



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la  
Recherches Scientifique



Université des Frères Mentouri de Constantine 1  
Faculté des Sciences de la Technologie  
Département d'Electrotechnique

N° de série :01/electech/2021

N° d'ordre :64/DS/2021

## Thèse

Pour l'obtention du diplôme de  
**DOCTORAT EN SCIENCES**  
Spécialité : ELECTROTECHNIQUE  
Option : ELECTROTECHNIQUE

Présenté par :

**CHAIB Ala Eddine**

Magister en Electrotechnique

## Thème

---

**ETUDE DE L'ECOULEMNT DE PUISSANCE OPTIMAL  
PAR LES METHODES D'OPTIMISATION  
METAHEURISTIQUES**

---

Soutenue le :24/06/2021

Devant le jury composé de :

<b>Président</b>	M. BOUCHERMA	Prof Université FM Constantine 1
<b>Rapporteur</b>	R. MEHASNI	Prof Université FM Constantine 1
<b>Examineur</b>	D. LABED	Prof Université FM Constantine 1
<b>Examineur</b>	H. ALLAG	Prof Université M S B Jijel
<b>Examineur</b>	T. HACIB	Prof Université M S B Jijel
<b>Invité</b>	H.R.E.H. BOUCHEKARA	Dr. Université Harf Al Batin, Arabie Saoudite

## Remerciements

Je tiens à remercier vivement mon directeur de thèse Pr R. MEHASNI, professeur au département d'électrotechnique de l'université des frères Mentouri Constantine 1, pour son acceptation d'encadrer mes travaux de thèse. Je lui adresse toutes mes reconnaissances pour sa disponibilité, ses qualités humaines et l'effort fourni pour atteindre les objectifs tracés.

Un remerciement très particulier à Mr H.R.E.H. BOUCHKERA, professeur à l'université Harf Al Batin (Arabie Saoudite) pour son accompagnement durant la réalisation de ce travail. Je lui adresse aussi toutes mes reconnaissances pour son suivi, ses orientations importantes, sa contribution inestimable notamment dans la tâche concernant le développement et la mise en œuvre des algorithmes d'optimisation.

Mes sincères remerciements aux membres du jury chacun à son nom :

Monsieur M. BOUCHERMA, professeur au département d'électrotechnique à l'université des frères Mentouri Constantine 1, pour l'honneur qui m'a fait par l'acceptation de présider le jury de soutenance.

Monsieur D. LABED, professeur à l'université des frères Mentouri Constantine 1, pour son acceptation d'être membre de ce jury et d'examiner mes travaux de thèse.

Monsieur H. ALLAG, professeur à l'université Mohamed Esseddik Benyahia Jijel pour son acceptation d'être membre de ce jury et d'examiner mes travaux de thèse.

Monsieur T. HACIB, professeur à l'université Mohamed Esseddik Benyahia Jijel, pour son acceptation d'être membre de ce jury et d'examiner mes travaux de thèse.

Tous mes remerciements les plus distinguées à mes enseignants de l'université des frères Mentouri Constantine 1, pour leurs contributions à ma formation et pour le savoir qu'ils m'ont appris.

À tous ceux, que je n'ai pas cités ici, je dirai MERCI !

## Table des Matières

Remerciements .....	<b>i</b>
Liste des figures.....	<b>ii</b>
Liste des tableaux .....	<b>vi</b>
Nomenclature .....	<b>x</b>
Introduction générale.....	<b>1</b>

### CHAPITRE I :

## REVUE SUR LES METHODES D'OPTIMISATION DU PROBLEME DE L'ECOULEMENT DE PUISSANCE OPTIMAL (OPF)

<b>I.1. Formulation du problème de l'écoulement de puissance optimal .....</b>	<b>4</b>
<b>I.2. Méthodes de résolution du problème de l'écoulement de puissance optimal.....</b>	<b>5</b>
I.2.1. Méthodes d'optimisation déterministes.....	5
I.2.1.1. La méthode du Gradient .....	5
I.2.1.2. Méthode de Programmation Quadratique.....	5
I.2.1.3. Les Méthodes de Points Intérieurs.....	6
I.2.2. Méthodes d'optimisation méta-heuristiques .....	6
I.2.2.1. Algorithmes Génétiques .....	7
I.2.2.2. Méthode d'Optimisation par Essaim de Particules.....	7
I.2.2.3. La méthode du Recuit Simulé.....	9
I.2.2.4. La méthode de Recherche Tabou.....	9
I.2.2.5. Algorithme d'optimisation des Colonies de Fourmis (ACO).....	11
I.2.2.6. Algorithme de Recherche Bactérienne .....	11
I.2.2.7. Algorithme d'Evolution Différentielle .....	12
I.2.2.8. Méthode de la Colonie d'Abeilles.....	14
I.2.2.9. Algorithme d'optimisation basé sur l'Electromagnétisme .....	14

I.2.2.10. Algorithme d'Optimisation Basé sur le Processus Enseignement-Apprentissage.....	15
I.2.2.11. Autres Méthodes d'Optimisation Metaheuristiques.....	16

## CHAPITRE II :

### FORMULATION DU PROBLEME DE L'ECOULEMENT DE PUISSANCE OPTIMAL (OPF)

<b>II. Formulation de l'écoulement de puissance optimal .....</b>	<b>22</b>
<b>II.1. Introduction générale sur le problème de l'écoulement de puissance .</b>	<b>22</b>
II.1.1. Problème de l'écoulement de puissance.....	22
II.1.2. Équations de l'écoulement de puissance .....	22
<b>II.2. Formulation de l'écoulement de puissance optimal (OPF) .....</b>	<b>25</b>
II.2.1. Classification des variables d'équations d'écoulement de puissance optimal.....	26
II.2.1.1. Les variables d'état.....	26
II.2.1.2. Les variables de contrôle .....	26
II.2.2. Contraintes de problème de l'écoulement de puissance.....	27
II.2.2.1. Contraintes d'égalités .....	27
a. Contraintes sur la puissance active.....	27
b. Contraintes sur la puissance réactive .....	27
II.2.2.2. Contraintes d'inégalités.....	27
a. Contraintes sur les générateurs.....	27
b. Contraintes sur les transformateurs .....	27
c. Contraintes sur les condensateurs de compensation .....	27
II.2.3. Limites de sécurité.....	28
II.2.4. Traitement de variables discrètes .....	29
II.2.5. Traitement de dépassement des contraintes .....	29
II.2.6. Fonction Objectif.....	30

II.2.6.1. Minimisation du coût de production .....	31
II.2.6.2. Coût de production avec l'introduction de l'effet d'ouverture des vannes .....	31
II.2.6.3. Caractéristique avec plusieurs types de combustibles.....	32
II.2.6.4. Caractéristique avec des zones de fonctionnements prohibées.....	32
II.2.6.5. Minimisation des pertes actives.....	34
II.2.6.6. Minimisation des pertes réactives .....	33
II.2.6.7. Minimisation de la déviation du profil de tension.....	34
II.2.6.8. Perfectionnement de la Stabilité de tension .....	34
II.2.6.9. Minimisation des émissions des gaz .....	35
<b>II.3. Formulation de l'écoulement de puissance optimal (OPF) avec introduction des dispositifs FACTS.....</b>	<b>36</b>
II.3.1. Modélisation des dispositifs FACTS.....	36
II.3.1.1. Les compensateurs statiques d'énergie réactive .....	37
II.3.1.2. Modélisation des compensateurs série contrôlés par thyristors TCSC .	37
II.3.1.3. Transformateur déphaseur commandé par thyristor (TCPST).....	38
<b>II.4. Ecoulement de puissance optimal avec les contraintes de sécurité .....</b>	<b>38</b>
II.4.1. Mode préventif .....	40
II.4.2. Mode correctif .....	40
<b>Conclusion .....</b>	<b>42</b>

## **CHAPITRE III :**

### **METHODES D'OPTIMISATION METAHEURISTIQUES**

III. Méthodes d'optimisation méta-heuristiques .....	43
III.1. Les Algorithmes Génétiques .....	43
III.2. Algorithme du Recuit Simulé.....	44
III.3. Algorithme d'optimisation par essaims de particules.....	45
III.4. Algorithme d'évolution différentielle .....	46
III.5. Algorithme d'Optimisation de Colonie de Fourmis .....	47
III.6. Algorithme de Recherche Harmonique.....	48

III.7. Electromagnétisme Comme Algorithme.....	49
III.8. Algorithme de Colonie d'Abeilles Artificielle .....	51
III.9. Algorithme de la Concurrence des Impérialistes .....	52
III.10. Algorithme d'Optimisation Basé sur la Biogéographie .....	54
III.11. Algorithme des Lucioles .....	55
III.12. Algorithme de la Ligue des Championnats.....	56
III.13. Algorithme d'Explosion des Grenades.....	57
III.14. Algorithme de recherche Gravitationnelle.....	58
III.15. Algorithme d'Optimisation de Mouvement de Vent.....	60
III.16. Algorithme d'Optimisation des Chauves-souris.....	60
III.17. La Recherche Cuckoos.....	62
III.18. Algorithme Génétique avec un Processus de Croisement Multi-Parent.	63
III.19. Algorithme d'Optimisation Basé sur le Processus Enseignement- Apprentissage .....	64
III.20. Algorithme du Cycle de l'Eau .....	66
III.21. Algorithme de Recherche Différentielle .....	67
III.22. Algorithme d'Explosion des Mines.....	69
III.23. Algorithme de Recherche par Retour en Arrière .....	70
III.24. Algorithme des Trous Noirs.....	71
III.25. Algorithme d'Optimisation des Loups Gris .....	71
III.26. Algorithme d'Optimisation de l'Essaim de Poulet.....	73
III.27. Algorithme d'Optimisation JAYA.....	74
III.28. Algorithmes d'optimisation de collision des corps.....	74
III.29. Algorithmes d'Optimisation de Feux d'Artifice .....	75
III.30. Algorithme de Meilleur Joueur (MVPA).....	76
Conclusion.....	78

## Chapitre IV

### Comparaison entre les différentes méthodes d'optimisation et techniques appliquées

<b>Introduction</b> .....	<b>79</b>
<b>IV.1. Les tests paramétriques</b> .....	<b>79</b>
<b>IV.2. Les tests non paramétrique</b> .....	<b>80</b>
IV.2.1. Test de signe.....	<b>80</b>
IV.2.2. Test de signe multiple .....	<b>81</b>
IV.2.3. Test de Wilcoxon .....	<b>81</b>
IV.2.4. Test de Friedman.....	<b>82</b>
IV.2.5. Le Test de la Page .....	<b>83</b>
IV.2.6. Le Profil de performance .....	<b>85</b>
<b>Conclusion</b> .....	<b>86</b>

## Chapitre V

### Mise en œuvre des techniques d'optimisation à l'OPF et résultats obtenus

<b>Introduction</b> .....	<b>87</b>
<b>V.1. Comparaison paramétrique des méthodes d'optimisation</b> .....	<b>88</b>
V.1.1. Comparaison des méthodes d'optimisation pour le réseau IEEE30 .....	<b>88</b>
V.1.1.1. Minimisation du coût de carburant (cas 1).....	<b>89</b>
V.1.1.2. Minimisation de la déviation du profil de tension (cas 2) .....	<b>97</b>
V.1.1.3. Amélioration de la stabilité de tension (cas 3).....	<b>104</b>
V.1.1.4. Minimisation de l'émission gaz (cas 4) .....	<b>109</b>
V.1.1.5. Minimisation du coût de carburant (cas 5 à 9).....	<b>115</b>
V.1.2. Comparaison des méthodes d'optimisation pour le réseau IEEE57 .....	<b>117</b>
V.1.3. Comparaison des méthodes d'optimisation pour le réseau IEEE 118 ...	<b>119</b>

V.1.4. Comparaison des méthodes d'optimisation pour le réseau Algérien 59 jeux de barres.....	120
<b>V.2. Comparaison non paramétrique des méthodes d'optimisation.....</b>	<b>124</b>
V.2.1. Application de la méthode de test de signes multiples .....	124
V.2.2. Application de test de Friedman.....	126
V.2.3. Application de la méthode de profile de performance .....	129
<b>Conclusion.....</b>	<b>131</b>
<b>CONCLUSION GENERALE .....</b>	<b>132</b>

## Liste des figures

Fig. II.1	Système à deux jeux de barre.....	23
Fig. II.2	Courbe du coût du combustible d'une unité thermique en considérant l'effet d'ouverture des vannes d'admission de vapeur.....	31
Fig. II.3	Caractéristique du cout d'un générateur avec plusieurs combustibles.....	32
Fig. II. 4	Courbe de coût du combustible d'une unité thermique avec des zones de fonctionnements prohibées.....	33
Fig. II.5	Modèle équivalent d'un SVC.....	38
Fig. II.6	Modèle équivalent d'un TCSC.....	38
Fig. II.7	Modèle équivalent d'un TCPST.....	39
Fig. V.1.a	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	92
Fig. V.1.b	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	93
Fig. V.1.c	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	94
Fig. V.1.d	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	95
Fig. V.1.e	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	96
Fig. V.1.f	Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30.....	97
Fig. V.2.a	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	98
Fig. V.2.b	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	99
Fig. V.2.c	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	100
Fig. V.2.d	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	101
Fig. V.2.e	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	102
Fig. V.2.f	Évolution du la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30.....	103
Fig. V.3	Profile de tension après optimisation de la déviation standard.....	103
Fig. V.4	Profile de tension avant optimisation de la déviation standard.....	103
Fig. V.5.a	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	104
Fig. V.5.b	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	105
Fig. V.5.c	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	106
Fig. V.5.d	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	107
Fig. V.5.e	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	108

Fig. V.5.f	Évolution du Lmax et de pénalité au cours des itérations pour le réseau IEEE30.....	109
Fig. V.1.a	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	109
Fig. V.1.b	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	110
Fig. V.1.c	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	111
Fig. V.1.d	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	112
Fig. V.1.e	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	113
Fig. V.1.f	Évolution du l'émission et de pénalité au cours des itérations pour le réseau IEEE30.....	114
Fig. V.7	Profile de performance pour tous les cas.....	130

## Liste des tableaux

Tableau II.1 Paramètres connus et inconnus liés à chaque type de jeux de barre.....	22
Tableau V.1 Cas de l'OPF étudiés pour chaque type de réseau.....	87
Tableau V.2 Caractéristiques principales du réseau IEEE30 considéré.....	88
Tableau V.3 Coefficients de coût et d'émission du réseau IEEE 30 considéré.....	88
Tableau V.4 Conditions de calculs liés à chaque méthode de minimisation.....	89
Tableau V.5. Valeurs optimales du coût de carburant des cas traités (cas 5 à 9).....	115
Tableau V.6 Caractéristiques du réseau IEEE57.....	117
Tableau V.7 Les coefficients de coût et d'émission pour le réseau IEEE57.....	117
Tableau V.8 Valeurs optimales des cas traités (cas 1 à 3) et classement des méthodes.....	118
Tableau V.9 Valeurs optimales des cas traités (cas 1 à 3).....	119
Tableau V.10 Caractéristiques du réseau Algérien 59.....	119
Tableau V.11 Les coefficients de coût et d'émission pour le réseau algérien.....	120
Tableau V.12 Valeurs optimales liées aux cas traités (cas 1 à 4).....	121
Tableau V.13 Sommes des différences par rapport au meilleur algorithme.....	123
Tableau V.14.a Nombre des gagnants (signes positifs) par algorithme de contrôle.....	124
Tableau V.14.b Nombre des pertes (signes négatifs) par algorithme de contrôle.....	125
Tableau V.15 moyenne des rangs des algorithmes.....	126
Tableau V.16. Partie 1 Valeur de probabilité P par paire d'algorithme.....	127
Tableau V.16. Partie 2 valeur de probabilité P par paire d'algorithme.....	128

## Nomenclature

$X$ :	Vecteur des variables d'état, ou les variables dépendants
$u$ :	Vecteur des variables de contrôle, ou les variables indépendants.
$J(x,u)$ :	Fonction objectif
$g(x,u)$ :	Contraintes d'égalités.
$h(x,u)$ :	Contraintes d'inégalités.
$P_t$ :	Fonction pénalité
$P_{Gi}$ :	Puissance active générée dans le jeu de barre dont $i=1 \dots NB$
$P_{Di}$ :	Puissance active demandée par le jeu de barre dont $i=1 \dots NB$
$Q_{Gi}$ :	Puissance réactive générée dans le jeu de barre $i$ dont $i=1 \dots NB$
$Q_{Di}$ :	Puissance réactive demandée dans le jeu de barre $i$ dont $i=1 \dots NB$
$NB$ :	Nombre de jeu de barre de réseau
$V_{Li}$ :	Modules des tensions des jeux de barres charges (load) dont $i=1 \dots NL$
$NL$ :	Nombre de jeu de barre charge (load)
$Q_{Gi}$ :	Puissance réactive générée par le générateur $i$ dont $i=1 \dots NG$
$NG$ :	Nombre de générateur
$S_{li}$ :	Puissance apparente de la ligne $i$ dont $i=1 \dots nl$
$Nl$ :	Nombre de lignes dans le réseau
$NT$ :	Nombre de transformateurs
$NC$ :	Nombre de compensateurs réactifs
$Q_{ci}$ :	Puissance réactive injecté par les compensateurs réactifs dont $i=1 \dots NC$
$T_i$ :	Facture de régulation des transformateurs régulateurs de charge $i=1 \dots NT$
$V_i$ :	Tension au niveau de jeu de barre $i$ dont $i=1 \dots NB$
$B_{ij}$ :	Suceptance de la ligne entre les deux jeux de barre $i$ et $j$
$G_{ij}$ :	Conductance de la ligne entre les deux jeux de barre $i$ et $j$
$V_{Gi}$ :	Tension au niveau des générateurs
$\lambda_G$ :	Facture de pénalité sur la puissance active générée par le générateur de référence
$\lambda_v$ :	Facture de pénalité sur la tension des jeux de barre charge
$\lambda_Q$ :	Facture de pénalité sur la puissance réactive des jeux de barre charge
$a_i$ :	Coût à vide du générateur $i$ .
$b_i$ :	Coefficient de coût linéaire du générateur $i$ .

$c_i$ :	Coefficient de coût quadratique du générateur $i$ .
$e_i$ :	Coefficient de coût représente l'effet d'ouverture des vannes du générateur $i$ .
$d_i$ :	Coefficient de coût représente l'effet d'ouverture des vannes du générateur $i$ .
$P_L$ :	Pertes en puissance active de la ligne
$Q_L$ :	Pertes en puissance réactive de la ligne
$VD$ :	Coefficient de la déviation de profil de tension
$J_{G1et G2}$ :	Fonction objectif pour les générateurs avec l'option multi fuels
$J_{G\text{ reste}}$ :	Fonction objectif pour les générateurs sans option multi fuels.
$(P_{G_i}^{inf})_m$ :	Limite inférieure pour la zone d'exploitation non-prohibée $m$ de générateur $i$ .
$(P_{G_i}^{sup})_m$ :	Limite supérieure pour la zone d'exploitation non-prohibée $m$ de générateur $i$ .
$L_j$ :	L'indice de stabilité $L$ pour le $j^{\text{th}}$ jeu de barre
$b_{SVC}$ :	Puissance réactive d'un SVC (Static Var Compensator)
$X_{TCSC}$ :	Réactance d'un TCSC.
$X_{ligne}$ :	Réactance de la ligne ( $i, j$ ) avant l'introduction de TCSC
$X_{i,j}$ :	Réactance de la ligne ( $i, j$ ) après l'introduction de TCSC
$r_{TCSC}$ :	Coefficient qui représente le degré de compensation de TCSC
$t$ :	Rapport de régularisation en phase de TCPST
$c$ :	Nombre des cas de contingence à considérer.
$GA$ :	Algorithme Génétique (Genetic Algorithm)
$SA$ :	Recuit Simulé (Simulated annealing)
$PSO$ :	Optimisation par Essaims de Particules (Particle Swarm Optimization)
$DE$ :	Evolution Différentielle (Differential Evolution)
$ACO$ :	Optimisation de Colonie de Fourmis (Ant Colony Optimization)
$HS$ :	Recherche Harmonique (Harmony Search)
$EM$ :	Electromagnétisme Comme Algorithme (Electromagnetism-like Mechanism)
$ABC$ :	Colonie d'Abeilles Artificielle (Artificial Bee Colony)
$ICA$ :	Algorithme de Concurrence Impérialiste (Imperialist Competitive Algorithm)
$BBO$ :	Optimisation Basé sur la Biogéographie (Biogeography-Based Optimization)
$FA$ :	Algorithme de Luciole (Firefly Algorithm)
$LCA$ :	Algorithme du Ligue des Champions (League Championship Algorithm LCA)
$GEM$ :	Méthode d'explosion de grenade Grenade Explosion Method
$GSA$ :	Algorithme de Recherche Gravitationnelle (Gravitational Search Algorithm)

<i>WDO</i> :	Optimisation de Mouvement de Vent (Wind Driven Optimization)
<i>BA</i> :	Algorithme de Chauves-souris (Bat Algorithm)
<i>CS</i> :	Recherche Cuckoos (Cuckoo Search)
<i>GA-MPC</i> :	Algorithme Génétique avec un Processus de Croisement Multi-Parent (GA- Multi Parent Crossover)
<i>TLBO</i> :	Algorithme d'Optimisation Basé sur le processus Enseignement-Apprentissage (Teaching-Learning-Based Optimization)
<i>WCA</i> :	Algorithme du Cycle de l'Eau (Water Cycle Algorithm)
<i>DSA</i> :	Algorithme de Recherche Différentielle (Differential Search Algorithm)
<i>MBA</i> :	Algorithme d'Explosion des Mines (Mine Blast Algorithm)
<i>BSA</i> :	Algorithme de Recherche par Retour en Arrière (Backtracking Search Algorithm)
<i>BHBO</i> :	Optimisation Basée sur les Trous Noirs (Black Hole Based Optimization)
<i>GWO</i> :	Optimisation des Loups Gris (Grey Wolf Optimization)
<i>CSO</i> :	Optimisation de l'Essaim de Poulet (Chicken Swarm Optimization)
<i>JAYA</i> :	JAYA
<i>CBO</i> :	(Optimisation de Collision des Corps (Colliding Bodies Optimization)
<i>FWA</i> :	Algorithme de Feux d'Artifice (Fireworks Algorithm)
<i>MVPA</i> :	Algorithme du Meilleur Joueur (The Most Valuable Player Algorithm)

# **Introduction Générale**

## **Introduction Générale**

Devant la demande croissant en énergie électrique, les réseaux électriques, clé des systèmes électriques, assurant la relation entre la production et les usagers finaux ne cessent d'évoluer en taille ainsi qu'en complexité. Sur le plan gestion, l'exploitation des réseaux devient un défi technique, économique et climatique à cause des contraintes physiques et sécuritaires à respecter pour éviter leur effondrement. Selon les capacités de production, la demande en énergie doit être couverte en assurant sa stabilité avec un coût moins. Grace à l'évolution rapide des calculateurs et le développement incessant des méthodes d'optimisation notamment celles basées sur l'intelligence artificielles rendent la réalisation d'une gestion optimale des systèmes électriques un sujet très maitrisable. Il s'agit en fait du calcul de l'écoulement de puissance optimal (OPF) cadre dans lequel entre notre sujet de thèse.

Le problème d'écoulement de puissance consiste à déterminer un plan de fonctionnement du réseau qui permet d'obtenir les valeurs optimales des variables de commande pour des valeurs données de la charge en minimisant, par exemple, le coût de production et de l'émission de gaz polluant tout en respectant des contraintes physiques et sécuritaires.

Introduit par Carpentier en 1962, la résolution d'un problème de l'OPF est réduite, généralement, à la minimisation d'une fonction objectif prédéfinie toute en respectant les limites de fonctionnement du système (contraintes physiques et opérationnelles) [1].

Les méthodes ou algorithmes utilisés pour résoudre les problèmes de l'OPF sont très divers allant des algorithmes conventionnels au métheuristiques modernes.

La plupart des algorithmes conventionnels sont déterministes où certains utilisent les informations de gradient d'où vient l'appellation d'algorithmes à base de gradient (ex. Algorithme de Newton). Ces algorithmes fonctionnent extrêmement bien pour les problèmes continus lisses par contre ils ne sont pas bien adaptés pour les problèmes discontinus [2].

Les méthodes ou algorithmes métheuristiques, inspirés par la nature, deviennent très populaires dans les applications d'optimisation grâce à l'imitation des processus les plus réussis dans la nature y compris les systèmes biologiques et les processus physiques et chimiques. Pour la plupart de ces algorithmes, on connaît leurs composants fondamentaux. Savoir comment ces composants interagissent pour atteindre l'efficacité reste en partie un mystère ce qui inspire des études plus actives.

Parmi les algorithmes d'optimisation existantes, et pour des raisons économiques et opérationnelles, il y a des algorithmes qui ne garantissent pas l'atteinte du meilleur optimum possible.

Pour cela, et afin d'identifier les méthodes les plus adéquates pour résoudre les problèmes de l'OPF, on procède dans ce travail à une étude comparative d'un ensemble de méthodes d'optimisation métaheuristiques. De telle comparaison représente une originalité dans le domaine de l'étude des problèmes de l'OPF, notamment les méthodes de comparaison appliquées.

Pour effectuer une étude comparative fiable du point de vue quantitative, trente (30) algorithmes métaheuristiques seront considérés et appliqués pour résoudre le problème de l'OPF pour différents types de réseaux, objectifs à minimiser et contraintes à prendre en compte. Du point de vue qualitative, et afin d'aboutir à des résultats valables, deux types de comparaison seront effectuées ; paramétrique et non paramétriques.

En ce qui concerne les algorithmes d'optimisation à appliquer, il s'agit du GA, SA, PSO, DE, ACO, HS, EM, ABC, ICA, BBO, FA, LCA, GEM, GSA, WDO, BA, CS, TLBO, GA-MPC, WCA, DSA, MBA, BSA, BHBO, GWO, CSO, JAYA, CBO, FWA et le MVPA (voir Nomenclature).

Pour estimer l'effet de la taille et les spécifications du réseau sur les résultats des algorithmes clé de la décision finale concernant les meilleurs algorithmes, quatre réseaux différents seront traités en l'occurrence le réseau IEEE30, IEEE57, IEEE 118 et le réseau Algérien réel 59 jeux de barres.

Les cas de fonctions objectif à minimiser au nombre de neuf concerneront le coût de carburant, la déviation du profil de la tension, perfectionnement de la stabilité de la tension, les émissions du gaz polluant, le coût de carburant avec la considération de plusieurs types de combustibles, le coût de carburant avec l'introduction de l'effet d'ouverture des vannes, le coût de carburant avec la considération de plusieurs types de combustibles et l'effet d'ouverture des vannes, le coût avec introduction des dispositifs FACTS, et finalement la prise en compte des contraintes de sécurité (SCOPF) seront traités.

Les approches de comparaison paramétriques à appliquer sont celles basées sur le Minimum Global, la Moyenne et la Déviation standard. Pour la méthode non paramétrique qui tient en compte, en plus des résultats finaux, la vitesse de convergence des algorithmes, on appliquera les approches basées sur le Test de Signe Multiple, le Test de Friedman et les Profil de performance.

Pour une compréhension simple de la problématique traitée, les objectifs visés, la méthodologie d'étude suivie et les résultats obtenus, notre mémoire est organisé autour de cinq chapitres.

Le premier chapitre présente un revue sur le problème de l'OPF, nous exposons les diverses formulations classiques et récentes de l'OPF et les différentes méthodes de résolution. Il s'agit d'une discussion complète des techniques d'optimisation déterministes et métaheuristiques appliquées à l'OPF en mettant l'accent sur les caractéristiques, les avantages et les inconvénients de chacune.

Dans le deuxième chapitre, et pour familiariser le lecteur avec les problèmes de l'écoulement de puissance optimal, on définit tout d'abord le problème de l'écoulement de puissance et on expose les différentes équations et modèles qui le régissent. Ensuite, on présente la formulation du problème de l'OPF en définissant les variables d'état et de contrôle, les différentes contraintes d'égalité et d'inégalité, les limites de sécurité, le traitement de variables discrètes, la fonction de pénalité et les différentes fonctions objectif.

On parle aussi de la formulation de l'OPF avec introduction des dispositifs FACTS. On présente la modélisation des trois types considérés en l'occurrence les compensateurs statiques de l'énergie réactive, les compensateurs séries contrôlés par thyristors TCSC et le transformateur déphaseur commandé par thyristor(TCPST). A la fin de ce chapitre, on fait un exposé sur l'écoulement de puissance optimal avec les contraintes de sécurité où on présente les modèles liés aux deux modes préventif et correctif.

Pour le troisième chapitre, il s'agit d'une revue détaillée sur les méthodes métaheuristiques d'optimisation adoptées et appliquées dans ce travail (30 méthodes). Après une définition du principe de base de ce type de méthodes, on décrit en détail ces méthodes en précisant les particularités de chacune, ces applications antérieures et l'algorithme permettant sa mise en œuvre.

Dans le quatrième chapitre, on présente les méthodes appliquées pour effectuer la comparaison entre les algorithmes utilisés pour résoudre le problème de l'OPF. Après une définition du principe des comparaisons paramétrique et non paramétrique, on expose les différentes approches liées à chacune. Il s'agit d'une comparaison entre les algorithmes sur la base du minimum global, de la moyenne, du maximum et de la déviation standard pour la méthode paramétrique et le test de signe, test de signe multiple, test de Wilcoxon, test de Friedman, test de la Page et Profil de performance pour la méthode non paramétrique.

Dans le cinquième et dernier chapitre qui représente notre contribution réelle dans le domaine du calcul des problèmes de l'OPF, on présente les conditions de calcul adoptées pour appliquer les algorithmes d'optimisation adoptés, les critères de convergence choisis et les résultats de chacun en termes d'optimum obtenu, vitesse de convergence et stabilité. A la fin de ce chapitre, on présente les résultats obtenus par chaque méthode de comparaison, la

validité de ces résultats ainsi que la décision finale concernant le ou les algorithmes les plus adéquats pour résoudre les problèmes de l'OPF.

A la fin, notre mémoire sera couronné par une conclusion générale dans laquelle, on rappelle les objectifs de notre travail, la méthodologie suivie, les conditions de calcul et les hypothèses introduites et analyser les résultats obtenus et exposer les perspectives de ce travail.

# **CHAPITRE I**

## **Revue sur les méthodes d'optimisation du problème de l'écoulement de puissance optimal OPF**

## **I. Revue sur les méthodes d'optimisation du problème de l'écoulement de puissance optimal**

L'écoulement de puissance optimal (OPF) est un des problèmes d'optimisation non-linéaires les plus étudiés. L'OPF cherche, généralement, à optimiser la production d'énergie électrique et sa transmission dans les réseaux de distribution soumis aux contraintes du système et aux limites de contrôle. Il existe une très grande variété de formulations de l'OPF ainsi que leurs méthodes de résolution. La nature de l'OPF ne cesse d'évoluer en raison de la modernité des marchés de l'électricité et l'intégration des ressources renouvelables.

Dans cette revue, nous exposons les diverses formulations classiques et récentes de l'OPF et les différentes méthodes de résolution. Il s'agit d'une discussion complète des techniques d'optimisation spécifiques appliquées à l'OPF en mettant l'accent sur les caractéristiques, les avantages et les inconvénients de chacune.

### **I.1. Formulation du problème de l'écoulement de puissance optimal (OPF)**

Proposé par Carpentier en 1979, la résolution de l'OPF vise à optimiser une fonction objectif sélectionnée via un ajustement optimal des variables de contrôle du réseau électrique tout en satisfaisant diverses contraintes d'égalités et d'inégalités [1].

Les principales objectifs de l'OPF sont :

1. Minimisation de coût de carburant,
2. Minimisation La déviation du profil de tension,
3. Minimisation de l'indice de stabilité,
4. Minimisation des émissions du gaz,
5. Minimisation de coût de carburant avec la considération de plusieurs types de combustibles,
6. Minimisation de coût de carburant avec l'introduction de l'effet d'ouverture des vannes,
7. Minimisation de coût de carburant avec la considération de plusieurs types de combustibles et l'effet d'ouverture des vannes,
8. Minimisation de coût de carburant avec l'introduction des dispositifs FACTS,
9. Minimisation du coût de carburant avec prise en compte des contraintes de sécurité (SCOPF),

## **I.2. Méthodes de résolution du problème de l'écoulement de puissance optimal**

Pour résoudre les problèmes de l'OPF, plusieurs techniques d'optimisation peuvent être appliquées.

Ces techniques sont classées en deux groupes, déterministes et métaheuristiques.

### **I.2.1. Méthodes d'optimisation déterministes**

Les méthodes d'optimisation dites déterministe ou classique sont généralement basées sur les dérivées partielles de premier et deuxième ordre des fonctions à minimiser, une revue sur ces dernier est détaillé dans [3], ainsi certaines méthodes déterministes ont montrés une excellence convergence dans le traitement de plusieurs problèmes de l'OPF. Dans cette classe, on trouve principalement :

#### **I.2.1.1. La méthode du Gradient**

La méthode du gradient est l'une des premières tentatives pour résoudre les problèmes pratiques de l'OPF [3]. Elle est basée sur l'utilisation du vecteur dérivé du premier ordre de la fonction objectif d'une optimisation non linéaire pour déterminer des directions améliorées pour la résolution par étapes itératives [3]. La méthode du gradient est fiable, facile à mettre en œuvre et convergente pour des fonctions bien déterminées mais elle est lente comparativement à celles d'ordre supérieur.

Dans Dommel et Tinney [4], ont appliqué une méthode du gradient au problème de l'OPF en utilisant des techniques de pénalité pour appliquer les limites sur les variables dépendantes et les contraintes fonctionnelles. Ce travail a été étendu par Alsac et Stott au problème de l'OPF avec contraintes de sécurité (SCOPF) en ajoutant des cas d'urgence prédéterminés [5].

#### **I.2.1.2. Méthode de Programmation Quadratique**

La programmation quadratique (PQ) est une forme particulière de programmation non linéaire dont la fonction objectif est quadratique et les contraintes sont linéaires [3].

J.A. Momoh [6] a présenté un modèle quadratique généralisé pour l'OPF. La construction de problème de l'OPF inclut les conditions de faisabilité, la convergence et l'optimalité.

L'algorithme utilise la sensibilité des fonctions objectifs avec des ajustements optimaux des contraintes ce qui donne une solution optimale globale. Le coût de calcul en termes d'espace mémoire nécessaire et temps d'exécution requis a été réduit.

G. F. Reid et les autres Dans [7], ont appliqué la méthode PQ pour un problème de l'OPF en utilisant la méthode de multiplicateur de Lagrange.

La méthode a montré sa capacité de gérer à la fois les contraintes d'égalité et d'inégalité, le temps de résolution est suffisamment petit pour permettre l'utilisation de la méthode pour la répartition en ligne à des intervalles de temps pratiques.

N. Grudinin [8] a présenté un modèle d'optimisation de la puissance réactive basé sur des méthodes de programmation quadratique successive (SQP). Par rapport aux méthodes classiques, les méthodes SQP fournissent une optimisation plus rapide et fiable.

### **I.2.1.3. Les Méthodes de Points Intérieurs**

Les méthodes de points intérieurs (MPI) sont une famille d'algorithmes utilisée pour la résolution linéaire et non linéaire des problèmes d'optimisation qui contraignent la recherche à la région faisable en introduisant des termes de barrière à la fonction objectif [3].

Sergio Granville [9], a appliqué la MPI pour résoudre un problème de répartition optimale de la puissance réactive pour de grands réseaux électriques, la méthode a convergé après 40 itérations à un temps de calcul (CPU) de 398,9 secondes. Son avantage majeur réside dans le fait que le nombre d'itérations n'est pas très sensible à la taille du réseau ou au nombre de variables de contrôle.

De nouvelles versions des méthodes de points intérieurs appliquées à l'écoulement optimal de puissance sont présentées dans [10,11], dans leur travail et afin de résoudre les problèmes d'optimisation à travers la MPI par programmation non linéaire, on introduit un paramètre de perturbation dans les conditions complémentaires.

Dans [12], un nouvel algorithme de point intérieur de programmation non linéaire a été appliqué pour résoudre un problème de l'OPF, les auteurs étendent cet algorithme à l'écoulement de puissance classique (OPF) et un problème approximatif à l'OPF. Par rapport aux données classiques de la structure de Newton, le nombre des données utilisées est réduit à la moitié et le CPU est réduit d'environ 15% pour des systèmes à grande échelle.

L'algorithme proposé comprend quatre types des fonctions objectifs et deux structures de données différentes.

### **I.2.2. Méthodes d'optimisation métaheuristiques**

Les algorithmes métaheuristiques sont des méthodes non déterministe, de fait que leur processus d'optimisation n'est pas liée au forme de la fonction objectif et leur dérivée, et plutôt il est heuristique c'est à dire basé sur l'essai à plusieurs tentatives en améliorant la résultat en se basant sur des processus inspiré des phénomène naturel ou physique.

Une présentation des méthodes d'optimisation méta heuristique les plus reconnus depuis leur première création par [13] sera détaillée dans cette section.

