intervalles de recherche.

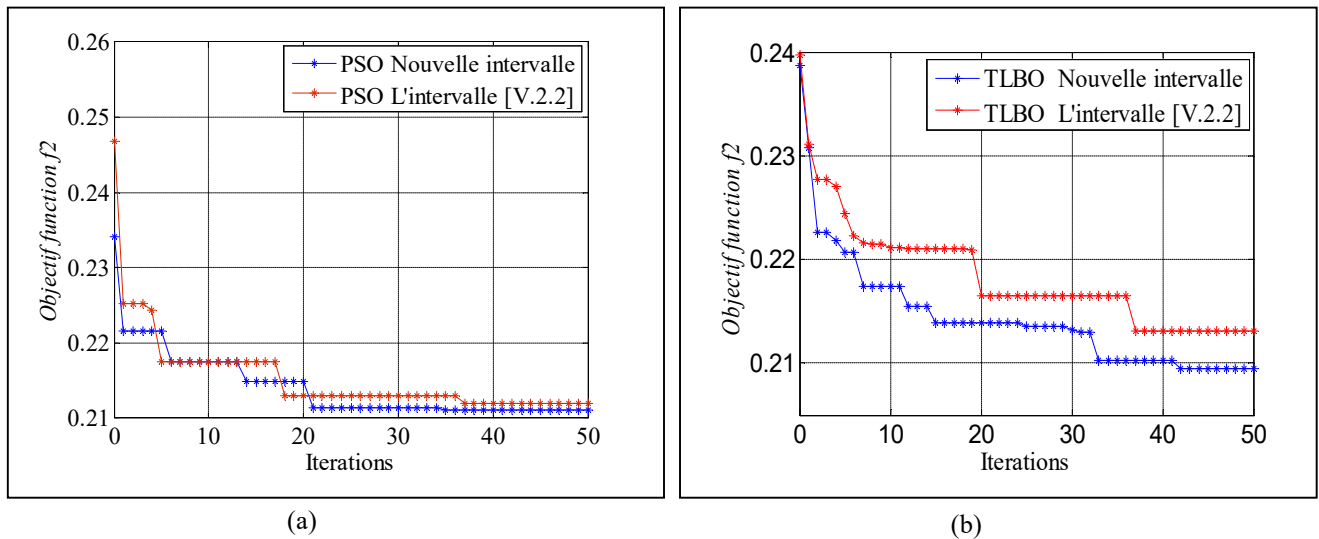


Figure 13. Variation de la fonction objectif f_2 en fonction des itérations des algorithmes PSO (a) et TLBO (b) pour les deux intervalles de recherche.

A partir des résultats obtenus, le nouvel intervalle de variation des paramètres de conception de la MSR_V a été maintenu. Dans ce qui suit les résultats d'applications des algorithmes PSO, CS, BA, NM-BA, C-BA, TLBO, TLBO-RLS et TLBO-NM sont présentés.

V.3.2.3 Problème 2 cas du nouvel intervalle

On a appliqué les algorithmes PSO, CS, BA, NM-BA, C-BA, TLBO, TLBO-RLS, et TLBO-NM, pour l'optimisation de la structure rotorique de la MSR_V avec barrières du flux, sous la considération du cas du problème 2 avec les nouvelles positions rotoriques et le nouvel intervalle de variation des paramètres de conception. Pour tous les algorithmes, la taille de la population initiale est choisie égale à 20, le nombre maximal d'itérations est fixé à 50.

Les résultats obtenus sont donnés dans les tableaux V.22 et V.23

Résultats d'optimisation	Fonction objectif f_2	CPU [H]	Ecart-type	Valeur maximale	Valeur moyenne	N° d'appels à f_2
Valeur initiale	0.3469	-	-	-	-	-
PSO	0.2127	49.3705	0.0037	0.2261	0.2174	940
TLBO	0.21538	86.7557	0.0015	0.2226	0.2192	1620
BA	0.2135	43.8499	0.0065	0.2396	0.22129	1020
C-BA	0.2100	56.5297	5.6953e-17	0.2100	0.2100	1188
NM-BA	0.2138	180.30	0.0045	0.2337	0.2259	1893
TLBO-RLS	0.2149	73.1014	0.0017	0.2204	0.2188	2865
TLBO-NM	0.2092	162.586	0.0034	0.2215	0.2157	1242
CS	0.2211	88.4746	0.0161	0.2962	0.0161	820

Tableau V.22 Résultats d'optimisation.

Résultats d'optimisation	Paramètres		Valeur moyenne du couple [N.m]	Valeur maximale du couple [N.m]
	$\beta_r/2$ [°]	e_b [mm]		
Valeur initiale	14.80	1.2857	0.0639	0.9493
PSO	16.452203	1.4444	0.40996	1.80425
TLBO	16.45198	1.68900	0.40941	1.79921
BA	16.45239	1.6020	0.40945	1.79440
C-BA	16.45236	1.58898	0.40971	1.79337
NM-BA	16.45231	1.82858	0.40855	1.79299
TLBO-RLS	16.4527	1.4939	0.4094	1.7951
TLBO-NM	16.4523	1.6747	0.4106170	1.778566
CS	16.5513	1.43618	0.4072	1.80141

Tableau V.23 Paramètres optimaux et valeurs maximale et moyenne du couple électromagnétique.

La figure V.14 illustre la variation du couple électromagnétique, cas des paramètres initiaux et ceux optimaux, obtenus avec l'algorithme TLBO-NM. Cet algorithme a fourni le meilleur résultat par rapport aux autres algorithmes.

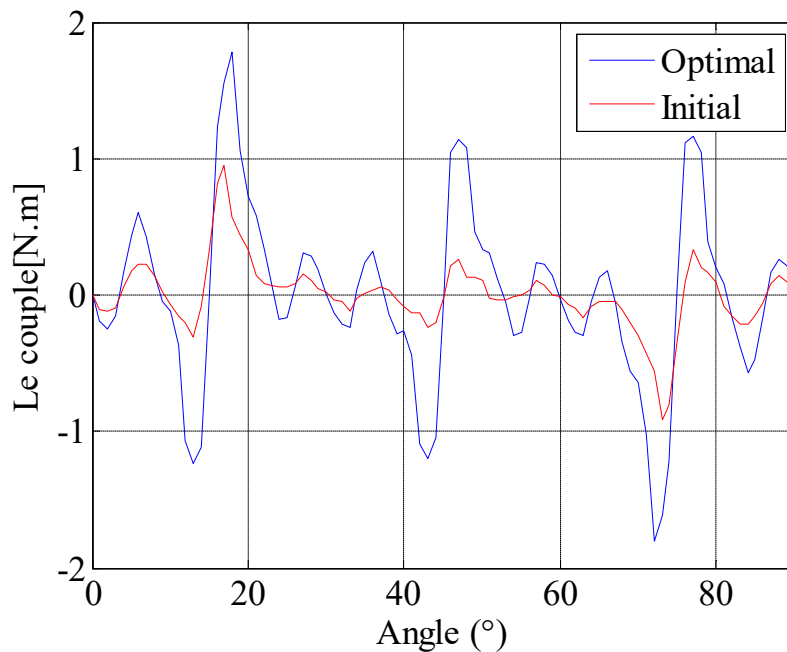


Figure V.14 Couple électromagnétique dans la configuration initiale et celle optimale obtenue avec le TLBO-NM.

La variation de la fonction objectif avec les itérations des algorithmes utilisés est présentée dans la figure V.15.