### **I.2.2.1. Algorithmes Génétiques**

Les algorithmes génétiques (GA) fonctionnent sur la base du codage binaire de la chaîne des paramètres des variables constituant la fonction objectif. Chaque chaîne est prise comme un chromosome qui décrit une solution au problème [13].

Un simple algorithme génétique est une procédure itérative. Au cours de chaque itération (génération) trois opérations génétiques (sélection, croisement et mutation) sont effectuées pour générer de nouvelles populations (progéniture).

La méthode AG a été utilisée par Bakritzs [14], pour résoudre les problèmes de répartition économique, les mérites de la méthode résident dans leur habilité pour résoudre des problèmes non convexes et le codage efficace pour travailler sur des machines parallèles. Elle a aussi montré sa supériorité à la programmation dynamique conformément à la performance observée pour le problème de répartition économique.

Dans [15], M.Younes et les autres ont présenté un GA utilisé pour résoudre un problème de l'OPF en tenant compte des pertes de puissance actives et réactives. La méthode utilise le GA pour obtenir une proche de la solution globale et le paquet de fichiers MATLAB pour résoudre le l'écoulement de puissance. La validation de la méthode sur un système IEEE 57 modifié a montré que l'approche hybride offre une bonne solution.

M. Sailaja et les autres en [16], ont appliqué l'algorithme génétique pour un problème de l'OPF formulé comme un problème d'optimisation multi-objectif, plusieurs problèmes d'optimisation ont été traités à savoir, la minimisation simultanée des pertes active et réactive avec le coût du carburant et l'indice de stabilité de la tension.

### **I.2.2.2. Méthode d'Optimisation par Essaim de Particules**

L'optimisation par essaim de particules (PSO) est une optimisation métaheuristiques introduite par Kennedy et Eberhart [17]. Le PSO est basé sur les processus qui se produisent naturellement dans les colonies socialement organisées telles que les troupes d'oiseaux et les poissons. La PSO exploite une population d'individus pour explorer des régions prometteuses au sein de l'espace de recherche.

Dans la procédure de recherche, chaque individu (particule) se déplace dans l'espace de décision au cours du temps et change sa position en fonction de sa propre meilleure expérience et la meilleure particule actuelle [18]. Le PSO est capable d'évoluer vers un optimum global avec une vitesse aléatoire par son mécanisme de mémoire [18].

Par rapport à d'autres méthodes d'optimisation, la PSO a des taux de convergence et stabilité supérieurs pour une grande partie de problèmes d'optimisation difficiles [19]. Cependant, comme avec de nombreuses approches heuristiques, un inconvénient principal de la PSO traditionnelle est sa convergence prématurée lorsque les paramètres ne sont pas choisis correctement, surtout en traitant des problèmes avec beaucoup d'optimums locaux [20].

Dans [20], Wang a démontré la faisabilité d'un algorithme PSO modifié où chaque particule obtient des informations non seulement d'elle-même et du meilleur du groupe mais aussi des autres membres du groupe. Les auteurs affirment que cette technique accélère la convergence vers un optimum global.

Dans [21], Swapur a utilisé la PSO pour résoudre un problème de l'OPF avec des variables continues et discrètes en tenant compte à la fois des états normaux et des états de contingence. Il a démontré la vitesse de l'algorithme en termes de temps de calculs de l'écoulement de puissance exécutés ainsi que la supériorité des résultats par rapport aux GA.

Kim et les autres [22], ont implémenté avec succès un algorithme PSO parallèle. De telle approche a réduit le temps de calcul par rapport aux algorithmes séquentiels.

Dans [23], les auteurs ont présenté un algorithme PSO utilisé pour minimiser le coût total du carburant tout en considérant les limites de puissance active et réactive des générateurs, les tensions des jeux de barre charge et les limites de puissance des lignes.

L'idée clé de leur méthode PSO modifiée est d'exploiter les informations contenues dans les pires expériences de chaque individu et au sein du groupe.

Ceci est en contraste avec les algorithmes PSO conventionnels, où les meilleures expériences seulement sont utilisées. Les résultats de simulation ont montré que leur PSO modifié surpasse le PSO classique en termes de convergence vers de meilleures solutions.

Zhang et Liu [24], Coath et les autres dans [25] ont rapporté l'application réussie de PSO à pour résoudre un problème de l'écoulement optimal de puissance réactive (ORPF), et de contrôle de tension incorporant des parcs éoliens.

Zhao [26], ont utilisé un algorithme similaire à celui de Wang. Le terme d'influence dans la formule d'actualisation de la vitesse pour chaque particule comprend l'information de plusieurs particules, par rapport à la meilleure particule dans le PSO classiques seulement.

Ils ont établi une actualisation adaptative de règle des paramètres qui a permis une preuve de convergence globale de leur PSO améliorée. Des comparaisons avec les GA, les algorithmes évolutionnaire et les PSO classiques ont montrés que leur amélioration de la PSO nécessite moins de temps de calcul pour obtenir de meilleures solutions.

Dans [27], Vlachogiannis et Lee ont proposés trois algorithmes PSO modifiés pour la puissance réactive et le contrôle de la tension, la comparaison avec un algorithme OPF basé sur la MPI, un algorithme PSO conventionnel a montré la performance des algorithmes proposés.

Li et les autres dans [28], ont proposé un algorithme adaptatif de PSO pour résoudre un problème de l'ORPF en introduisant le concept d'espèce dans la mesure de la diversité de la population. Les résultats de la simulation ont montré un taux de convergence rapide avec un calcul robuste.

Pouya et Lesani dans [29], ont mis en œuvre une méthode PSO modifiée pour l'approvisionnement optimal de la puissance réactive dans un marché d'électricité ouverte où ils ont rapporté la rapidité de leur méthode. Dans ce cas, le problème de gestion de l'énergie réactive a été formulé sous la forme d'un problème non linéaire, et la contrainte de stabilité de la tension a été implémentée comme une contrainte douce pour garantir la sécurité du système.

#### **I.2.2.3. La méthode du Recuit Simulé**

Proposé par Kirpatrick, le recuit simulé (RS ou SA) est une métaheuristique générique et probabiliste pour l'optimisation globale [30].

Hsiao et les autres [31], ont utilisé le RS pour résoudre les valeurs optimales des compensateurs réactifs pour le problème de planification des contingences tout en considérant les équations réelles et le bilan de puissance réactive. Lors des tests, les auteurs affirment que l'algorithme est adapté pour les réseaux électriques à grande échelle.

Wong et Fung [32], ont développé un algorithme général de répartition économique basé sur l'algorithme RS qui intègre les pertes de transmission en les approchant par une formule quadratique. Bien que les résultats des tests démontrent que l'algorithme est capable de trouver une solution optimale globale ou quasi globale, son temps de calcul est élevé.

La technique RS a été utilisée pour résoudre un problème de répartition économique où les contraintes de transmission ont été modélisées sous forme polaire où un temps de calcul long a été rapporté [33]. Les approches du RS pure utilisées pour les problèmes de l'OPF ont été remplacées par des méthodes hybrides au cours de la dernière décennie.

#### **I.2.2.4. La méthode de Recherche Tabou**

La méthode de recherche Tabou (RT) est une procédure d'amélioration itérative introduite par Glover [34]. Elle a été appliquée pour résoudre divers problèmes d'optimisation des

systèmes électrique [35]. Au début des années 90, la méthode est jugée prometteuse avec d'autres algorithmes de recherches heuristiques telles que les AG.

Une méthode parallèle basée sur la RT a été utilisée pour le contrôle de la tension et de la puissance réactive. Le schéma parallèle améliore la qualité de la solution en calculant le voisinage de manière parallèle en utilisant deux cœurs avec deux longueurs de tabou (un noyau par longueur de tabou). Cette méthode est jugée de très efficace par rapport à la RT, le RS et l'AG classiques [36].

Abido [37] a présenté une Approche efficace et fiable basée sur la RT pour définir les variables de contrôle optimales du problème de l'OPF général en examinant diverses fonctions et contraintes objectifs.

Dans [38], un algorithme RT efficace a été développé pour résoudre le problème de l'OPF en tenant compte à la fois de la puissance active et de la puissance réactive. Pour évaluer l'utilité ainsi que les avantages de l'algorithme, les auteurs ont comparé leurs résultats avec ceux obtenus par la programmation quadratique.

Les contraintes de l'écoulement de puissance sont données sous forme polaire. Les émissions en termes de tonnes totales de polluants sont modélisées par des fonctions quadratiques et exponentielles. Les émissions totales sont combinées via une pondération avec le coût du carburant en une fonction objectif à minimiser. Leur RT réactive améliore le processus de recherche par rapport à la méthode standard de RT par implémentation d'une modification adaptative de la longueur de la liste Tabou.

Les auteurs ont signalé que la RT réactive peut donner de meilleures solutions tout en réduisant le temps de calcul par rapport à l'algorithme standard.

Altun et Yalcinoz [39], ont étudié le problème de répartition économique avec une fonction quadratique de perte de transmission de puissance dans les variables de production d'énergie. Quatre méthodes ont été discutées et comparées y compris la RT et les AG qui ont donné de bonnes solutions. Les auteurs n'ont pas incorporé l'ensemble des contraintes OPF.

Un algorithme RT a été appliqué pour résoudre un problème de répartition économique d'énergie avec contrainte de sécurité [40]. De telle formulation a incorporé les contraintes de l'écoulement de puissance de base et de contingence. Les auteurs ont signalé que l'algorithme était suffisamment efficace et fiable pour être appliqué aux systèmes à l'échelle des services publics.

### **I.2.2.5. Algorithme d'optimisation des Colonies de Fourmis (ACO)**

La technique ACO est inspirée de la stratégie des fourmis dans leur recherche de la nourriture par la communication de phéromone pour faire leur chemin [41]. Elle est appliquée à un certain nombre de problèmes OPF avec plusieurs fonctions objectifs.

Swarup [42], a appliqué l'ACO aux problèmes de répartition des charges et de basculement des générateurs.

Kalil [43], ont proposé des techniques ACO pour un problème de répartition optimale de la puissance réactive afin d'améliorer les conditions de stabilité de tension et de réduire les pertes de transmission tout en assurant la surveillance du profil de tension. Les auteurs indiquent que l'ACO a surperformé plusieurs méthodes déterministes en termes de qualité de la solution et de temps de calcul.

Allaoua et Laoufi [44], ont utilisé l'ACO pour minimiser le coût total du combustible des unités de production thermique tout en conservant un niveau acceptable de performance du système en termes de limites sur les sorties de puissance réelle et réactive du générateur, les tensions, les condensateurs, réglages de prise de transformateur et flux de puissance sur les lignes de transmission.

Leur méthode est remarquable car elle permet de minimiser le temps de calcul en décomposant les contraintes en ensembles de contraintes actifs et passifs. Les contraintes actives sont utilisées pour calculer l'ensemble de solutions optimales en utilisant l'ACO tandis que les contraintes passives sont appliquées par un algorithme d'écoulement de puissance de Newton-Raphson.

Les résultats de la simulation ont montré que leur méthode ACO surpasse les algorithmes évolutionnaire comme le AG en termes de rapidité de calcul et qualité de solution.

Gasbaoui et Allaoua [45], ont également utilisé l'ACO pour résoudre un problème OPF combinatoire avec plusieurs objectifs, y compris la minimisation du coût du carburant, l'amélioration du profil de tension et l'amélioration de la stabilité de la tension. Les auteurs rapportent que l'approche ACO a obtenu de meilleurs résultats mieux que les techniques classiques et les AG.

### **I.2.2.6. Algorithme de Recherche Bactérienne**

L'algorithme de recherche bactérienne (ARB) est inspiré des modèles exposés par les bactéries tout en recherchant de la nourriture. Bien qu'elle est relativement récente, la ARB a suscité de l'intérêt chez la communauté travaillant sur les systèmes d'énergie électrique.

Dans [46] et [47], l'ARB a été appliquée pour optimiser les pertes de puissance active et les limites de stabilité de tension d'un réseau électrique maillé, le problème a été formulé comme un problème de l'OPF multi-objectif avec le contrôleur de flux de puissance unifiée, les résultats de simulation montrent que l'ARB est supérieur par rapport l'algorithme de point intérieur.

Li et les autres dans [48], ont développé un ARB avec une population variable pour le problème OPF. Elle a été utilisée pour la première fois pour explorer les mécanismes de la chimio taxie bactérienne et la prolifération.

Un inconvénient de l'ARB est qu'il n'est pas toujours capable de suivre efficacement la solution optimale dans des environnements dynamiques [49]. Afin de remédier à cette lacune, Tang et les autres [50], ont présenté une approche appelée ARB dynamique (ARBD) pour résoudre le problème de l'OPF avec les charges dynamiques.

Les variations des charges de puissance et de la topologie du système ont été simulées comme des changements environnementaux réguliers et irréguliers. Les résultats ont démontré l'adaptabilité de l'ARBD aux divers changements environnementaux. Les auteurs ont signalé que l'ARBD présente d'avantages par rapport aux méthodes ARB traditionnelle et aux autres méthodes.

Tang et les autres [51], ont ensuite appliqué un ARBD pour minimiser le coût du carburant du système d'énergie électrique avec une OPF intégrée dans un environnement à charges dynamiquement. Les résultats de simulation montrent que par rapport aux méthodes ARB et PSO, les ARBD peuvent s'adapter plus rapidement pour charger les changements et suivre plus étroitement le coût global du carburant.

#### **I.2.2.7. Algorithme d'Evolution Différentielle**

L'évolution différentielle (ED) est un algorithme de recherche stochastique directe proposé par Storn et Price pour résoudre des problèmes d'optimisation sur un domaine continu [52-53]. L'ED combine des opérateurs arithmétiques simples avec les opérateurs évolutifs classiques de croisement, de mutation et de sélection pour évoluer d'une population de départ générée aléatoirement à une solution finale.

Le mécanisme de mutation différentielle est l'élément clé qui distingue le DE des autres techniques basées sur la population. Pour générer des vecteurs de paramètres d'essai, le DE ajoute la différence pondérée entre deux vecteurs de population à un troisième vecteur. Aucune distribution de probabilités distincte n'est requise [53].

La compétition un-à-un de la progéniture fait de l'ED beaucoup plus rapide en convergence que les autres algorithmes évolutionnaire. Malheureusement, cette convergence plus rapide aboutit à une probabilité plus élevée de converger vers un optimum local plutôt que global. Afin de surmonter cet inconvénient et pour éviter d'avoir une population importante, Chiou et Wang [54], ont ajouté deux phases à l'ED, la phase accélérée et la phase migrante.

L'ED a été appliqué à plusieurs problèmes d'ingénierie dans différents domaines, y compris l'OPF [55-56]. Pour l'OPF, l'ED a été appliqué avec des courbes de coûts complexes ou des caractéristiques de générateur non conventionnel.

Dans [57], les auteurs ont mis en œuvre des algorithmes ED pour résoudre les problèmes d'acheminement économique avec les contraintes et les pertes de ligne de transmission. Les algorithmes prennent en compte les caractéristiques non linéaires du générateur telles que les limites de taux de rampe et les zones d'exploitation interdites.

Vaisakh et Srinivas [58] ont appliqué l'ED à l'OPF avec des caractéristiques de coût conventionnelles et non conventionnelles.

Sayah et Zehar [59] ont développé des algorithmes ED pour résoudre les OPF avec des courbes de coût de carburant de générateur non lisse et non-convexe. Les auteurs ont suggéré des modifications efficaces dans la règle de mutation, améliorant le taux de convergence tout en améliorant la qualité de la solution. Les auteurs ont également montré que leur algorithme ED modifié, perfectionne les algorithmes ED classiques en vitesse de convergence globale et obtenait des résultats similaires par rapport aux algorithmes évolutionnaire.

L'ED est également adapté aux problèmes OPF qui incluent des contraintes de stabilité transitoire ou de contrôles complexes.

Bakare [60] a appliqué l'ED au réseau électrique nigérian pour optimiser les profils de tension et les pertes du système via le contrôle de la puissance réactive du système. L'approche a permis de réduire considérablement les pertes de puissance réelles tout en maintenant les profils de tension dans les limites acceptables.

Basu [61] a utilisé un algorithme ED pour minimiser le coût du carburant de la génératrice dans le contrôle optimal du flux de puissance avec des dispositifs de systèmes de transmission flexibles (FACTS), y compris des condensateurs série et des déphaseurs à thyristors. Les résultats des tests ont montré que l'approche ED proposée peut obtenir de meilleures solutions nécessitant moins de temps CPU que les algorithmes génétique.

#### **I.2.2.8. Méthode de la Colonie d'Abeilles**

Présenté par Sumpavakup, l'algorithme de la colonie d'Abeilles (ABC) a été appliqué aux problèmes de l'OPF [62]. Vérifié par des tests sur des systèmes IEEE-14 et IEEE-30 et comparé à d'autres techniques d'intelligence artificielle (algorithmes génétiques et essaim des particules), l'algorithme montre une convergence relativement rapide vers la meilleure solution.

Khorsandi [63] a présenté un algorithme à base floue modifiant celui de colonie d'abeilles artificielles (MABC) pour résoudre le problème de l'OPF discret ayant à la fois des variables discrètes et continues en tenant compte des effets de l'ouverture des vannes. Le problème OPF est formulé en tant qu'un problème non linéaire multi-objectif mixte-entier où les paramètres optimaux des variables de contrôle recherchés lors de la minimisation simultanée sont le coût total du carburant des unités thermiques, émission totale, pertes de puissance active totale et la déviation tension. Testé sur des problèmes d'OPF IEEE 30 et IEEE 118 et comparé aux méthodes conventionnelles, l'algorithme présente plus d'efficacité et rapidité dans l'exploration de recherche globale.

Dans le but d'améliorer les performances de résolution en termes de mécanisme de recherche et vitesse de convergence de la technique de colonie artificielle d'abeilles, Afandi et Miyauchi [64] ont développé un algorithme dont l'efficacité a été approuvée par un test sur un système IEEE-62 et comparaison des résultats à ceux obtenus par des autres technique modifié de l'algorithme ABC. Leur algorithme présente l'avantage de réduire le nombre d'itérations.

#### **I.2.2.9. Algorithme d'optimisation basé sur l'Electromagnétisme**

L'algorithme d'optimisation basé sur l'Electromagnétisme nommé (EM) qui simule la théorie d'électromagnétisme a été utilisé par plusieurs auteurs pour résoudre les problèmes de l'OPF.

Boucekara et les autres [65], ont développé une version améliorée de l'algorithme EMA afin de trouver la solution optimale pour le problème de l'écoulement de puissance optimal dans un système électrique. Pour montrer l'efficacité de la méthode développée, il a été testé sur des systèmes IEEE 30-bus et IEEE 57-bus pour sept (7) objectifs différents reflétant la minimisation des coûts de carburant avec des générateurs pouvant avoir des caractéristiques de coût de carburant convexe ou non convexe, amélioration du profil de tension, amélioration de la stabilité de la tension et minimisation des pertes de transmission de puissance active et réactive. La comparaison des résultats obtenus avec ceux obtenus par d'autres algorithmes a

montré qu'elle représente une approche assez prometteuse pour résoudre les problèmes d'OPF.

Jeddi B et les autres [68], ont présenté un algorithme de type électromagnétisme multi-objectif (EMA) basé sur le concept de dominance de Pareto et la stratégie d'archivage externe. Tenant compte de sa mauvaise capacité d'exploration de l'espace de recherche, ils ont proposé un EMA amélioré avec un processus de recherche locale modifié.

La faisabilité et l'efficacité de la méthode proposée sont testées sur les systèmes IEEE 30 et IEEE 118 pour différents fonctions objectifs et comparées à l'EMA originale et d'autres méthodes heuristiques rapportées dans la littérature.

#### **I.2.2.10. Algorithme d'Optimisation Basé sur le Procès Enseignement-Apprentissage**

Shabanpour [67] ont introduit un algorithme d'optimisation enseignement-apprentissage (TLBO) pour résoudre le problème de l'OPF multi-objectif en considérant le coût total du carburant et l'émission totale des unités. La phase modifiée de l'algorithme d'optimisation utilise une stratégie de mutation auto-adaptative. De plus, une technique de logique flou a été proposée pour éviter une taille de référentiel extrêmement importante en plus d'une sélection de population intelligente pour l'itération suivante.

Les performances de la méthode ont été vérifiées par comparaison des résultats obtenus avec d'autres algorithmes pour les problèmes IEEE 30-Bus et IEEE 57-Bus.

Dans le même contexte, Ghasemi [68] ont travaillé à améliorer l'algorithme TLBO en utilisant la stratégie de mutation de Lévy pour les paramètres optimaux des variables de contrôle des problèmes d'OPF. Les performances de leur approche sont étudiées et évaluées par résolution des problèmes de test standard IEEE 30-bus et IEEE 57-bus avec différentes fonctions objectifs. La confrontation de leurs résultats avec ceux obtenus par d'autres méthodes comparaison a montré une grande efficacité de leur approche pour résoudre les problèmes de l'OPF.

Dans [69], les auteurs ont proposé une TLBO modifiée pour déterminer l'emplacement optimal et la taille des unités de génération distribuée dans les systèmes de distribution. Par souci de clarté, et sans perte de généralité, la fonction objectif considérée à minimiser représente les pertes totales de puissance électrique. Bien que le problème puisse être facilement configuré comme multi-objectif (d'autres fonctions objectifs peuvent être considérées en même temps), l'emplacement optimal des systèmes de génération distribuée ainsi que leurs tailles sont obtenus simultanément. Pour valider leur approche, ils ont comparé leurs résultats avec ceux obtenus par d'autres méthodes.

Mandala et les autres [70] ont résout un problème d'optimisation de la puissance réactive optimale (ORPD) multi-objectif en minimisant la perte de puissance réelle, l'écart de tension et l'indice de stabilité de la tension. Pour accélérer la vitesse de convergence et améliorer la qualité de la solution, le concept d'apprentissage par quasi-opposition (QOBL) est incorporé dans l'algorithme TLBO original.

Les approches TLBO et TLBO quasi-oppositionnelles (QOTLBO) sont mises en œuvre sur les problèmes de test IEEE 30-bus et IEEE-118 standards. Les résultats ont démontré que l'approche QOTLBO présente une supériorité en termes de qualité de la solution par rapport au TLBO original et d'autres techniques d'optimisation et confirment son potentiel à résoudre le problème ORPD.

Sultana et les autres [71], ont utilisé la TLBO pour trouver l'emplacement optimal des condensateurs dans les systèmes de distribution radiale afin de minimiser les pertes de puissance et le coût de l'énergie. Leur algorithme a été validé sur les systèmes de distribution radiale standard 22, 69, 85 et 141 bus en comparant leurs résultats avec ceux obtenus par l'AG, PSO et l'ED.

#### **I.2.2.11. Autres Méthodes d'Optimisation Metaheuristiques**

La méthode dite Algorithmes d'optimisation des Grenades multi objectif (GEM) a été appliqué pour résoudre un problème de l'écoulement de puissance optimal multi objectif [72]. Ce dernier est formulé d'une manière où on a supposé que le décideur peut avoir un objectif flou pour chacune des fonctions objectif. Les six objectifs considérés sont la minimisation du coût du combustible de production, l'amélioration du profil de tension, l'amélioration de la stabilité de la tension, la réduction des émissions et la minimisation des pertes de transmission actives et réactives. L'approche a été testée sur un système IEEE 30 bus où les résultats obtenus ont montré une grande efficacité.

La méthode d'explosion de grenades multi-objectifs (MO-GEM) est utilisée comme algorithme de résolution. Les fonctions objectifs minimisées sont le coût du carburant concurrentiel et les pertes. L'approche a été vérifiée sur un système de bus IEEE 30 [73].

Comme contribution personnelle, nous avons utilisé un algorithme BSA "Backtracking Optimisation Algorithme" pour résoudre un problème d'OPF pour 16 cas différents des problèmes IEEE 30-bus, IEEE 57-bus et IEEE-118. En plus du coût du combustible de production traditionnel, des options multi-carburantes, l'effet d'ouverture des vannes et d'autres complexités ont été considérés [74].

En outre, différents objectifs tels que l'amélioration du profil de tension, l'amélioration de la stabilité de la tension et la réduction des émissions ont été pris en compte. Nos résultats ont été comparés à ceux obtenus par d'autres algorithmes où le temps de calcul a été comparé par une nouvelle approche. De telle comparaison a mis en évidence l'efficacité de la méthode BSA pour résoudre différents problèmes d'OPF avec des fonctions objectifs compliquées et non lisses.

La technique BSA a été utilisée pour affecter les positions optimales des générateurs distribués (GD) le long de la distribution radiale des réseaux [75].

Dans [76], les auteurs ont appliqué l'algorithme de la ligue des champions (LCA) pour résoudre le problème de l'OPF du réseau algérien pour différentes fonctions objectifs. Les résultats obtenus ont montré une haute efficacité de la méthode par rapport à l'ABC, l'AG et l'ACO.

Les mêmes auteurs ont introduit le concept Pareto afin de résoudre le problème de l'OPF multi objectif en utilisant la méthode d'explosion de grenade (GEM). Aussi, ils ont utilisé le concept de la logique floue en supposant que le décideur peut avoir un objectif flou pour chacune des fonctions objectifs. L'efficacité de l'approche a été validée par le système IEEE-30 où les six objectifs sont la minimisation du coût du combustible de production, l'amélioration du profil de tension, l'amélioration de la stabilité de la tension, la réduction des émissions et la minimisation des pertes de transmission actives et réactives [76].

Dans [77], les auteurs ont combiné les deux algorithmes GEM et ABC pour résoudre les problèmes de l'OPF et cela pour de nombreux cas de fonctions objectifs tels que le coût total du carburant correspondant aux différentes courbes du coût (quadratique, quadratique avec effets des vannes et quadratique avec plusieurs types de combustibles).

Dans [78] et [79], l'algorithme d'optimisation du cycle de l'eau (WCA) a été appliqué pour un problème de l'OPF afin de minimiser le coût de combustible du réseau IEEE-57 bus et la résolution du problème de répartition de puissance réactive optimale pour le système IEEE 30 bus où il a montré une grande efficacité de calcul.

Dans [80], l'algorithme WCA a été appliqué pour résoudre le problème de répartition économique de la charge dynamique.

Afin d'éviter le problème des optimaux locaux et la convergence prématurée, Heidari et les autres [81], ont proposé dans une version améliorée de la WCA pour résoudre le problème de l'OPF lié aux systèmes IEEE 30, 57 et 118 bus.

L'algorithme basé sur la loi de Newton de la gravité et des interactions de masse appelé Algorithme de Recherche Gravitationnelle (GSA) a eu beaucoup d'attention dans la résolution

des problèmes de l'OPF [82-83]. Dans [84], la GSA a été utilisée pour un problème combiné d'économie et d'émission (CEED) dans les réseaux électriques.

Le concept FACTS (Flexible AC Transmission System) a été introduit dans le problème de l'OPF [85]. Un algorithme hybride PSO-GSA a été développé et utilisé dans [86] dont les performances ont été évaluées sur le système de test IEEE 30-bus standard avec différentes fonctions objectives.

Un autre algorithme appelé BBO a été rigoureusement étudié dans la résolution de problème d'OPF [87, 88, 89, 90, 91], les auteurs ont consolidé l'efficacité de l'approche pour la résolution des différents problèmes OPF ainsi leur supériorité par rapport à d'autres algorithmes tels que PSO, ABC, GA, FA, DE.

Afin d'améliorer les performances de l'algorithme BBO, une version adaptative codée réelle de BBO (ARC-BBO) a été présentée et implémentée dans la résolution de problème de l'OPF [92].

L'opérateur de perturbation pour améliorer la capacité d'exploration et d'exploitation a été introduit dans la BBO [95]. L'apprentissage par opposition est incorporé dans l'algorithme d'optimisation basé sur la biogéographie originale [95].

Une résolution de l'écoulement optimal de la puissance réactive avec l'incorporation des systèmes de transmission AC flexibles (FACTS) a été effectuée par l'algorithme BBO [96]. Un algorithme hybride BBO-DE a été appliqué pour résoudre le problème d'OPF dans le réseau algérien [97].

L'algorithme basé sur la recherche harmonique (HS) est un algorithme inspiré de la nature imitant l'improvisation des musiciens. Son application aux problèmes de l'OPF a été introduite par plusieurs auteurs [98], [99], [100].

Une version multi-objectif de l'algorithme HS a été développée et utilisée dans [100]. Le tri élitiste non dominé et la distance d'encombrement ont été utilisés pour trouver et gérer le front optimal de Pareto et un mécanisme à base floue a été utilisé pour sélectionner une solution de compromis de l'ensemble de Pareto.

Un algorithme de recherche harmonique multi-objectif « MOHS » a été proposé et testé sur le système de bus IEEE 30 avec des objectifs différents. Les résultats de la simulation montrent clairement que la méthode proposée est capable de générer des solutions optimales de Pareto pour le problème d'OPF [101].

L'algorithme d'optimisation des loups gris (GWO) inspiré du mécanisme des loups gris dans la chasse et des versions améliorées GWA et IGWA ont eu beaucoup d'attention dans la résolution des problèmes d'OPF [102, 103, 104, 105, 106, 107].

L'algorithme d'optimisation de l'essaim de poulet (CSO) a été appliqué pour résoudre le problème de la répartition économique optimal avec des zones de fonctionnement interdites en tenant compte des pertes dans le réseau [108]. Une nouvelle version de cet algorithme appelée CSO multi-étages a été proposée pour une optimisation globale [109]. La modification a réduit les étapes de l'algorithme CSO en éliminant les paramètres coqs, poules et poussins. La CSO multi-étage présente une grande efficacité par rapport à l'ancienne version pour la résolution des problèmes d'OPF.

Un algorithme dit des chauves-souris (BA) a été utilisé pour résoudre le problème de l'OPF lié aux systèmes IEEE 9 bus, 14 bus, 30 bus et 57 bus en minimisant le coût du combustible comme fonction objectif. Les résultats ont été comparés à ceux obtenus par les algorithmes PSO, HS. De telle comparaison a montré que cette méthode donne une réponse efficace et robuste de haute qualité [110].

Un algorithme de chauve-souris amélioré a été utilisé pour résoudre le problème de l'OPF multi-objectif (MOPF) basé sur la méthode pondérée en le formulant comme problème d'optimisation non linéaire à deux fonctions objectifs (coût et émission du carburant) et plusieurs contraintes [111]. Après avoir trouvé les solutions de Pareto par l'algorithme de chauve-souris, la théorie des ensembles flous est utilisée pour trouver la solution de compromis. Les performances de l'algorithme ont été vérifiées par résolution du problème de l'OPF multi objectif « MOPF » du système IEEE 57-bus.

L'algorithme de luciole (FA) est un algorithme métaheuristique introduit par Xin-She en 2009. Depuis son introduction, plusieurs auteurs ont étudié son application pour la résolution des problèmes de l'OPF [112, 113, 114, 115, 116, 117, 118, 119].

Dans [120], l'algorithme FA est utilisé pour la résolution d'un problème de l'OPF multi-objectif dans un système de puissance adopté avec l'énergie éolienne en minimisant le coût du carburant, les pertes de puissance active et la déviation de la tension. La fonction du coût du carburant est incorporée à la variabilité de la vitesse du vent en termes de sûreté et sous-estimation lors de l'estimation du coût de la génération d'énergie éolienne.

Les résultats ont été comparés avec ceux obtenus par les algorithmes de Cuckoo modifié et le PSO modifié.

Le problème SCOPF a été traité en utilisant l'algorithme FA dans [121] et [122]. Un Algorithme FA auto-adaptatif a été proposé pour identifier le placement optimal du compensateur à thyristors (TCSC) dans un réseau d'alimentation en minimisant comme fonction objectif les pertes de transmission [123].

L'Algorithme de la Concurrence des Impérialistes (ICA) est un des algorithmes d'optimisation évolutifs qui s'appuient sur le colonialisme en tant que phénomène sociopolitique.

Ranjbar [124], ont l'utilisé pour résoudre le problème d'OPF en comparant les résultats avec ceux obtenus par l'AG et le PSO. La technique a été utilisée pour résoudre le problème de l'OPF en minimisant les pertes de puissance et les coûts d'installation des dispositifs UPFC.

Un ICA modifié compétitif multi-objectif a été utilisé pour résoudre le problème de l'OPF en considérant le coût, l'émission du carburant, la déviation de la tension et les pertes de puissance [125].

Dans [126], un algorithme hybride ICA-TLBO a été proposé pour résoudre le problème d'OPF.

L'algorithme de recherche de coucou (CS) est un algorithme basé sur la population. Il est beaucoup utilisé pour l'étude des problèmes de l'OPF [127, 128, 129]. Ces performances ont été examinées et testées sur un système IEEE 14 bus en minimisant le coût de carburant comme fonction objectif.

Pour résoudre un problème de répartition de puissance réactive multi-objectif, un algorithme de recherche de coucou amélioré a été utilisé [130]. L'analyse modale du système est utilisée pour l'évaluation de la stabilité de la tension statique. La minimisation des pertes et la maximisation de la marge de stabilité de la tension ont été prises comme fonctions objectifs.

La planification optimale de l'énergie réactive et la minimisation de la perte de puissance active à l'aide de l'algorithme de recherche Coucou est étudié [131].

Une étude comparative entre la CSA et le PSO pour le problème d'optimisation dans un système de distribution actif déséquilibré a été effectué dans [132].

Au cours de la préparation de cette thèse, plusieurs algorithmes ont été originalement étudiés dans ce travail pour résoudre le problème de l'OPF notamment :

- ✓ Algorithme génétique avec un nouveau croisement multi parent (GA-MPC),
- ✓ Algorithme d'optimisation des trous noirs (BHBO),
- ✓ Algorithme de recherche en retour (BSA),
- ✓ Algorithmes d'optimisation des grenades (GEM),
- ✓ Algorithme de la ligue des champions(LCA),
- ✓ Optimisation de collision des corps (CBO),
- ✓ Algorithme d'optimisation JAYA,
- ✓ Algorithme d'optimisation de vent (WDO),

- ✓ Algorithme d'optimisation des mines (MBA),
- ✓ Algorithme d'optimisation des chauves-souris (ABA),
- ✓ Algorithme de concurrence impérialiste (ICA),
- ✓ Algorithme de Meilleur Joueur (MVPA)

## **Conclusion**

Afin de bien voir de l'intérêt des travaux de cette thèse, dans cette première partie nous avons présenté un état synthétique de domaine de l'OPF mettons l'axe sur les méthode de résolution classique et métaheuristiques.

Les méthodes d'optimisation classiques ou déterministe sont dépend fortement au forme de la fonction objectif qui doit être strictement convexe, continue et dérivable.

Les algorithmes métaheuristiques sont des méthodes non déterministes, de fait que leur processus d'optimisation n'est pas lié au forme de la fonction objectif et leur dérivée, leur principe est basé généralement sur une procédure stochastiques itératifs, qui progressent vers un optimum global, ils sont largement appliqués dans le traitement de problème de l'OPF, néanmoins il n'existe aucun algorithme meilleur pour tous les cas et les problèmes, et il n'y a pas de ligne de conduite convenue pour choisir un algorithme adéquat pour le traitement des différents problème de l'OPF efficacement.

**CHAPITRE II**

**FORMULATION DU PROBLEME**  
**DE L'ECOULEMENT DE**  
**PUISSANCE OPTIMAL (OPF)**

## **II. Formulation de l'écoulement de puissance optimal**

### **II.1. Introduction générale sur le problème de l'écoulement de puissance**

Afin de procéder à l'étude et la formulation de l'écoulement de puissance optimal, nous avons vu utile de présenter les notions et les formulations de bases qui concernent le problème de l'écoulement de puissance. Il s'agit de la définition des différentes parties constituant le système électrique, l'exposition de leur modélisation et les méthodes de résolution.

#### **II.1.1. Problème de l'écoulement de puissance**

L'objectif de la résolution d'un problème de l'écoulement de puissance est la détermination des tensions et des courants dans les jeux de barres ainsi que les transites des puissances active et réactive dans les lignes sous certaines conditions de fonctionnement [134].

L'étape fondamentale pour la résolution d'un problème de l'écoulement de puissance est la modélisation des éléments constituant le réseau électrique. Après le développement de ces modèles et la mise sous forme d'équations matricielles du problème, on procède à la résolution de ces équations et cela généralement par application des méthodes itératives (Gauss Seidel, Newton Raphson, Levenberg-Marquardt, ...).

Les grandeurs à déterminer dépendent du type de jeux de barre. Le tableau II.1 présente les paramètres connus et inconnus pour chaque type de Jeux de barre.