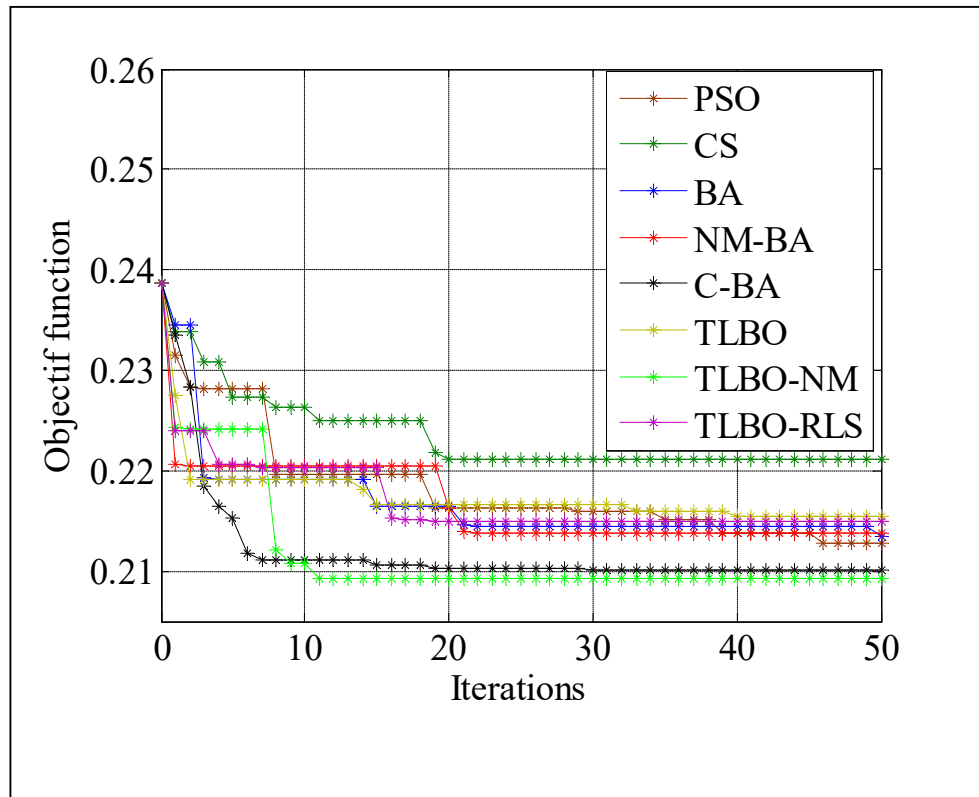


Figure 15. Variation de la fonction objectif avec les itérations des algorithmes d'optimisation, utilisés.

D'après les résultats obtenus, on remarque que les algorithmes TLBO-NM et C-BA ont donnés les meilleurs résultats. Pour le TLBO-NM, 1242 appels à la fonction objectif ont été effectués pour atteindre la solution optimale. Par contre, le TLBO a effectué 1620 appels, malgré ça, le temps de calcul pour le TLBO est moins important que celui du TLBO-NM.

L'algorithme C-BA a permis d'aboutir à des résultats meilleurs à ceux de l'algorithme NM-BA, soit au niveau de la qualité de la solution, soit au niveau du nombre d'appels à la fonction objectif, égale à 1188, avec un temps de calcul aussi réduit. Par comparaison avec l'algorithme BA original, la qualité de la solution fournie par le C-BA est meilleure, cela confirme que le processus d'optimisation avec le C-BA assure un bon équilibre entre les modes de recherche local et global.

L'algorithme NM-BA figure parmi les algorithmes qui ont fourni les meilleurs résultats. La valeur finale de la fonction objectif obtenue avec cet algorithme est très proche de celle obtenue avec l'algorithme BA classique. En contre partie, il consomme un temps de

calcul plus supérieur que celui du BA classique. Cela est dû à l'utilisation de l'algorithme de Nelder Mead, dont le processus est basé sur un mode de recherche local, l'algorithme hybride NM-BA trouve alors beaucoup de difficultés pour sortir de la région de la recherche locale.

Pour l'algorithme TLBO-RLS, la solution optimale est atteinte après 2865 appels à la fonction objectif, elle est meilleure que celle du TLBO original. Comme on peut le voir sur la figure V.15, le TLBO-RLS converge plus rapidement vers la solution optimale ; après il change lentement jusqu'à fin du processus d'optimisation, ce qui explique son nombre élevé d'appels au calcul de la fonction objectif par rapport à l'algorithme TLBO. Cependant, les résultats du TLBO-RLS, restent inférieurs à ceux des autres algorithmes comme le PSO, le TLBO-NM et le BA, sans oublier de rappeler que cet algorithme a donné le meilleur résultat dans les cas des fonctions test et le TEAM 25. Cela revient essentiellement aux paramètres qui caractérisent l'équation de la recherche locale, intégrée dans l'algorithme TLBO original (ε qui dépend de la valeur du pas, l'intensité A et le paramètres α pour diminuer l'intensité A), c'est-à-dire les paramètres qui définissent le pas de la recherche locale. Cet algorithme nécessite un réglage approprié des paramètres de la recherche locale, surtout ε ; lorsque le pas est grand, alors la phase de la recherche locale est inutile. Une autre explication possible de ce résultat, est la divergence de l'algorithme TLBO-RLS vers une solution locale lorsqu'il trouve des difficultés pour sortir de la région de la recherche locale.

Le résultat de l'algorithme PSO appliqué à la MSRV est acceptable. Par contre, l'algorithme CS a fourni une solution de qualité inférieure par rapport aux autres algorithmes.

L'étude statistique des résultats obtenus avec les différents algorithmes appliqués peut donner une idée sur les meilleurs algorithmes ou les algorithmes les plus adaptés au problème traité. Dans le cas de la MSRV, on peut constater que les algorithmes TLBO-NM, C-BA, PSO, NM-BA et TLBO sont les plus adaptés pour cette application.

En réalité, il n'y a aucune loi ou règle générale pour choisir l'algorithme le plus adapté et qui assure le meilleur résultat du problème considéré, mais, tous ça dépend des paramètres de réglage et de contrôle de ces algorithmes qui dépendent étroitement de la nature et des spécificités du problème posé.

La figure V.16 présente les lignes équipotentielles du potentiel vecteur et la densité du flux dans la configuration optimale de la MSRV, obtenue avec l'algorithme C-BA.

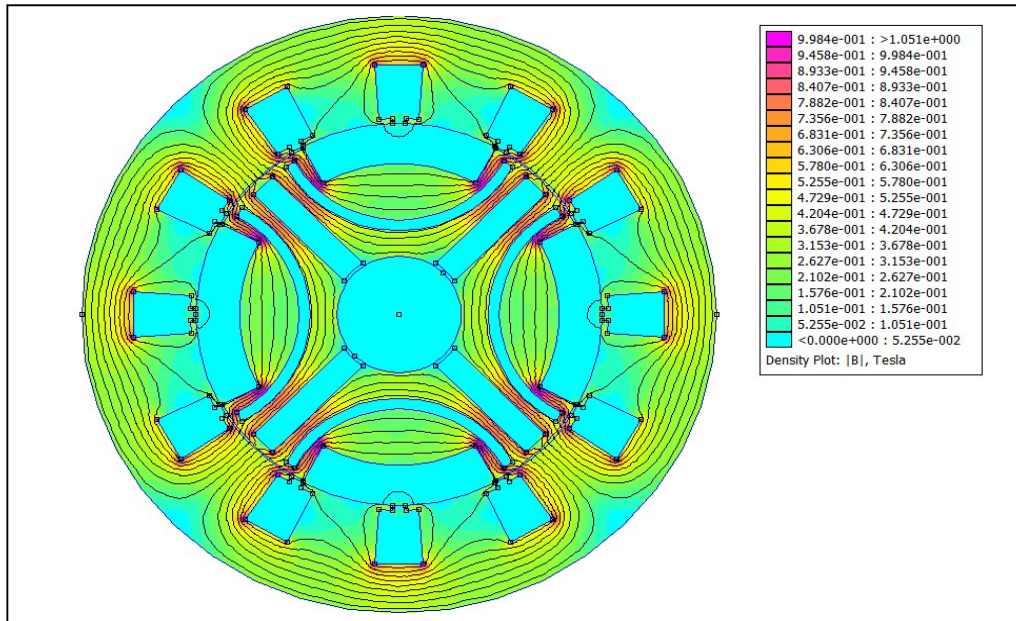


Figure V.16 Densité du flux dans la configuration optimale de la MRV obtenue avec l'algorithme C-BA.

V.3.3 Machine Synchrones à Double Excitation (MSDE)

Le problème traité dans cette application est bi-objectif (donné par l'équation III.42), il consiste à maximiser le couple électromagnétique tout en minimisant le volume de l'aimant au rotor. Le problème a été transformé à un autre mono-objectif où la fonction objectif a été mise sous la forme suivante :

$$F = w_1 \cdot f_1 + w_2 \cdot f_2$$

f_1 étant l'équation du couple (voir l'équation III.39) et f_2 celle du volume de l'aimant (voir l'équation III.40).

La qualité du résultat dépend du choix des deux poids, w_1 et w_2 , et aussi de la normalisation des valeurs des deux fonctions, f_1 et f_2 .