**Tableau II.1. Paramètres connus et inconnus liés à chaque type de jeux de barre**

Type de jeu de barre	Puissance active	Puissance réactive	Tension	Phase
Générateur	connue	inconnue	connue	inconnue
Charge	connue	connue	inconnue	inconnue
Equilibre	inconnue	inconnue	connue	connue

#### **II.1.2. Équations de l'écoulement de puissance**

Pour montrer les différentes étapes essentielles pour résoudre un problème d'écoulement de puissance de base, on considère le système réduit à deux jeux de barre de la figure suivante :

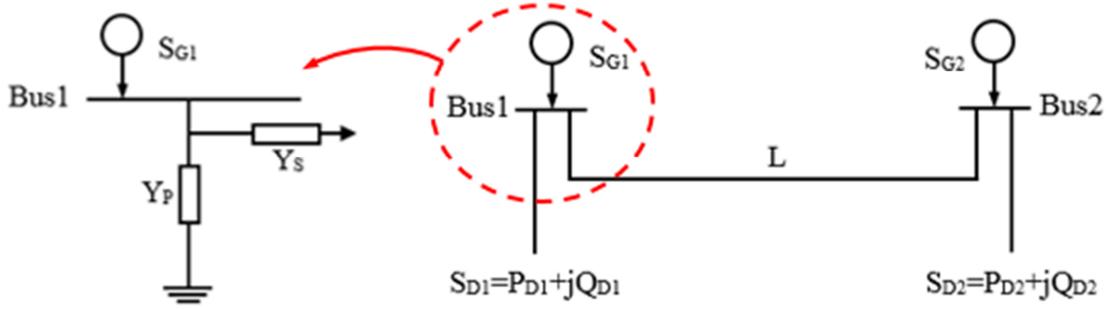


Fig. II.1. Système a deux jeux de barre [135]

On note par  $S_1$  et  $S_2$  les puissances apparentes des bus 1 et 2,  $S_{G1}$  et  $S_{G2}$  les puissances générées et  $S_{D1}$  et  $S_{D2}$  les puissances demandées on peut écrire [135] :

$$S_1 = S_{G1} - S_{D1} \quad (\text{II.1})$$

$$S_2 = S_{G2} - S_{D2} \quad (\text{II.2})$$

Au jeu de barre  $i$ , on a :

$$S_i = P_i + j Q_i = (P_{Gi} + j Q_{Gi}) - (P_{Di} + j Q_{Di}) = (P_{Gi} - P_{Di}) + j (Q_{Gi} - Q_{Di}) \quad (\text{II.3})$$

En fonction de la tension  $V_i$  et du courant  $I_i$  aux jeux de barre  $i$ , la puissance  $S_i^*$  est donnée par :  $S_i^* = V_i I_i^* = P_i - j Q_i$

Selon la figure II.1, on voit que le courant  $I_i$  à deux composantes,  $Y_p \times V_i$  et  $(V_i - V_j) \times Y_s$  tels que  $Y_p$  et  $Y_s$  sont respectivement les admittances parallèles et shunts de la ligne.

L'application des lois de Kirchhoff aux deux jeux de barre donne [136] :

$$I_1 = Y_p \times V_1 + Y_s \times (V_1 - V_2) = (Y_p + Y_s) \times V_1 - Y_s \times V_2 = Y_{11} V_1 + Y_{12} V_2 \quad (\text{II.4})$$

$$I_2 = Y_p \times V_2 + Y_s \times (V_2 - V_1) = (Y_p + Y_s) \times V_2 - Y_s \times V_1 = Y_{21} V_1 + Y_{22} V_2 \quad (\text{II.5})$$

Sous forme matricielle, (II.4) et (II.5) deviennent :

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (\text{II.6})$$

Pour un système à  $n$  jeux de barre, on peut écrire [136] :

$$I_i = \frac{S_i^*}{V_i^*} = Y_{i1} V_1 + Y_{i2} V_2 + \dots + Y_{ii} V_i + \dots + Y_{in} V_n \quad (\text{II.7})$$

$$Y_{ii} = \sum_{k=1, k \neq i}^n (Y_{pik} + Y_{sik}) \text{ et } Y_{ik} = Y_{ki} = -Y_{sik} \quad (\text{II.8})$$

Pour  $V_i = |V_i| \angle \delta_i$  et  $Y_{ij} = |Y_{ij}| \angle \theta_{ij}$ , on aura pour la puissance complexe:

$$S_i^* = P_i - jQ_i = V_i^* \sum_{\substack{j=1 \\ j \neq i}}^n Y_{ij} V_j = \sum_{\substack{j=1 \\ j \neq i}}^n |Y_{ij}| |V_i| |V_j| e^{j(\delta_j - \delta_i + \theta_{ij})} \quad (\text{II.9})$$

$$P_i = \sum_{j=1}^n |V_i| |V_j| |Y_{ij}| \cos(\delta_j - \delta_i + \theta_{ij}) \quad (\text{II.10})$$

$$Q_i = \sum_{j=1}^n |V_i| |V_j| |Y_{ij}| \sin(\delta_j - \delta_i + \theta_{ij}) \quad (\text{II.11})$$

Pour un système à  $n$  jeux de barre, les équations de l'écoulement de puissance seront formulées comme suit :

$$\begin{aligned} I_1 &= \sum_{i=1, i \neq n}^n Y_{1i} V_1 + (-Y_{2n}) V_2 + \dots \dots \dots (-Y_{1n}) V_n \\ &\quad \vdots \\ &\quad \vdots \end{aligned} \quad (\text{II.12})$$

$$I_n = (-Y_{n1}) V_1 + (-Y_{n2}) V_2 + \dots \dots \dots + \left( \sum_{i=1, i \neq n}^m Y_{ni} \right) * V_n$$

Sous forme matricielle, on aura :

$$\begin{pmatrix} I_1 \\ \vdots \\ \vdots \\ I_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1, i \neq n}^n Y_{1i} & \cdot & \cdot & -Y_{1n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ -Y_{n1} & \cdot & \cdot & \sum_{i=1, i \neq n}^n Y_{ni} \end{pmatrix} \times \begin{pmatrix} V_1 \\ \vdots \\ \vdots \\ V_n \end{pmatrix} \quad (\text{II.13})$$

Le système d'équation de l'écoulement de puissance est un système non linéaire nécessitant une résolution faisant appel à des méthodes numériques itératives telles que Gauss\_Seidel et Newton-Raphson et qui fournissent une solution approximative.

Pour la méthode de Newton-Raphson adoptée dans notre cas, l'algorithme de résolution est donnée par [136] :

$$x^{(k+1)} = x^{(k)} - [J^{(k)}]^{-1} f(x^{(k)}) \quad (\text{II.14})$$

$x^{(k+1)}$  est la solution à l'itération  $k+1$ ,  $f(x^{(k)})$  est la fonction évaluée pour  $x^{(k)}$  et  $J^{(k)}$  est la matrice Jacobienne  $(J^{(k)} = \frac{\partial f}{\partial x} \Big|_{x=x^{(k)}})$ .

Les itérations sont faites jusqu'au moment où la différence entre  $x^{(k+1)}$  et  $x^{(k)}$  est inférieure à l'erreur ou à une précision définie à l'avance.

Soient  $P_i$  et  $Q_i$  les puissances active et réactive au jeu de barre  $i$ . La formule de base qui caractérise l'écoulement de puissance est donné par [136]:

$$f(x) = P_i(x) - P_i = \Delta P_i = 0 \quad (\text{II.15})$$

$$Q_i(x) - Q_i = \Delta Q_i = 0 \quad (\text{II.16})$$

Il est important aussi de souligner que la complexité de la méthode appliquée dans les réseaux électriques réside dans le calcul efficace de la matrice Jacobienne (calcul des dérivées de la puissance active et réactive par rapport aux angles et aux modules des tensions).

La matrice Jacobienne dans l'écoulement de puissance est présentée en quatre sous matrices comme suit [136]

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta P_i}{\partial \Delta \theta_i} & \frac{\partial \Delta P_i}{\partial \Delta V_i} \\ \frac{\partial \Delta Q_i}{\partial \Delta \theta_i} & \frac{\partial \Delta Q_i}{\partial \Delta V_i} \end{bmatrix} \quad (\text{II.17})$$

## II.2. Formulation de l'écoulement de puissance optimal (OPF)

La répartition économique de puissance ou l'écoulement de puissance optimal (Optimal Power Flow OPF) est parmi les outils utilisés dans l'exploitation et la planification des systèmes d'énergie [134]. Introduit par Carpentier en 1962, l'OPF est progressivement reconnue et devenu l'outil le plus utilisé par les opérateurs des systèmes énergétiques.

Pour un système énergétique reliant un ensemble de consommateurs à un ensemble de centrales électriques par l'intermédiaire d'un réseau de transport, l'OPF est un problème d'écoulement de puissance qui consiste à obtenir les valeurs optimales des variables de commande pour des valeurs données de la charge. Pour cela, on réduit généralement au minimum une fonction objectif prédéfinie en respectant les limites de fonctionnement du système.

Un problème de l'OPF peut être alors formulé comme un problème d'optimisation basé sur la minimisation d'une fonction objectif  $J$  avec des contraintes linéaire (d'égalités)  $g$  et non-linéaire (inégalités)  $h$  [137].

$$\begin{cases} \min J(x, u) \\ g(x, u) = 0 \\ h(x, u) \leq 0 \end{cases} \quad (\text{II.18})$$

$x$  est le vecteur des variables d'état (variables dépendants) et  $u$  est le vecteur des variables de contrôle (variables indépendants).

La formulation la plus utilisée est celle qui consiste à minimiser comme fonction objectif le coût du carburant ou le coût de production. Il existe d'autres formulations basées sur la minimisation de l'investissement pour les capacités shunts, la maximisation de production en énergie réactive, la minimisation des écarts de déviation des tensions, la minimisation de l'émission gaze [139].

## II.2.1. Classification des variables d'équations d'écoulement de puissance optimal

### II.2.1.1. Les variables d'état (variables dépendants)

Il s'agit des variables qui décrivent n'importe quel état unique du système. De tels variables sont [76]:

$$x = [P_{G_1}, V_{L_1} \cdots V_{L_{NL}}, Q_{G_1} \cdots Q_{G_{NG}}, S_{l_1} \cdots S_{l_{nl}}]^T \quad (\text{II.19})$$

$P_{G_i, i=1:NG}$ : puissance active générée de générateur  $i$ .

$Q_{G_i, i=1:NG}$ : puissance réactive générée de générateur  $i$ .

$P_{GI}$ : la puissance active générée par le jeu de barre de référence ou le balancier (slack bus).

$V_{L_i, i=1:NL}$ : les modules des tensions des jeux de barres charges (load).

$S_{l_i, i=1:nl}$ : la puissance apparente de la ligne.

$NL$  : nombre des jeux de barre charge.

$NG$  : nombre de générateur.

$nl$  : nombre de ligne.

### II.2.1.2. Les variables de contrôle (variables indépendants)

Les variables de contrôle sont les variables qu'on peut les modifier afin d'avoir la solution optimale de l'écoulement de puissance. Elles sont généralement les modules des tensions des jeux de barre auxquels sont connecté les générateurs  $V_G$ , les puissances actives générées aux jeux de barres générateurs, les rapports de transformation des transformateurs régulateurs de charge, et les puissances réactives générées par les différents compensateurs d'énergie réactive  $Q_C$  [138].

$$u = [P_{G_2} \cdots P_{G_{NG}}, V_{G_1} \cdots V_{G_{NG}}, Q_{C_1} \cdots Q_{C_{NC}}, T_1 \cdots T_{NT}]^T \quad (\text{II.20})$$

Dont :  $NG$ ,  $NT$  et  $NC$  sont respectivement : le nombre de générateurs, le nombre de transformateurs de régulation et le nombre de compensateurs réactifs.

$Q_{c_i, i=1:NC}$  : puissance réactive injecté par les compensateurs réactifs.

$T_{i, i=1:NT}$ : facture de régulation des transformateurs régulateurs de charge.

## II.2.2. Contraintes de problème de l'écoulement de puissance

### II.2.2.1. Contraintes d'égalités

Les contraintes d'égalité de l'OPF reflètent les lois physiques gouvernant le système électrique. Elles sont représentées par les équations non-linéaires de l'écoulement de puissance qui exigent que la somme des puissances actives et réactives injectées dans chaque jeu de barre soit nulle [138].

#### a. Contraintes sur la puissance active

$$P_{Gi} - P_{Di} - V_i \sum_{j=i}^{NB} V_j [G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij})] = 0 \quad (\text{II.21})$$

NB est le nombre de jeux de barres,  $P_{Gi}$  est la puissance active générée au jeu de barre  $i$ ,  $P_{Di}$  est la puissance active demandée au jeu de barre  $i$ ,  $G_{ij}$  et  $B_{ij}$  sont respectivement la conductance et la susceptance de la ligne entre les jeux de barres  $i$  et  $j$ , ( $Y_{ij} = G_{ij} + j\beta_{ij}$ ) et  $\theta_{ij} = \theta_j - \theta_i$  est la différence des phases des deux jeux de barre  $i$  et  $j$ .

#### b. Contraintes sur la puissance réactive

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=i}^{NB} V_j [G_{ij} \sin(\theta_{ij}) + B_{ij} \cos(\theta_{ij})] = 0 \quad (\text{II.22})$$

Tels que :

$$\theta_{ij} = \theta_j - \theta_i.$$

NB : nombre de jeux de barres.

$Q_{Gi}$  : puissance réactive générée dans le jeu de barre  $i$ .

$Q_{Di}$  : puissance réactive demandée dans le jeu de barre  $i$ .

### II.2.3. Contraintes d'inégalités

Les contraintes d'inégalités reflètent les limites physiques et de sécurité sur les dispositifs constituant le réseau électrique. Les contraintes les plus habituels sont les limites sur les tensions des jeux de barres, transformateurs à prises de charge, transformateurs déphaseurs et les limites introduites pour assurer la sécurité du système.

En plus, d'autres contraintes d'inégalités doivent être considérées telles que les limites des puissances réactives des compensateurs, les limites des puissances actives et réactives générées par les centrales qui doivent être maintenues à des valeurs admissibles.

#### a. Contraintes sur les générateurs

Au niveau des jeux de barres auxquels sont liés les générateurs incluant le jeu de barre de référence, les modules de tension, les puissances actives et réactives générées doivent être dans leurs limites inférieures et supérieures admissibles [139].

$$V_{G_i}^{min} \leq V_{G_i} \leq V_{G_i}^{max}, \quad i = 1, \dots, NG \quad (II.23)$$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max}, \quad i = 1, \dots, NG \quad (II.24)$$

$$Q_{G_i}^{min} \leq Q_{G_i} \leq Q_{G_i}^{max}, \quad i = 1, \dots, NG \quad (II.25)$$

NG est le nombre de générateurs.

#### b. Contraintes sur les transformateurs

Les transformateurs à prises de charge ou régulateur en charge ont des commutateurs max et min pour régler le niveau de tension par rapport à la tension nominale. De même, les transformateurs régulateurs de phase ont des décalages max et min pour le réglage des phases des tensions et la régulation d'écoulement de puissance active. Les deux types de transformateurs forment les contraintes d'inégalités suivantes :

$$T_i^{min} \leq T_i \leq T_i^{max}, \quad i = 1, \dots, NT \quad (II.26)$$

#### c. Contraintes sur les condensateurs de compensation

Les batteries et les compensateurs statiques de la puissance réactive sont cernés par des valeurs minimales et maximales comme suit :

$$Q_{C_i}^{min} \leq Q_{C_i} \leq Q_{C_i}^{max}, \quad i = 1, \dots, NC \quad (II.27)$$

NC est le nombre de batteries.

### II.2.4. Limites de sécurité

Les limites sur les transits de la puissance apparente au niveau des lignes de transport et les amplitudes des tensions au niveau des jeux de barres charges (PQ bus) sont créées pour assurer la sécurité du système contre les pertes thermiques dans les conducteurs et/ou la stabilité du réseau. On peut les représenter par les expressions suivantes [137]:

$$V_{L_i}^{min} \leq V_{L_i} \leq V_{L_i}^{max}, \quad i = 1, \dots, NL \quad (II.28)$$

$$S_{l_i} \leq S_{l_i}^{max}, \quad i = 1, \dots, nl \quad (II.29)$$

$S_{l_i}$  est la puissance apparente de la ligne,  $NL$  et  $nl$  sont respectivement les nombres de jeu de barre charge et de lignes.

### II.2.5. Traitement de variables discrètes

Afin de fournir des solutions significatives de l'OPF, les variables discrètes (les batteries de condensateurs et les transformateurs régleurs de la charge) sont traitées par une technique d'arrondi progressive [74]. La technique la plus utilisée est basée sur le traitement de toutes les variables discrètes comme des variables continues. Ensuite, ils sont progressivement arrondis en leurs plus proches valeurs des variables discrètes. La solution de l'OPF est réévaluée et à chaque fois, un ensemble de variables discrètes est fixé à des valeurs continues.

De telle technique offre un bon compromis entre la précision et la vitesse. Néanmoins, ceci ne garantit pas que la solution arrondie soit une solution optimale, car quelques contraintes peuvent être violées. Aussi, cette méthodologie ne conviendrait pas bien avec les variables discrètes qui ont de très grand pas telles que les batteries de condensateurs [74].

Une solution possible pour remédier à ces problèmes reflète l'arrondissement des valeurs continues avant l'intégration dans le calcul de l'écoulement de puissance, cela veut dire que la solution trouvée est une valeur discrète.

### II.2.6. Traitement de dépassement des contraintes

Un problème parfois rencontré lors de la résolution de l'OPF est qu'on n'arrive pas à avoir une solution physiquement réalisable au terme des contraintes à satisfaire. Une solution pour remédier à ce problème et d'ajouter une fonction comme terme de pénalité quadratique à la fonction objectif.

L'idéale pour une fonction de pénalité est d'être très faible à proximité d'une limite et d'augmenter rapidement lorsque cette limite est dépassée. Dans ces conditions, un facteur de pénalité multiplié par le carré de la valeur de crête de la variable dépendante est ajouté à la fonction objectif ce qui conduit au fait que toute solution impossible obtenue est diminuée. Mathématiquement, la fonction de pénalité peut être exprimée par [139] :

$$f_p = \lambda_P (P_{G_1} - P_{G_1}^{lim})^2 + \lambda_V \sum_{i=1}^{NL} (V_{L_i} - V_{L_i}^{lim})^2 + \lambda_Q \sum_{i=1}^{NG} (Q_{G_i} - Q_{G_i}^{lim})^2 + \lambda_S \sum_{i=0}^{nl} (S_{l_i} - S_{l_i}^{max})^2 \quad (II.30)$$

Tels que  $\lambda_P$ ,  $\lambda_V$ ,  $\lambda_Q$  sont des facteurs de pénalité et  $P_{G_1}^{lim}$ ,  $V_{L_i}^{lim}$  et  $Q_{G_i}^{lim}$  sont respectivement les composants limites de vecteur des variables indépendants  $x$  tels que :

$P_{G_1}^{lim}$  : Les valeurs limites de la puissance active de générateur de référence.

$V_{L_i}^{lim}$  : Les valeurs limites de tension des jeux de barre charge.

$Q_{G_i}^{lim}$  : Les valeurs limites de la puissance réactive générer.

D'une manière générale, la valeur limite de toute variable dépendante  $x$  notée  $x^{lim}$  ne peuvent être que  $x^{max}$  ou  $x^{min}$ .

$$x^{lim} = \begin{cases} x^{max}; & x > x^{max} \\ x^{min}; & x < x^{min} \end{cases} \quad (II.31)$$

## II.2.7. Fonction objectif

Le problème de dispatching économique consiste à appliquer plusieurs formulations pour déterminer le coût minimal de la production pour répondre à la puissance totale demandée. Cependant, ces formulations simplifiées ignorent les contraintes de l'écoulement de puissance [141].

Dans le problème de l'OPF, plusieurs fonctions objectives soumises aux différentes contraintes d'égalités et d'inégalités sont considérées. En réalité, la puissance générée par les unités de production n'est pas linéaire, elle est cernée par les limitations physiques des différents composants de la centrale [142].

A cet effet, plusieurs formulations ont été ajoutées à la fonction objectif basique de coût, à savoir l'effet des vannes, les zones interdites, les pertes de puissance actives et réactives, les émissions de gaz, la marge de sécurité, la déviation du profil de tension, la minimisation de la taille des condensateurs shunt et l'optimisation de leur emplacement.

### II.2.7.1. Minimisation du coût de production

La fonction objectif classique pour un problème de l'OPF est le coût de production des unités thermiques des centrales électriques. Généralement, la fonction coût peut être exprimée par une fonction quadratique représentée par [137].

$$J(x, u) = \left( \sum_{i=1}^{NG} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \right) + f_p \quad (\text{II.32})$$

Où :

$NG$ : Nombre total des générateurs.

$a_i$ : Coût à vide du générateur  $i$ .

$b_i$ : Coefficient de coût linéaire du générateur  $i$ .

$c_i$ : Coefficient de coût quadratique du générateur  $i$ .

$P_{Gi}$ : Puissance active générée par l'unité  $i$ .

### II.2.7.2. Coût de production avec l'introduction de l'effet d'ouverture des vannes

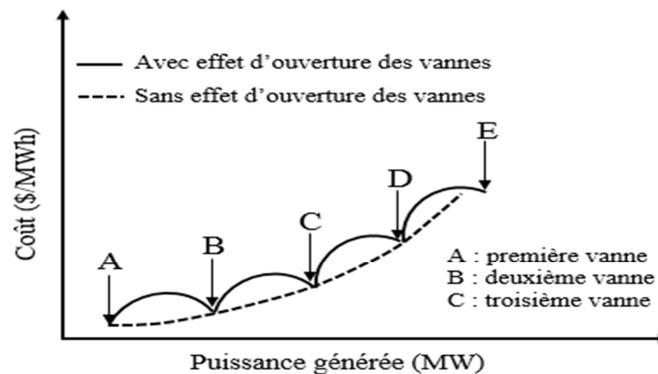
Dans le problème d'écoulement de puissance basique, la fonction objectif est rapprochée en utilisant des fonctions quadratiques en se basant sur l'idée que les courbes progressives de coût des unités sont des fonctions représentées par des morceaux-linéaires. Cependant, la représentation réelle de la caractéristique du coût est discontinue à cause des vannes de vapeur qui sont utilisées pour le contrôle de la puissance délivrée par l'unité. L'ouverture des vannes conduit par conséquent à une augmentation des pertes ce qui introduit des effets de répercussions en cascade à la courbe de coût.

L'effet d'ouverture des vannes est modélisé par un terme sinusoïdal qui est ajoutée à la fonction du coût quadratique [140]:

$$J(x, u) = \left( \sum_{i=1}^{NG} \left( a_i + b_i \times P_{Gi} + c_i \times P_{Gi}^2 + \left| d_i \times \sin \left( e_i \times (P_{Gi}^{min} - P_{Gi}) \right) \right| \right) \right) + f_p \quad (\text{II.33})$$

Ici  $a_i, b_i, c_i, e_i$ , sont les coefficients du coût de la  $i^{ème}$  unité de production.

La figure II.2 montre la courbe du coût en considérant l'effet des vannes d'admission de vapeur [140].



**Fig. II.2.** Courbe du coût du combustible d'une unité thermique en considérant l'effet d'ouverture des vannes d'admission de vapeur [140]

### II.2.7.3. Caractéristique avec plusieurs types de combustibles

Dans certains cas, les centrales thermiques peuvent avoir des sources multi-combustibles comme le gaz naturel et le pétrole [140]. Par conséquent, la fonction objectif de coût est représentée par plusieurs morceaux des fonctions quadratiques au nombre des combustibles utilisés. En général, une fonction quadratique par morceaux est utilisée pour représenter la courbe de coût d'un générateur à plusieurs combustibles, la fonction quadratique par morceaux est décrite dans l'équation (II.34) et la fonction de coût est illustrée sur la Figure II.3.

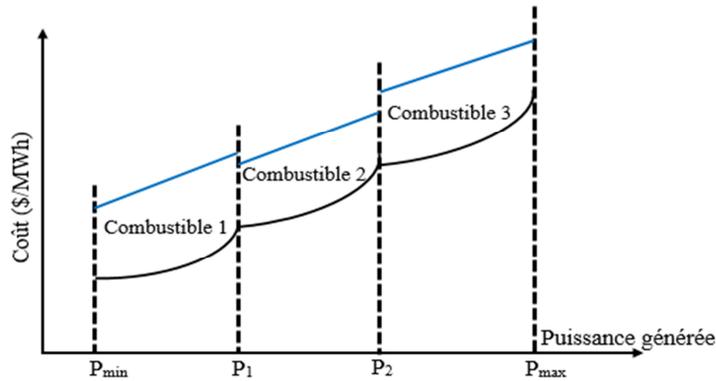


Fig. II.3. Caractéristique du cout d'un générateur avec plusieurs combustibles [146]

$$J(x, u) = J_{Cout_{G1 \text{ et } G2}} + J_{Cout_{Reste}} + f_p \quad (II.34)$$

$$\text{Avec : } P_{G_{ik}}^{min} \leq P_{G_i} \leq P_{G_{ik}}^{max}, \quad k = 1, 2, \dots \quad (II.35)$$

k est l'indice du combustible.

$$J_{Cout_{Reste}} = \sum_{i=3}^{NG} a_i + b_i P_{G_i} + c_i P_{G_i}^2, \quad i=3, \dots, NG \quad (II.36)$$

NG est le nombre total des générateurs.

### II.2.7.4. Caractéristique avec des zones de fonctionnements prohibées

Dans la pratique, les générateurs ont des zones d'exploitation interdites pour des raisons de certains défauts dans le palier d'arbre ou des vibrations de machines ou de leurs accessoires tels que les pompes ou les chaudières [143], [144].

Ces phénomènes conduisent à un espace de solution avec des régions disjointes impossibles pour chaque générateur. La fonction de coût d'un générateur en considérant l'effet des zones interdites POZ est représentée dans la figure II.4.

Par conséquent, les producteurs ont un ensemble de zones d'exploitation déductibles (zones dites non-prohibées) où ils doivent travailler dans une d'elles. La fonction POZ peut être incluse en tant que contrainte de la manière suivante [143]:

$$(P_{Gi}^{inf})_m \leq P_{Gi} \leq (P_{Gi}^{sup})_m \quad \forall i \in NG, \quad m = 1, 2, \dots, M_i \quad (II.37)$$

Tells que :  $(P_{Gi}^{inf})_m$  et  $(P_{Gi}^{sup})_m$ , sont les limites inférieures et supérieures pour une zone d'exploitation non-prohibée ou autorisée,  $M_i$  est le nombre de zones d'exploitation non-prohibées pour le  $i^{ème}$  générateur. Il est évident que chaque centrale comporte au moins une zone de fonctionnement autorisée [144].

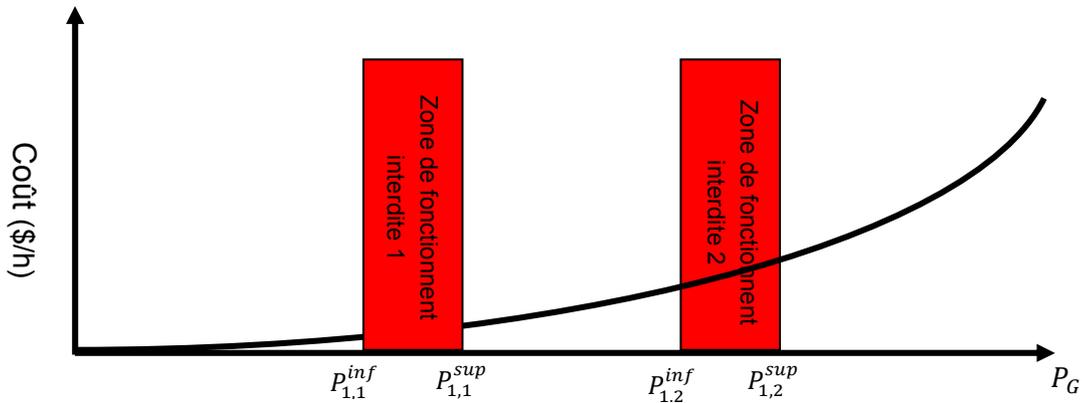


Fig. II. 4. Courbe de coût du combustible d'une unité thermique avec des zones de fonctionnements prohibées [144]

### II.2.7.5. Minimisation des pertes actives

La minimisation des pertes actives peut être formulée de la manière suivante [145] :

$$P_L = \sum_{i=1}^{NB} P_{Gi} - \sum_{i=1}^{NB} P_{Di} \quad (II.38)$$

Où :

$P_{Gi}$  : Puissance active générée au nœud  $i$ .

$P_{Di}$  : Puissance active demandée au nœud  $i$ .

### II.2.7.6. Minimisation des pertes réactives

Un réseau électrique consiste à l'acheminement de la puissance active de la source à la charge. Ainsi, la disponibilité de la puissance réactive est essentielle pour la stabilité de la marge de tension. Partant de ce constat, la minimisation des pertes réactives est un critère important dans la planification des réseaux électriques [145].

Les pertes réactives totales sont exprimées par l'équation (II.39).

$$Q_L = \sum_{i=1}^N Q_{Gi} - \sum_{i=1}^N Q_{Di} \quad (\text{II.39})$$

$Q_{Gi}$  est la puissance réactive générée au nœud  $i$  et  $Q_{Di}$  est la puissance réactive demandée au nœud  $i$ .

### II.2.7.7 Minimisation de la déviation du profil de tension

La tension des jeux de barres est l'un des indicateurs les plus importants et significatifs de la qualité et de la sécurité de service. Le moyen d'évaluation de la qualité du profil de tension pour un jeu de barre de charge donné est basé sur le calcul de l'écart entre la tension de bus PQ et l'unité 1. Pour le réseau complet, l'amélioration de profils de tension consiste à minimiser la somme des écarts exprimée par [143] :

$$VD = \sum_{j=1}^{NL} |V_{L_i} - 1| \quad (\text{II.40})$$

Où  $V_{L_i}$  est le module de tension aux nœuds de charge en PU et NL est le nombre total des jeux de barre de charge.

### II.2.7.8. Perfectionnement de la Stabilité de tension

Récemment, et vu l'augmentation de la demande en puissance, les réseaux électriques sont stressés à fonctionner à proximité de leurs limites. A cet effet, la stabilité de la tension devient une condition fondamentale.

La sensibilité d'un indice de stabilité de la tension appelé  $L$  développé par Kessel et Glavitch est utilisé pour assurer la stabilité de la tension [147]. Cet indicateur varie entre les deux valeurs 0 et 1 qui correspondent respectivement aux cas d'absence de charge et d'effondrement de la tension.

L'augmentation de la valeur de l'indice  $L$  signifie que le bus correspondant se déplace vers l'état instable. A cet effet, l'objectif dans ce cas est de minimiser l'indice  $L$  dont l'expression pour le  $j^{\text{ième}}$  bus est donnée par [155] :

$$L_j = \left| 1 - \sum_{i=1}^{NG} F_{ji} \frac{V_i}{V_j} \right|, \quad j = 1, 2, \dots, NPQ \quad (\text{II.41})$$

$$\text{Tels que } :F_{ji} = [Y_1]^{-1}[Y_2] \quad (\text{II.42})$$

Où NG est le nombre des jeux de barre générateur, NPQ est le nombre des jeux de barre charge,  $Y_1$  est  $Y_2$  sont des sous matrices de la matrice impédance obtenue après la séparation des paramètres des jeux de barre générateur et charge.

De telle matrice impédance relie les courants aux potentiels selon l'équation suivante :

$$\begin{bmatrix} I_{PQ} \\ I_{PV} \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} \begin{bmatrix} V_{PQ} \\ V_{PV} \end{bmatrix} \quad (\text{II.43})$$

$I_{PQ}$  : Représente les courants au niveau de jeux de barre charge,

$I_{PV}$  : sont les courants au niveau de jeux de barre générateurs,

$V_{PQ}$  : sont les tensions au niveau de jeux de barre charge,

$V_{PV}$  : sont les tensions de jeux de barre générateurs.

L est calculé pour tous les jeux de barre charge par :

$$L = \max(L_j), \quad j = 1, 2, 3, \dots, NPQ \quad (\text{II.44})$$

### II.2.7.9. Minimisation des émissions des gaz

Après le protocole de Kyoto signé en 1990 et qui vise la réduction des émissions de gaz à effet de serre, la sécurité des réseaux n'est plus devenue le critère suffisant pour la distribution de l'énergie électrique. La minimisation des gaz polluant devient également indispensable pour les compagnes d'électricité dans de nombreux pays [148].

La fonction objectif d'émission des gaz peut être présentée comme la somme de tous les types d'émission  $NOx$ ,  $SOx$ , ... avec un facteur approprié à la pondération de l'émission de chaque polluant.

Dans notre étude, deux types importants d'émission de gaz sont pris en compte  $NOx$  et de  $SOx$ . La quantité  $Eg$  de ces émissions est représentée par une fonction de termes exponentiels et quadratiques de la puissance générée par les générateurs:

$$Eg = \sum_{i=1}^{NG} 10^{-2} (\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2) + (\omega_i e^{(\mu_i P_{Gi})}) \quad (\text{II.45})$$

Les constants  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ ,  $\mu_i$  et  $\omega_i$  ont les coefficients des émissions pour l'unité i.

### **II.3. Formulation de l'écoulement de puissance optimal (OPF) avec introduction des dispositifs FACTS**

Durant ces dernières années et avec l'augmentation de la demande de l'énergie électrique, les réseaux électriques sont confrontés à des problèmes poussés par de nouvelles contraintes qui peuvent dégrader la qualité de l'énergie électrique fournie au terme de stabilité des tensions et de l'augmentation des pertes [150].

Face à ces problèmes, le concept FACTS (Flexible AC Transmission System) a été lancé dont l'objectif principale est d'augmenter la capacité de transfert des lignes de transmission par leur fonctionnement près de leurs limites de stabilité sans avoir des risques d'effondrement de tension [150],[151].

Selon l'IEEE, la terme FACTS signifie Système de Transmission en Courant Alternatif incorporant des dispositifs basés sur l'électronique de puissance et d'autres régulateurs statiques utilisés pour améliorer la contrôlabilité et augmenter la capacité de transfert de puissance du réseau [152].

Le développement des dispositifs FACTS ouvre des nouveaux aboutissements pour une meilleure exploitation des réseaux grâce à leur impact continu et rapide sur les différents paramètres du réseau et cela par consommation ou compensation de la puissance réactive en agissant sur l'impédance des lignes ou en modifiant les angles des tensions [153].

Grâce à leur faible temps de réponse favorisé par l'utilisation des éléments de l'électronique de puissance et des systèmes à thyristors, les FACTS sont bien adaptés aux changements rapides des différentes grandeurs du réseau électrique [153].

Il existe grande famille de dispositifs FACTS selon leur mode de connexion à la ligne. On distingue les modes parallèles, séries et hybrides (série- parallèles).

Dans ce travail, on s'intéresse aux dispositifs FACTS les plus connus qui sont les compensateurs statiques d'énergie réactive (SVC), les compensateurs série contrôlés par thyristors (TCSC) et les transformateurs déphaseurs commandés par thyristor (TCPS) [150].

#### **II.3.1. Modélisation des dispositifs FACTS**

Plusieurs approches ont été adoptées pour modéliser les dispositifs FACTS. Ces approches ont été développées pour étudier les impacts des dispositifs FACTS sur les réseaux électriques en termes de stabilité des tensions et influence sur l'écoulement de puissance. Trois modélisations sont fréquemment rencontrées dans la littérature [151].

La première modélisation est basée sur l'injection de puissance équivalente où l'effet des FACTS est résumé dans la modification des équations de calcul des puissances actives et réactives de la ligne [151].

La deuxième est la création d'un nœud fictif où les puissances fournies par les FACTS sont injectées. Suite au rajout de ce nouveau nœud, on effectue des modifications de la structure de la matrice jacobéenne [152].

Pour la troisième tendance, l'effet des FACTS conduit à des changements directs sur la matrice admittance du réseau sans avoir la nécessité de modifier la matrice jacobéenne [152].

Dans ce travail, la troisième tendance sera adoptée pour résoudre le problème de l'OPF avec la considération des trois types des FACTS.

### II.3.1.1. Les compensateurs statiques d'énergie réactive

Les compensateurs statiques d'énergie réactive(SVC) sont des éléments de compensation shunts destinés à garder les niveaux de tension dans ses limites exigées par injection de la puissance réactive dans les jeux de barre où ils sont insérés. Ils sont constitués d'une inductance et d'une capacité branchés en parallèle (voir Fig. II.5). La puissance réactive injectée est contrôlée par des thyristors [153].

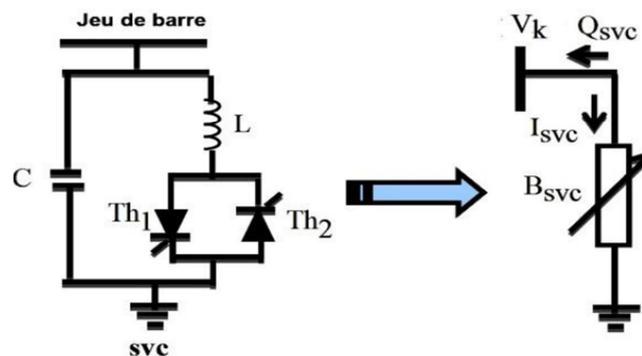


Fig. II.5. Modèle équivalent d'un SVC [153]

Selon la figure II.5, les SVC sont modélisés par des admittances variables. La puissance réactive doit être limitée par deux valeurs max et min :

$$Q_{SVC}^{min} \leq Q_{SVC} \leq Q_{SVC}^{max} \quad (II.46)$$

### II.3.1.2. Modélisation des compensateurs série contrôlés par thyristors TCSC

Ce sont des dispositifs FACTS connectés en série avec le réseau pour contrôler le transit de la puissance et l'amortissement de ses oscillations. Ils sont formés d'une inductance en parallèle avec une capacité variable contrôlée par un thyristor [154].

Le modèle d'un TCSC est illustré par le schéma de la figure. II.6.

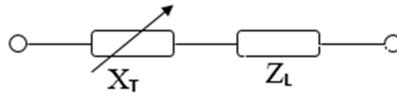


Fig. II.6. Modèle équivalent d'un TCSC

L'introduction d'un TCSC va modifier l'impédance série de la ligne. Plus précisément, c'est la réactance de la ligne qui va changer selon l'équation suivante [154]:

$$X_{ij} = X_L + X_T \quad (\text{II.47})$$

$$X_T = r_T \cdot X_L \quad (\text{II.48})$$

$r_T$  est le coefficient qui représente le degré de compensation du TCSC pour éviter les surcompensations  $r_{T\min} = 0.2$ ,  $r_{T\max} = 0.7$ , la gamme de fonctionnement du TCSC est choisie pour être entre  $-0,2 \cdot X_T$  à  $0,7 \cdot X_T$  [154].

### II.3.1.3. Transformateur déphaseur commandé par thyristor (TCPST)

Le TCPST est modélisé par un transformateur déphaseur ajusté par thyristor afin de fournir un angle de phase variable [155].

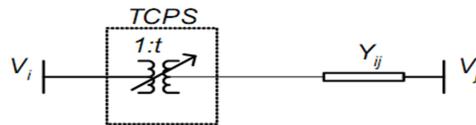


Fig. II.7. Modèle équivalent d'un TCPST

L'introduction de TCPS modifie la matrice  $Y$  pour être comme suit [156] :

$$Y_{\text{bus}} = \begin{pmatrix} t^2 Y_{ii} & -t Y_{ij} e^{-j\phi} \\ -t Y_{ij} e^{-j\phi} & Y_{jj} \end{pmatrix} \quad (\text{II.49})$$

$t$  est le rapport de régularisation en phase.

## II.4. Ecoulement de puissance optimal avec les contraintes de sécurité

Avec l'augmentation de la demande d'énergie, les réseaux électriques sont utilisés à proximité de leurs limites de sécurité. Cela peut conduire au déclenchement d'une ligne ou surcharge de ses voisins par son report de transit qui a par conséquence un effet cascade de surcharges. Ces dernières peuvent conduire à des dégâts dans les équipements et l'instabilité du réseau. Pour éviter ces problèmes, une forme étendue à l'OPF appelée écoulement de puissance optimale avec contraintes de sécurités (SCOPF) cette nouvelle problématique est nommée au premier lieu « l'état stable de sécurité » par Alsac et les autres [5], et sa

première définition comme écoulement de puissance optimale avec contraintes de sécurités (SCOPF) a été introduite par [157].

Dans le problème SCOPF, les contraintes relatives aux cas d'urgences ou les cas de contingences seront ajoutés au cas de base de l'OPF. La solution obtenue assure que le réseau reste dans ses limites de stabilité et de sécurité non seulement en état de fonctionnement normal mais aussi pendant différentes éventualités tel que le déclenchement d'une ligne [157], [158].

On distingue deux modes de problème SCOPF, mode préventif (PSCOPF) et mode correctif (CSCOPF) [157]. La principale différence entre ces deux modes est que dans la PSCOPF, le changement des variables de contrôle dans le cas de contingence n'est pas autorisé. Les valeurs des variables de contrôle obtenues en résolvant le cas de base doit être satisfaisantes aussi et simultanément pour les cas de contingence [159].

Le SCOPF est formulé comme un problème d'optimisation non linéaire et non-convexe à grande échelle, Son plus important défi est sa grande taille notamment pour les systèmes à grande échelle. Quand il y a un grand nombre de cas de contingences à considérer, l'utilisation d'une approche directe pour obtenir la solution du problème avec la prise en compte de tous les scénarios nécessite un ressort de calcul énorme [160].

Pour remédier à cet inconvénient, plusieurs approches de simplification ont été proposées. Ces approches se concentrent sur :

- les méthodes de sélection de contingences ou de simplification du réseau comme les schémas de sélection des contingences itératives [161], [162],
- les techniques de filtrage des contingences [163],
- les méthodes de décomposition [164],
- la compression du réseau [165],
- les méthodes itératives combinant les méthodes de sélection et de compression du réseau [166].

Les méthodes de sélection itérative des contingences sont basées sur l'idée que la majorité des cas de contingence ne contraignent pas l'optimum et qu'aucune violation des contraintes ne peut avoir pour la grande majorité de cas.

La méthode de décomposition est une impressionnante technique largement utilisée pour résoudre le problème de SCOPF. Son idée principale consiste à décomposer le problème SCOPF en un problème maître qui correspond au fonctionnement normal et  $N$  sous problèmes correspondant aux cas imprévus.

### II.4.1. Mode préventif

Le mode préventif PSCOPF est initialement suggéré par Alsac [5]. Une extension de l'approche (Dommel-Tinney) a été présentée afin d'obtenir une formulation exacte de l'OPF en considérant un déclenchement d'une ligne comme un cas de contingence.

Dans le PSCOPF, le changement des variables de contrôle dans les cas de contingence n'est pas autorisé, la solution obtenue doit satisfaire le cas de base ainsi que les cas de contingences.

Le PSCOPF est formulé comme suit :

$$\left\{ \begin{array}{l} \min f(X_0, U_0) \\ g_k(X_k, U_0) = 0, \quad k = 1, \dots, c \\ h_k(X_k, U_0) \geq 0, \quad k = 0, \dots, c \end{array} \right. \quad \begin{array}{l} \text{(II.50)} \\ \text{(II.51)} \\ \text{(II.52)} \end{array}$$

$f, g, h$  sont respectivement la fonction objectif, les contraintes d'égalité et d'inégalité,  $X_0$  et  $U_0$  sont respectivement les variables d'état et de contrôle du cas de base,  $X_k$  et  $U_k$  sont respectivement les variables d'état et de contrôle du  $k^{\text{ème}}$  cas de contingence et  $c$  est le nombre des cas de contingence à considérer.

Il est fondamental de mentionner que la valeur des variables limites des contraintes d'inégalité dans les cas de contingences peuvent être différents de ceux du cas de base.

Quatre étapes essentielles sont utilisées dans la formulation du PSCOPF :

1. La résolution de problème de l'OPF pour le cas de base.
2. Identification des cas de contingences d'insécurité.
3. Les contraintes des cas de contingence seront ajoutées progressivement à la formulation de l'écoulement de puissance optimal jusqu' à l'obtention de la nouvelle solution sécurisée.
4. On revient à l'étape 2, les paramètres obtenus du système pour l'étape 3 seront testés.  
Si un état d'insécurité est détecté nous allons à l'étape 3.

Ce processus sera répété jusqu' à l'obtention d'une solution sécurisée pour tous les cas de contingence.

### II.4.2. Mode correctif

L'approche corrective CSCOPF est conçue pour être plus économique que la PSCOPF. Certaines variables de contrôle peuvent être arrangées en utilisant des actions correctives lors de l'occurrence d'une contingence.

Dans cette approche, on se concentre sur l'hypothèse que les limites de sécurité de

l'équipement du réseau peuvent être empiétées pendant plusieurs minutes sans endommager l'équipement correspondant. Ce temps est suffisant pour adopter des mesures correctives telles que le rééchelonnement de la génération pour éliminer les violations des contraintes [165].

le mode correctif du SCOPF a été traité par Monticelli [157]. Il s'est concentré sur la correction de la solution du problème de répartition économique avec la considération des contraintes de sécurité en utilisant une nouvelle formulation de l'approche de la décomposition.

Le mode correctif de la CSCOPF est formulé comme suit [157] :

$$\left\{ \begin{array}{l} \min f(X_0, U_0) \\ g_k(X_k, U_k) = 0, \quad k = 1, \dots, c \\ h_k(X_k, U_k) \geq 0, \quad k = 1, \dots, c \\ |U_k - U_0| \leq \Delta U_k^{\max}, k = 1, \dots, c \end{array} \right. \quad \begin{array}{l} \text{(II.53)} \\ \text{(II.54)} \\ \text{(II.55)} \\ \text{(II.56)} \end{array}$$

L'équation (II.56) représente les contraintes de couplage. Elle impose une condition utilisée pour empêcher des actions correctives irréalistes.

$\Delta U_k^{\max} = T_k \cdot dU_k/dt$  est le vecteur des valeurs maximales de tolérance d'ajustement des variables de contrôle entre le cas de base et le  $k^{\text{ème}}$  cas de contingence,  $T_k$  est le temps supposé nécessaire pour l'action corrective et  $dU_k/dt$  est la proportion de l'ajustement des valeurs de contrôle afin de satisfaire le cas de contingence.

Nous pouvons voir que la SCOPF préventive est un cas particulier de la SCOPF corrective pour  $\Delta U_k^{\max} = 0$  correspondant au cas où aucune mesure corrective n'est autorisée.

Deux étapes essentielles sont utilisées dans le CSCOPF :

1. Résoudre le problème de l'écoulement de puissance optimale pour le cas de base.
2. Résoudre le problème de l'écoulement de puissance optimale des cas de contingences  $k=1 \dots, c$  tout en satisfaisant les contraintes d'inégalité (II.55) et les contraintes de couplage (II.56).

## **Conclusion**

Dans ce chapitre nous avons présenté en premier lieu les notions et les équations de base du l'écoulement de puissance avec ces différents méthodes de résolution, la méthode de Newton Raphson est optée dans notre travail pour résoudre le problème.

En deuxième lieu nous avons présenté les concepts essentiels pour la formulation d'un problème de l'écoulement de puissance optimal, en occurrence ces différents variables d'état et de contrôle, ces contraintes et principalement ces différents objectifs traités dans la littérature.

Suite à l'étude effectuée dans ce chapitre, nous avons conclu que l'OPF est un problème d'optimisation non linéaire et non-convexe avec contraintes.

**CHAPITRE III**  
**METHODES D'OPTIMISATION**  
**META HEURISTIQUES**

### III. Méthodes d'optimisation métaheuristiques

Dans les domaines d'économie et d'ingénierie, l'épuisement des ressources naturelles en raison de la surexploitation, a donné une attention très sérieuse aux procédés d'optimisation. Depuis l'année 1947 où Danzig a développé la méthode d'optimisation *simplex* pour des problèmes de programmation linéaire, les techniques d'optimisation ne cessent de développer en ouvrant la voie à la programmation dynamique et à l'optimisation non linéaire et avec contraintes [167].

Généralement, on a deux types d'algorithmes stochastiques, heuristiques (du grec *eurisko* qui signifie découvrir par essais et erreurs) et méta heuristiques issus suite à la prospection du développement sur les algorithmes heuristique (le mot méta signifié au-delà ou niveau supérieur) [168].

Les algorithmes heuristiques sont guidés par des spécificités liées au problème posé. Ils sont donc dépendants [169]. En outre, tous les algorithmes métaheuristiques utilisent certains compromis entre la randomisation et le recherche locale [170]. Les processus utilisés par ces derniers sont souvent inspirés par des systèmes naturels dans de nombreux domaines tels que la biologie (algorithmes évolutionnaires et génétiques) et la physique comme le recuit simulé.

On souligne qu'aucune définition convenue des heuristiques et métaheuristiques existe dans la littérature. Certains utilisent heuristiques et métaheuristiques de manière interchangeable. Cependant, la tendance récente tend à nommer tous algorithmes stochastiques avec randomisation et recherche locale comme métaheuristiques. La randomisation offre un bon moyen de se déplacer loin des optimums locaux [171].

Ce chapitre présente un exposé sur un ensemble de méthodes d'optimisation programmées pour la résolution de notre problème de l'OPF. Dans de tel exposé, nous citons le principe de chaque méthode, ses avantages et les étapes (algorithme) de sa mise en œuvre.

#### III.1. Les Algorithmes Génétiques

Les Algorithmes Génétiques AG sont des algorithmes d'optimisation stochastique et évolutionnaire avec un potentiel de recherche global initialement proposés par Holland en 1975 [13]. Ils sont inspirés de la génétique des populations (hérédité et fréquences de gènes) et de leur évolution [171], [172].

Comme tous les algorithmes évolutionnistes, les AG utilisent les trois grands principes de l'évolution naturelle ; la reproduction, la sélection naturelle et la diversité des espèces [173].

La stratégie de recherche des AG est basée sur trois opérations principales: reproduction, croisement et mutation.

Dans un cycle d'exécution simple, l'opération de reproduction sélectionne une chaîne de chromosomes de la génération précédente sur la base de la valeur de fitness et sa probabilité de propagation à la prochaine génération [173]. Après que la population de la génération suivante est remplie, l'opération de croisement fonctionne en conjonction avec la reproduction. Elle sélectionne deux individus nommés parents puis coupe et permute l'information génétique de ces deux derniers.

L'opérateur de mutation modifie les valeurs des chromosomes aléatoires pendant la phase de reproduction croisée. Les probabilités d'occurrence de croisement et de la mutation lors d'une exécution sont sélectionnables [173]. L'opérateur de sélection détermine les individus qui survivront et continueront dans la prochaine génération [174].

La procédure de résolution par l'AG est résumée dans l'algorithme suivant :

- 1. Initialisation**
  - Génération de la population initiale, la probabilité de mutation et de croisement.
- 2. Sélection**
  - Evaluation e la valeur de fitness ou la fonction objectif entre tous les individus dans la population actuelle. On choisit ceux qui continueront dans les étapes suivantes pour produire le nouvel individu.
- 3. Recombinaison**
  - Elle consiste à la génération de la nouvelle population, la recombinaison typique pour l'algorithme génétique est une opération nécessitant deux parents, qui seront échangés les gènes correspondants pour former un enfant ou un nouvel individu.
- 4. Mutation**
  - Dans cette étape, certains éléments de l'ADN des individus de la nouvelle population seront changés. Dans la nature, ces changements sont principalement causés par des erreurs au cours de processus de copie des gènes des parents.  
Dans l'AG, la mutation signifie un changement aléatoire de la valeur d'un gène dans la nouvelle population.

On revient à l'étape 2 ou l'opération de sélection sera appliquée sur la nouvelle population générée dans l'étape 4.

Le processus est répéter jusqu'à un critère d'arrêt satisfaisons est atteint.

### III.2. Algorithme du Recuit Simulé

La méthode du recuit simulé (RS) ou SA en anglais, est une méthode d'optimisation inspirée du processus de recuit physique utilisé en métallurgie par la société IBM en 1983

pour améliorer la qualité d'un solide. La structure cristallographique des métaux sera améliorée en alternant des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau [30].

L'algorithme commence par la génération d'une population initiale, et par la suite (à chaque nouvelle itération), une solution s'est générée de manière aléatoire dans le voisinage de chaque individu. Si le nouvel individu améliore la valeur de la fonction objectif, elle sera acceptée directement. Sinon, elle ne sera acceptée qu'avec une valeur de probabilité donnée par le critère de Métropolis-Hastings. Le recuit simulé applique itérativement le critère de Métropolis jusqu'à l'obtention d'une structure stable [168].

L'application de l'algorithme RS est clarifiée dans le schéma suivant :

1. Générer la configuration initiale des individus  $x_0$  et de la température de départ  $T_0$ .
2. Générer des individus aléatoires dans le voisinage des positions initiales et évaluer leurs fonctions objectifs (énergie) nommées  $E$ .
3. Comparer les deux solutions selon le critère de Metropolis,
 

Si  $E(x_{new}) < E(x_{old})$ ,  $x = x_{new}$ , directement.

Sinon  $x = x_{new}$ , lorsque  $\frac{\Delta E}{T} = \frac{E(x_{new}) - E(x_{old})}{Kc \times T} > r$ ,  $r$  : nombre aléatoire  $< 1$ .
4. Répéter 2 et 3 jusqu'à ce que l'équilibre statistique soit atteint (un critère prédéfini est atteint).
5. Décroître la température d'une façon continue, le plus courant est d'utiliser la loi suivante :  $T_{i+1} = \alpha \times T_i$ , en générale  $\alpha = 0,9$  à  $0,99$  jusqu'à arriver à T-stop.

Le procès de 3 à 5 et répéter jusqu'à un critère d'arrêt satisfaisant est atteint.

### III.3. Algorithme d'optimisation par essais de particules

L'optimisation par essais de particules (PSO), proposé par Kennedy [17] en 1995, est un algorithme d'optimisation flexible et robuste. Il est basé sur la recherche stochastique et l'intelligence des essaims (groupes) initialement [17]. Il est inspiré du comportement des groupes d'oiseaux dans leurs recherches de la nourriture. Les particules dans l'essaim volent dans un environnement dirigé par les membres ajusteurs afin de trouver les bonnes zones préalablement connues.

L'algorithme commence par la génération des positions et des vitesses aléatoires aux particules de l'essaim. La nouvelle position d'un individu sera changée en fonction de la meilleure position de lui-même et des autres individus. Au cours des itérations (temps), les

meilleures positions seront actualisées en se basant sur le calcul de la fonction objectif. Ensuite, les particules convergent vers les endroits optimaux [177].

La procédure de l'algorithme (PSO) est résumée comme suit :

1. Initialisation de la population des essaims en attribuant une position et une vitesse aléatoire à chaque particule.
2. Pour chaque particule, évaluer la fonction objectif (fitness), ensuite les valeurs obtenues seront comparées, en déterminant la particule avec la meilleure valeur de fitness.
3. La position ainsi que la vitesse de chaque particule seront actualisées en fonction de sa meilleure position connue ainsi que la position de la particule qui a la meilleure valeur de fitness via les équations suivantes :

$$\begin{cases} x_{k+1}^i = x_k^i + v_{k+1}^i \\ v_{k+1}^i = w \times v_k^i + c_1 \times r_1 \times (p_k^i - x_k^i) + c_2 \times r_2 \times (p_k^g - x_k^i) \end{cases}$$

$x_k^i$ ,  $v_k^i$  et  $p_k^i$  représentent respectivement la position, la vitesse et la meilleure position connue de particule  $k$  à l'itération  $i$ ,  $c_1$  est son coefficient de pondération,  $c_2$  est le coefficient de pondération,  $p_k^g$  est la meilleure position de tous les particules connus jusqu'à l'itération  $i$ ,  $r_1$  et  $r_2$  sont finalement des nombres aléatoires uniformes compris entre 0 et 1.

Les étapes 2, 3 seront répétées jusqu'à un critère d'arrêt est atteint (un nombre maximale d'itération).

#### III.4. Algorithme d'évolution différentielle

L'algorithme d'évolution différentielle (DE) est un algorithme d'optimisation de recherche stochastique directe et global initialement proposé par Storn et Price en 1997 [176].

L'algorithme DE est basé sur une population comme algorithmes génétiques. Il utilise des opérateurs similaires au AG ; le croisement, la mutation et la sélection.

La principale différence entre l'AG et le DE réside dans l'étape de mutation ainsi que le processus de sélection. Dans le DE, toutes les solutions des candidats ont la même chance d'être sélectionnées en tant que parents sans avoir la dépendance de leur fitness.

L'avantage majeur du DE repose sur le fonctionnement de mutation qui est basée sur les différences entre les vecteurs choisis au hasard. Le DE a également un croisement non-uniforme qui utilise plus d'informations que les autres algorithmes évolutionnaire [177].

La procédure de l'algorithme DE peut se résumer comme suit [178], [179]:

**1. Initialisation**

→ Génération d'une population initiale de vecteurs de recherche d'une dimension  $X_i = [X_1, X_2, \dots, X_D]^T$ ,  $i=1, 2 \dots NP$ , ou NP est le nombre de population, et D est la dimension de problème ou nombres des variables de contrôle.

**2. Mutation**

→ Une nouvelle population des vecteurs sera générée, pour l'algorithme DE trois individus seront choisis arbitrairement afin de générer un nouveau vecteur via l'équation suivante :

$$X'_i = X_a + F \times (X_b - X_c), i=1, 2 \dots NP.$$

$X_a, X_b$  et  $X_c$  sont trois vecteurs choisis arbitrairement et  $F$  est le facteur de mutation choisi afin de contrôler la taille de population.

**3. Recombinaison (croisement) :**

→ L'opération de croisement génère des nouveaux vecteurs en mélangeant les paramètres des vecteurs mutants avec les vecteurs cibles.

$$\begin{cases} X_{i+1} = X'_i & \text{si } \text{rand}(0,1) < c_r \\ X_{i+1} = X_i & \text{si } \text{rand}(0,1) > c_r \end{cases}$$

*rand* est une fonction qui génère des nombres aléatoires dans l'intervalle [0 1], et  $c_r$  est coefficient de croisement à définir, typiquement  $c_r=0,8$ .

**4. Sélection**

→ La Sélection est l'opération par laquelle une meilleure progéniture est générée, l'évaluation de la fonction objectif d'une progéniture est comparée par rapport à celle de son parent, et la meilleure sera conservée.

Les étapes 2, 3, 4 seront répéter jusqu'à qu'un critère d'arrêt et satisfait.

**III.5. Algorithme d'Optimisation de Colonie de Fourmis**

L'algorithme d'optimisation de colonie de fourmis (ACO) est un algorithme d'optimisation qui a été initialement introduit par Dorigo et Gambardella [181], [182]. L'ACO est inspiré du comportement alimentaire des fourmis, en particulier la communication de la phéromone entre les fourmis concernant un bon chemin entre la colonie et une source de nourriture dans un environnement. Lors de la recherche de la nourriture, les fourmis explorent d'abord la zone entourant leur nid d'une manière aléatoire. Dès qu'une fourmi trouve une source de nourriture, elle évalue la quantité et la qualité de la nourriture et transporte la nourriture trouvée au nid [183].

Au cours du voyage de retour, la fourmi déposera une piste de phéromone chimique sur le sol. La quantité de phéromone déposée reflète la quantité et la qualité de la nourriture et guidera les autres fourmis à la source de nourriture.

La communication indirecte entre les fourmis via les pistes de phéromone permet de trouver les chemins les plus courts entre leurs sources de nidification et de nourriture.

Cette fonctionnalité d'une colonie de fourmis réelle est exploitée dans les colonies de fourmis artificielles dans le but de résoudre des problèmes d'optimisation globale [183].

L'ACO est un algorithme stochastique où la construction d'une solution est effectuée d'une façon probabiliste en ajoutant des composants des solutions de façon itérative à des solutions partielles en prenant en compte les pistes des phéromones artificielles. Ces dernières changent dynamiquement lors de l'exécution pour tenir compte de l'expérience de recherche acquise par les agents.

La procédure de l'algorithme (ACO) est résumée comme suit [183], [184] :

1. Les fourmis sont positionnées sur différents chemins, et des valeurs initiales  $\tau_{ij}$ , pour la quantité des phéromones pour chaque chemin sont fixées.
2. Une fourmi choisie un chemin avec une probabilité  $P$  donnée ci-après :

$$P_{i,j} = \frac{(\tau_{i,j}^\alpha) \times (n_{i,j}^\beta)}{\sum (\tau_{i,j}^\alpha) \times (n_{i,j}^\beta)}$$

$\tau_{i,j}$  représente la quantité de phéromone dans un chemin (définie par deux nœuds  $i, j$ ),  $\alpha$  est un paramètre pour le contrôle de  $\tau_{i,j}$ ,  $n_{i,j}$  est l'opportunité de bord  $(i, j)$  typiquement  $1/d_{i,j}$ , ou  $d_{i,j}$  est la distance entre les deux fourmis (chemins  $i$  et  $j$ ) ou  $\beta$  est un paramètre pour le contrôle de l'influence de  $n_{i,j}$ .

3. La quantité des phéromones dans chaque chemin sera mise à jour via l'équation suivante :

$$\tau_{i,j} = (1 - \rho) \times \tau_{i,j} + \Delta\tau_{i,j}$$

$\tau_{i,j}$  est la quantité de phéromone dans un chemin  $(i,j)$ ,  $\rho$  est le taux d'évaporation de phéromone.

$\Delta\tau_{i,j}$  est la quantité de phéromone déposée typiquement donnée par :

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{Q}{L_k} & \text{Si l'ant } k \text{ se déplace sur le chemin } (i, j) \\ 0 & \text{Autrement} \end{cases}$$

$L_k$  est la fonction objectif évaluée pour la  $k^{\text{ième}}$  fourmi et  $Q$  est un constant.

Les étapes de 2 à 3 seront répétées jusqu'à qu'un critère d'arrêt est satisfait.

### III.6. Algorithme de Recherche Harmonique

L'algorithme de recherche harmonique (HS) initialement développé par Zong [185] est un algorithme d'optimisation inspiré de l'observation des musiciens qui cherchent à avoir un état parfait d'harmonie ce qui est analogue à chercher l'optimalité dans un processus d'optimisation. Cet algorithme méta-heuristique est basé sur l'analogie avec le processus

d'improvisation musicale où les musiciens improvisent les pas de leurs instruments pour obtenir une meilleure harmonie [186].

Dans le HS, une nouvelle harmonie (c'est-à-dire un nouveau vecteur de solution) est générée suivant les trois règles qui sont la considération de la mémoire, ajustement de pas, sélection aléatoire, si le nouveau vecteur de solution (nouvelle harmonie) est meilleur que le pire qui a été stocké dans un vecteur de mémoire appelé HM, il le remplace. On répète le processus itératif jusqu'à ce que le critère d'arrêt soit satisfait [186] Les étapes de mise en œuvre de l'algorithme HS sont les suivantes [187], [188]:

#### 1. Initialisation

→ Générer la mémoire d'harmonie initial de manière aléatoire avec une distribution uniforme :

$$X_i^j = L_{xi} + r \times (U_{xi} - L_{xi})$$

$j = 1, 2, 3$  jusqu'à la taille de la mémoire et  $i$  est l'indice de la variable de décision,  $r$  est un nombre aléatoire compris entre 0 et 1,  $U_{xi}$  et  $L_{xi}$  sont les limites supérieures et inférieures.

#### 2. Etablissement de la mémoire d'harmonie HS

Dans cette étape les meilleures solutions enregistrées seront utilisées pour former la mémoire d'harmonie HS, un paramètre  $r_{accep} \in [0, 1]$ , appelé critère d'acceptation de mémoire d'harmonie est utilisé pour définir le nombre d'harmonie à choisir, nous utilisons  $r_{accep} = 0,7 \sim 0,95$ .

#### 3. Ajustement de pas :

Les individus de la population seront actualisés en ajustant leurs pas via l'équation suivante :

$$x_{new} = x_{old} + b_{rang} \times \varepsilon$$

$b_{rang}$ ,  $r_{pas}$  sont la largeur de bande de pas et la vitesse d'ajustement de pas respectivement,  $\varepsilon$  est un nombre aléatoire dans la plage de  $[-1, 1]$ .

#### 4. La mémoire d'harmonie HS sera mise à jour, le vecteur d'harmonie généré remplace la plus mauvaise harmonie dans le HM (mémoire d'harmonie) seulement si sa valeur de fitness est meilleure que la pire harmonie.

Les étapes 3 et 4 sont répétées, jusqu'à que le critère d'arrêt (généralement le nombre d'itérations) est atteint.

### III.7. Electromagnétisme Comme Algorithme

Electromagnétisme Comme Algorithme (ECM) est un algorithme d'optimisation stochastique inspiré du mécanisme de la théorie d'électromagnétique initialement développé par Birbal et Fang [189] L'ECM simule le mécanisme d'attraction-répulsion de la théorie de l'électromagnétisme qui est basée sur la loi de Coulomb. La force électrostatique entre deux

charges ponctuelles est directement proportionnelle à l'amplitude de chaque charge et inversement proportionnelle au carré de la distance entre elles. Les points à grande valeur de charge sont utilisés pour trouver la direction de recherche dans les itérations suivantes [190].

Dans l'ECM, chaque point est considéré comme une particule chargée qui occupe un espace dont la charge est relative à la fonction objectif. De telle charge détermine également l'ampleur de l'attraction ou de répulsion du point par rapport aux autres. Plus la valeur de la fonction objectif est meilleure plus le point a une grande valeur de charge et par conséquent une grand force d'attraction, le mécanisme de l'algorithme ECM repose sur les quatre étapes principales suivantes [190] :

### 1. Initialisation

Génération aléatoire de la population initiale des points dans les limites supérieures et inférieure, la fonction objectif  $F$  de chaque point est équivalente à sa charge.

2. **Calcul de la Force** : La procédure de calcul de  $F$  consiste à déterminer la valeur de la force exercée sur un point par les autres points. Au début, la charge  $q^i$  assignée à chaque point est :

$$q^i = \exp\left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^m (f(x^k) - f(x^{best}))}\right)$$

$m$  et  $n$  sont respectivement le nombre de population et le nombre des coordonnées du point  $x^i$  (dimension de problème).

Le vecteur total des forces  $F^i$  exercée sur chaque point est calculé en ajoutant la composante individuelle des forces,  $F_j^i$  entre chaque paire de points  $x^i$  et  $x^j$  :

$$F^i = \sum_{j \neq i}^m F_j^i, \quad i = 1, 2, \dots, m,$$

Tels que :

$$F_j^i = \begin{cases} (x^j - x^i) \frac{q^i \times q^j}{|x^j - x^i|^2} & \text{lorsque: } f(x^j) < f(x^i) \\ (x^i - x^j) \frac{q^i \times q^j}{|x^j - x^i|^2} & \text{lorsque: } f(x^i) \leq f(x^j) \end{cases}$$

### 3. Etape de déplacement

Le mouvement de point  $x^i$  est donnée par :

$$x^i = x^i + \lambda \times \frac{F^i}{|F^i|} \times RNG, \quad \text{Pour } i = 1, \dots, m \text{ et } i \neq \text{best},$$

$RNG$  est un vecteur qui définit la plage de mouvement autorisée,  $\lambda$  est un paramètre aléatoire uniformément,  $0 \leq \lambda \leq 1$ ,  $F^i$  est la somme des forces exercée sur le point  $x^i$ .  $|F^i|$  est la norme du vecteur force.

Le meilleur point  $x_{best}$  ne se déplace pas et il est porté aux itérations suivantes.

4. **La procédure de recherche locale** : Les point ayant la meilleure fonction objectif explore leur voisinage via un algorithme de recherche local en utilisant un paramètre « Delta » à définir par l'utilisateur.

Ce processus sera itéré jusqu'à l'obtention d'un critère d'arrêt satisfaisant.

### **III.8. Algorithme de Colonie d'Abeilles Artificielle**

L'algorithme de colonie d'abeilles artificielle (ABC) initialement proposé par Karaboga [191] est un algorithme d'optimisation basé sur l'intelligence des essaims.

Il simule le comportement de recherche de nourriture des abeilles [191]. Les abeilles recueillent le nectar des vastes zones autour de leur ruche (plus de 10 kilomètres).

La quantité de nectar dans chaque source sera communiquée avec les autres abeilles dans la ruche par une danse frétilante qui leur informe de la direction, la distance et la qualité des sources de nourriture.

Dans l'algorithme ABC, les sources alimentaires et les abeilles travailleuses sont deux composantes essentielles, la position d'une source d'alimentation représente une solution possible pour le problème d'optimisation et la quantité de nectar d'une source d'alimentation correspond à la qualité (fitness) de la solution.

Les abeilles se composent de deux groupes, les abeilles employeur et les abeilles sans emploi (types spectateurs et scouts). Les spectateurs choisissent une source de nourriture en observant la danse des abeilles employeur alors que les scouts cherchent de nouvelle source de nourriture. Une abeille employeur dont la source de nourriture est épuisée devient un scout [191].

La stratégie de l'algorithme est basée sur la localisation et l'exploration des bons sites dans un espace de recherche. Les abeilles dites scouts sont envoyées d'une façon aléatoire dans l'espace de recherche et localisent les bons sites qui seront exploités avec l'application d'une stratégie de recherche locale.

Un petit nombre de bons sites sera exploité plus que les autres. Ainsi, à chaque itération, des scouts seront envoyés pour chercher de nouveaux sites [192]. La procédure de l'algorithme ABC peut se résumer comme suit :

**1. Initialisation**

Génération de la population initiale des individus :

$$x_{i,j} = x_{j,min} + rand \times (x_{j,max} - x_{j,min})$$

$i=1.....NP$  (nombre de (population),  $j=1.....D$  (dimension de problème),

$x_{j,min}$  et  $x_{j,max}$  sont les limites inférieure et supérieure de la  $j^{ième}$  variable de décision.

**2. Phase des abeilles employeurs**

La fonction objectif est calculée, les bons sites seront exploités et enregistrés dans la mémoire des abeilles employeurs, ainsi ils cherchent dans le voisinage des bons sites,

$$v_{i,j} = x_{i,j} + \emptyset \times (x_{i,j} - x_{k,j})$$

$x_{k,j}$  est une solution choisie aléatoirement,  $v_{i,j}$  la nouvelle solution et  $\emptyset$  est un nombre aléatoire choisi dans l'intervalle  $[-1,1]$ . La nouvelle solution obtenue est comparée avec la première et la meilleure sera conservée.

**3. Phase des abeilles spectateurs**

La probabilité qu'une abeille spectateur poursuive une source  $i$  est

$$P_i = \frac{fit(i)}{\sum_{n=1}^{SN} fit(n)}$$

$fit(i)$  est la valeur de fitness de l'individu  $i$  et  $SN$  est le nombre de sources de nourriture, une nouvelle source est choisie pour le point  $x_i$  et la meilleure entre les deux est conservée.

**4. Phase des abeilles scoutes**

Les sources de nourriture (solution) ou abeilles employeur qui n'évoluent pas au cours d'un nombre prédéterminé d'itérations appelé limite seront abandonnées et transformées à des abeilles scoutes et on commence de nouveau la recherche aléatoire de nouvelles solutions.

Le processus est répété pour un nombre prédéfini de cycles.

**III.9. Algorithme de la Concurrence des Impérialistes**

L'algorithme de la concurrence impérialiste (ICA) est une nouvelle approche d'optimisation globale qui simule le processus de la concurrence des pays puissants impérialistes pour dominer les colonies [193], [194]. La concurrence impérialiste entraînera progressivement une augmentation du pouvoir des empires puissants et une diminution du pouvoir des plus faibles jusqu'à leur effondrement.

Ce processus est continué jusqu'à ce que tous les pays convergent vers un état dans lequel il n'existe qu'un seul empire au monde et tous les autres pays soient ses colonies. Dans ce monde nouveau idéal, les colonies auront les mêmes positions et pouvoir que l'impérialiste.

Dans l'ICA, la population initiale composée d'individus nommés pays est ensuite séparée en deux sortes différentes selon leur fonction objectif impérialiste ou coloniale. Tous les

impérialistes et colonies se rassemblent pour former les empires selon la domination. Les colonies sont divisées en empires proportionnellement à leurs valeurs de la fonction objectif.

Après avoir divisé toutes les colonies entre les impérialistes, les colonies commencent le mouvement vers leurs pays impérialistes. Le pouvoir total de tous les empires dépend de la puissance des pays impérialistes et du pouvoir de leurs colonies. Ce fait est défini avec la puissance totale d'un empire par la puissance du pays impérialiste plus un pourcentage de la puissance moyenne de leurs colonies [194]. La procédure de l'ICA est définie comme suit [195] :

1. Génération de la population initiale des pays aléatoirement.

Les meilleurs individus (avec les coûts minimum) sont choisis comme impérialistes, les autres pays sont des colonies. Les colonies sont distribuées entre les impérialistes comme suit :

$$NC_n = rand \{p_n \cdot (N_{col})\}$$

Avec  $P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right|$ , le pouvoir d'une empire,  $C_n = \max(c_i)_{i=\{1,2,\dots,N_{imp}\}} - cost_n$ , est son coût normalisé,  $cost_n$  est le coût du  $n^{\text{ième}}$  impérialiste,  $\max(c_i)$  est l'empire avec la valeur de fonction objectif la plus grande et  $N_{col}$  est le nombre de toutes les colonies.

2. Pour augmenter la recherche autour de l'impérialiste, une quantité de déviation aléatoire est ajoutée à la direction du mouvement de la colonie comme suit :

$$P_{colonie} = P_{colonies} + \beta \times (Position_{empire} - Position_{colonies})$$

Où  $\beta$  est un paramètre qui ajuste l'écart par rapport à la direction originale une valeur de 1.5 est choisie.

Compte tenu de la fonction de coût, l'impérialiste et la colonie changent de position. Ensuite, les autres colonies se déplacent vers cette nouvelle position.

3. Le pouvoir des empires les plus faibles diminue progressivement et le pouvoir des plus puissants augmente. Quelques-unes des colonies les plus faibles des empires les plus faibles (généralement une) sera à la possession d'une autre avec une probabilité  $P$ .