V.3.3.1 Normalisation des fonctions f_1 et f_2

Pour le couple, la fonction f_1 , on divise les valeurs du couple obtenue pour chaque position sur la valeur du couple initial (couple obtenu avec les paramètres de conception initiaux) correspondante à la même position choisie.

Pour le volume de l'aimant, la fonction f_2 , on divise les valeurs du volume, à chaque évaluation ou itération, sur le volume initial de l'aimant (volume obtenu avec les paramètres de conception initiaux) ;

V.3.3.2 Choix des poids w_1 et w_2

La valeur des poids w_1 et w_2 dépend de l'objectif souhaité, si w_1 est plus grand que w_2 , alors on favorise la maximisation du couple au détriment de la minimisation du volume de l'aimant, et contrairement à ça dans le cas inverse. Donc à chaque fois en changeant les poids w_1 et w_2 , la solution optimale est pour le même algorithme change aussi.

Dans cette étude, on a considéré une seule valeur des poids. Où, plusieurs essais pour fixer cette valeur ont été effectués. Pour ce faire, l'algorithme TLBO a été appliqué, il a donné un résultat acceptable avec un nombre réduit d'itérations.

Après plusieurs essais, les valeurs 0.935 et 0.065 pour w_1 et w_2 , respectivement sont maintenues. Pour confirmer ce choix, on a appliqué l'algorithme génétiques GA pour résoudre le problème. La solution optimale obtenue, pour 40 itérations et une taille de la population fixée à 20, est la suivante :

$$F=0.9001, X=[52.0706 \ 0.7010 \ 43.5494]$$

Où

F est la fonction objectif ;

$X(1)$ est le Rayon interne de l'aimant R1;

$X(2)$ est l'épaisseur de l'entre fer e ;

$X(3)$ est l'Ouverture angulaire de l'aimant τ_m .

Le couple moyen et le volume de l'aimant correspondants sont :
646.4554 [N .m] et $1.31702e-4 \text{ m}^3$, respectivement.

La figure V.17 illustre la variation de la fonction objectif avec les itérations de l'algorithme génétique.

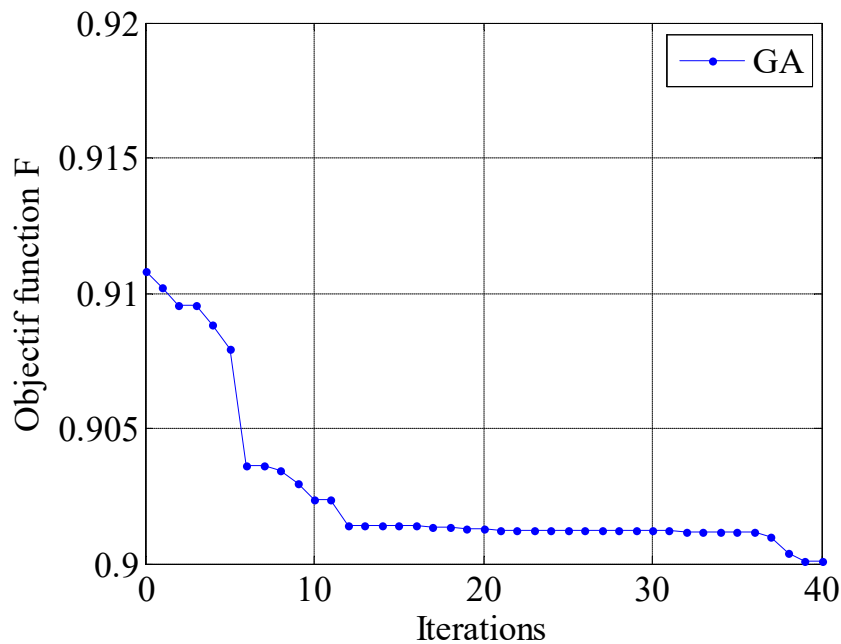


Figure V.17 Variation de la fonction objectif avec les itérations de l'algorithme génétique.

V.3.3.3 Résultats d'optimisation

Les algorithmes PSO, TLBO, BA, C-BA, NM-BA, TLBO-RLS, TLBO-NM et CS, ont été appliqués pour résoudre le problème d'optimisation relatif à la MSDE.

L'analyse de la variation de la fonction objectif, F , avec les itérations de l'algorithme génétique, figure V.17, où le processus d'optimisation change lentement à partir de la douzième itération, nous a permis, sous le souci des temps de calcul prohibitifs, de choisir un nombre maximal d'itérations égal à 30 pour tous les algorithmes utilisés et une taille de la population fixée à 20. La population initiale est identique pour tous les algorithmes.

Les résultats d'optimisation sont présentés dans les tableaux V.24 et V.25. Ils sont comparés avec ceux du GA pour 40 itérations.

Résultats d'optimisation	Fonction objectif	CPU [H]	Ecart-type	Valeur maximale	Valeur moyenne	N° d'appels à F
PSO	0.8914	45.9222	2.0995e-4	0.8920	0.8916	580
TLBO	0.8914	41.9262	1.1570e-4	0.8917	0.8915	1220
BA	0.9064	27.8528	9.9009e-4	0.9099	0.9069	300
C-BA	0.9042	27.9970	3.9862e-4	0.9054	0.9048	837
NM-BA	0.8965	74.3065	8.1178e-5	0.8969	0.8966	1980
TLBO-RLS	0.8913	51.1982	1.6110e-04	0.8919	0.8916	1307
TLBO-NM	0.8961	94.0603	2.0127e-4	0.8968	0.8963	2608
CS	0.8921	62.7407	0.0192	0.9703	0.9069	1180
GA	0.9001	59.5262	9.4920e-4	0.9043	0.9013	800

Tableau V.24 Résultats d'optimisation pour la MSDE.

Résultats d'optimisation	Paramètres			Volume de l'aimant	Valeur moyenne du couple [N.m]	Valeur maximale du couple [N.m]
	R1[mm]	e[mm]	τ_m [°]			
Valeurs initiales	50.0000	1.5000	48	1.79241e-4	595.1507	705.7500
Algorithmes						
PSO	55.0000	0.7000	33.4953	3.96426e-5	605.6935	842.5981
TLBO	55.0000	0.7000	33.4956	3.96429 e-5	605.6904	842.5987
BA	50.6116	0.7000	44.9858	1.75637e-4	656.98359	797.95390
C-BA	51.0857	0.7000	44.7893	1.62190e-04	654.8550	795.35031
NM-BA	52.8794	0.7000	41.6108	1.05141e-04	636.3699	811.3032
TLBO-RLS	55.0000	0.7000	33.4958	3.9643e-5	605.6845	842.5919
TLBO-NM	53.4763	0.7000	38.4318	8.2802e-5	623.1857	807.7890
CS	55.0000	0.7000	33	3.90582e-5	604.97218	840.31555
GA	52.0706	0.7010	43.5494	1.31702e-4	646.455491	793.246632

Tableau V.25 paramètres de conception optimaux de la MSDE.

La figure V.18 présente la variation de la fonction objectif avec les itérations des algorithmes utilisés.