Le pouvoir total d'un empire est calculé comme suit :

$$TC_n = f_{cost}^{(imp,n)} + \xi \cdot \frac{\sum_{i=1}^{NC_n} f_{cost}^{(col,i)}}{N C_n}$$

Où  $TC_n$  est le coût total du  $n^{\text{ième}}$  empire et  $\xi$  est un nombre positif inférieur à 1. La valeur de 0,1 pour  $\xi$  est une bonne valeur dans la plupart des implémentations, le coût total normalisé est défini comme :  $NTC_n = TC_n - \max_i(TC_i)$

$NTC_n$  est le coût total normalisé du  $n^{\text{ième}}$  empire ayant le coût total normalisé, la probabilité de possession de chaque empire est évaluée par :

$$P_n = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right|$$

Le processus de 2 à 3 sera répété jusqu'à ce que le critère d'arrêt soit satisfait.

### III.10. Algorithme d'Optimisation Basé sur la Biogéographie

L'algorithme d'optimisation basé sur la biogéographie (BBO) introduit par Simon est un algorithme d'optimisation évolutionnaire stochastique inspiré de la science de la biogéographie [196]. Il imite le processus de migration des animaux ou d'autres espèces. Certaines îles ont la tendance à accumuler plus d'espèces que d'autres parce qu'elles possèdent certaines caractéristiques environnementales plus appropriées pour le maintien de ces espèces que des îles avec moins d'espèces.

Cette capacité à maintenir un plus grand nombre d'espèces peut être calculée par une mesure de fitness que l'on peut quantifier en attribuant un indice de qualité ( $ISI$ ) à chaque île qui dépend aussi de nombreux indices de qualité ( $SIV$ ) [197]. Les espèces dans une île ont une tendance d'émigration désigné par un indice d'émigration  $\mu$ , ainsi qu'un taux d'immigration  $\lambda$ .

Afin d'appliquer le concept du BBO à un problème d'optimisation, les  $n$  rang ( $SIV_1, SIV_2, \dots, SIV_n$ ) associé à la caractéristique d'une île est considérée comme une solution du problème. En d'autres termes, l'ensemble de tous ces  $n$  rang est l'espace de recherche à partir duquel une solution optimale va être déterminé.

La valeur de  $ISI$  pour une île donnée est considérée comme la valeur de la fonction objectif associée à la solution. Chaque  $SIV$  (Caractéristique) dans chaque île (solution) est choisie pour être immigrer de manière probabiliste selon le taux de migration. Si l'immigration est choisie pour un  $SIV$  donné, alors l'île émigrant est sélectionnée de façon probabiliste en utilisant  $\mu$ . Après l'opération de migration, une opération de mutation est appliquée de manière probabiliste à l'île pour accroître la population [90].

Les étapes principales de l'algorithme BBO sont les suivants [90], [91]:

1. Initialisation de la population initiale avec une taille  $N$  de candidats ou chaque un est constituée d'un nombre  $D$  de  $SIV$ .
2. Calcul de la fonction objectif ( $I/S$ ) pour chaque candidat.
3. Calcul des taux d'immigration  $\lambda$  et d'émigration  $\mu$  de chaque candidat.

$$\mu = E \times \left( \frac{S}{S_{max}} \right), \quad \lambda = I \times \left( 1 - \frac{S}{S_{max}} \right)$$

$I$  est le taux d'immigration maximal possible lié à zéro espèce sur l'île,  $E$  est le taux maximal d'émigration possible correspondant à l'île avec le plus grand nombre d'espèces  $S_{max} = N$  (nombre de population).

4. Sélectionnez l'île  $x_i$  à modifier en fonction du taux d'immigration  $\lambda$ , et l'île  $x_j$  à partir desquelles  $SIV$  va être émigré en fonction du taux d'émigration  $\mu$ .  
Remplacer la variable de décision  $SIV$  sélectionnée au hasard  $x_i$  par son variable correspondant en  $x_j$ .
5. Effectuer une mutation basée sur la probabilité de mutation de chaque île :

$$M_s = M_{max} \times (1 - P_s) / P_{max}$$

$M_{max}$  est un paramètre défini par l'utilisateur,  $P_s$  la probabilité de mutation de l'île  $s$  et  $P_{max}$  est la probabilité maximale de tous les candidats.

Si le critère d'arrêt n'est pas satisfait on revient l'étape 4.

### III.11. Algorithme des Lucioles

L'algorithme de luciole (FA) est inspiré de la phase d'accouplement du cycle de vie de luciole [198]. Les lucioles (Firefly) sont des petits coléoptères ailés capables de produire une lumière clignotante froide. La lumière produite par un processus bioluminescence a des flashes dont les deux fonctions fondamentales sont la communication et l'attraction des proies potentielles.

En outre, ces clignotants peuvent également servir dans le mécanisme d'avertissement et de protection. Le flash rythmique, le taux de clignotement et la quantité de temps font partie du système de signal qui réunit les deux sexes [199].

Pour des problèmes d'optimisation, les lucioles ayant des grandes valeurs de luminosité et d'attraction attire les autres lucioles. La luminosité  $I$  est la valeur de la fonction objectif d'une luciole à un emplacement particulier  $x$  notée  $I(x)$ , alors que l'attractivité  $\beta$  est relative.  $B$  varie avec la distance  $r_{i,j}$  entre les deux lucioles  $i$  et  $j$ , l'intensité lumineuse diminue avec l'augmentation de la distance par rapport à sa source alors que l'attractivité varie avec le degré d'absorption de la lumière dans l'environnement.

L'algorithme FA est résumé comme suit : [199], [200] :

1. Générer une population initiale de lucioles  $x_i (i=1\dots, d)$ , et déterminer les intensités de lumière  $I_i$  à  $x_i$  via  $f(x_i)$ .
2. Pour chaque deux lucioles  $i$  et  $j$ , l'attractivité  $\beta_{i,j}$  est calculée en fonction de la distance entre elles  $r_{i,j}$ ,

$$\beta_{i,j} = \beta_0 \times e^{-\gamma \cdot r_{i,j}}$$

$\gamma$  est un coefficient d'absorption lumineuse fixe et  $\beta_0$  est l'attractivité pour  $r_{i,j}=0$ .

La distance entre deux lucioles  $i$  et  $j$  à est donné comme suit :

$$r_{i,j} = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$$

$x_{i,k}$  est le  $k^{\text{ème}}$  composant de la  $i^{\text{ème}}$  luciole.

Le mouvement d'une luciole attirée vers une autre plus attrayant (plus lumineuse)  $j$ , est:

$$x_i = (1 - \beta_{i,j}) \times x_i + \beta_{i,j} \times x_j + \alpha \times (\text{rand} - 1/2)$$

$\text{rand}$  est la fonction générateur de nombres aléatoires dans  $[0, 1]$  et  $\alpha$  est le paramètre de randomisation.

3. Si  $I_j > I_i$ , déplacer la luciole  $i$  vers  $j$  avec l'attractivité  $\beta_{i,j}$ , Sinon sera déplacé aléatoirement.
4. Evaluation de la nouvelle solution  $I_i$ , et conserver la meilleure.

Ce procès est répété pour un nombre d'itération maximal ou l'atteinte d'un critère d'arrêt satisfaisant.

### III.12. Algorithme de la Ligue des Championnats

L'algorithme de la ligue des champions (LCA) est un algorithme évolutionnaire inspiré de la concurrence des équipes de sport dans la ligue des champions sportive [201].

Un certain nombre d'individus représente des équipes de sport qui concurrencent entre eux dans une ligue artificielle pendant plusieurs semaines (itérations). Basé sur le programme de la ligue en chaque semaine, les équipes jouent par paires d'équipes et les résultats sont déterminés en termes de victoire ou de perte. Le niveau de force de jeu d'une équipe est déterminé en se basant sur leur évaluation de la fonction objectif [201].

Dans la période de récupération, chaque équipe conçoit les changements nécessaires dans leur formation / style de jeu (une nouvelle solution) et la concurrence reconduit pour la prochaine semaine. Avec cette nouvelle configuration des équipes, le championnat se poursuit pendant un certain nombre de saisons (critère d'arrêt).

Le LCA est un algorithmique d'optimisation évolutionnaire basée sur l'interaction entre les différentes particules d'une population sur un espace de recherche continue. Une caractéristique commune pour tous les algorithmes basés sur la population comme le LCA est qu'ils tentent de déplacer une population de solutions possibles à des zones prometteuses de l'espace de recherche [202]. Au cours des itérations, l'objectif est d'améliorer l'aptitude de l'ensemble des individus. Un tel résultat est obtenu en simulant le mécanisme de sélection qui régit en soutenant la survie des individus les plus forts selon la théorie Darwinienne. La sélection dans l'algorithme LCA est une sélection gourmande qui remplace la meilleure formation actuelle avec une formation d'équipe plus productive ayant une meilleure résistance à jouer, L'algorithme LCA est résumé comme suit [201]:

1. Génération de la population initiale (L) des équipes et le nombre des saisons (S).  
On utilise  $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$  pour représenter la formation de l'équipe  $i$  pendant la semaine  $t$  (itération).
2. Déterminer la force de jeu de chaque équipe  $F(X_i^t)$  qui est la valeur de la fonction objectif, ainsi que la formation de chaque équipe.
3. Déterminer l'équipe gagnant / perdant : chaque équipe est marquée par 3 points pour la victoire, 0 pour la perte et 1 pour l'égalité. Nous abordons la meilleure formation pour l'équipe  $i$ , connu jusqu'à la semaine  $t$  par :

$$B_i^t = (b_{i1}^t, b_{i2}^t, \dots, b_{in}^t)$$

4. Pour chaque équipe  $i$ , concevoir une nouvelle formation à sa prochaine rencontre à la semaine  $t+1$ . Puis évaluer la force de jeu avec la formation résultante. Si la nouvelle formation est le plus fort, envisager la comme meilleure formation actuelle de l'équipe  $i$ .

On revient à l'étape 3 jusqu'à que le critère d'arrêt est satisfait.

### III.13. Algorithme d'Explosion des Grenades

La méthode d'explosion de grenade (GEM) est une technique métaheuristique basée sur le concept de population [203].

Le GEM est inspiré du phénomène d'explosion de grenade ou chaque grenade génère des shrapnels qui endommagent tous objet qui se trouvent dans la zone d'explosion.

L'idée de l'algorithme est basée sur l'observation de l'explosion de grenade dans laquelle les morceaux jetés des shrapnels endommagent les objets proche de la location de l'explosion. La perte provoquée par chaque morceau de shrapnel est calculée. Une valeur élevée pour cette dernière dans un secteur donnée indique l'existence des objets objectifs valables dans ce secteur. Alors et pour causer plus de dégât, la prochaine grenade est jetée vers les zones qui ont enregistré plus de dommages. [204].

Bien que des objets près de l'endroit de la grenade soient les plus envisagés pour être endommagés, la probabilité de la destruction est encore gardée pour des objets plus lointains en choisissant une valeur élevée pour la longueur de l'explosion ( $L_e$ ) [203].

Le processus de l'algorithme GEM est résumé comme suit :

1. Sélectionner les paramètres de problèmes (nombre de grenade, nombre des shrapnels, le rayon de territoire ( $Rt$ ), la plage d'explosion le long de chaque coordonnée ( $L_e$ ).
2. Générer un nombre ( $N_g$ ) de grenades dans des endroits aléatoires dans un espace à  $n$  dimensions  $X_i \in [-1, 1].n$ , ( $i = 1, \dots, N_g$ ), qui sont éloignées les unes des autres par une distance d'au moins ( $Rt$ ).
3. Classer les grenades en fonction de leur valeur de fitness (La meilleure grenade est la première).
4. Générer un nouveau point  $X'_i$  autour d'une grenade selon l'équation :

$$X'_i = \{X_i + \sin(r_m) \times |r_m|^p \times L_e\}, \quad j = 1, 2, \dots, N_g$$

$r_m \in [-1, 1]$ , et  $p = cte$ .

Chaque collision en dehors de la zone faisable est transportée vers un nouvel emplacement à l'intérieur de la région réalisable. Si ( $X'_i$ ) est en dehors de  $[-1, 1]^*n$ , la transporter dans  $[-1, 1]^*n$  selon l'équation :

$$\text{Si } X'_i \notin [-1, 1] \times n \rightarrow B'_i = r_i \times \left( \frac{X'_i}{|X'_i(\max)|} - X'_i \right) + (X'_i)$$

$B'_i$  et la nouvelle position dans le domaine faisable, et  $r_i$  est un nombre aléatoire dans l'intervalle  $[0, 1]$ .

5. Calculer la valeur de fitness des nouvelles positions (grenade) générées, et les comparer avec l'ancien et la meilleure sera conservée.
6. La longueur de l'explosion  $L_e$  et le rayon de territoire ( $Rt$ ) seront mise à jour.

On revient à l'étape 4 jusqu'à ce que le critère d'arrêt soit satisfait.

### III.14. Algorithme de recherche Gravitationnelle

L'algorithme de recherche gravitationnelle GSA est un algorithme d'optimisation stochastique basé sur la loi de la gravité et des interactions de masse [205]. Il est basé sur la gravité newtonienne dont l'énoncé est que chaque particule dans l'univers attire toutes les autres particules avec une force qui est directement proportionnelle au produit de leurs masses et inversement proportionnelle au carré de la distance entre elles.

Dans le GSA, les individus sont considérés comme des objets dont la fonction objectif est mesurée par leurs masses. Les objets avec une meilleure fonction objectif ont par conséquent une masse importante et attire les autres objets [206]. Tous les individus s'attirent

par la force de gravité qui provoque un mouvement global de tous les objets vers ceux ayant les masses les plus lourdes. Ainsi, la direction de recherche s'oriente vers les emplacements des solutions optimales.

Les objets ayant une grande masse exercent alors un mouvement lent et par conséquent leur processus de recherche est local. Contrairement les masses légères qui correspondent à des mauvaises solutions se déplacent vers les masses lourdes beaucoup plus rapidement ce qui garantit l'étape d'exploitation de l'algorithme. Alors, le GSA qui est un algorithme sans mémoire fonctionne efficacement comme les algorithmes ayant une mémoire [84], leur procédure est comme suit [205] :

1. Génération de la population initiale des individus :

$$X_i = (x_i^1, \dots, x_i^d), \text{ pour } i=1, 2, 3 \dots N$$

$N$  et  $d$  sont respectivement la taille de population et la dimension du problème.

2. Evaluation de la fonction objectif pour chaque individu (masse de chacun).
3. Evaluation de la force gravitationnelle exercée par la masse  $i$  ( $M_i$ ) sur la masse  $j$  ( $M_j$ ) au moment  $t$  :

$$F_{i,j}(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (x_j(t) - x_i(t))$$

$G(t)$  est la constante de gravitation à l'itération  $t$ ,  $\varepsilon$  est constante ( $\varepsilon \ll$ ), et  $R_{ij}(t)$  est la distance entre les objets  $i$  et  $j$  :

$$R_{ij}(t) = (\text{norm}(\|x_i(t) - x_j(t)\|))^{R_{\text{power}}}$$

4. Calcul de la force totale  $F_i(t)$  agissant sur l'individu :

$$F_i^{\text{total}}(t) = \sum_{j \neq i}^N \text{rand}_j * F_{i,j}(t)$$

$\text{rand}_j$  est un nombre aléatoire dans l'intervalle  $[0, 1]$ , L'accélération de chaque individu  $i$  ayant une masse  $M_i(t)$  est donnée par :

$$a_i(t) = \frac{F_i^{\text{total}}(t)}{M_i(t)}$$

5. Calcul de la position  $x_i^d$  et de la vitesse  $v_i^d$  suivantes d'un agent :

$$\begin{cases} v_i(t+1) = \text{rand}_i \times v_i(t) + a_i(t) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases}$$

$G_0$  est la valeur initiale de gravité,  $\alpha = \text{cte}$  choisie par l'utilisateur,  $t$  est l'itération courante et  $T$  le nombre total des itérations.

6. Calcul des masses à l'aide de la fonction objectif et les mettre à jour par les équations suivantes :

$$m_i(t) = \frac{\text{fit}_i(t) - V_w(t)}{V_b(t) - V_w(t)}, \quad M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}$$

$\text{fit}_i(t)$  est la valeur de la fonction objectif de l'individu  $i$  à l'instant  $t$ ,  $V_b(t)$  est la meilleure valeur de la population alors que  $V_w(t)$  est la mauvaise valeur.

Si le critère d'arrêt est atteint, le processus s'arrête, sinon on revient à 2.

### III.15. Algorithme d'Optimisation de Mouvement de Vent

L'algorithme d'optimisation de mouvement de vent (WDO) est inspiré du procès de mouvement de vent [207] Les équations fondamentales du WDO sont inspirées du comportement du vent dans l'atmosphère terrestre en se basant sur l'analyse du mouvement d'une parcelle infinitésimale.

Quatre forces peuvent être la cause du déplacement du vent dans une certaine direction ou déviation de son chemin existant. Elles sont la force du gradient de pression, la force de frottement, la force de gravité et la force de Coriolis [208].

Le WDO est une technique itérative utilisée pour l'optimisation globale des problèmes multidimensionnelle. Une population de parcelles infinitésimales d'air est distribuée à travers un espace du problème à N dimensions. Les vitesses sont assignées au hasard de telle sorte que les positions des parcelles d'air sont mises à jour à chaque itération sur la base des équations physiques qui régissent le mouvement atmosphérique à grande échelle [209].

L'algorithme WDO est décrit comme suit [207]:

1. Génération des paramètres initiaux, taille de la population, nombre maximal des itérations, différents coefficients utilisés et définition de la fonction de pression (fonction objectif).
2. Attribution des positions et des vitesses initiales à chaque parcelle d'air.
3. Evaluation de la fonction objectif pour chaque parcelle d'air, la position et la pression de la particule et définition de la meilleure valeur (fitness value).

4. Actualisation de la vitesse et la position des particules,

$$\begin{cases} u_{new} = (1 - \alpha) \times u_{old} - g \times (x_{old}) + \left|1 - \frac{1}{i}\right| \times R T \times (x_{max} - x_{old}) + \frac{c \times u_{old}^d}{i} \\ x_{new} = x_{old} + (u_{new} \times \Delta t) \end{cases}$$

$\alpha$ ,  $g$ ,  $R$ ,  $T$  et  $c$  sont respectivement le coefficient de frottement, la constante gravitationnelle, la constante universelle des gaz, la température et constante qui représente la rotation de la terre,  $i$  est le nombre d'itération,  $x_{max}$  la position du meilleur individu et  $u_{old}^d$  est une autre composante du vecteur de vitesse de la particule  $x$ . Pour simplifier,  $\Delta t = 1$ ,

5. Pour chaque dimension, le WDO permet aux parcelles d'air de se déplacer seulement dans leurs limites autorisées :

$$U_{new}^* = \begin{cases} U_{max} & \text{if } U_{new} > U_{max} \\ -U_{max} & \text{if } U_{new} < -U_{max} \end{cases}$$

Cette procédure sera répétée jusqu'à qu'un nombre max des itérations est atteint.

### III.16. Algorithme d'Optimisation des Chauves-Souris

L'algorithme d'optimisation des chauves-souris (BA) est une méthode métaheuristique inspirée du comportement des micros chauves-souris lors de la recherche de la nourriture et

qui utilisent l'écholocation pour se guider en générant des ondes sonores avec certaines impulsion et fréquences [210]. En cas de rencontre d'objets, les ondes seront déviées et réfléchies portant ainsi l'information sur la distance et l'emplacement des objets (proies). Lorsque les chauves-souris rapprochent de l'objet, l'impulsion augmente et la fréquence diminue.

Pour une itération prescrite  $t$  du BA, un ensemble de chauve-souris est généré où chacun est défini par sa position, sa vitesse, sa fréquence, longueur et taux d'émission de l'impulsion. La longueur d'onde, la fréquence et la vitesse d'émission de leurs impulsions peuvent être automatiquement ajustés en fonction de la proximité de leur cible. Leurs positions sont ainsi actualisées en fonction de l'emplacement de la chauve-souris le plus proche de la cible (meilleure valeur de fitness) [211]. Il semble que le BA essaie de fusionner les principales caractéristiques des deux algorithmes d'optimisation essaim des particules (PSO) et le recuit simulé. L'algorithme BA peut être clarifié par le schéma suivant [210], [111]:

1. initialisation de la population des chauves-souris et attribution à chaque individu  $i$  la position  $x_i$ , la vitesse  $v_i$ , la fréquence d'impulsion  $f_i$ , le taux d'émission  $r_i$  et la longueur d'impulsion  $A_i$ .

2. Mise à jour des positions et fréquence :

$$\begin{cases} x_i^{t+1} = x_i^t + v_i^{t+1} \\ v_i^{t+1} = v_i^t + (x_i^t - x_*) \times f_i \\ f_i = f_{min} + (f_{max} - f_{min}) \times \beta \end{cases}$$

$\beta \in [0, 1]$  est un vecteur aléatoire tiré d'une distribution uniforme,  $f_{min} = 25kHz$ ,  $f_{max} = 150 kHz$  et  $x_*$  est le meilleur emplacement (solution globale),

Une nouvelle solution pour chaque chauve-souris est générée localement à l'aide d'une balade aléatoire locale :

$$x_{new} = x_{old} + \varepsilon \times A^t$$

$\varepsilon \in [-1,1]$  est un nombre aléatoire,  $A^t$  est la longueur d'impulsion moyenne de tous les chauves-souris à cette itération.

3. Actualisation du volume sonore  $A_i$  et du taux d'émission d'impulsions  $r_i$  au cours des itérations:

$$A_i^{t+1} = \alpha \times A_i^t, \quad r_i^{t+1} = r_i^0 \times [1 - \exp(-\gamma t)]$$

$\alpha$  et  $\gamma$  sont des constantes.

$$\text{Pour } 0 < \alpha < 1, 0 < \gamma, t \rightarrow \infty, \quad A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0$$

Pour simplifier, on prend  $\alpha = \gamma$ . Dans notre cas,  $\alpha = \gamma = 0.9$ .

Les étapes de 2 à 3 sont répétées jusqu'à ce que le critère d'arrêt est atteint.

### **III.17. La Recherche Cuckoos**

La méthode de recherche Cuckoos (CS) proposée par Yang s'inspire du comportement de reproduction d'une espèce spéciale d'oiseaux parasites de nids appelés Coucous [212]. Ces oiseaux sont fascinants, non seulement en raison des beaux sons qu'ils peuvent faire, mais également en raison de leur stratégie de reproduction agressive.

Certaines espèces telles que les cuckoos et les Guira pondent leurs œufs dans les nids d'autres espèces, bien qu'ils puissent enlever les œufs d'autres pour augmenter la probabilité d'éclosion de leurs propres œufs.

Certains oiseaux hôtes peuvent engager un conflit direct avec les cuckoos intrus. Si un oiseau hôte découvre que les œufs ne le sont pas, il va soit éliminer les œufs étrangers, soit tout simplement abandonner son nid et construire un nouveau nid ailleurs.

Certaines espèces de cuckoos telles que le tapera-parasite ont évolué de telle sorte que les cuckoos parasites féminins sont souvent très spécialisés dans le mimétisme de la couleur et le schéma des œufs de quelques espèces hôtes choisies ce qui réduit la probabilité que leurs œufs soient abandonnés et augmente ainsi leur reproductivité [213].

Dans l'algorithme CS, la population possède des œufs qu'elle met d'abord dans d'autres types de nids d'oiseaux. Certains des œufs qui ont moins de similitude (mauvaise valeur de la fitness) avec les œufs de l'oiseau hôte seront reconnus et détruits alors que les autres continuent dans les nouvelles générations [214].

En fait, les cuckoos s'améliorent et apprennent à pondre les œufs de manière continue comme les œufs d'oiseaux de l'hôte ciblé et les oiseaux hôtes apprennent à reconnaître les faux œufs. Un plus grand nombre d'œufs survivants dans chaque zone montre plus d'adéquation de cette zone et plus d'œufs survivants, plus d'attention est accordée à cette zone. En fait c'est le paramètre que l'algorithme d'optimisation cuckoo veut optimiser, l'algorithme de recherche cuckoos est décrit comme suit [214] :

**1. Générer une population initiale des œufs :**

$$E^k = (\{e_1^k, e_2^k, e_3^k, \dots, e_N^k\}),$$

tels que  $k$  est le nombre d'itération,  $N$  nombre de population, La qualité de chaque œuf  $e_i^k$  (Solution candidate) est évaluée en utilisant la fonction objectif  $f(e_i^k)$ .

**2. Génération de la nouvelle solution :**

Une nouvelle solution candidate  $e_i^{k+1}$  est produite en perturbant le courante  $e_i^k$  avec un changement de position  $c_i$  (approche des vols Lévy).

$$\begin{cases} e_i^{k+1} = e_i^k + c_i \\ c_i = 0.001 \times s_i \times (e_i^k - e^{best}) \\ s_i = \frac{u}{|v|^{1/\beta}} \end{cases}$$

$u = \{u_1, \dots, u_n\}$ ,  $v = \{v_1, \dots, v_n\}$  sont des vecteurs aléatoire à  $n$  dimensions, et  $\beta = 3/2$ , Chaque élément de  $u$  et  $v$  est calculé en considérant les distributions normales suivantes :

$$\begin{cases} u = \text{rand} \times \sigma_u \\ v = \text{rand} \\ \sigma_u = \left( \frac{\Gamma(1 + \beta) \times \sin(\pi \times \frac{\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\beta-1)/2}} \right), \Gamma: \text{distribution gamma.} \end{cases}$$

**3. Remplacement de certains nids en construisant de nouvelles solutions**

$$e_i^{k+1} = \begin{cases} e_i^k + \text{rand} \times (e_{d_1}^k - e_{d_2}^k) & \text{si } p_a > r_i \\ e_i^k & \text{si } p_a < r_i \end{cases}$$

$d_1 = 1, \dots, N$  et  $d_2 = 1, \dots, N$  sont des nombres entiers aléatoires,  $\text{rand}$  est un nombre aléatoire normalement distribué,  $r_i \in [0,1]$  est un nombre aléatoire uniforme et  $p_a$  est la probabilité de changement ( $p_a = 0.1$  dans la plupart des implémentations).

**4. Stratégie de sélection élitiste**

Comparaison des valeurs  $e_i^{k+1}$  et  $e_i^k$  et conservation de la meilleure valeur.

Les étapes de 2 à 3 sont répétées jusqu'à ce que le critère d'arrêt soit atteint.

**III.18. Algorithme Génétique avec un Processus de Croisement Multi-Parent**

L'Algorithme Génétique avec un Processus de Croisement Multi-Parent (GA-MPC) est un algorithme d'optimisation stochastique et évolutionnaire dans lequel plus de deux parents sont impliqués dans l'opération de croisement [215], il utilise les trois opérations principales des algorithmes évolutionnaire qui sont la reproduction, le croisement et la mutation. Néanmoins, il diffère des algorithmes génétiques dans l'opération de croisement où il utilise trois parents au lieu de deux. Le GA-MPC est développé pour améliorer les performances des algorithmes génétiques par l'introduction d'un croisement avec un opérateur aléatoire pour remplacer la

mutation [216], de tel croisement utilise trois parents pour générer trois enfants dont deux pour aider à l'exploitation tandis que le troisième pour assurer l'exploration. L'opérateur de randomisation peut aider à s'éloigner de l'optimal local ainsi que la convergence prématurée. Le croisement devrait avoir la capacité d'exploiter les informations sur l'espace de recherche dans la génération de nouveaux individus dont leur répartition ne doit pas être extrêmement étroite ou extrêmement larges par rapport à leurs parents [217]. L'algorithme du GA\_MPC est décrit comme suit [215]:

### 1. Génération de la population initiale

$$x_{i,j} = L_j + u \times (U_j - L_j)$$

$L_j, U_j$  sont les limites inférieure et supérieure de la variable  $x_j$  et  $u \in [0, 1]$  est un nombre aléatoire.

### 2. Sélection

Remplissage d'une matrice d'archives avec une taille  $tc$  par les meilleurs individus (en se basant sur leurs valeurs de la fonction objectif).

### 3. Croisement

En utilisant trois parent successif  $(x_1, x_2, x_3)$  de la matrice archive qui sont rangé basant sur leur fitness, et qui sont différents entre eux (en cas d'égalité l'un des parents est remplacé par un autre aléatoire), trois enfants sont générés comme suit :

$$\begin{cases} O_1 = x_1 + \beta \times (x_2 - x_3) \\ O_2 = x_2 + \beta \times (x_3 - x_1) \\ O_3 = x_3 + \beta \times (x_1 - x_2) \end{cases}$$

$\beta$  est un nombre aléatoire qui suit une distribution normale avec une valeur moyenne  $u$  et une déviation standard  $\sigma$ .

Fusion des individus des archives avec ceux de la nouvelle population pour s'éloigner des optimaux locaux. Pour chaque  $o_i^{j'}$ , on génère un nombre aléatoire  $u \in [0, 1]$ . Si  $u < p$  (paramètre de probabilité),  $o_i^{j'} = x_{arch}^i$ .

Si le critère d'arrêt n'est pas atteint, on revient à l'étape 2.

## III.19. Algorithme d'Optimisation Basé sur le Processus Enseignement-Apprentissage

L'algorithme d'optimisation basé sur le processus enseignement-apprentissage (TLBO) est semblable aux algorithmes évolutionnaires dont les opérations principales sont similaires aux opérations évolutionnaire tels que la mutation et la sélection [218], [219].

L'idée principale du TLBO est la simulation du processus d'apprentissage de l'école classique basé sur deux étapes. Dans la première étape appelée phase d'enseignement, un enseignant transmet des connaissances directement à ses élèves. Dans la deuxième étape d'apprentissage, les connaissances seront partagés entre les élevés [220].

Toutefois, dans la première étape, il n'y a que les étudiants très rares qui peuvent comprendre tous les documents présentés par l'enseignant. La plupart des étudiants acceptent partiellement les informations fournis et dans certains cas l'enseignant n'aura presque aucun effet direct sur les connaissances des élèves [221]. Cependant, la possibilité pour la plupart des étudiants à acquérir de nouvelles connaissances n'est pas perdue. Au cours de la deuxième étape (phase d'apprentissage), un étudiant peut apprendre avec l'aide d'autres.

Dans l'ensemble, la quantité de connaissances transférée à un étudiant ne dépend pas seulement de son professeur mais aussi des interactions entre les étudiants par le biais de l'apprentissage [218]. Le TLBO est un algorithme évolutionnaire basé sur une population qui se compose d'élèves dans une classe et les variables de décision qui sont les cours offerts. Pendant la phase de l'enseignement, le rôle de l'enseignant est attribué à la meilleure personne ( $X_{teacher}$ ). L'algorithme tente à améliorer d'autres individus  $X_i$  en déplaçant leur position vers la position de  $X_{teacher}$  en tenant compte de la valeur moyenne des individus ( $X_{mean}$ ) [221].

L'algorithme TLBO ne possède pas des paramètres de contrôle spécifiques qui doivent être soigneusement choisis. Seuls les paramètres de contrôle communs sont nécessaires ce qui est un important distinctif par rapport aux autres algorithmes évolutionnaires [220].

L'algorithme TLBO est décrit comme suit [220]:

1. Initialisation des paramètres du problème et de l'algorithme.
2. Calculer la fonction objectif.
3. Calculer la moyenne de la population.
4. Déterminer la meilleure solution  $X_{teacher}$ .
5. La nouvelle population sera générée, les candidats (étudiants) seront mis à jour

$$X_{new} = X_i + r \times (X_{teacher} - T_F \cdot X_{mean})$$

$r \in [0,1]$  est un paramètre aléatoire,  $T_F$  est un facteur qui peut prendre les valeurs 1 ou 2 et  $X_{mean}$  est la moyenne des individus.

Calcul de la fonction objectif des nouvelles populations et conserver la meilleur.

6. Pour l'étape d'apprentissage, un individu  $X_{ii}$  choisi au hasard est utilisé pour actualiser  $X_i$ ,

$$\begin{cases} X_i^{new} = X_i + r \times (X_i - X_{ii}) & \text{si } f(X_i) < f(X_{ii}) \\ X_i^{new} = X_i + r \times (X_{ii} - X_i), & \text{si } f(X_i) > f(X_{ii}) \end{cases}$$

La fonction objectif des nouvelles populations sera calculée, est la meilleur est conservée et  $r$  est nombre aléatoire dans l'intervalle  $[0, 1]$ .

En revient à l'étape 4 jusqu'à ce que le nombre d'itérations maximal soient atteint.

### **III.20. Algorithme du Cycle de l'Eau**

L'idée de l'algorithme du cycle de l'eau (WCA) est inspirée de l'observation du processus de l'eau et comment les rivières et les ruisseaux descendent vers la mer dans la nature [222].

Une rivière ou un cours d'eau est formé à chaque fois que l'eau se déplace en descente d'un endroit à un autre. Cela signifie que la plupart des rivières sont formées en haut dans les montagnes où la neige de l'hiver ou les anciens glaciers sont en train de fondre.

L'eau dans les rivières est évaporée tandis que les plantes se déchargent (transpirer) de l'eau à travers le processus de la photosynthèse. L'eau évaporée est dispersée en l'air pour produire des nuages qui se condensent ensuite dans le temps plus froid. Ensuite, l'eau retourne à la terre sous forme de pluie. Cette procédure naturelle est connue par le nom de cycle hydrologique [223].

Les plus petites branches de la rivière sont les petits cours d'eau où les rivières commencent à se former. Ils s'appellent ruisseaux de premier ordre. Un ruisseau de second ordre est produit lorsque deux cours d'eau de premier ordre sont assemblés alors qu'un ruisseau de troisième ordre est formé lorsque deux de second ordre se joignent. Ce processus se poursuit jusqu'à ce que la rivière coule dans la mer. Enfin, toutes les rivières coulent vers la mer (à savoir l'endroit le plus en descente).

Dans l'algorithme WCA, on suppose qu'il ya des phénomènes de pluie ou de précipitation qui conduit à la création des ruisseaux. Le WCA est un algorithme basé sur la population. Par conséquent, une population initiale des variables à savoir les ruisseaux d'eau est générée de façon aléatoire. Le meilleur individu classé en termes de fonction de coût minimum ou de fitness maximale est choisi comme la mer [224]. Ensuite, et en se basant sur la valeur de la fonction objectif, un certain nombre des individus sont choisis comme des rivières alors que tous les autres ruisseaux sont appelés des cours d'eau qui coulent dans les rivières et la mer.

L'algorithme WCA utilise un concept simple de ruissellement de surface c'est-à-dire le mouvement des cours d'eau vers les rivières et les rivières à la mer.

L'étape d'évaporation et le processus de formation de pluie sont correspond à la phase d'exploration pour l'algorithme WCA décrit comme suit [223].

1. Génération de la population initiale des individus et les paramètres de l'algorithme ( $N_{sr}$  est le nombre des ruisseaux et  $D_{max}$  est taux d'évaporation, a choisie  $<0,1$ ).
2. Évaluer la fonction objectif pour chaque individu. Les individus représentant la mer, les rivières et les ruisseaux sont identifiés respectivement du meilleur au plus mauvais.
3. Tous les individus prennent de nouvelles positions et se dirigent vers la meilleure solution de sorte que les ruisseaux qui sont créés à partir des gouttes de pluie ( $X_{Stream}^{i+1}$ ) s'écoulent vers une rivière spécifique ( $X_{River}^i$ ) qui est sélectionnée à l'aide d'une distance aléatoire, et les rivières s'écoulent vers la mer ( $X_{Sea}^i$ ):

$$\begin{cases} X_{Stream}^{i+1} = X_{Stream}^i + rand \times C \times (X_{River}^i - X_{Stream}^i) \\ X_{River}^{i+1} = X_{River}^i + rand \times C \times (X_{Sea}^i - X_{River}^i) \end{cases}$$

$C \in [1,2]$  est une variable aléatoire.

4. Afin d'améliorer le taux de convergence, des ruisseaux se jettent directement à la mer via l'équation :

$$X_{Stream}^{new} = X^{new} + \sqrt{\mu} \times rand(1, Nvar)$$

Tels que  $\mu$  est un coefficient qui montre la plage de recherche près de la région du mer, une valeur de 0.1 est une valeur appropriée.

5. Évaluer la fonction objectif pour ces nouvelles positions, les cours d'eau, les rivières et la mer sont redéfinis. Il convient de mentionner que les positions de la mer, les rivières et les ruisseaux peuvent être échangés.
6. Vérifier la condition d'évaporation en utilisant :

$$|X_{Sea}^i - X_{River}^i| < d_{max}, \quad i = 1, 2, 3, \dots, N_{sr} - 1$$

$d < 0$  est un variable aléatoire dont la valeur augmente au cours des itérations,

$$d_{max}^{i+1} = d_{max}^i - \frac{d_{max}^i}{\max(iter)}$$

Si la condition d'évaporation est vérifiée, le processus de pluie se produit et les gouttes de pluie créent de nouveaux cours d'eau dans différents emplacements de l'espace de recherche en utilisant l'équation suivante :

$$X_{Stream}^{new} = LB + rand \times (UB - LB)$$

Où  $UB$  et  $LB$  sont les limites supérieure et inférieure du problème.

7. Evaluer la fonction objectif pour cette nouvelle population.

Si le critère d'arrêt (un nombre maximum des itérations) est atteint, on arrête l'algorithme. Sinon, on revient à l'étape 3.