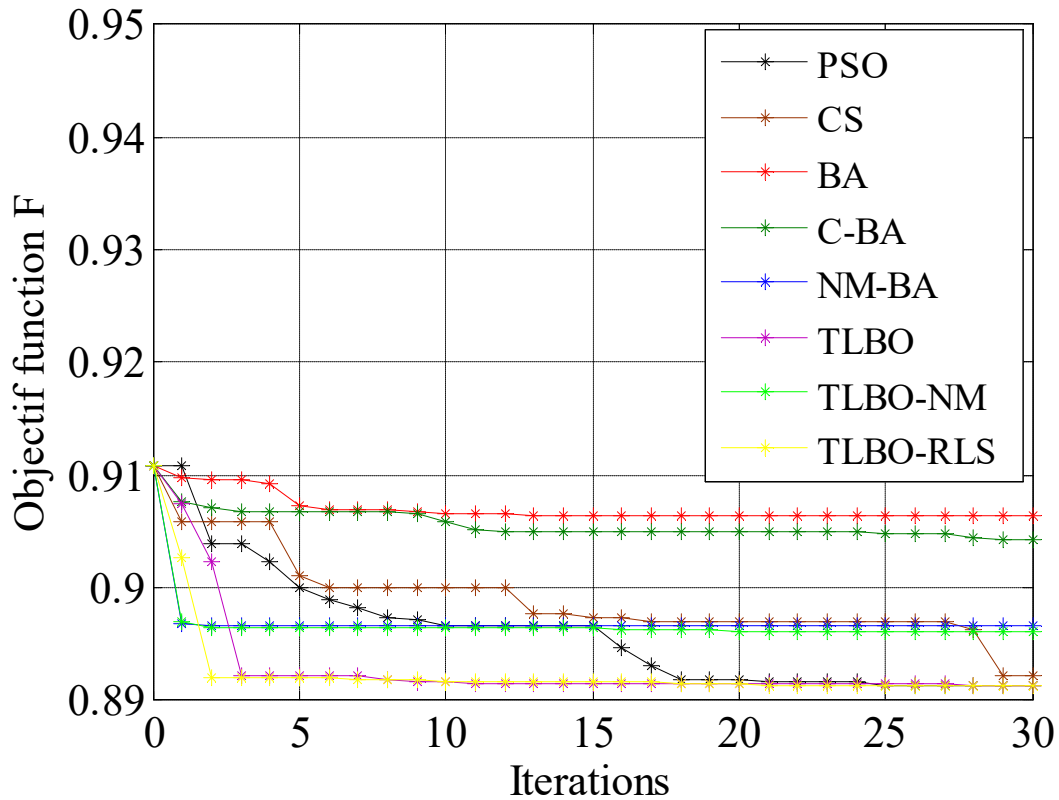


Figure V.18 Variation de la fonction objectif avec les itérations des algorithmes appliqués à la MSDE.

Les algorithmes TLBO-RLS, TLBO, PSO et CS ont donné les meilleurs résultats. Cependant, les algorithmes C-BA et le BA original ont donné des résultats, presque identiques, mais de qualité inférieure aux précédents.

Les résultats des algorithmes C-BA et NM-BA sont meilleurs que ceux de l'algorithme BA original ayant un nombre d'appels à la fonction objectif réduit (300 appels) par rapport aux deux versions hybrides. Parmi ces trois versions, le NM-BA a donné le meilleur résultat avec un nombre d'appels à la fonction objectif et un temps de calcul plus importants.

Pour cette application, l'algorithme TLBO-RLS a donné le meilleur résultat avec une convergence rapide vers la solution optimale, comme on peut le voir sur la figure V.18. Par contre, le temps de calcul et le nombre d'appels à la fonction objectif nécessaires pour atteindre la solution optimale sont plus importants que ceux du TLBO original.

La qualité du résultat de l'algorithme TLBO-NM est au dessous de celle des algorithmes TLBO-RLS et TLBO original, et avec un grand nombre d'appels à la fonction objectif.

L'échec des deux algorithmes hybrides TLBO-NM et NM-BA à fournir le meilleur optimum, est dû à l'utilisation de la méthode de Nelder-Mead simplex, NM, qui converge vers une solution locale, par conséquent les algorithmes hybrides trouvent des difficultés pour sortir de la zone de la solution locale, surtout si le mode de recherche de l'algorithme original est basé sur l'exploitation et pas sur l'exploration.

L'algorithme C-BA a également échoué à fournir le meilleur résultat comme dans les précédentes applications, TEAM 25 et la MSRV, malgré qu'il possède des performances similaires ou supérieures aux celles de l'algorithme TLBO-RLS. Cela est dû au réglage des paramètres de contrôle de l'algorithme BA, opération qui demande plusieurs essais, voire donc des moyens de calcul plus puissants et plus performants que ceux avec lesquels on a réalisé nos travaux de simulation et de calcul. Sachant que nous avons utilisé trois machines de caractéristiques différentes pour effectuer les différents calculs, elles sont données ci-dessous :

Pour le TEAM 25, on a utilisé un PC I7, avec une RAM de 6 Go et un processeur de 2.5 GHz.

Pour la MSR_V on a utilisé un PC I5, avec une RAM de 8 Go et un processeur de 2.5 GHz, à l'exception pour l'algorithme TLBO-RLS on a utilisé un PC I7, avec une RAM de 16 Go et un processeur de 2.5 GHz.

Pour la MSDE on a utilisé deux machines, la première est la même qui a été utilisée pour la MSR_V, tandis que la deuxième est un PC I5, avec une RAM de 4 Go et un processeur de 2.7 GHz (pour l'application des algorithmes PSO, CS, TLBO-RLS).

La figure V.19 montre la variation du couple électromagnétique en fonction de la position rotorique dans les configurations optimales obtenues avec les différents algorithmes utilisés.

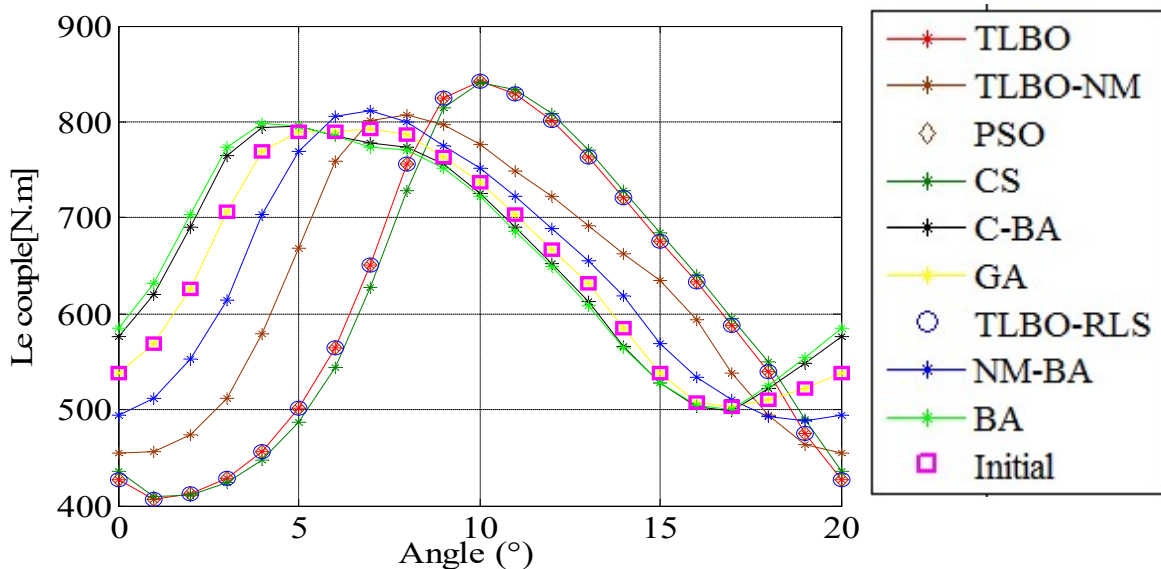


Figure V. 19 Variation du couple électromagnétiques en fonction de la position angulaire du rotor, dans les différentes configurations optimales de la MSDE.

Les valeurs du couple électromagnétique, par ordre ascendant, et celles du volume de l'aimant correspondantes, pour les paramètres de conception optimaux obtenus avec tous les algorithmes appliqués sont données dans le tableau V.26.

Les lignes équipotentielles du potentiel vecteur magnétique et la densité du flux dans la configuration optimale de la MSDE, obtenue avec l'algorithme TLBO-RLS, sont données dans la figure V.20. Tandis que la figure V.21 correspond à celle obtenues avec l'algorithme BA.

	Volume de l'aimant	Valeur moyenne du couple [N.m]
Valeurs initiales	1.7924e-4	595.1507
CS	3.9058e-5	604.9721
TLBO-RLS	3.9643e-5	605.6845
TLBO	3.9642 e-5	605.6904
PSO	3.9642e-5	605.6935
TLBO-NM	8.2802e-5	623.1857
NM-BA	1.0514e-04	636.3699
GA	1.3170e-4	646.4554
C-BA	1.6219e-04	654.8550
BA	1.7563e -4	656.9835

Tableau V.26 Valeurs du couple et du volume de l'aimant dans les différentes configurations optimales.