### III.21. Algorithme de Recherche Différentielle

L'algorithme de recherche différentielle (DSA) est un algorithme d'optimisation évolutionnaire inspiré du mouvement brownien utilisé par un organisme lors de migration [225]. Beaucoup d'espèces des êtres vivants montrent un comportement de migration saisonnier tout au long de l'année qui s'explique par le fait que la capacité et l'efficacité des zones alimentaires existant dans la nature (pâturages, approvisionnement en eau...etc.) varient

souvent en raison des changements climatiques périodiques au cours de l'année. Par conséquent, le comportement migratoire permet aux êtres vivants de passer d'une zone à capacité et diversité de sources naturelles épuisées à une zone plus efficace.

Dans le mouvement migratoire, les espèces migratrices constituent un super organisme contenant un grand nombre d'individus. Ensuite, le super organisme commence à changer de position en se dirigeant vers les zones les plus fructueuses [225].

Au cours de ce processus, le super organisme artificiel vérifie quelles positions choisies au hasard peuvent être conservées temporairement. Si une position testée est apte à être conservée pendant un certain temps, le super organisme artificiel utilise ce modèle de migration pour s'installer à la position découverte puis poursuit sa migration à partir de cette position [226]. La procédure de résolution par DSA est résumée dans l'algorithme suivant [226]:

1. Génération d'un super organisme artificiel  $X$  de dimension  $(NP \times D)$  contenant  $NP$  organisme  $x_i$  limités par les contraintes minimale et maximale suivantes :

$$\begin{cases} x_{min} = \{x_{1,min}, x_{2,min}, \dots, x_{D,min}\} \\ x_{max} = \{x_{1,max}, x_{2,max}, \dots, x_{D,max}\} \end{cases}$$

La  $i^{\text{ème}}$  composante du  $j^{\text{ème}}$  vecteur est générée par :

$$x_{j,i} = x_{j,min} + rand_{i,j} \times (x_{j,max} - x_{j,min}), \text{ ou } rand_{i,j} \in [0,1], i=1,\dots, NP \text{ et } j=1,\dots,D.$$

2. Evaluation de la fonction objectif pour chaque individu  $x_i$ .
3. Génération d'un vecteur d'escala pour la population  $X_i$  comme suit :

$$S_{i,G} = X_{i,G} + Scale \times (X_{r1,G} - X_{i,G})$$

$Scale$  est généré par un générateur de nombres aléatoires dans l'intervalle  $[0,1]$ ,  $G$  indique l'itération alors que  $X_{r1,G}$  est le vecteur de la population dont l'ordre des composants est aléatoirement changé.

Si l'un des sites escale est trouvé en dehors de la zone faisable, il sera aléatoirement remplacé par un autre.

4. Recherche du site d'escala (nouvelle solution). Le processus est décrit comme suit :

$$s'_{i,j,G} = \begin{cases} s_{i,j,G} & \text{si } r_{i,j} < rand \\ X_{i,j,G} & \text{si } r_{i,j} = 0 \end{cases}$$

$j = [1, \dots, D]$ ,  $r_{i,j}$  est nombre aléatoire prend des valeurs entre 0 ou 1, et  $s'_{i,j,G}$  désigne le vecteur Scale de la  $j^{\text{ème}}$  particule dans la  $i^{\text{ème}}$  dimension à la  $G^{\text{ème}}$  itération.

5. Effectuer la sélection en se basant sur l'expression suivante :

$$\begin{cases} X_{i,j,G+1} = s'_{i,j,G} & \text{si } f(s'_{i,j,G}) \leq f(X_{i,j,G}) \\ X_{i,j,G+1} = X_{i,j,G} & \text{si } f(s'_{i,j,G}) > f(X_{i,j,G}) \end{cases}$$

Mémoriser la meilleure solution actuelle.

Si le critère d'arrêt n'est satisfait on revient à l'étape 3.

### III.22. Algorithme d'Explosion des Mines

L'algorithme d'explosion des mines (MBA) est une méthode d'optimisation développée et utilisée pour traiter les problèmes d'optimisation complexes dont les concepts fondamentaux et les idées sont dérivés de l'explosion des mines dans le monde réel où les morceaux de shrapnel jetés suite à l'explosion d'une mine se heurtent avec d'autres mines près de l'endroit d'explosion en provoquant leur explosion [228], [229].

La stratégie du MBA est basée sur la localisation de la mine avec le plus grand effet explosif. Chaque morceau de shrapnel a des directions et des distances définies pour entrer en collision avec d'autres mines. Les pertes causées par l'explosion d'une mine est équivalent à la valeur de la fonction objectif (la valeur de fitness).

La procédure de résolution par MBA est résumée dans l'algorithme suivant [229] :

1. Génération des paramètres initiaux : nombre des morceaux de shrapnel  $N_s$ , le facteur d'exploration  $\mu$ , la constante de réduction  $\alpha$ .
2. Génération de la population initiale :  $X_0 = LB + rand \times (UB - LB)$
3. Génération de la nouvelle population : l'explosion d'une mine à l'endroit  $X_{e(n+1)}^f$  va générer des morceaux de shrapnels à une distance  $d_{n+1}^f$  qui causent l'explosion d'une autre mine à l'endroit  $X_{n+1}^f$  qui va causer leur explosion (déplacement à un autre endroit  $X_{n+1}^f$ ) et  $m_{n+1}^f$  et la différence de fonction objectif entre la mine explosif et le nouvel endroit :

$$\left\{ \begin{array}{l} X_{n+1}^f = X_{e(n+1)}^f + \exp\left(-\sqrt{\frac{m_{n+1}^f}{d_{n+1}^f}}\right) \times X_n^f \\ X_{e(n+1)}^f = d_{n+1}^f \times rand \times \cos(\theta) \end{array} \right. , \left\{ \begin{array}{l} d_{n+1}^f = \sqrt{(X_{e(n+1)}^f - X_n^f)^2 + (F_{n+1}^f - F_n^f)^2} \\ m_{n+1}^f = \frac{F_{n+1}^f - F_n^f}{X_{e(n+1)}^f - X_n^f} \end{array} \right.$$

$rand$  est un nombre aléatoire uniformément distribué,  $\theta = 360/N_s$ , et  $F$  est la valeur de la fonction objectif pour le point  $X$ .

4. Exploration de l'espace de solution en se servant des équations suivantes :

$$si \mu > k, \left\{ \begin{array}{l} d_{n+1}^f = d_n^f \times (|rand|)^2 \\ X_{e(n+1)}^f = d_{n+1}^f \times \cos(\theta) \end{array} \right. , K \text{ est le nombre d'itération}$$

La distance initiale des morceaux des shrapnels est réduite progressivement pour donner aux mines la possibilité de chercher la location de minimum globale. La réduction dans  $d_0^f$  est donné par :

$$d_n^f = \frac{d_{n-1}^f}{\exp\left(\frac{k}{\alpha}\right)}$$

$\alpha$  dépendant de la complexité du problème et choisi par l'utilisateur a comme effet réduire la distance de chaque morceau de shrapnel.

Si le critère d'arrêt n'est pas vérifié, on revient à l'étape 2.

### III.23. Algorithme de Recherche par Retour en Arrière

L'Algorithme de recherche par retour en arrière (BSA) est un algorithme évolutionnaire développé par Civicioglu [230]. Il construit de manière incrémentielle des candidats aux solutions et abandonne chaque candidat partiel ("BACKTRACKS") dès qu'il détermine que ce candidat ne peut pas éventuellement être muni d'une solution valable, le BSA utilise un seul paramètre de contrôle. Ainsi, la performance de résolution du problème n'est pas sensible à la valeur de ce paramètre. [231], [232].

La stratégie du BSA pour générer une population d'essai comprend deux nouveaux opérateurs de croisement et de mutation. Les stratégies du BSA ont pour objet de générer des populations d'essai et de contrôler l'amplitude de la matrice de recherche. En particulier, l'algorithme BSA possède une mémoire dans laquelle il stocke une population d'une génération précédente choisie au hasard afin de générer la matrice de recherche-direction.

L'algorithme BSA est montré ci-dessous [230]:

#### 1. Initialisation :

Génération de la première population ancienne basée sur l'expression :

$$P_{ij} = U(\text{low}_j, \text{up}_j), \text{ Pour } i = 1, 2, \dots, N \text{ et } j = 1, 2, \dots, D.$$

$N$  et  $D$  sont respectivement la taille de la population et la dimension du problème et  $U$  est la distribution uniforme.

#### 2. Sélection-I

La première population historique est déterminée en utilisant l'équation suivante :

$$\text{old}P_{ij} = U(\text{low}_j, \text{up}_j)$$

Au début de chaque itération, le BSA peut redéfinir la population ancienne en utilisant l'*if-then* conditionnelle comme suit : *if*  $\alpha < \delta$ , *then*  $\text{old}P = P$ ,  $\alpha$  et  $\delta$  sont des nombres aléatoires.

Les individus dans l'ancienne population sont ensuite mélangés en permutant leurs ordres :

$$\text{old } P = \text{permuting}(\text{old}P)$$

#### 3. Croisement et mutation

Le processus de croisement comporte deux étapes distinctes. La première implique l'initialisation d'une matrice  $\text{map}_{n,m}$  de valeurs entières binaires de taille  $(N, D)$  et le deuxième est le choix du nombre d'individus à muter par un critère de croisement  $\text{DIM\_RATE}=1$ .

Le processus de mutation génère la forme initiale de la population d'essai  $T$  appelée Mutant en utilisant l'expression suivante :

$$\left\{ \begin{array}{l} \text{Mutant} = P + (\text{map} \times F) \times (\text{old}P - P), \\ F = 3 \times \text{rand} \end{array} \right. , P \text{ est la population actuelle et } \text{old}P \text{ est la population historique actuelle}$$

#### 4. Sélection-II

Comparer la nouvelle population  $T$  et l'initiale  $P$ .

Si  $\text{fitness}(T_i) < \text{fitness}(P_i)$  pour deux individus ayant le même ordre,  $P_i$  est remplacé par  $T_i$ .

Si le critère d'arrêt est satisfait le processus est arrêté. Sinon, on revient à l'étape 3.

### III.24. Algorithme d'optimisation des Trous Noirs

L'algorithme des trous noirs (BHBO) est un algorithme heuristique qui s'inspire du phénomène des trous noirs [233]. Un trou noir dans l'espace est ce qui se forme quand une étoile de taille massive s'effondre. La force gravitationnelle du trou noir est trop haute que même la lumière ne peut pas y échapper [234].

L'algorithme de trou noir (BHBO) commence par une population initiale de solutions à un problème d'optimisation et une fonction objectif qui est calculée pour eux. A chaque itération du trou noir, le meilleur candidat est choisi pour être le trou noir qui commence alors à tirer d'autres candidats appelés étoiles autour de lui. Si une étoile se rapproche trop du trou noir, elle va être avalée par ce trou. Dans ce cas, une nouvelle étoile est générée au hasard et placée dans l'espace de recherche et démarre une nouvelle recherche [235].

L'algorithme BHBO est décrit comme suit :

1. Initialiser une population d'étoiles avec des lieux aléatoires dans l'espace de recherche.
2. Pour chaque étoile, évaluer la fonction objectif.
3. Sélectionner la meilleure étoile qui sera le trou noir on se basant sur sa valeur de fitness.
4. Changer l'emplacement de chaque étoile selon l'équation suivante :

$$x_i(t+1) = x_i(t) + rand \times (x_{BH} - x_i(t)), \quad i = 1, 2, \dots, N$$

$x_i(t)$  et  $x_i(t+1)$  sont respectivement les emplacements de la  $i^{\text{ème}}$  étoile aux itérations  $t$  et  $t+1$ ,  $x_{BH}$  est l'emplacement du trou noir,  $rand \in [0, 1]$  est un nombre aléatoire et  $N$  est le nombre d'étoiles.

5. Si une étoile atteint un endroit trop proche du trou noir, son emplacement sera échangé. Si une étoile traverse l'horizon du trou noir  $R$ , on la remplace par une nouvelle étoile.

$$R = \frac{f_{BH}}{\sum_{i=1}^N f_i}$$

$f_{BH}$  est la valeur de la fonction objectif du trou noir et  $f_i$  celle de  $i^{\text{ème}}$  candidat.

Le processus est arrêté si le critère d'arrêt est atteint (maximum d'itérations ou aptitude suffisante). Sinon, on revient à l'étape 2.

### III.25. Algorithme d'Optimisation des Loups Gris

L'algorithme d'optimisation des loups gris (GWO) introduit par Mirjalili est un algorithme méta heuristique inspiré du mécanisme utilisé par les loups gris dans la chasse [237]. Les loups gris sont considérés comme des prédateurs apex ce qui signifie qu'ils sont au sommet de la chaîne alimentaire. Les loups gris préfèrent surtout vivre dans des troupeaux dont la taille du groupe est en moyenne de 5-12 individus.

Il est particulièrement intéressant qu'ils aient une hiérarchie sociale très stricte qui comprend quatre niveaux des loups, *alpha* ( $\alpha$ ), *beta* ( $\beta$ ), *delta* ( $\delta$ ) et *gamma* ( $\omega$ ). Les chefs sont un mâle et une femelle appelés *alpha*. Ils sont principalement responsables de prendre des décisions concernant la chasse et toutes autres choses concernant le groupe [238].

Le deuxième niveau dans la hiérarchie des loups gris est *bêta*. Ils sont des loups subordonnés qui aident l'*alpha* dans la prise de décision ou d'autres activités de groupe. Le *bêta* est probablement le meilleur candidat pour être l'*alpha* dans le cas où l'un des loups *alpha* disparaît ou devient très vieux [238].

Les loups du niveau *delta* appelé subordonné doivent se soumettre au *alpha* et au *bêta* mais dominant l'*oméga*. Ce dernier doit toujours se soumettre à tous les autres niveaux [239].

Afin de modéliser mathématiquement la hiérarchie sociale des loups, nous considérons la solution la plus adaptée comme *alpha*, les deuxièmes et troisièmes meilleures solutions respectivement *bêta* et *delta*. Le reste des solutions candidates sont supposées être *oméga*.

Dans l'étape d'exploration, les loups gris recherchent principalement la proie en fonction de la position de l'*alpha*, *bêta* et *delta*. Ils divergent à la recherche des proies et convergent pour attaquer des proies. La procédure de résolution par l'algorithme GWO est décrite comme suit [237] :

1. Génération de la population initiale (position des loups gris).
2. Evaluation de la fonction objectif pour chaque agent et on détermine par la suite les  $\alpha$ , les  $\beta$ , les  $\delta$ , et les  $\omega$  respectivement du meilleur au plus mauvais.
3. Les positions des trois meilleurs individus seront utilisées pour l'actualisation des positions des autres individus comme suit :
 
$$X(t+1) = (X_1 + X_2 + X_3)/3$$

$$D_\alpha = (C_1 \times X_\alpha(t) - X(t)), C_1 = 2 \times r_1$$

$$D_\beta = (C_2 \times X_\beta(t) - X(t)), C_2 = 2 \times r_2$$

$$D_\delta = (C_3 \times X_\delta(t) - X(t)), C_3 = 2 \times r_3$$

$$X_1 = X_\alpha - A_1 \times D_\alpha$$

$$X_2 = X_\beta - A_2 \times D_\beta$$

$$X_3 = X_\delta - A_3 \times D_\delta$$

$r_2$  sont des vecteurs contenant des aléatoires de l'intervalle [0, 1],  $A_1$ ,  $A_2$  et  $A_3$  sont des valeurs calculées par une distribution uniforme.
4. Evaluation de la fonction objectif pour cette nouvelle population.  $\alpha$ ,  $\beta$  et  $\delta$  seront ainsi actualisés.

Si le critère d'arrêt n'est pas atteint (généralement le nombre d'itérations), on revient à l'étape 2.

### III.26. Algorithme d'Optimisation de l'Essaim de Poulet

L'algorithme d'optimisation des essaims de poulets (CSO) est une méthode d'optimisation qui simule le comportement hiérarchique dans l'essaim de poulet [240].

En général, les comportements des poulets varient selon le sexe. Le coq tête serait positivement chercheur de la nourriture et se battra avec les poulets qui envahissent le territoire du groupe habité. Dans le même territoire, il existe des concours entre différents poulets. Quant aux poussins, ils cherchent la nourriture autour de leur mère [241].

Dans le CSO, le principe consiste à deviser l'espace de recherche sur des groupes. Les individus avec les meilleures valeurs de la fonction objectif sont les coqs alors que les autres individus sont les poulets et les poussins qui ont la plus mauvaise valeur. Les poulets suivent leur coq dans un même groupe (recherche local). Ainsi, elles peuvent examiner d'autre endroit. La nouvelle position de l'individu est déterminée en utilisant celle connue dans leur propre groupe et celles des autres groupes [242]. L'algorithme CSO est décrit comme suit :

1. Génération de la population initiale de poulets en divisant l'essaim en plusieurs groupes. Chaque groupe comprend un coq dominant, un couple de poules et de poussins définis sur la base de la fonction objectif.
2. Actualisation de la population, Les équations de mise à jour des positions des coqs sont :

$$x_{i,j}^{t+1} = x_{i,j}^t \times (1 + \text{randn}(0, \sigma^2))$$

$$\sigma^2 = \begin{cases} 1 & \text{si } f_i \leq f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \varepsilon}\right) & \text{pour } k = 1, \dots, N \text{ et } k \neq i \end{cases}$$

$\text{randn}(0, \sigma)$  est une distribution gaussienne avec une moyenne 0 et une déviation standard  $\sigma$ ,  $\varepsilon$  est une constante très petite,  $k$  est un indice de coq choisi au hasard parmi le groupe des coqs ( $k \neq i$ ),  $f_i$  est la valeur de la fonction objectif du coq  $i$ .

La mise à jour de position des poules est réalisée selon les équations suivantes :

$$X_{i,j}^{t+1} = X_{i,j}^t + C_1 \times \text{randn} \times (X_{r_1,j}^t - X_{i,j}^t) + C_2 \times \text{randn} \times (X_{r_2,j}^t - X_{i,j}^t)$$

$$C_1 = \exp\left(\frac{f_i - f_{r_1}}{|f_i| + \varepsilon}\right), \quad C_2 = \exp((f_{r_2} - f_i))$$

$\text{randn}$  est un nombre aléatoire uniforme sur  $[0, 1]$ ,  $r_1$  est l'index du coq qui est la tête groupe du poulet et  $r_2$  est l'indice du poulet qui est choisi au hasard parmi l'essaim de poulet  $r_1 \neq r_2$ .

La mise à jour de position des poussins suivant leur mère à la recherche de la nourriture s'est fait selon l'équation suivante:

$$X_{i,j}^{t+1} = X_{i,j}^t + FL \times (X_{m,j}^t - X_{i,j}^t)$$

$FL$  est un paramètre choisi par l'utilisateur généralement supposé égale à 0,2.

Les coqs, les poulets et les poussins seront identifiés en se basant sur les nouvelles valeurs de la fonction objectif.

Répétez l'étape 2 jusqu'à ce que le critère d'arrêt soit atteint.

### III. 27. Algorithme d'Optimisation JAYA

Introduit par Rao, JAYA est un algorithme d'optimisation basé sur le principe de population [243]. Cet algorithme a un caractère distinctif par rapport à d'autres algorithmes évolutifs réside dans le fait qu'il n'y a pas des paramètres de contrôle spécifiques qui doivent être soigneusement choisis et que seules les paramètres de contrôle communs (nombre de population et nombre des itérations) sont requis [244].

Son processus d'optimisation est basé sur l'idée que la solution déterminée pour un problème spécifique doit passer à la solution optimale et échapper à la l'optimum locale et cela en assurant la prise en considération de la plus mauvaise et de la meilleure solution de la population actuelle. L'algorithme basique du JAYA a une seule phase selon le concept susmentionné ce qui le rendre une technique d'optimisation simple.

Les étapes essentielles de l'algorithme JAYA sont décrites comme suit :

1. Génération aléatoire de la population initiale.
2. Evaluation de la fonction objectif pour toutes les variables de décision, enregistrer les candidats correspondants à la meilleure et à la pire solution d'optimisation et les utiliser pour mettre à jour la valeur de toute la population dans la prochaine itération ( $i+1$ ) :

$$x_{new} = x(i) + r_1 \times (X_{best} - abs(x(i))) - r_2 \times (X_{worst} - abs(x(i)))$$

$X_{best}$  et  $X_{worst}$  sont respectivement les valeurs des variables correspondants à la meilleure et à la pire solution obtenue à la  $i^{ème}$  itération.

$r_1$  et  $r_2$  sont deux nombres aléatoires générés dans l'intervalle  $[0, 1]$  pour les variables de décisions correspondant à la meilleure et à la pire solution à la  $i^{ème}$  itération.

3. Si la nouvelle variable a une meilleure valeur que la précédente, les valeurs des variables de décision sont mises à jour avec la nouvelle valeur. Sinon, on garde les valeurs précédentes.

Le processus de 2 à 3 sera répété jusqu'à ce qu'un critère d'arrêt soit atteint.

### III.28. Algorithmes d'optimisation de collision des corps

L'algorithme d'optimisation de collision des corps CBO est inspiré de l'observation du phénomène de collision des objets. Après une collision de deux corps en mouvement ayant des masses et vitesses spécifiées, ils sont séparés à de nouvelles vitesses et directions [245].

Selon leurs masses, les objets peuvent être classés en deux catégories, des objets fixes et des objets en mouvement. Les objets en mouvement se déplacent pour suivre des objets stationnaires où des collisions se produisent entre des paires d'objets afin d'améliorer leurs positions [142].

Le CBO est basée sur des collisions entre les corps. Chaque candidat de la population est considéré comme un objet avec une masse qui est définit en utilisant la fonction objectif. La vitesse et la direction des corps seront actualisées ou changées après chaque collision [246].

La procédure de l'algorithme CBO est décrite comme suit [245]:

**1. Génération de la population initiale**

$$x_i^0 = x_{min} + rand(x_{max} - x_{min})$$

**2. Calcul de la masse de chaque objet  $k$**

$$m_k = \frac{1/fitness(k)}{\sum_{i=1}^n 1/fitness(i)}$$

$fitness(k)$  est la valeur de la fonction objectif de l'objet  $k$  et  $n$  est le nombre de population.

**3. Répartition de la population en deux parties**

→ Les objets stationnaires avant collision avec les meilleures fonctions objectif avec :

$$v_i = 0, i=1... n/2.$$

→ Les objets en mouvement (non stationnaire) avec les mauvaises fonctions objectif avec :

$$v_i = x_i - x_{i-\frac{n}{2}}, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots \dots \dots n$$

$v_i$ , et  $x_i$  sont la vitesse et la position du  $i^{\text{ème}}$  objet alors que  $x_{i-\frac{n}{2}}$  est son équivalent dans le groupe des objets stationnaires.

**4. Après collision, les vitesses des objets en mouvement seront mises à jour ;**

$$v'_i = \frac{(m_i - \varepsilon \times m_{i-\frac{n}{2}}) \times v_i}{m_i + m_{i-\frac{n}{2}}}, \quad i = \frac{n}{2} + 1, \dots \dots \dots n$$

$\varepsilon$  est un paramètre d'autorisation de mouvement.

**5. Des nouvelles positions seront assignées à chaque objet**

$$\begin{cases} x_i^{new} = x_{i-\frac{n}{2}} + rand \times v'_i, & i = \frac{n}{2} + 1, \dots \dots \dots n, & \text{Pour les objets en} \\ & & \text{mouvement} \\ x_i^{new} = x_i + rand \times v'_i, & i = 1, \dots \dots \dots \frac{n}{2}, & \text{Pour les objets stationnaires} \end{cases}$$

On revient à l'étape 2 jusqu'à qu'un critère d'arrêt est atteint.

### III.29. Algorithmes d'Optimisation de Feux d'Artifice

L'algorithme d'optimisation de feux d'artifice (FWA) est un algorithme inspiré de l'observation du phénomène d'explosion d'artifice proposé par yang [247].

Lorsque qu'un feu d'artifice explose, il génère une pluie d'étincelles dans une zone adjacente. Ces étincelles explosent de nouveau et génèrent d'autres spectacles d'étincelles dans une zone plus petite. Progressivement, les étincelles vont explorer tout l'espace de recherche et se concentrer sur un petit endroit pour trouver la solution optimale [248].

Le FWA utilise une population initiale des individus où chacun explose et génère des étincelles autour de lui. Le nombre et l'amplitude d'étincelles de chaque individu sont déterminés par certaines stratégies dans laquelle, une génération gaussienne est utilisée pour générer des étincelles pour garder la diversité de la population. A chaque itération, l'algorithme garde le meilleur individu dans la population et sélectionne le reste (N-1) individus basés sur la distance pour la prochaine génération [249].

Le processus de résolution par le FWA est le suivant :

1. Génération de la population initiale (locations initiales des artifices).
2. Lorsqu'une étincelle éclate, elle donne, avant sa disparition, de nombreuses étincelles autour d'elle et donc de nouveaux individus seront générés en utilisant les formules suivantes :

$$S_i = S \times \frac{Y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^N (Y_{max} - f(x_i)) + \varepsilon}$$

$$A_i = A \times \frac{f(x_i) - Y_{min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - Y_{min}) + \varepsilon}$$

$S_i$  est le nombre d'étincelles générées par un individu de la population,  $A_i$  est l'amplitude pour un individu pour générer les étincelles d'explosion,  $S$  et  $A$  sont des constants,  $Y_{max}$  et  $Y_{min}$  sont respectivement la mauvaise et la meilleure valeur obtenues,  $f(x_i)$  est la fonction objectif de l'individu  $i$  et  $\varepsilon$  est nombre utilisé pour empêcher le dénominateur de s'annuler.

3. Application d'une stratégie de cartographie qui assure que tous les individus restent dans l'espace réalisable. Les étincelles éloignées de la frontière seront dispersés à leurs portées autorisées.
4. Garder la diversité de la population en utilisant une stratégie gaussienne de la génération des étincelles

$$x'_k = g \times x_k^i$$

$g$  est un nombre aléatoire dans la distribution gaussienne et  $x_k^i$  est la position de l'individu.

5. Pour sélectionner les individus pour la prochaine génération, le meilleur individu est toujours conservé au début. Puis les individus suivants sont sélectionnés en fonction de leur distance par rapport aux autres individus. L'individu qui est loin des autres aura plus de chance d'être sélectionné.

Ce processus est répété jusqu'à un critère d'arrêt est atteint.

### III. 30. Algorithme de Meilleur Joueur (MVPA)

L'algorithme du meilleur joueur (MVPA) est inspiré de l'observation du comportement des équipes de sport où des équipes sont formées de joueurs qui concourent individuellement et collectivement (en équipes) afin de gagner le championnat [250].

Le MVPA est donc basé sur la génération d'une population de joueurs repartis sur plusieurs équipes où chacune a son meilleur joueur dit local. En se basant sur le calcul de fonction objectif, l'ensemble des équipes ont un joueur MPV (avec meilleurs fonction objectif). Les autres joueurs améliorent leurs compétences en faisant suivre leur meilleur joueur local ainsi que le MPV (certain meilleur joueur des autres équipes est choisi arbitrairement) [251]. La procédure de l'algorithme MVPA est décrite comme suit :

1. Initialisation au hasard de la population de joueurs et équipes dans l'espace de recherche:

$$P_k = [S_{k,1}, S_{k,2}, \dots \dots \dots S_{k,D}]$$

$P_k$  désigne le joueur  $k$ ,  $S$  est les variables de décision,  $D$  est la dimension de problème.

2. Formation des équipes par répartition au hasard des joueurs. Les premières équipes ( $nT1$ ) ont un nombre de joueurs ( $nP1$ ) tandis que les autres équipes ( $nT2$ ) ont un nombre de joueurs ( $nP2$ ) calculés de la manière suivante :

$$\begin{aligned} nP1 &= \text{ceil}\left(\frac{N}{T}\right) \\ nP2 &= nP1 - 1 \\ nT1 &= N - nP2 \times T \\ nT2 &= T - nT1 \end{aligned}$$

$N$  est le nombre total des joueurs générés,  $T$  est le nombre d'équipes,  $\text{ceil}$  est une fonction qui arrondit un nombre réel au plus petit entier suivant.

3. Chaque équipe suivra deux étapes de concurrence, une compétition individuelle entre les joueurs dans la même équipe et une compétition entre équipe.

→ Les variables de décision de chaque joueur seront mises à jour comme suit :

$$P_{k,i}^{t+1} = P_{k,i}^t + \text{rand} \times (P_i^* - P_{k,i}^t) + 2 \times \text{rand} \times (MPV - P_{k,i}^t)$$

$t$  désigne l'itération,  $i$  désigne l'indice de l'équipe sectionné,  $k$  est l'indice du joueur dans l'équipe sélectionné,  $P_i^*$  est le meilleur individu dans l'équipe sélectionné et  $MPV$  est le meilleur joueur de toutes les équipes.

→ Les formations des équipes seront actualisées deux par deux. Une équipe  $i$  est actualisée en utilisant les paramètres d'une autre choisie au hasard  $j$ . Si cette dernière performe mieux que l'équipe  $i$  selon un calcul statistique basé sur la valeur de fonction objectif normalisée de chaque équipe, les joueurs de l'équipe  $i$  sont actualisés une autre fois comme suit :

$$P_{k,i}^{t+1} = P_{k,i}^t + \text{rand} \times (P_{k,i}^t - P_j^*)$$

Si l'inverse et l'équipe  $i$  performe mieux que l'équipe  $j$ , on a :

$$P_{k,i}^{t+1} = P_{k,i}^t + \text{rand} \times (P_j^* - P_{k,i}^t)$$

4. Comparaison des nouvelles solutions avec les précédentes et conservation des meilleurs.
5. Dans le cas de deux joueurs successifs de la population sont exactement les mêmes, le deuxième joueur est remplacé par un autre.

On revient à l'étape 3 jusqu'à un critère d'arrêt est atteint.

## **Conclusion**

Dans ce chapitre nous avons effectué une analyse descriptive de 30 algorithmes d'optimisation métaheuristiques les plus répondus dans la littérature.

Nous avons étudié ces algorithmes en mettant l'axe sur ces différents aspects de recherche à savoir : la source d'inspiration et leur métaphore, le principe d'optimisation et les différentes étapes optées pour l'obtention de l'optimum global.

Suite à l'étude effectuée, les algorithmes sélectionnés sont classés en cinq principales catégories :

- ✓ Algorithmes évolutionnaires : GA, DE, GA\_MPC, DSA, TLBO, BSA, BBO, LCA.
- ✓ Algorithmes basés sur la recherche directionnelle : GEM, MBA, FWA.
- ✓ Algorithme basés sur la concurrence entre groupes de particule : CSO, ICA, MVPA.
- ✓ Algorithmes basés sur l'intelligence des essaims et recherche de groupe de particules: PSO, ACO, ABC, FA, BA, GWO, CS, JAYA.
- ✓ Algorithmes basés sur les lois physiques : SA, HS, EM, GSA, WDO, WCA, BHBO, CBO.

# **Chapitre IV**

## **Comparaison entre les différentes méthodes d'optimisation et techniques appliquées**

#### **IV. Comparaison entre les performances des différentes méthodes d'optimisation et techniques appliquées**

Plusieurs techniques d'optimisation métaheuristiques sont utilisées pour résoudre les problèmes de l'OPF [252], [253]. Ces techniques sont connues pour leurs capacités à décrire la recherche de grands espaces de solutions, de trouver des optimums globaux et d'échapper de ceux locaux. En raison de la variabilité des objectifs lors de la résolution des problèmes de l'OPF, aucun algorithme ne présente les mêmes performances pour tous les problèmes. Pour cela, et afin de définir le degré de fiabilité de chaque algorithme, une évaluation comparative des performances de plusieurs algorithmes basée sur un protocole statistique rigoureux doit être effectuée.

Généralement, la manière la plus appliquée est celle basée sur l'analyse des meilleurs résultats finaux obtenus par application des algorithmes sur un ensemble prédéfini des fonctions de référence dont les minimums globaux sont connus au préalable ce qui n'est pas le cas pour les problèmes rencontrés dans l'écoulement de puissance optimale.

Actuellement, des tests statistiques sont utilisés pour améliorer le processus d'évaluation et de comparaison des algorithmes. Selon le type de données employées pour les analyses, les procédures statistiques sont regroupées en deux classes : paramétriques et non paramétriques [252]. Les tests paramétriques sont basés sur la comparaison entre les critères paramétriques des algorithmes à savoir, le minimum global, la moyenne, la médiane, le maximum et la déviation standard.

Les tests non paramétriques peuvent donner lieu à des comparaisons entre deux ou plusieurs algorithmes. L'algorithme le plus performant est mis en surbrillance avec l'application du test. Parmi les tests non paramétriques, on trouve le test de signe, le test de Wilcoxon, le test de signe multiple, le test de Friedman, le test de la page et le rapport de performance [253].

##### **IV.1. Les tests paramétriques**

Pour les tests paramétriques, il s'agit de la comparaison entre les algorithmes sur la base du minimum global, de la moyenne, du maximum et de la déviation standard.

Pour un problème de minimisation, le minimum et le maximum globaux sont respectivement la valeur la plus petite et la valeur la plus grande de la fonction objectif que l'algorithme donne après 30 exécutions.

La moyenne est le rapport de la somme des valeurs des fonctions objectifs obtenues et le nombre des exécutions effectuées.

La déviation standard permet d'évaluer la dispersion des mesures autour de la valeur moyenne. Elle est donnée par [252] :

$$\delta = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - x_m)^2} \quad (\text{IV.1})$$

Ici  $N$  est le nombre d'exécutions,  $x_i$  est la valeur optimale de la fonction objectif de la  $i^{\text{ème}}$  exécution et  $x_m$  est la valeur moyenne de la fonction objectif pour les  $N$  exécutions.

## IV.2. Les tests non paramétriques

Dans le domaine des statistiques différentielles, les tests d'hypothèses peuvent être utilisés pour tirer des conclusions sur une ou plusieurs populations à partir d'échantillons donnés (résultats). Deux hypothèses sont définies, hypothèse nulle  $H_0$  et hypothèse alternative  $H_1$  [252]. L'hypothèse nulle est une déclaration sans effet et sans aucune différence, alors que l'hypothèse alternative présente un effet ou une différence (dans notre cas, il s'agit de la différence significative entre les résultats obtenus par les algorithmes).

Dans une procédure statistique, l'application des tests conduit au calcul d'un paramètre qui peut être utilisé pour rejeter l'hypothèse nulle à un niveau de signification donné  $\alpha$  [257].

### IV.2.1. Test de signe

Une façon populaire de comparer les performances globales des algorithmes est de compter le nombre de cas sur lesquels un algorithme est le gagnant global [258]. Si deux algorithmes comparés sont comme supposés sous l'hypothèse nulle, l'équivalent de chacun devrait gagner approximativement  $n/2$  sur  $n$  problèmes. Appliqué au cas de deux échantillons associés par paires, ce test tire son nom du fait qu'il utilise les signes + et - aux différences entre les deux populations au lieu des données quantitatives. Ce test est basé uniquement sur l'étude des signes des différences observées entre les paires d'individus quelles que soient les valeurs de ces différences. Il peut se résumer dans les étapes suivantes [259] :

1. Soustraire l'ensemble 2 de l'ensemble 1 et sauvegarder les résultats,
2. Ajouter une quatrième colonne indiquant le signe de la soustraction,
3. Compter le nombre de points positifs et négatifs,
4. Calculer la valeur de la probabilité. L'hypothèse nulle est vraie lorsqu'il existe un nombre égal de signes (c'est-à-dire 50/50). Ainsi, alors une probabilité de 0.5 de signe négatif et 0.5 de signe positif, dans ce cas on peut dire qu'il n'est pas une différence significative entre les deux algorithmes.

Dans le cas où l'hypothèse nulle est rejetée, et afin de tirer des conclusions, on présente le nombre critique des cas qu'un algorithme doit gagner pour dire qu'il est meilleur pour les deux niveaux de signification  $\alpha=0.1$  et  $\alpha=0.05$  [259].

#### IV.2.2. Test de signe multiple

Une comparaison multiple des différents algorithmes doit d'abord être effectuée en utilisant une méthode statistique pour tester les différences (résultats obtenus par chaque algorithme) [259], [260]. Une fois que ce test rejette l'hypothèse d'équivalence, la détection des différences concrètes entre les algorithmes peut être faite avec l'application des procédures statistiques.

Étant donné un algorithme. Le test de signe pour plusieurs comparaisons permet de mettre en évidence celles dont les performances sont statistiquement différentes par rapport à l'algorithme témoin ou de contrôle. La procédure se présente comme suit [261] :

1. représenter par  $x_{i,1}$  et  $x_{i,j}$  les performances de l'algorithme de contrôle et le  $j^{\text{ième}}$  algorithme dans le  $i^{\text{ième}}$  problème.
2. calculer les différences  $d_{ij} = x_{i,j} - x_{i,1}$  par paire pour chaque algorithme avec celui de contrôle, de façons similaire au test de signe en utilisant les signes + et - aux différences au lieu de données quantitatives.
3. pour tester  $H_0 (M_j \geq M_1)$  contre  $H_1 (M_j < M_1)$ , rejeter  $H_0$  si le nombre de signes positifs est inférieur ou égal à la valeur critique de  $R_j$  [262].
4. pour tester  $H_0 (M_j \geq M_1)$  contre  $H_1 (M_j > M_1)$ , rejeter  $H_0$  si le nombre de signes négatifs est inférieur ou égal à la valeur critique de  $R_j$  [262].