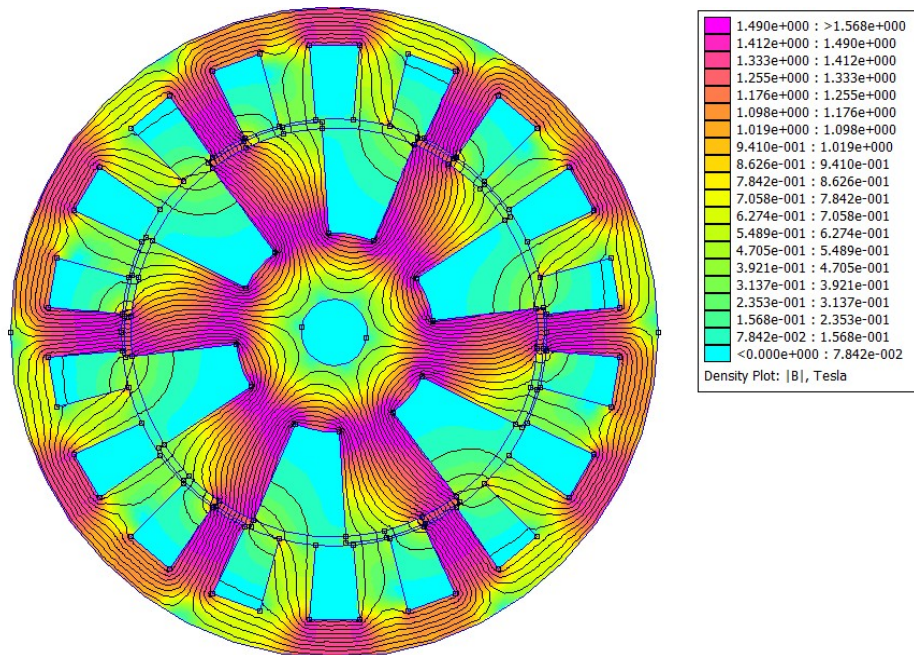


Figure 20. Densité du flux et lignes équipotentielles dans la configuration optimale de la MSDE, obtenue avec l'algorithme TLBO-RLS.

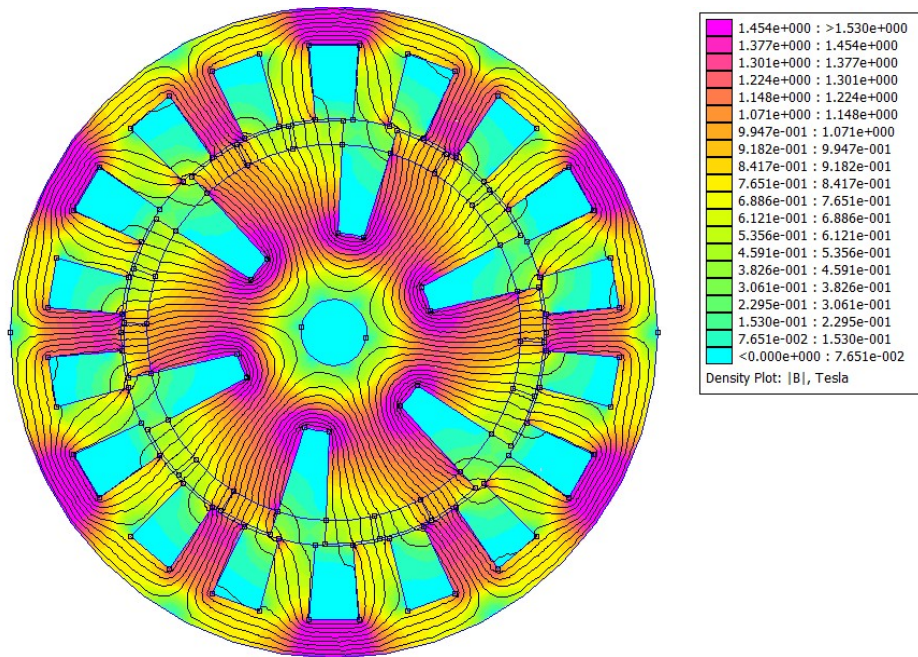


Figure 21. Densité du flux et lignes équipotentielles dans la configuration optimale de la MSDE, obtenue avec l'algorithme BA.

V.4 Conclusion

Dans ce chapitre on a présenté les résultats des différentes stratégies d'optimisation, développées dans ce travail de thèse, pour résoudre trois problèmes de la conception optimale des machines électriques. Ces applications concernent une presse à aimant, TEAM 25, un moteur synchrone à reluctance variable avec barrières du flux et une machine synchrone à double excitation. Les trois applications consistent à trouver les meilleurs paramètres géométriques de la structure étudiée pour atteindre un objectif visé.

Les résultats obtenus ont permis de ressortir l'intérêt de suivre et d'analyser les différentes étapes de la démarche de conception optimale des machines électriques. Où, nous avons observé que l'analyse des résultats et le retour en arrière sont nécessaires pour juger la pertinence des résultats et la stratégie d'optimisation adoptée, cas par exemple de la MSRv et de la MSDE.

L'intérêt des stratégies d'hybridation appliquées aux différents algorithmes de l'Intelligence d'Essaim se révèle par la bonne adaptation des algorithmes hybrides résultants à la résolution des problèmes posés. Où les algorithmes NN-TLBO, NN-PSO, TLBO-RLS et C-BA ont montré une grande efficacité, soit au niveau de la qualité de la solution, soit au niveau des temps de calcul nécessaires pour la convergence vers une solution optimale.

L'objectif principal des hybridations développées est d'établir un certain équilibre entre les deux phases de la recherche de l'optimum, l'exploitation et l'exploration. Le meilleur algorithme est celui qui peut procurer cet équilibre, et qui dépend généralement du réglage de ses paramètres de contrôle.

*Conclusion Générale et
perspectives*

Le travail présenté dans cette thèse concerne le développement des stratégies d'optimisation, par des techniques de l'intelligence artificielle, pour la conception optimale des machines électriques. La démarche adoptée est basée sur l'association de l'outil d'optimisation par algorithmes d'intelligence artificielle et l'outil de modélisation du champ électromagnétique par la méthode des éléments finis. Le travail est une réelle contribution dans les deux domaines, celui de l'optimisation et celui de la conception des machines électriques.

Nous avons développé un outil d'optimisation basé sur différents algorithmes qui appartiennent à la classe de l'intelligence artificielle, connus sous le nom des algorithmes méta-heuristiques ou les algorithmes de l'Intelligence d'Essaim tels que le PSO, le TLBO, le CS, et le BA. La plupart de ces algorithmes sont récents, où leur première apparition se date à partir de 2010. En outre, plusieurs stratégies d'hybridation ont été appliquées à ces algorithmes dans le but de renforcer l'exploration de l'entourage de la solution optimale et réduire le temps de calcul prohibitif, surtout lorsque ces algorithmes sont associés à un calcul numérique par éléments finis. La première stratégie développée consiste à coupler les algorithmes PSO et TLBO avec le réseau de neurones de type MLP, capable d'approximer la réponse de n'importe quel système modélisé, ce qui permet de surpasser des multiples appels à la routine éléments finis pour évaluer la fonction objectif et les fonctions contraintes à chaque itération de l'algorithme d'optimisation. Deux nouveaux algorithmes sont obtenus, le NN-PSO et le NN-TLBO. Dans la deuxième stratégie, les deux algorithmes stochastiques BA et TLBO sont couplés avec d'autres algorithmes déterministes ou exacts, la méthode de Nelder-Mead simplexe et celle de Quasi-Newton ont été considérées. La troisième stratégie est une hybridation entre méthodes méta-heuristiques, où la technique de croisement utilisée dans les algorithmes génétiques a été intégrée dans un processus d'optimisation par le BA, ainsi le nouveau algorithme résultant est nommé algorithme de chauves souris avec le croisement, C-BA. Partant de la même idée, la partie de la recherche locale aléatoire de l'algorithme BA a été intégrée dans le processus d'optimisation par l'algorithme TLBO, il en résulte un nouvel algorithme nommé TLBO avec recherche locale aléatoire, TLBO-RLS. L'objectif de ces deux dernières stratégies d'hybridation est d'intensifier la phase d'exploitation des algorithmes de l'Intelligence d'Essaim, accélérer leur convergence et améliorer la qualité de la solution obtenue. Les différents algorithmes utilisés, classiques ou améliorés, sont implémentés sous environnement MATLAB.

Pour tester la validité et l'efficacité des stratégies développées, elles étaient premièrement, appliquées à trois fonctions test connues. Les résultats obtenus étaient

largement satisfaisants. Après, on a intégré ces stratégies dans une démarche de conception optimale appliquée à trois problèmes électrotechniques différents.