#### IV.2.3. Test de Wilcoxon

Le test de somme de rang signé de Wilcoxon est un autre exemple de test non paramétrique pour tester l'hypothèse nulle. La médiane de différence entre les résultats de deux algorithmes doit être égale à une certaine valeur [263]. Contrairement au test de signe qui utilise seulement le signe de la différence entre deux populations, dans le test de Wilcoxon, la valeur de la différence et son signe sont pris en considération [264]. Cela implique que l'on puisse dire quel membre d'une paire est plus grand que l'autre et ranger les différences en ordre croissant.

Le test de Wilcoxon peut être résumé par les étapes suivantes [263] :

1. Enoncer l'hypothèse nulle qui dans ce cas signifie que la valeur moyenne des différences égales à zéro.
2. Calculer la différence de chaque paire,  $d_i = x_i - y_i$  ( $x_i$  et  $y_i$  sont les paires d'observations).
3. Classer les  $d_i$  en ignorant les signes (attribuer le rang 1 au plus petit  $|d_i|$  et le rang 2 au suivant etc.).
4. Aligner chaque rang avec son signe selon le signe de  $d_i$ .
5. Calculer les sommes des rangs dites positive  $W^+$  et négative  $W^-$ . (Pour vérifier le total,  $W^+ + W^- \leq \frac{n(n+1)}{2}$ ,  $n$  est le nombre de paires d'observations dans l'échantillon).
6. Pour rejeter l'hypothèse nulle ou l'accepter, on fait recours à des tableaux présentant les valeurs critiques de  $W^+$  et  $W^-$  pour des tests de deux populations avec une taille  $n \leq 20$  [263]. Pour  $n \geq 20$ , l'hypothèse nulle sera rejetée si une valeur calculée de  $Z_{stat}$  est inférieure ou supérieure à des valeurs critiques [263].

$$Z_{stat} = \frac{\max(W^+, W^-) - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \quad (IV.2)$$

7. Tirer des conclusions concernant la comparaison entre les deux algorithmes ou populations selon la valeur de probabilité  $p$  définie des tableaux précités [264].

#### IV.2.4. Test de Friedman

Le test de Friedman est un test statistique non paramétrique développé par Milton Friedman [265, 266]. Il est utilisé pour comparer les populations dont leur nombre est supérieur à deux. Il vise à trouver les différences significatives entre deux ou plusieurs populations [265].

Les principales étapes de ce test sont [259] :

1. Rassembler les résultats observés pour chaque paire (algorithme, problème).
2. Pour chaque problème  $i$ , classer les valeurs du meilleur résultat au plus pire de 1 à  $k$  en notant ces rangs comme  $r_i^j$  ( $1 \leq j \leq k$ ).

3. Pour chaque algorithme  $j$ , calculer la moyenne  $R_j$  des rangs obtenus dans tous les problèmes  $n$ .

$$R_j = \frac{1}{n} \sum_{i=1}^n r_i^j \quad (\text{IV.3})$$

4. Classer les algorithmes pour chaque problème séparément. Le rang 1 pour l'algorithme le plus performant, le rang 2 pour le deuxième meilleur et ainsi de suite. En cas d'équivalence où deux algorithmes ont le même rang pour un problème donné, nous recommandons l'informatique de rang moyen  $R_j$ .

5. La statistique de test suggérée par Friedman est donnée par [267] :

$$F_D^2 = \frac{12n}{k(k+1)} \left( \sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right) \quad (\text{IV.4})$$

Ici  $k$  est le nombre total des algorithmes et  $n$  est le nombre total de problèmes.

A un niveau de signification  $\alpha$  donnée, l'hypothèse nulle est rejetée si  $F_D^2 \geq \gamma^2$  (des valeurs critiques de  $\gamma^2$  sont données dans [268]).

Cette dernière mesure produit souvent un effet conservateur non désiré. Pour cela, une dérivation de la Friedman  $F_{ID}$  est proposée [267]

$$F_{ID} = \frac{(n-1)F_D^2}{n(k-1) - F_D^2} \quad (\text{IV.5})$$

6. Lorsque des différences significatives sont détectées entre les résultats, c'est-à-dire l'hypothèse nulle est rejetée, il existe plusieurs tests post-hoc (disponible par exemple dans Matlab) qui peuvent être appliqués pour calculer la valeur de probabilité  $p$  qui permet de découvrir l'algorithme différent des autres.

#### IV.2.5. Le Test de la Page

Le test de tendance de la Page pour les alternatives ordonnées peut être classé dans la famille des tests d'association similaire au test de Friedman [269], [270]. L'alternative ordonnée doit être définie par le praticien avant de commencer l'analyse. Une commande entre les  $k$  traitements doit être fournie. Elle devrait refléter l'ordre attendu pour les populations. Par conséquent, les mesures devraient être numérotées de 1 à  $k$  où le 1 est attribué au plus petite somme de rangs et  $k$  au plus grand.

Une fois qu'un tel ordre et données (composé de  $n$  échantillons des  $k$  traitements) sont fournis, les  $n$  échantillons (lignes de données) peuvent être classés du meilleur au pire en donnant le rang 1 à la meilleure mesure de l'échantillon, le 2 à la suivante et ainsi de suite jusqu'au rang  $k$  attribué au pire. S'il existe des liens (même valeur) pour un échantillon donné, les rangs moyens peuvent être attribués (par exemple, un lien entre le premier et le deuxième résultat produirait un rang moyen de  $(1+2)/2 = 1,5$  qui serait assigné aux deux mesures) [257].

Après avoir obtenu les rangs, la statistique de la page  $L$  peut être calculée en utilisant l'expression suivante [257] :

$$L = \sum_{j=1}^k jR_j = R_1 + 2R_2 + \dots \dots kR_k \quad (\text{IV.6})$$

Ici  $r_i^j$  est le rang de la  $j^{\text{ième}}$  des  $k$  mesures sur le  $i^{\text{ième}}$  des  $n$  problèmes.

Les valeurs critiques  $L$  peuvent être calculées pour les petites valeurs de  $k$  et  $n$  [257]. Dans le cas où des valeurs plus grandes sont requises, une approximation normale devrait être considérée. Pour la statistique de la page  $L$  de telle approximation est donnée par :

$$z = \frac{12(L - 0.5) - 3nk(k + 1)^2}{k(k + 1)\sqrt{n(k - 1)}} \quad (\text{IV.7})$$

Ici  $n$  est le nombre des problèmes et  $k$  est le nombre des algorithmes.

Les étapes de la méthode sont [257], [270] :

1. Pour  $n$  fonctions objectifs et  $k$  algorithmes, on exécute pour chaque problème chaque algorithme 30 fois et à chaque fois 500 itérations. Pour prendre en considération la vitesse de convergence des algorithmes, le nombre d'itération est divisé en 10 points de coupure en sauvegardant les valeurs de la fonction objectif de chaque point pour les 30 exécutions.
2. La meilleure valeur objectif atteinte à chaque point de coupure est recueillie pour chaque série de chaque algorithme et fonction. Les valeurs recueillies sont moyennées le long des analyses (moyenne de 10 point de coupure) de sorte qu'une seule valeur agrégée s'obtienne pour chaque algorithme et fonction.
3. Calcul des valeurs de  $L$  et de la probabilité  $p$ .
4. Selon les valeurs de  $p$  on dit que l'algorithme avec le meilleur ordre global  $R_1$  est significativement meilleur que les autres avec un niveau de signification  $\alpha$  donnée (0.1 ou 0.5).

#### IV.2.6. Le Profil de performance

La méthode de profil de performance est une évaluation systématique des algorithmes d'optimisation globale via un profil de performance modifié de nature similaire à la méthode suggéré par Dolan et More [271]. Ils ont proposé un profil de performance d'un algorithme sur la base de trois éléments, le temps  $t_{(p,s)}$ , le rapport de performance  $r_{(p,s)}$  et le taux de performance fractionnaire  $\rho_s(\tau)$  tels que  $p \in P$  l'ensemble de tous les problèmes et  $s \in S$  l'ensemble de tous les algorithmes.

Le temps de calcul  $t_{(p,s)}$  requis pour résoudre le problème  $p$  par le solveur  $s$  est utilisé mais on recommande également l'utilisation de diverses mesures au lieu de calculer le temps.

On propose de remplacer  $t_{(p,s)}$  par  $m_{(p,s)}$  qu'est un critère qui mesure l'amélioration du résultat après un certain nombre d'itérations prédéfini [272]:

$$m_{(p,s)} = \frac{f^{\wedge}_{(p,s)} - f^*}{f_w - f^*} \quad (IV.8)$$

Pour le problème  $p$  avec le solveur  $s$ ,  $f_w$  désigne la pire (plus grande) valeur de la fonction trouvée parmi tous les solveurs sur ce problème particulier,  $f^{\wedge}_{(p,s)}$  est l'estimation moyenne de la valeur optimale de la fonction objectif après un nombre d'itérations  $k$  sur les 30 exécutions trouvées par le solveur  $s$  pour le problème  $p$ ,  $f^*$  est le meilleur résultat de tous les algorithmes.

En d'autre manière, pour réduire les résultats de 30 réplifications à un seul nombre, la moyenne des résultats optimaux de 30 exécutions est  $f^*$ .

Le deuxième élément à calculer est le rapport de performance  $r_{(p,s)}$  qui est défini par [272]:

$$r_{(p,s)} = \frac{m_{(p,s)}}{\min\{m_{(p,s)} \text{ pour tous } s \in S\}} \quad (IV.9)$$

L'élément final à calculer est  $\rho_s(\tau)$  est la fraction du nombre total de problèmes que le solveur  $s$  à un rapport de performance  $r_{(p,s)}$  inférieur à un facteur  $\tau$ .

$$\rho_s(\tau) = \frac{1}{R} \times NC \quad (IV.10)$$

Tel que  $R$  est le nombre total des problèmes et  $NC$  le nombre des cas où les valeurs de  $r_{(p,s)}$  sont égales ou inférieures à  $\tau$  (valeur prédéfinie), le graphique du profil de performance  $\rho_s(\tau)$  compare la façon dont les solveurs peuvent estimer l'optimum global l'un par rapport à

l'autre. Les valeurs de  $\rho$  sont calculées pour des différentes valeurs de  $\tau$  de 1 jusqu'à 50. Une grande valeur de  $\rho$  pour un algorithme donné signifie que cette dernière performe mieux que les autres.

## **CONCLUSION**

Dans la littérature, il existe plusieurs tests statistiques, Il est alors nécessaire d'employer une méthode rationnelle pour choisir le test le plus approprié pour un problème donné.

Nous avons vu que l'un des critères de choix est la puissance du test utilisé, mais d'autres critères sont importants pour déterminer l'adéquation d'un test lors de l'analyse de données particulières comme la façon dont l'échantillon a été réalisé et la nature de la population ainsi que la nature des mesures réalisées.

Plusieurs tests non paramétriques peuvent être utilisés pour comparer les performances des algorithmes évolutionnaires dans des problèmes d'optimisation.

Le Test de signe et le test de signe de Wilcoxon et le test de la page peuvent aider à faire face à des comparaisons par paires, tandis que le test de Friedman, le test de signes multiples, et le test de profil de performance peuvent être utilisés pour effectuer des comparaisons multiples.

## **Chapitre V**

### **Mise en œuvre des techniques d'optimisation à l'OPF et résultats obtenus**

## V. Mise en œuvre des techniques d'optimisation à l'OPF et résultats obtenus

### Introduction

L'objectif de ce chapitre est l'application des algorithmes d'optimisation présentés précédemment sur des problèmes de l'OPF et la vérification de leurs performances pour plusieurs cas de réseaux électriques. Afin de classer ces algorithmes et identifier le meilleur, une approche de comparaison basée sur l'application de trois méthodes statistiques a été appliquée sur les neuf cas de l'OPF existants en l'occurrence :

1. Minimisation de coût de carburant,
2. Minimisation de la déviation du profil de tension,
3. Amélioration de la stabilité de tension,
4. Minimisation des émissions du gaz,
5. Minimisation de coût de carburant avec la considération de plusieurs types de combustibles,
6. Minimisation de coût de carburant avec l'introduction de l'effet d'ouverture des vannes,
7. Minimisation de coût de carburant avec la considération de plusieurs types de combustibles et l'effet d'ouverture des vannes,
8. Minimisation de coût de carburant avec l'introduction des dispositifs FACTS,
9. Minimisation du coût de carburant avec prise en compte des contraintes de sécurité (SCOPF),

Les cas traités pour chaque type de réseau électrique sont illustrés dans le tableau ci-après :

**Tableau V.1 Cas de l'OPF étudiés pour chaque type de réseau**

cas Réseau	Cas1	Cas2	Cas3	Cas4	Cas5	Cas6	Cas7	Cas8	Cas9
IEEE30	✓	✓	✓	✓	✓	✓	✓	✓	✓
IEEE57	✓	✓	✓						
IEEE 118	✓	✓	✓						
Réseau Algérien	✓	✓	✓	✓					

L'approche proposée est justifiée par le fait que la meilleure méthode qui peut être obtenue par une comparaison paramétrique simple des minimums globaux, des moyennes ou des déviations standard peut changer avec le changement du type du réseau ou du cas considéré.

Les techniques statistiques adoptées dans notre approche sont le test de signe multiple, le test de Friedman et le profil de performance.

Pour vérifier la dépendance de la meilleure méthode du type du réseau et du cas de l'OPF, on fait une comparaison paramétrique des méthodes d'optimisation pour les réseaux et cas du tableau V.1.

## V.1. Comparaison paramétrique des méthodes d'optimisation

### V.1.1. Comparaison des méthodes d'optimisation pour le réseau IEEE30

Ce réseau a une puissance totale générée de 900,2 MW et une puissance active totale demandée de 283,400MW.

Les caractéristiques principales de ce réseau sont illustrées dans le tableau V.2.

**Tableau V.2 Caractéristiques principales du réseau IEEE30 considéré [66]**

Caractéristiques	Nombre	Details
Lignes	30	-
Branches	41	-
Générateurs	6	Jeux de barre 1, 2, 5, 8, 11 et 13.
Shunts	9	Jeux de barre 10, 12, 15, 17, 20, 21, 23, 24 et 29.
Transformateurs	4	Branches 11, 12, 15 et 36.
Variables de contrôle	24	-

Les coefficients de coût et d'émission liés à ce réseau sont donnés dans le tableau V.3.

**Tableau V.3 Coefficients de coût et d'émission du réseau IEEE 30 considéré**

Jeu de barre	$a$	$b$	$c$	$d$	$\beta$	$\gamma$	$\omega$	$\mu$
1	0	2	$3.75 \times 10^{-3}$	18	-5.554	6.49	$2 \times 10^{-4}$	2.857
2	0	1.75	$1.75 \times 10^{-2}$	16	-6.047	5.638	$5 \times 10^{-4}$	3.333
5	0	1	0.0625	14	-5.094	4.586	$1 \times 10^{-6}$	8
8	0	3.25	$8.34 \times 10^{-3}$	12	-3.55	3.38	$2 \times 10^{-3}$	2
11	0	3	0.025	13	-5.094	4.586	$1 \times 10^{-6}$	8
13	0	3	0.025	13.5	-5.555	5.151	$1 \times 10^{-5}$	6.667

Pour effectuer notre calcul, des programmes ont été développés sous l'environnement Matlab et exécutés par une machine i5, 2.20 GHz et 8.00 Go de RAM en effectuant 30 exécutions d'une manière parallèle (cinq exécutions à la fois).

#### V.1.1.1. Minimisation du coût de carburant (cas 1)

Pour minimiser le cout de carburant exprimé par la fonction objectif quadratique ( II.16), nous avons mis en œuvre les 30 algorithmes précités (GA, SA, PSO, DE, ACO, HS, EM, ABC, ICA, BBO, FA, LCA, GEM, GSA, WDO, BA, CS, GA-MPC, TLBO, WCA, DSA, MBA, BSA, BHBO, GWO, CSO, JAYA, CBO, FWA et MVPA). Les conditions de calcul liées à chaque méthode de minimisation sont mentionnées dans le tableau V.4.

**Tableau V.4. Conditions de calcul liées à chaque méthode de minimisation**

ALGO	Paramètre	Symbole	Valeur
GA	Critère de mutation	Mu	0.05
	Critère de sélection	Cr	0.75
	Nombre de bits pour le codage binaire	N bits	16
SA	Température initiale	T0	1
	Température de stabilité	T_stop	0.0001
	Constant de Métropolies	Kc	5
	Coefficient de diminution de température	Alpha	0.9
	Nombre de répétition dans une température	Rep_t_max	100
PSO	Coefficient de pondération 1	C1	1,5
	Coefficient de pondération 2	C2	1,5
	Coefficient d'accélération	W	0.5
DE	Facteur de croisement	Cr	0.8
	Facteur de mutation	F	0.890
ACO	Sensibilité aux phéromones	Alpha	2
	Sensibilité à la visibilité	Beta	10
	Taux d'évaporation de phéromone	Rho	0.5
	Constante de mise à jour des phéromones	Q	50
HS	Critère d'acceptation de mémoire	Raccep	0.9
	Mémoire d'harmonie	Hs	25
	Largeur de bande de pas	$B_{rang}$	0.1
	Vitesse d'ajustement de pas	Rpas	0.995
EM	Paramètre de recherche local	Delta	0.1
	Paramètre randomisation	$\lambda$	rand
ABC	Nombre limite d'itération d'abandonne de site	L	100
	Nombre d'abeilles employeurs	Nemp	50

ICA	Nombre des empires	Nemp	10
	Nombre des colonies	Ncol	90
	Paramètre d'ajustement de pouvoir total de l'empire	$\zeta$	0,1
	Coefficient d'assimilation	Beta	1,5
BBO	Taux d'immigration max pour chaque île	$I$	1
	Taux d'émigration max, pour chaque île	$E$	1
	Paramètre de probabilité de mutation	$M_{max}$	0.005
FA	Attractivité pour une distance égale a zéro	B0	1
	Pas de déplacement	Alpha	2
	Coefficient d'absorption lumineuse	$\Gamma$	1
LCA	Nombre des saisons qui est le nombre d'itération	S	500
	Liste de des équipes concoure par semaine	L	100
GEM	Nombre grenade	Ng	10
	Nombre des shrapnels	Nq	30
	La longueur de l'explosion initiale	le	1,5
	Le rayon de territoire initial	Rt	0,9
	Réduction du rayon de territoire	Rrd	100
	Taux d'exploitation	mmin	0,3
GSA	Constante de gravité	G0	100
	Constant	alfa	20
	Point d'influence de distance	$R_{power}$	1
	Constant	eps	Fournit par Matlab
WDO	Coefficient de frottement	$\alpha$	0.4
	Constante gravitationnelle	$g$	0.1
	La constante universelle des gaz	$RT$	3
	Coefficient de Coriolis	$c$	0.1
BA	Volume sonore initiale	$A$	0.5
	Taux d'émission d'impulsions initiale	$r$	0.5
	Fréquence minimum	$fmin$	0
	Fréquence maximum	$fmax$	40
	Constant 1	eps	0.001
	Constant 2	$\alpha$	0.9
	Constant 3	$\gamma$	0.9

CS	Constante de vole Lévy	beta	3/2
	Probabilité de changement	P	0.1
	Distributeur gamma	gamma	Fournit par Matlab
GA-MPC	Taille de la population d'archive	Tc	50
	Critère de croisement	cr	0.5
	Nombre aléatoire qui suit une distribution normale	beta	Normrnd (0.7, 0.5)
	Probabilité de mutation	p	0.05
TLBO	Facteur de croisement	T_F	Round (1+rand)
WCA	Nombre des revirées	Nre	8
	Ruisseaux d'eaux	Nsr	92
	Taux d'évaporation	Dmax	1.00E-05
	Constant	C	2
	Constant	$\mu$	0,1
DSA	Constant	P1	0.3*rand
	Constant	Scale	rand
MBA	Nombre des morceaux de shrapnels	N_s	50
	Facteur d'exploration	$\mu$	0.1
	Constante de réduction	$\alpha$	500/5
	Ongle d'explosion	$\theta$	360/N_s
BSA	Facteur de mutation	F	3*randn
	Constant	$\alpha$	rand
	Constant	$\delta$	rand
	Constant de croisement	DIM_RATE	1
BHBO	Nombre des étoiles	P	100
GWO	Nombre de population	P	100
CSO	Nombre de coqs	N coq	20%
	Nombre de poules	N poule	60%
	Nombre mères poules représente	N mère poule	10%
	Population de poussins	N poussins	10%
	Constant	FL	20%
JAYA	Nombre de population	P	100
CBO	Nombre de population	P	100
FWO	Constant d'amplitude d'explosion	A	40
	Nombre d'étincelles Générées Par Un Individu	S	50
MVPA	Nombre des joueurs	N	90
	Nombre des équipes	T	20

Les résultats obtenus en terme de variation du coût et de pénalité en fonction des itérations sont présentés dans le groupe des figures V.1.a, b, c, d, e.

L'inspection des résultats des figures V.1.a, b, c, d, e montre que les algorithmes PSO, GA, DE, CBO, TLBO, GA-MPC, GSA, GWO, WCA, ICA, MVPA, WDO et FA ont convergé à partir de la 20<sup>ème</sup> itération, tandis que les algorithmes CSO, ABA et SA ont atteint leurs minimums après 50 itérations. Pour les algorithmes FWA, CSO, SCA, BHBO, ABC, BBO, BSA, GEM, LCA et CBO ont nécessité un nombre plus élevé d'itérations pour atteindre leurs minimums.

Qualitativement, l'algorithme ICA a fourni le meilleur résultat (798.98 \$/h). Le deuxième meilleur résultat (799.034 \$/h) a été obtenu par les algorithmes DE, TLBO, GA-MPC, JAYA, WCA et MVPA, alors que les mauvais résultats (812.76 \$/h, 814.39 \$/h et 820.29 \$/h) ont été obtenus respectivement par les algorithmes SA, FWA et GWO.

Il est à noter qu'après 500 itérations tous les algorithmes ont donné un terme de pénalité nul ce qui vérifie leur convergence sûre pour ce cas.

Pour vérifier la robustesse des algorithmes, nous les avons exécutés 30 fois pour les mêmes conditions de calcul. La déviation standard moyenne a montré une robustesse significative des algorithmes GA-MPC et TLBO avec une déviation standard de l'ordre de  $10^{-5}$  malgré que l'ordre moyenne de coût est de 800 \$/h. Les deux algorithmes SA et HS ne sont pas robuste pour le traitement de ce problème à cause de leur valeur de déviation standard élevée ( $10^3$ ) par rapport aux autres donnant une déviation standard de l'ordre de 10 et qui jugé acceptable.

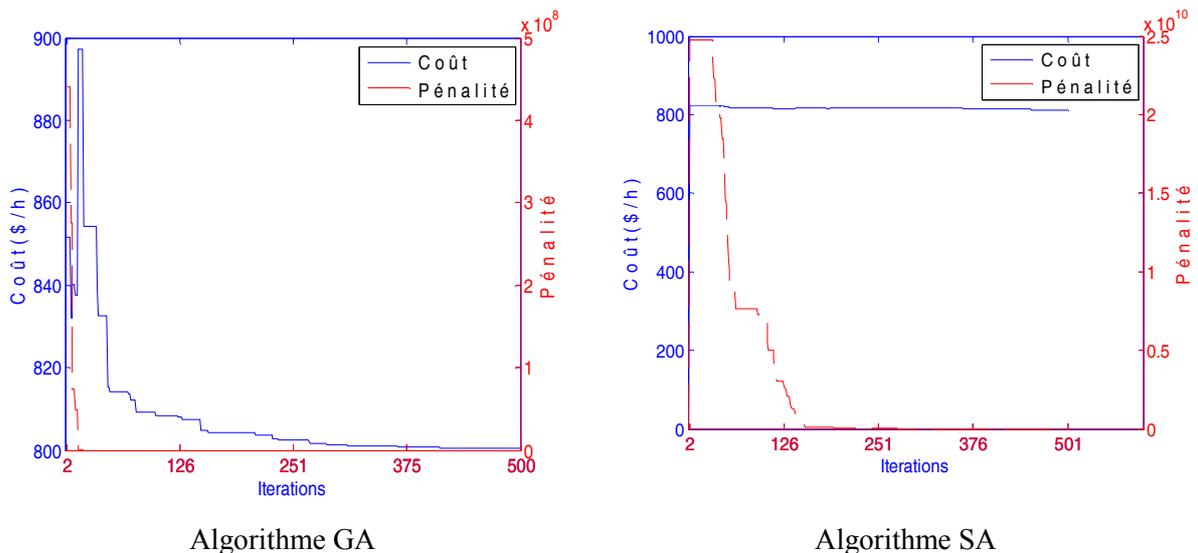
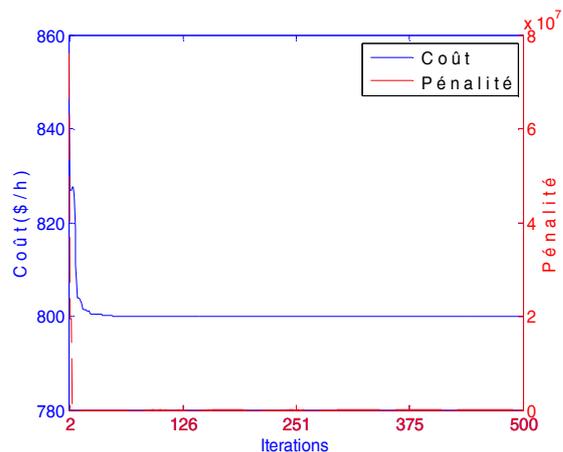
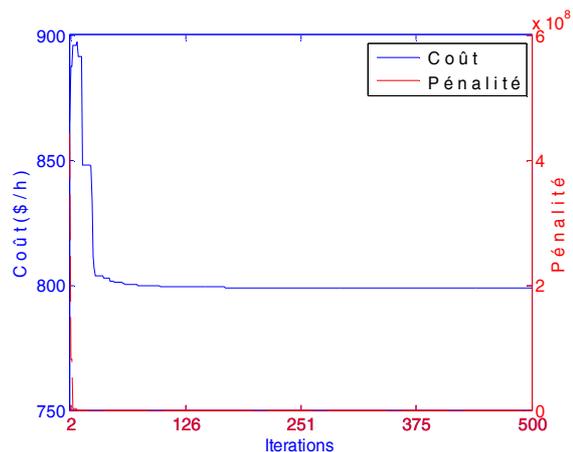


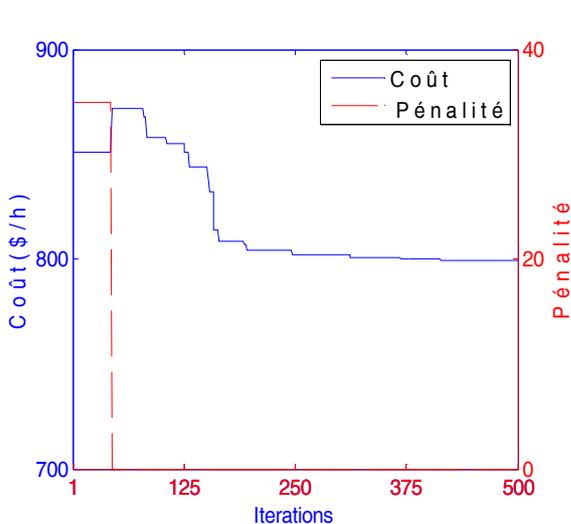
Fig. V.1.a Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30



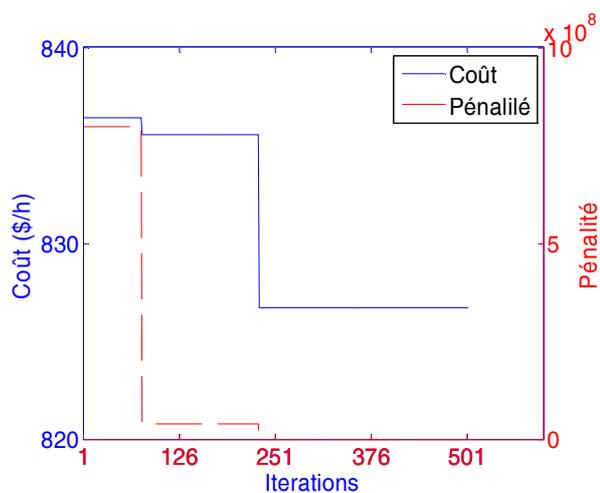
Algorithme PSO



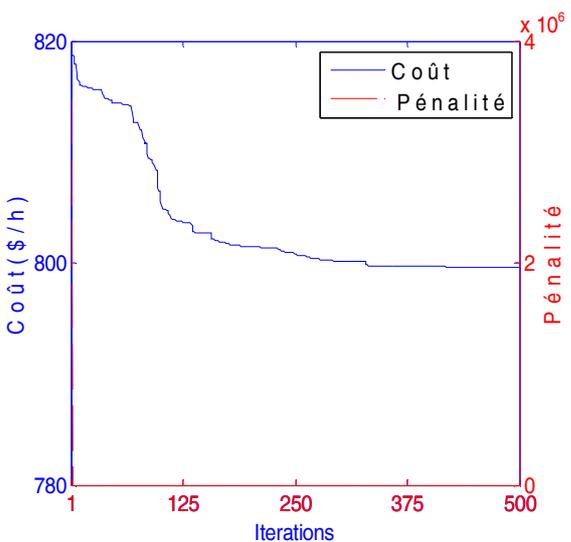
Algorithme DE



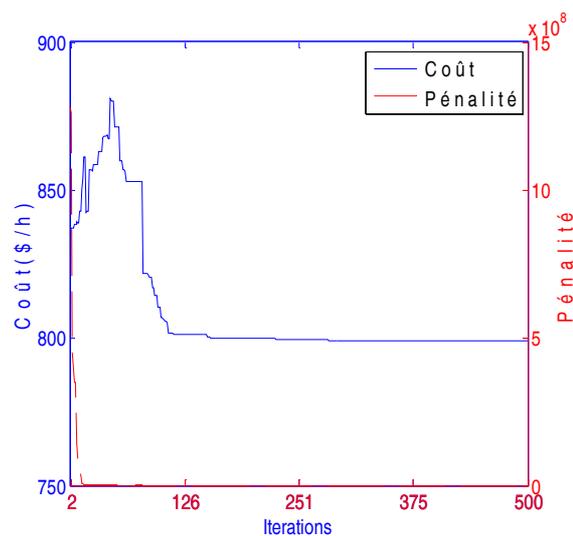
Algorithme ACO



Algorithme HS

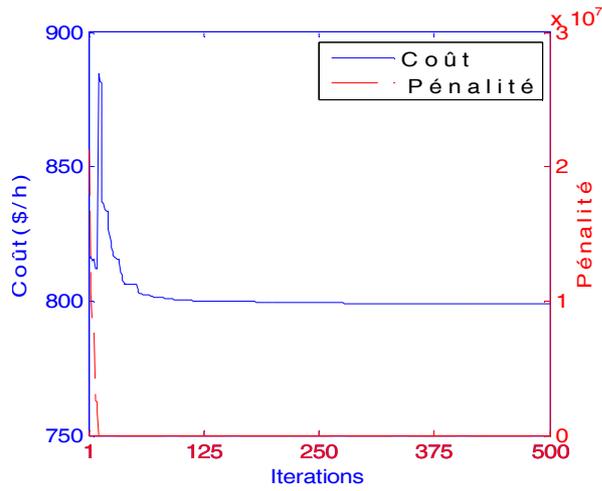


Algorithme EM

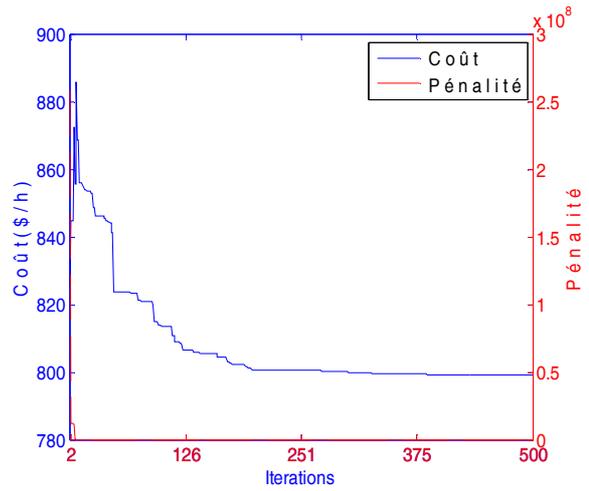


Algorithme ABC

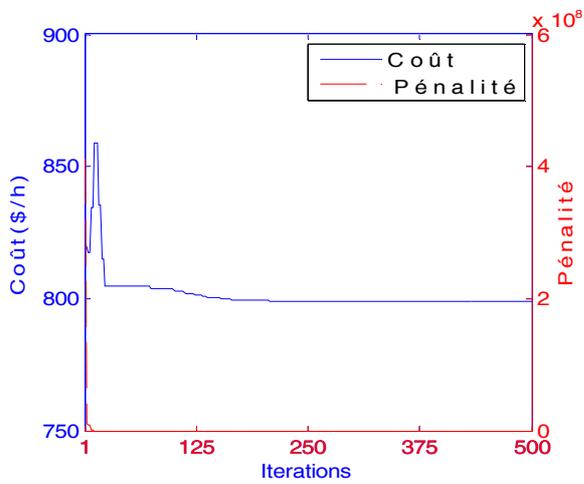
Fig. V.1.b Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30



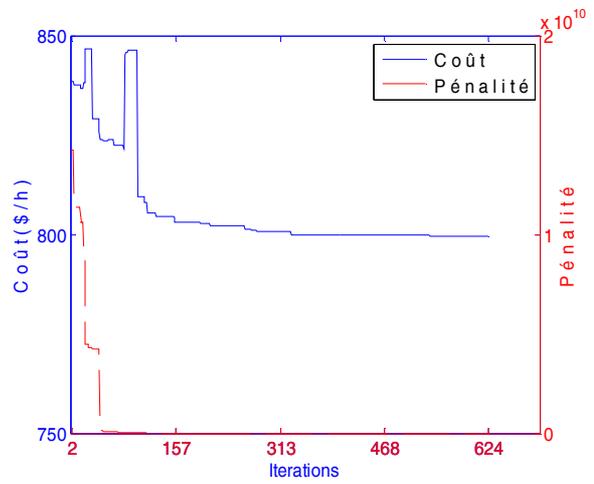
Algorithme ICA



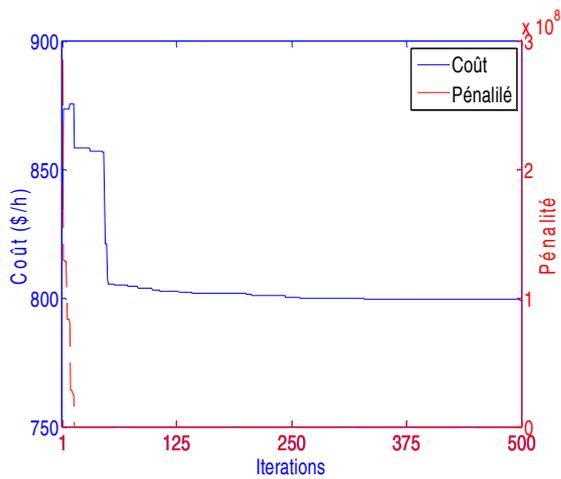
Algorithme BBO



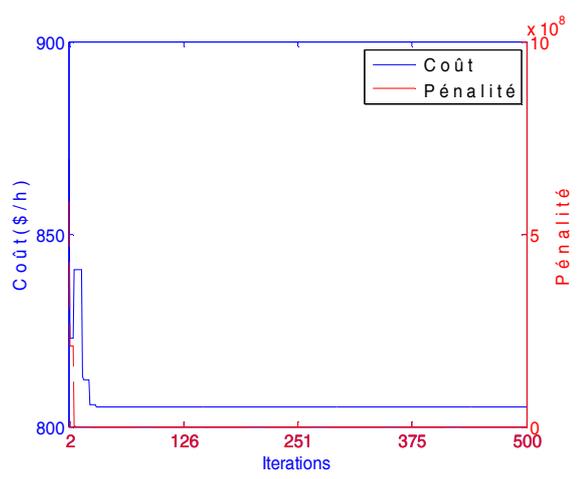
Algorithme FA



Algorithme LCA

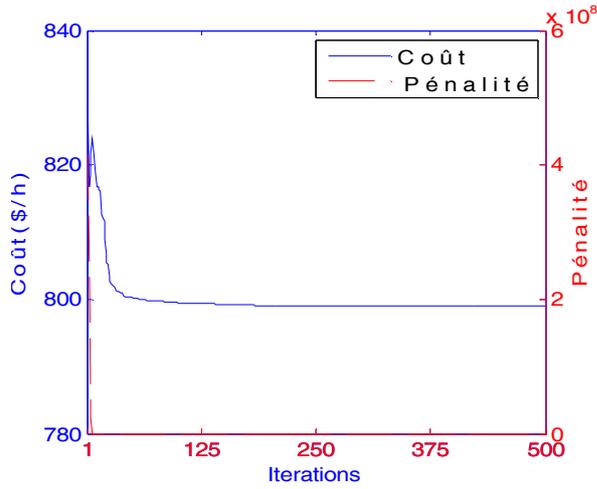


Algorithme GEM

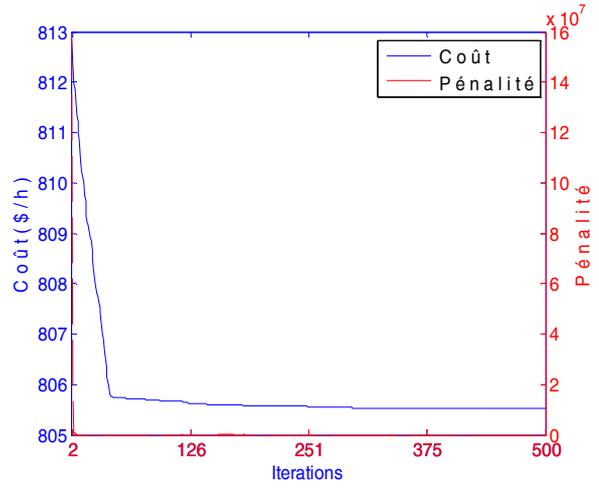


Algorithme GSA

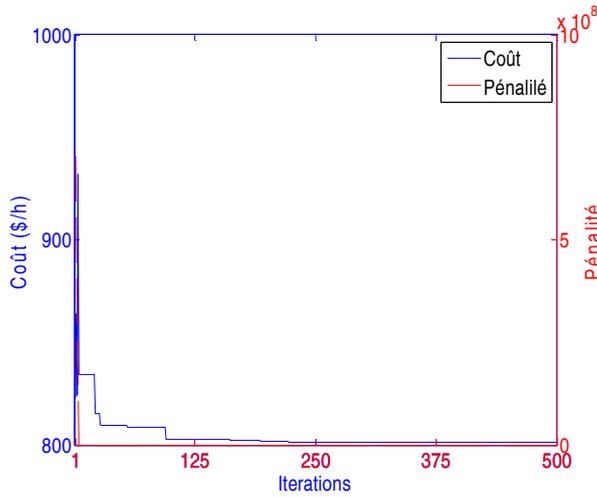
Fig. V.1.c Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30