Les applications électrotechniques considérées dans ce travail sont, une presse à aimant, utilisée comme problème test international qui sert pour la validation des méthodes d'optimisation. Les deux autres applications portent sur des structures innovantes, dont la première est un moteur synchrone à reluctance variable avec barrières du flux, et la deuxième concerne une machine synchrone à double excitation. La modélisation des dispositifs étudiés par le calcul du champ électromagnétique a été effectuée par la méthode des éléments finis en utilisant le logiciel libre FEMM.

L'application de la première stratégie d'optimisation utilisant les algorithmes NN-PSO et NN-TLBO pour résoudre ces trois problèmes a permis l'obtention des résultats possédants la même qualité que ceux obtenus avec le PSO et le TLBO. Cela, confirme bien l'efficacité des réseaux de neurones MLP à approximer le problème à partir d'une base de données sans passer par un calcul éléments finis à chaque fois que l'algorithme nécessite l'évaluation des fonctions objectifs et des contraintes. Par conséquent, le temps de calcul, grand handicap des méthodes méta-heuristiques en électrotechnique, avec cette stratégie est vraiment infinitésimal.

L'application des algorithmes hybrides issus de deux autres stratégies d'hybridation ont fourni des résultats de meilleure qualité que ceux classiques. Les résultats obtenus mettent en évidence l'intérêt de combiner les deux phases de la recherche de l'optimum : l'exploration et l'exploitation.

Les stratégies d'optimisation par intelligence artificielle, développées dans ce travail de thèse, ont permis l'accélération du tau de la convergence des méthodes méta-heuristiques et l'amélioration de la qualité de la solution obtenue. On peut confirmer donc qu'elles sont bien adaptées pour la résolution des problèmes de la conception optimale des machines électriques.

Le travail s'ouvre sur plusieurs perspectives, à savoir :

- Considération des problèmes d'optimisation multi objectifs en sens de front de Pareto pour les mêmes algorithmes développés dans ce travail, et des problèmes avec contraintes
- Développer de nouvelles hybridations, entre méthodes méta-heuristiques ou entre méthodes méta-heuristiques et autres déterministes.

- Proposition de nouveaux critères d'arrêt plus performants ayant la capacité de déciller la convergence de l'algorithme vers la solution optimale et éviter les calculs inutiles, ce qui permet de gagner un gain du temps important.
- Utilisation des techniques de la logique floue pour mener une étude comparative plus solide et plus fiable entre les différents algorithmes d'optimisation.
- Exploitation de nouvelles méthodes pour la réduction du temps de calcul, telles que la méthode de transformation réductrice Aliénor [w4], qui permet de ramener le problème d'optimisation dans R^n à un problème d'optimisation univariante, et la méthode dé-compositionnelle d'Adomian [KONF-05], à la place de la méthode des éléments finis. Ces techniques, à ma connaissance est jusqu'à maintenant, ne sont pas encore investiguées dans le domaine électrotechnique.

Bibliographie

- [ABDE-16] M.A.EL AZIZ, and A.E. HASSANIEN, «Modified cuckoo search algorithm with rough sets for feature selection», *Neural Computing and Applications*, vol.29,no.4, pp.925-934,2018.
- [ADAR-16] B. R. ADARSH, T. RAGHUNATHAN, T.JAYABARATHI and X. S. YANG, «Economic dispatch using chaotic bat algorithm», *Energy*, vol.96, pp. 666-675, 2016.
- [AMMA-07] M.Y. AMMAR, « Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu », Thèse de doctorat, Université de l'institut national polytechnique de Toulouse, 2007.
- [AMRA-15] I. AMRAOUI,« Allocation de ressources dans un réseau de radio cognitive en se basant sur les méta-heuristiques: Bat Inspired Algorithm and Bee Colony Algorithm », Thèse de Doctorat, Université de Tlemcen, 2015.
- [ANDR-13] R. ANDREUX, « Modélisation et optimisation des démarreurs à inducteur bobiné pour l'application " stop-start" ou micro-hybride », Thèse de doctorat, Université de Lorraine, 2013.
- [BARA-11] R. BARATI, « Parameter estimation of nonlinear Muskingum models using Nelder-Mead simplex algorithm », *Journal of Hydrologic Engineering*, vol. 16, no 11, pp. 946-954, 2011.
- [BELA-09] M. BELATEL, «Etude des machine électriques par CAO », Thèse de Doctorat, Université Constantine, 2009.
- [BELL-08] Z. BELLI,« Adaptation de méthodes d'optimisation associées au calcul du champ électromagnétique pour l'étude de dispositifs électrotechniques», Thèse de Doctorat, Université de Jijel, 2008.
- [BENE-07] Y. BÉNÉDIC, « Approche analytique pour l'optimisation de réseaux de neurones artificiels », Thèse de Doctorat, Université de Haute Alsace-Mulhouse, 2007.
- [BOUA-10] S.BOUALLAGUI,«Techniques d'optimisation déterministe et stochastique pour la résolution de problèmes difficiles en cryptologie»,Thèse de doctorat, institut national des sciences appliquées de rouen 2010.
- [BOUR-18] K. BOURAHLA , Z. BELLI, T .HACIB and I.ACHOUI, «Application of PSO and TLBO algorithms with neural network for optimal design of electrical machines», *COMPEL-The international journal for computation and*

- mathematics in electrical and electronic engineering, vol. 37, no 2, pp. 549-564, 2018.
- [BOUR-19a] K. BOURAHLA , Z. BELLI and T .HACIB, «Improvements of Bat algorithm using crossover technique and hybridization with Nelder-Mead simplex method», COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, 2019, vol. 38, no. 3, pp. 977-989.
- [BOUR-19b] K. BOURAHLA , Z. BELLI and T .HACIB, «Improved version of teaching learning-based optimization algorithm using random local search: TLBO-RLS». COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, vol. 38, no. 3, pp. 1048-1060, 2019.
- [BENM-08] S. BENMOSTEFA, « Conception de la Technique des Chauves-souris pour la Classification Automatique des Images », Thèse de Doctorat, USTO, 2012.
- [BOUH-12] S. BOUHOUS, « Utilisation de la méthode de la phase stationnaire et des réseaux de neurones artificiels pour la modélisation d'un résonateur micro-ruban à structure ouverte », Thèse de Doctorat, Université de Batna 2, 2012.
- [BRIN-14] I. BRINSTER, « Multi-objective Algorithms for Coupled Optimization of Mechanical and Electromagnetic Systems», Thèse de Doctorat , Carnegie Mellon University, 2014 .
- [BRIS-07] S. BRISSET, « Démarches et outils pour la conception optimale des machines électriques », Rapport de Synthèse, Université des sciences et technologies de Lille, 2007.
- [BURE-91] G. BUREL, « Réseaux de neurones en traitement d'images-des Modèles théoriques aux Applications Industrielles », Thèse de Doctorat, Université de Bretagne occidentale-Brest, 1991.
- [CAI-18] X. CAI, H. WANG,Z.CUI,J.CAI,Y.XUE and L. Wang,«Bat algorithm with triangle-flipping strategy for numerical optimization». International Journal of Machine Learning and Cybernetics, vol. 9, no.2, 199-215, 2018.
- [CHAM-07] A. CHAMEKH, « Optimisation des procédés de mise en forme par les réseaux de neurones artificiels », Thèse de Doctorat, Université d'Angers, 2007.
- [CHEN-18a] X. CHEN, B. XU, C. MEI, Y. DING and K. LI, « Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation», Applied

- energy, vol.212, pp.1578-1588, 2018.
- [CHEN-18b] X. CHEN, B. XU, K.YU, and W. DU, «Teaching-Learning-Based Optimization with Learning Enthusiasm Mechanism and Its Application in Chemical Engineering», *Journal of Applied Mathematics*, pp.1-15, 2018.
- [CHI-17] R. CHI , Y.X. SU, D.H. ZHANG, X.X.CHI, and H.J.ZHANG, «A hybridization of cuckoo search and particle swarm optimization for solving optimization problems». *Neural Computing and Applications*, vol.31,no.1, pp. 653-670, 2017.
- [CULI-94] J.C. CULIOLI, « Introduction à l'optimisation », Ellipses, ISBN 26729-89428-4, 1994.
- [COST-01] M. C. COSTA, « Optimisation de dispositifs électromagnétiques dans un contexte d'analyse par la méthode des élément finis », Thèse de Doctorat, Institut Notionnel Polytechnique de Grenoble, 2001.
- [CREV-10] G .CREVECOEUR, P. SERGEANT, L.DUPRE and R. Van de Walle, «A two-level genetic algorithm for electromagnetic optimization», *IEEE Transactions on Magnetics*, vol. 46, no.7, pp.2585-2595, 2010.
- [CUI-17] Z. CUI, B. SUN, G. WANG, Y. XUE, and J. CHEN, «A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems». *Journal of Parallel and Distributed Computing*, vol.103, pp.42-52, 2017.
- [DIRI-15] A.DIRIYE, S. OUAGUED, Y.AMARA, et G. BARAKAT, «Performance analysis of a series hybrid excited synchronous machine by a hybrid analytical model», *Tenth International Conference on Ecological Vehicles and Renewable Energies (EVER 2015)*, IEEE, pp. 1-6, 2015.
- [DIRI-18] A. A. DIRIYE, «Modélisation analytique tridimensionnelle de nouvelles structures de génératrices électriques destinées à l'éolienne de forte puissance», Thèse de Doctorat, Université de Normandie, 2018.
- [DOC-10] C. DOC, « Contribution à la conception et au dimensionnement d'un actionneur d'embrayage », Thèse de Doctorat, Université de Technologie de Compiègne, 2010.
- [DOGA-13] H. DOGAN, « Méthodologie de conception des machines synchrones à aimants », Application au véhicule électrique avec chargeur rapide embarqué », Thèse de Doctorat, Université de Grenoble, 2013.

- [DORI-04] M. DORIGO and T. STÜTZLE, « Ant Colony Optimization », A Bradford Book, The MIT Press Cambridge, Massachusetts London, England, 2004.
- [DUAN-18] P.Y. DUAN, J.Q. LI, Y. WANG, H.Y. SANG and B.X. JIA, « Solving chiller loading optimization problems using an improved teaching-learning-based-optimization algorithm », *Optimal Control Applications and Methods*, vol.39, no.1, pp. 65-77, 2018.
- [ESPA-99] C. ESPANET, « Modélisation et conception optimale de moteurs sans balais à structure inversée-Application au moteur-roue », Thèse de Doctorat, Université de Franche-Comté, 1999.
- [FAN-06] S. K. S. FAN, Y. C. LIANG and E. ZAHARA, « A genetic algorithm and a particle swarm optimizer hybridized with Nelder–Mead simplex search », *Computers & industrial engineering*, vol. 50, no 4, pp. 401-425, 2006.
- [FEYE-15] P. FEYEL, « Optimisation des correcteurs par les méta-heuristiques. Application à la stabilisation inertielle de ligne de visée », Thèse de Doctorat, Centrale Supélec, 2015.
- [FONT-07] J. FONTCHASTAGNER, « Résolution du problème inverse de conception d'actionneurs électromagnétiques par association de méthodes déterministes d'optimisation globale avec des modèles analytiques et numériques », Thèse de Doctorat, Université de Toulouse, 2007.
- [FORG-09] C. FORGEZ, « Contribution à la modélisation et à l'identification temps réel pour les systèmes à énergie électrique embarquée », Habilitation à Diriger des Recherches de l'Université de Technologie de Compiègne, 2009.
- [FU-18] H. FU, Z. LI, Z. LIU, and Z. WANG, « Research on big data digging of hot topics about recycled water use on micro-blog based on particle swarm optimization », *Sustainability*, vol.10, no.7, p.2488, 2018.
- [GHER-13] A. GHERBOUDJ, « Méthodes de résolution de problèmes difficiles académiques », Thèse de Doctorat, Université de Constantine2, 2013.
- [GUER-04] V. GUERIN, « Prédiction et compréhension de la densification des poudres commerciales d'alumine et de fer grâce à une approche par réseau de neurones artificiels », Thèse de Doctorat, Verlag nicht ermittelbar, 2004.
- [HACI-04] T. HACIB, « Utilisation de la méthodologie du problème inverse pour la caractérisation et l'optimisation des dispositifs électromagnétiques », Mémoire de Magister, Université de Jijel, 2004.

- [HACI-08] T. HACIB, « contribution à l'étude du problème électromagnétique inverse par une méthode couplée éléments finis réseaux de neurones », Thèse de Doctorat, Université de Jijel, 2008.
- [HAJJ-03] O. HAJJI, « Contribution au développement de méthodes d'optimisation stochastiques », Thèse de Doctorat, Ecole Centrale de Lille, 2003.
- [HAMO-12] A. HAMOUDA, «Elaboration d'un programme de machines synchrones sous Matlab en vue d'une étude d'optimisation», mmoire de Magister, Université de tizi-ouzou, 2012.
- [HASA-17] M. HASANIPANAH, R.NADERI, J.KASHIR, S.NOORANI and A.Z.A QALEH, «Prediction of blast-produced ground vibration using particle swarm optimization», *Engineering with Computers*, vol.33, no.2, pp.173-179, 2017.
- [ILEA-11] D. ILEA, « Conception optimale des moteurs à réluctance variable à commutation électronique pour la traction des véhicules électriques légers », Thèse de Doctorat, Ecole Centrale de Lille, 2011.
- [JANN-10] X. JANNOT, « Modélisation et optimisation d'un ensemble convertisseur-machine. Application aux systèmes d'entraînement à haute vitesse », Thèse de Doctorat, Centrale Supélec, 2010.
- [KANK-17] M. KANKAL, and E. UZLU, «Neural network approach with teaching-learning-based optimization for modeling and forecasting long-term electric energy demand in Turkey», *Neural Computing and Applications*, vol.28,no.1,pp.737-747, 2017.
- [KAMI-08] A. KAMIL, « Application d'un algorithme hybride à colonies de fourmis au problème d'affectation quadratique », Thèse de Doctorat, Université du Québec, 2008.
- [KAMM-06] R. KAMMARTI, « Approches évolutionnistes pour la résolution du 1-PDPTW statique et dynamique », Thèse de Doctorat, Ecole Centrale de Lille, 2006.
- [KONF-05] B.O. KONFE, « Nouvelles méthodes mathématiques Alienor et Adomian, pour la Biomédecine », Thèse de Doctorat, Université de Ouagadougou, 2005.
- [KENN-95] J. KENNEDY, J. and R. EBERHART, « Particle swarm optimization», *Proceedings of the IEEE International Conference on Neural Networks*, IEEE, Vol. 4, pp. 1942-1948, 1995.

- [LAGA-98] J.C. LAGARIAS, J.A. REEDS, M.H. WRIGHT and P.E. WRIGHT, « Convergence properties of the Nelder-Mead simplex method in low dimensions », *SIAM Journal on optimization*, vol. 9, no.1, pp.112-147, 1998.
- [LE RO-08] N. LE ROUX, « Avancées théoriques sur la représentation et l'optimisation des réseaux de neurones », université de Montréal, 2008.
- [MENG-15] K. MENGHOUR, « Approches Bio-inspirées pour la Sélection d'Attributs », Thèse de Doctorat, Université Badji Mokhtar-Annaba, 2015.
- [MERH-13] A. MERHOUM, « Optimisation des techniques de MLI pour les équipements FACTS à base d'onduleur », Thèse de Doctorat, Université Mohamed Boudiaf des sciences et de la technologie d'Oran, 2013.
- [MOKR-05] L. MOKRANI, « contribution à la C.A.O optimisée des machines électriques, application au moteur linéaire à induction », Thèse de Doctorat, Université de Batna 2, 2005.
- [MOGH-16] M. A. MOGHADDAM, R. GOLMEZERGİ and F. KOLAHAN, « Multi-variable measurements and optimization of GMAW parameters for API-X42 steel alloy using a hybrid BPNN-PSO approach », *Measurement*, vol. 92, p. 279-287, 2016.
- [MONM-00] N. MONMARCHÉ, « Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation », Thèse de Doctorat, Université François Rabelais-Tours, 2000.
- [NGUY-11] P. H. NGUYEN, « Impacts des modèles de pertes sur l'optimisation sur cycle d'un ensemble convertisseur-machine synchrone : applications aux véhicules hybrides », Thèse de Doctorat, École normale supérieure de Cachan-ENS Cachan, 2011.
- [NELD-65] J.A. Nelder, and R. Mead, "A simplex method for function minimization", *The Computer Journal*, Vol. 7 No. 4, pp. 308-313, 1965.
- [N'TSH-11] B.SESANGA, « Optimisation de Gammes: Application à la conception des machines synchrones à concentration du flux », Thèse de Doctorat, Institut National Polytechnique de Grenoble (INPG), Grenoble, France, 2011.
- [OUAG-15] S.OUAGUED, A. A.DIRIYE, Y.AMARA, and G. BARAKAT, « A general framework based on a hybrid analytical model for the analysis and design of permanent magnet machines », *IEEE Transactions on Magnetics*, 2015, vol. 51, no.11, pp. 1-4.

- [OSAB-19] E. OSABA, X.S. YANG, I.FISTER JR, J. DEL SER, P. LOPEZ-GARCIA, et A.J.VAZQUEZ-PARDAVILA, « A discrete and improved bat algorithm for solving a medical goods distribution problem with pharmacological waste collection», *Swarm and evolutionary computation*, vol.44, 273-286,2019.
- [PRAk-18] R. Prakash, B. Lokeshgupta and S. Sivasubramani, «Optimal Site and Size of DG with different Load Models using Cuckoo Search Algorithm», In 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES) pp. 1-6,December 2018.
- [PATE-16] V. PATEL, and V. SAVSANI, « Multi-objective optimization of a Stirling heat engine using TS-TLBO (tutorial training and self learning inspired teaching-learning based optimization) algorithm», *Energy*, vol.95, pp.528-541, 2016.
- [POHL-06] H. Pohlheim, «Geatbx examples of objective functions», http://www.Geatbx.com/download/GEATbx_ObjFunExpl_v37.Pdf. 2006.
- [QUAR-04] A. QUARTERONI, R.SACCO, F.SALERI, «Méthodes numériques Algorithmes analyse et applications, Springer_Verlag Italia, Milano, 2007.
- [RAO-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, no.3, 303-315, 2011.
- [RAO-13] R.V. RAO, and V. PATEL, « An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problem», *Scientia Iranica*, Vol. 20 No. 3, pp. 710-720, 2013.
- [RAO-17] P.S.RAO, P.K.JANA, and H.BANKA, «A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks», *Wireless networks*, vol. 23, No.7, pp. 2005-2020, 2017.
- [RAJI-17] V. RAJINIKANTH, S.C.SATAPATHY, S.L. FERNANDES, and S. NACHIAPPAN, «Entropy based segmentation of tumor from brain MR images—a study with teaching learning based optimization. *Pattern Recognition Letters*, vol. 94, pp. 87-95,2017.
- [REDJ-06] R. REDJEM, « Calcul et conception assistée par ordinateur des machines électriques : application aux moteurs asynchrones», *Mémoire de Magister, Université de Constantine*, 2006.

- [RODR-16] D. P. RODRIGUEZ, « Modélisation et optimisation des machines synchro-réductantes à aimants permanents et de leur électronique », Thèse de Doctorat, Centrale Supélec, 2016.
- [SANT-12] SANTOS, C. A. G. FREIRE, P. MACHADO and S. K. MISHRA, «Cuckoo search via Lévy flights for optimization of a physically-based runoff-erosion model», Journal of Urban and Environmental Engineering (JUEE), vol. 6, no 2, pp.123-131,2012.
- [SALD-1992] R. R. SALDANHA, « Optimisation en électromagnétisme par application conjointe des méthodes de programmation non linéaire et de la méthode des éléments finis », Thèse de Doctorat, Institut Polytechnique de Grenoble, 1992.
- [SAUV-00] C .SAUVEY, « Contribution méthodologique à la modélisation pour le dimensionnement de moteurs à reluctance variable», Thèse de Doctorat. Institut National Polytechnique de Grenoble-INPG, 2000.
- [SERH-09] H. SERHOUD, « Contribution à l'étude de la machine synchrone à réductance variable », Mémoire de Magister, Université de Batna, 2009.
- [TAKA-96] N. TAKAHASHI, «Optimization of die press model», Proc. TEAM Workshop 6th Round, Okayama, Japon, Mar, 20-21. 1996.
- [TARA-19] C. TARAMASCO, R.OLIVARES, R. MUNOZ, R. SOTO, M.VILLAR, V.HUGO C. de ALBUQUERQUE, «The patient bed assignment problem solved by autonomous bat algorithm»,Applied Soft Computing Journal, vol.81, pp.1-15,2019.
- [TEY-18] K.S.TEY, S.MEKHILEF, and M. SEYEDMAHMOUDIAN, «Implementation of BAT Algorithm as Maximum Power Point Tracking Technique for Photovoltaic System Under Partial Shading Conditions», In 2018 IEEE Energy Conversion Congress and Exposition (ECCE), pp. 2531-2535, September 2018.
- [TFAI-07] W. TFAILI, « Conception d'un algorithme de colonie de fourmis pour l'optimisation continue dynamique », Thèse de Doctorat, Université de Paris val de Marne, 2007.
- [TRAN-09] T. TRAN, « Problèmes combinatoires et modèles multi-niveaux pour la conception optimale des machines électriques », Thèse de Doctorat, École Centrale de Lille, 2009.
- [TUNG-16] H. D. TUNGADIO, A. JORDAAN, and M.W. SITI, «Power system state

- estimation solution using modified models of PSO algorithm: Comparative study», *Measurement*, vol. 92, pp. 508-523, 2016.
- [VASC-94] J. VASCONCELOS, «Optimisation de forme des structures électromagnétiques », Thèse de Doctorat, Ecole Centrale de Lyon, 1994.
- [VANA-15] C.VANARET, « Hybridation d'algorithmes évolutionnaires et de méthodes d'intervalles pour l'optimisation de problèmes difficiles », Thèse de doctorat, École Doctorale Mathématiques, Informatique et Télécommunications (Toulouse), 2015 .
- [WANG-11] L.Wang, Y. Xu, and L.Li, « Parameter identification of chaotic systems by hybrid Nelder–Mead simplex search and differential evolution algorithm », *Expert Systems with Applications*, vol.38,no.4, pp. 3238-3245, 2011.
- [WANG-19] Y.WANG, P.WANG, J.ZHANG, Z.CUI, X.CAI, W.ZHANG, et J. CHEN, «A novel bat algorithm with multiple strategies coupling for numerical optimization», *Mathematics*, vol. 7, no .2, pp. 135,2019.
- [YANG-09] X.S.YANG and S.DEB, « Cuckoo Search Via Levy Flight », *Proc. Of World Congress on Nature & Biologically Inspired Computing*, India, IEEE Publications, pp. 210-214, USA, December 2009.
- [YANG-13] X.S. NG and X. HE, «Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*», vol. 5, no 3, p. 141-149, 2013.
- [YANG-10] X. S. YANG, «A new metaheuristic bat-inspired algorithm», In *Nature inspired cooperative strategies for optimization (NICSO 2010)* Springer, pp. 65-74, 2010.
- [YANG-11] X.S.YANG, «Bat algorithm for multiobjective optimization», *International Journal of Bio-Inspired, Computation*, vol.3, no.5, pp.267-274, 2011.
- [YANG-18] X.S.YANG , S. DEB, and S.K. MISHRA, «Multi-species cuckoo search algorithm for global optimization» *Cognitive Computation*, vol.10, no.5, pp.1085-1095,2018.
- [ZEMO-03] R. ZEMOURI, «Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance », Thèse de Doctorat, Université de Franche-Comté, 2003.
- [ZHAN-18] M. ZHANG, H.WANG, Z. CUI, and J. CHEN, «Hybrid multi-objective cuckoo search with dynamical local search», *Memetic Computing*, vol.10,

- no.2, pp.199-208, 2018.
- [ZHU-19] X. ZHU, and N. WANG, « Cuckoo search algorithm with onlooker bee search for modeling PEMFCs using T2FNN», Accepted for publication in *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 740-753, 2019.
- [WEI-06] Z. WEI, G. LI, and L. QI, « New quasi-Newton methods for unconstrained optimization problems», *Applied Mathematics and Computation*, vol. 175, no 2, p. 1156-1188, 2006.
- [W1] <http://www.compumag.org/jsite/images/stories/TEAM/problem25.pdf>
- [W2] [\[http://www.femm.info/wiki/Download\]](http://www.femm.info/wiki/Download)
- [W3] [\[http://thesis.univ-biskra.dz/2124/4/chapitre02.pdf\]](http://thesis.univ-biskra.dz/2124/4/chapitre02.pdf)
- [W4] <https://bu.umc.edu.dz/theses/math/HAM6543.pdf>