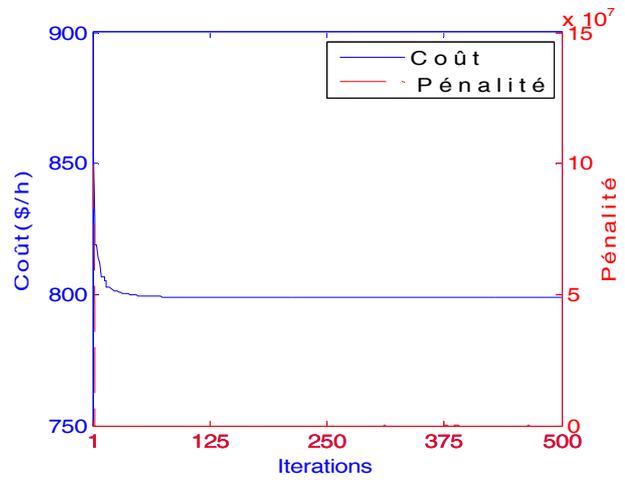
Algorithme WDO



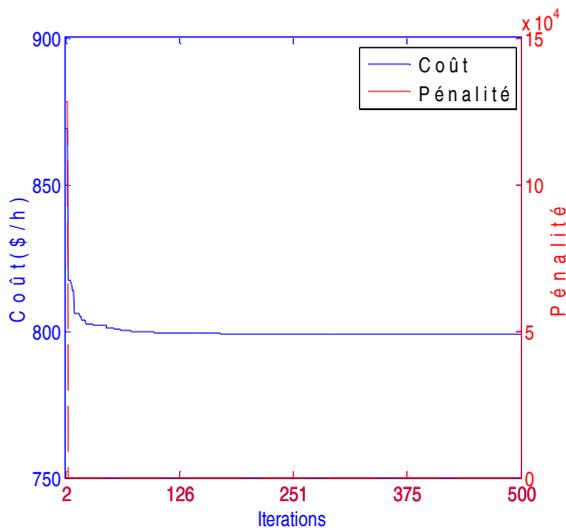
Algorithme BA



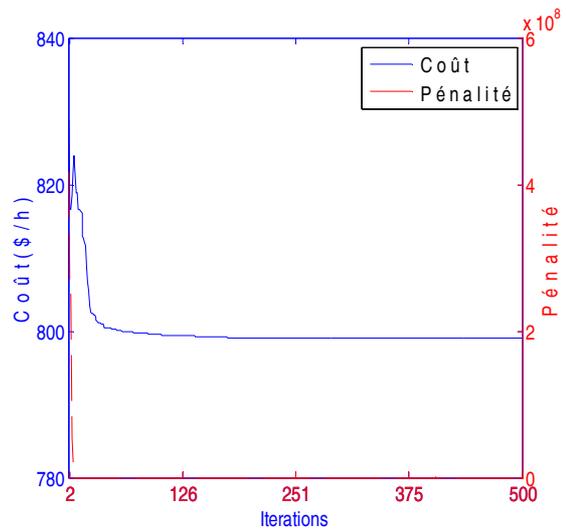
Algorithme CS



GA-MPC

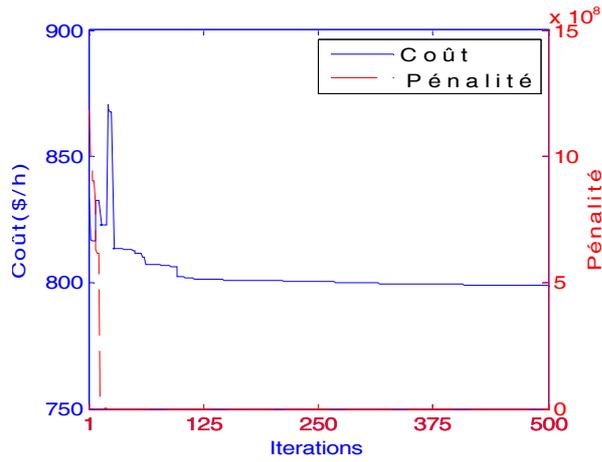


Algorithme TLBO

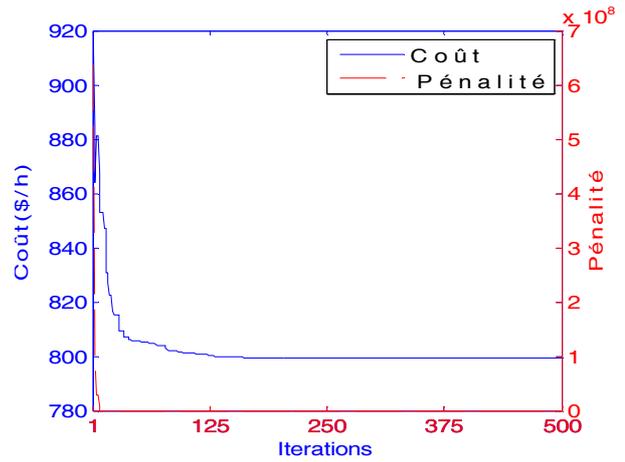


Algorithme WCA

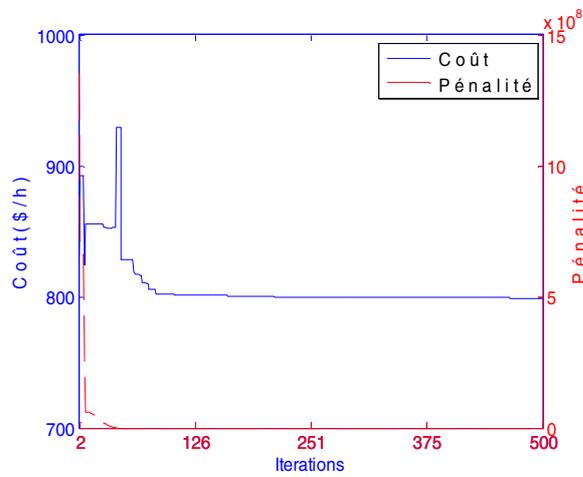
Fig. V.1.d Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30



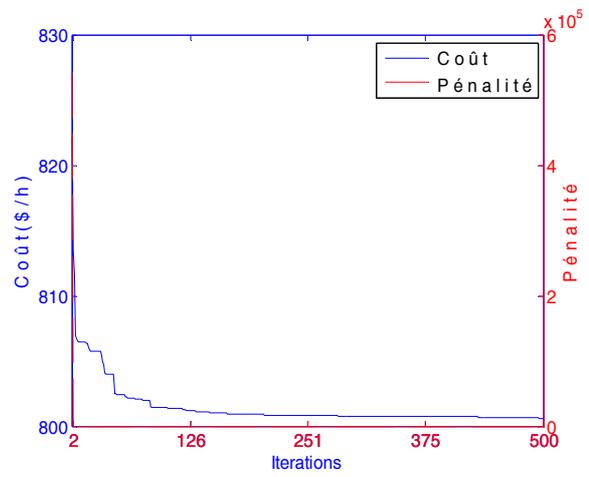
Algorithme DSA



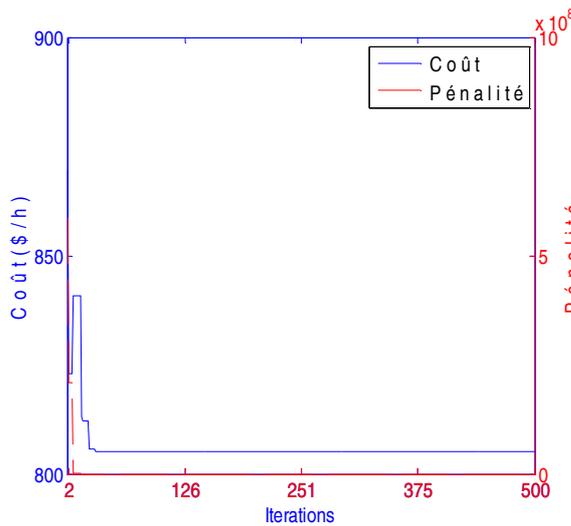
Algorithme MBA



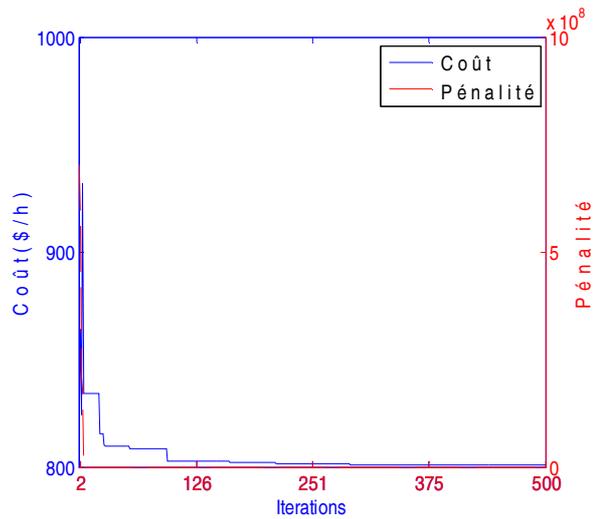
Algorithme BSA



Algorithme BHBO



Algorithme GWO



Algorithme CSO

Fig. V.1.e Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30

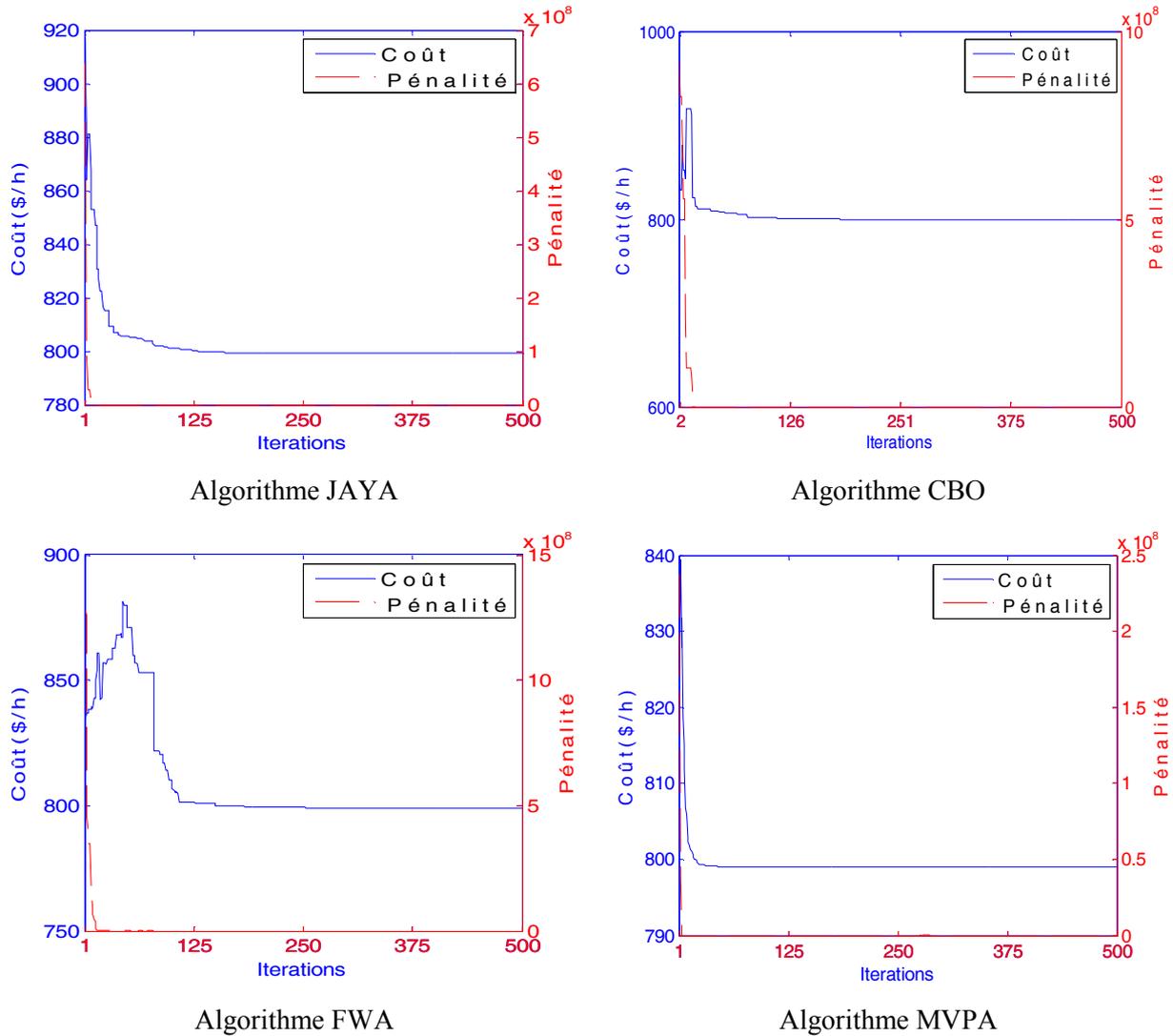


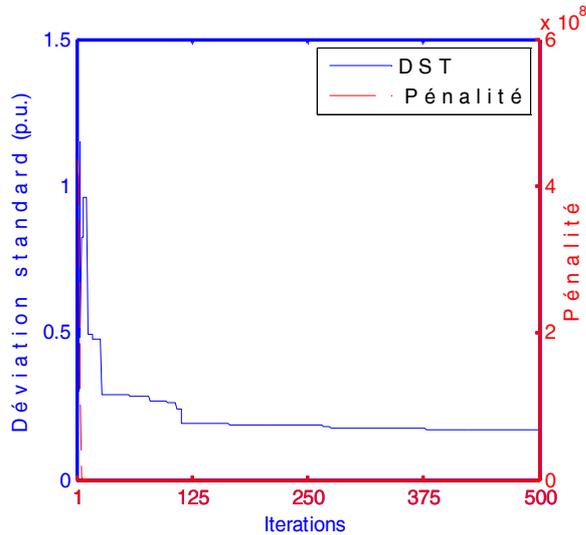
Fig. V.1.f Évolution du coût et de pénalité au cours des itérations pour le réseau IEEE30

### V.1.1.2. Minimisation de la déviation du profil de tension (cas 2)

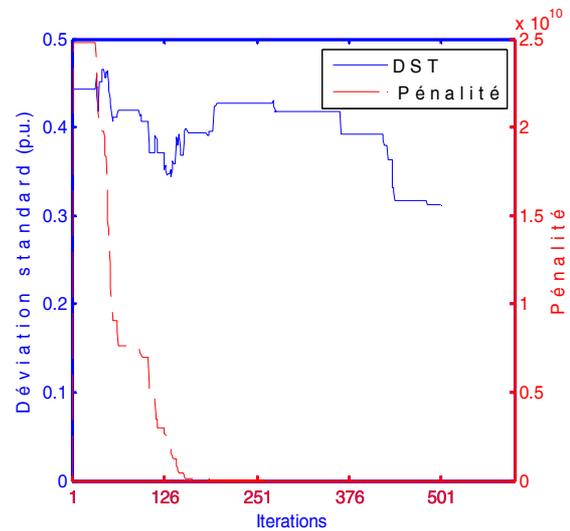
Pour ce deuxième cas, il s'agit en fait d'une minimisation double objective. La minimisation de la déviation du profil de tension (minimisation de l'écart entre la tension des jeux de barres charges et l'unité 1.0) et le coût du carburant. Pour ce cas, les mêmes conditions de calcul précédentes ont été adoptées.

Les résultats concernant la déviation standard de tension et la pénalité sont présentés dans le groupes des figures V.2.a, b, c, d, e. Ces résultats montrent que les algorithmes PSO, GA-MPC, BHBO, TLBO, ABC, BBO et MVPA ont convergé avant d'atteindre l'itération 100 par rapport aux autres.

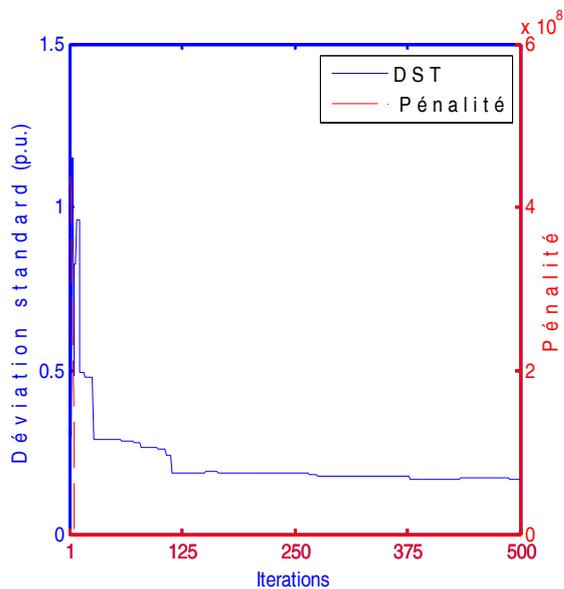
La meilleure valeur optimale 0.0986 PU a été obtenue par les algorithmes GA-MPC et TLBO alors que les MVPA, DE, DSA et JAYA ont donné juste des valeurs proches de cet optimum global. Les valeurs les plus mauvaises ont été obtenues par les FWA, SA et GWO. Pour ce deuxième cas, et tenant compte des résultats des 30 exécutions effectuées, on a vu que tous les algorithmes sont convergents et robustes.



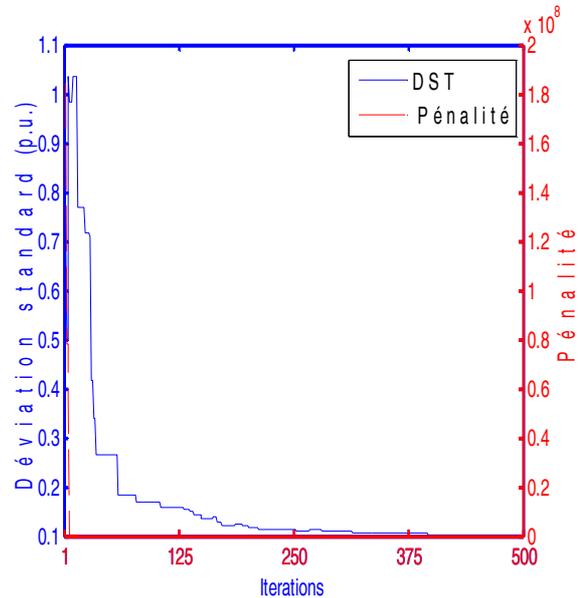
Algorithme GA



Algorithme SA

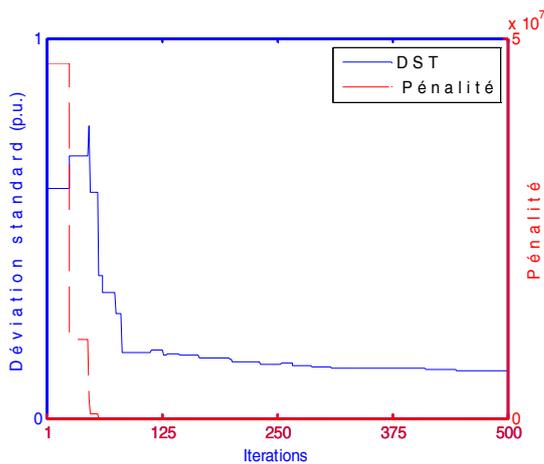


Algorithme PSO

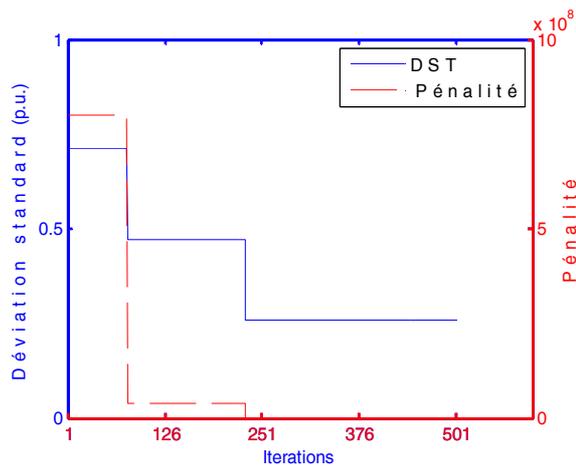


Algorithme DE

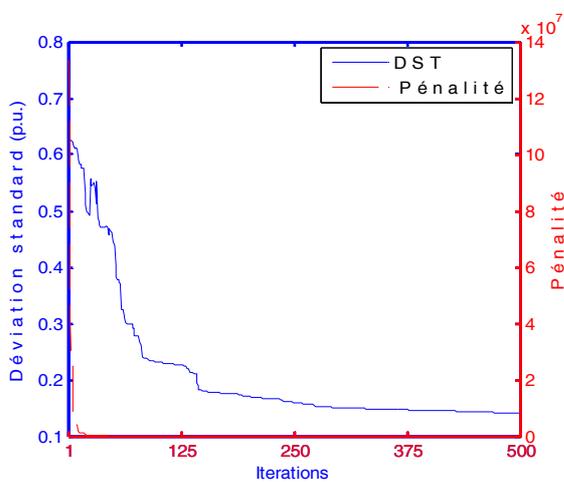
Fig. V.2.a Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30



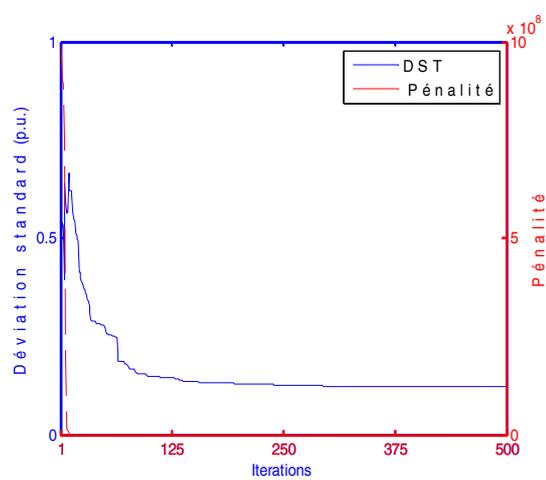
Algorithme ACO



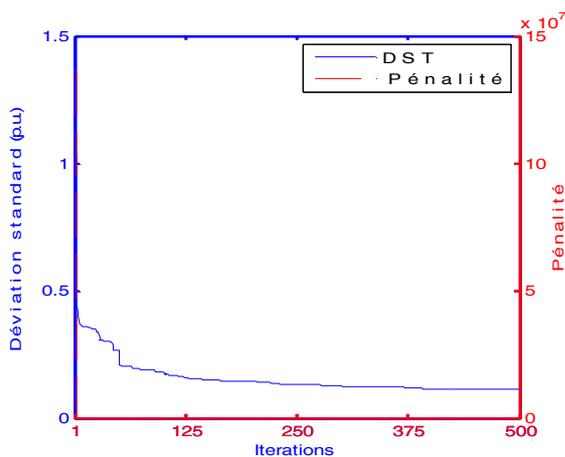
Algorithme HS



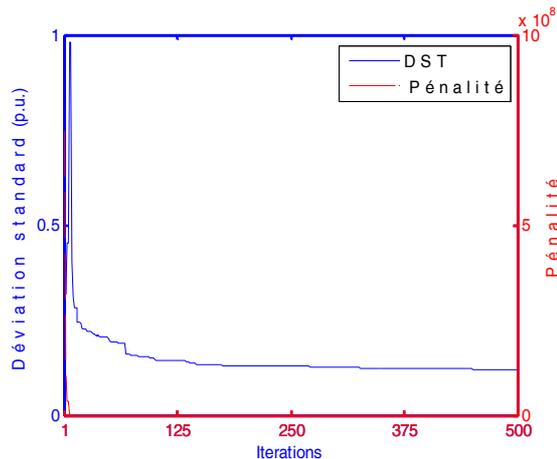
Algorithme EM



Algorithme ABC

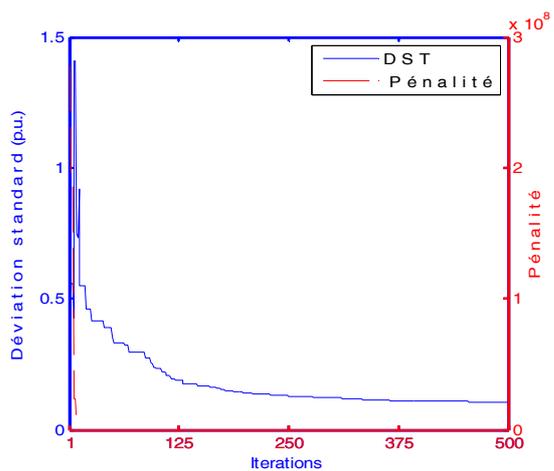


Algorithme ICA

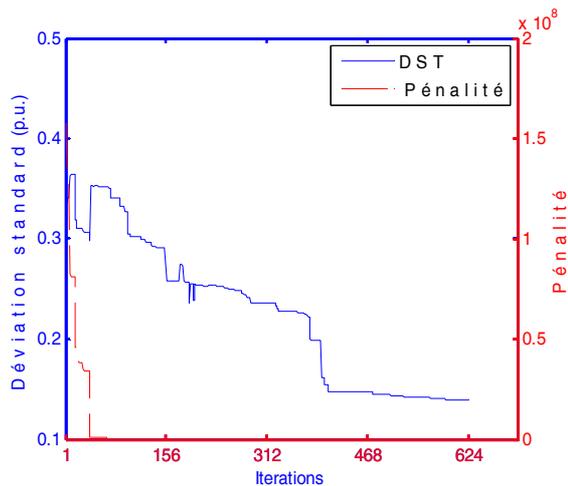


Algorithme BBO

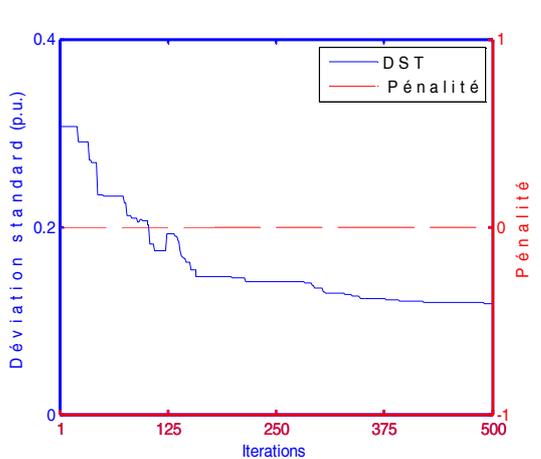
Fig. V.2.b Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30



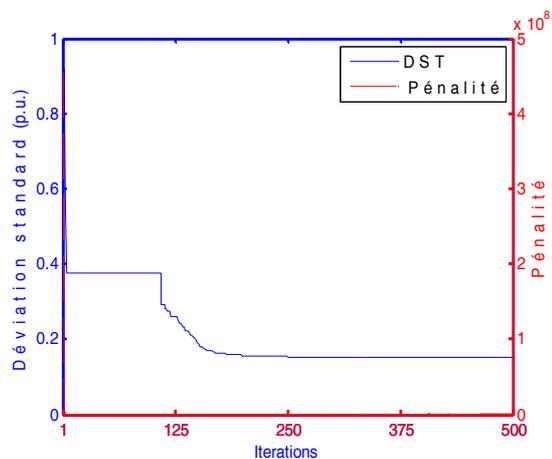
Algorithme FA



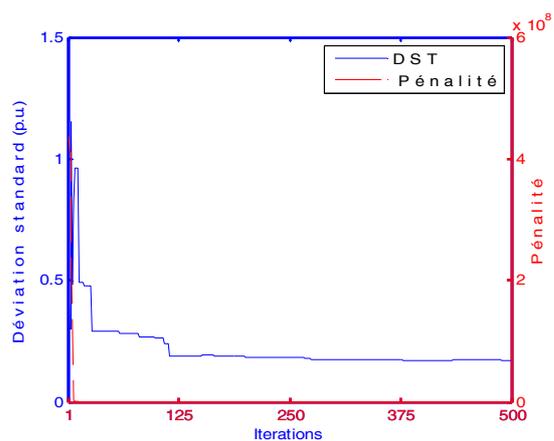
Algorithme LCA



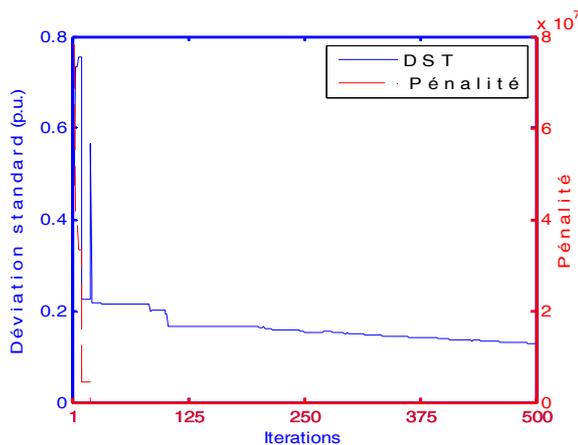
Algorithme GEM



Algorithme GSA

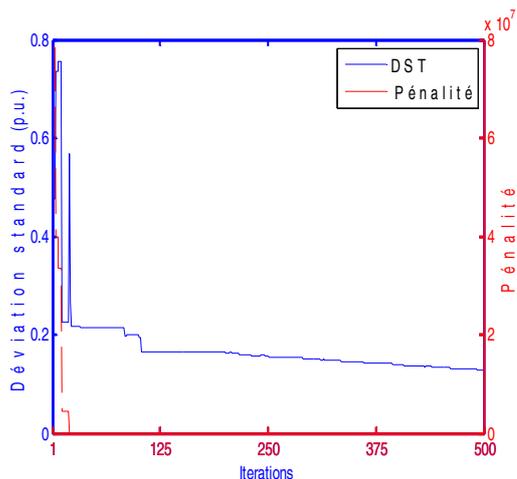


Algorithme WDO

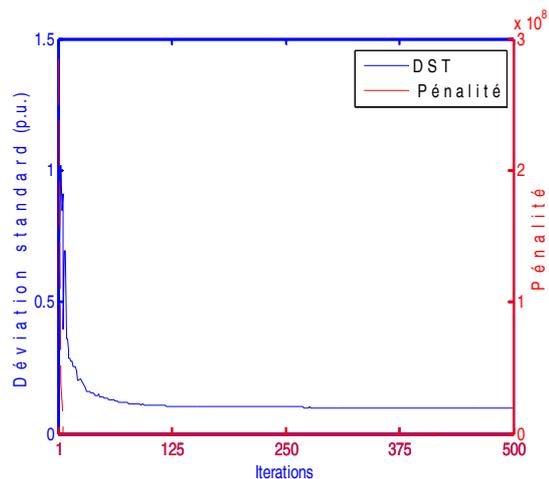


Algorithme BA

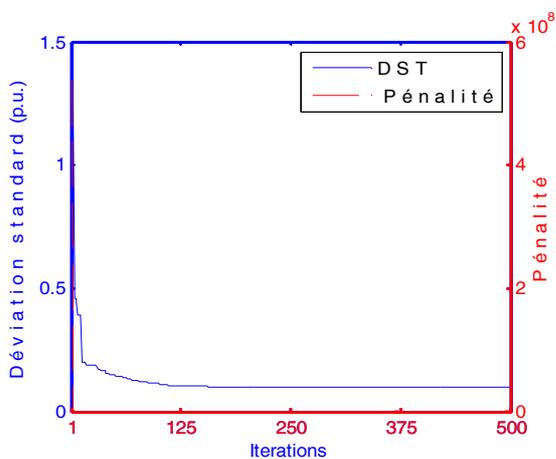
Fig. V.2.c Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30



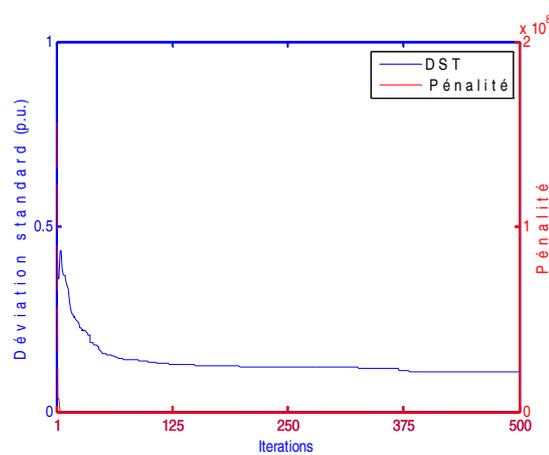
Algorithme CS



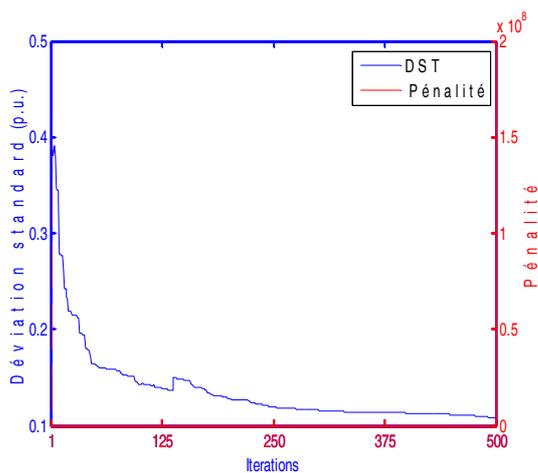
GA-MPC



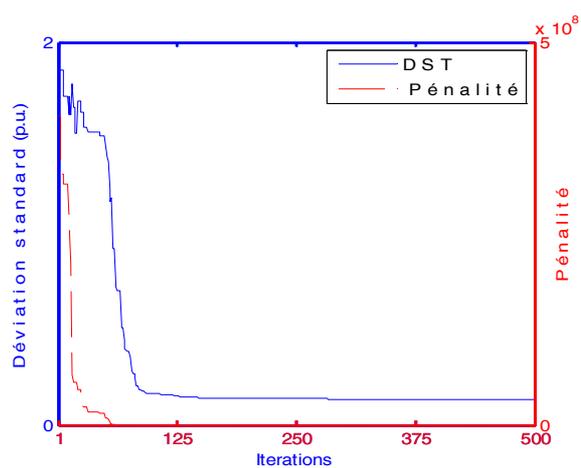
Algorithme TLBO



Algorithme WCA

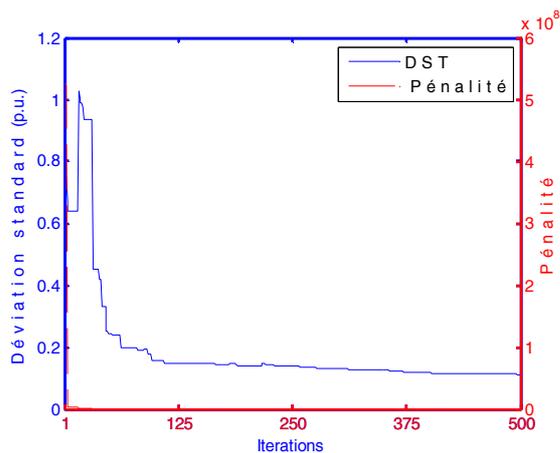


Algorithme DSA

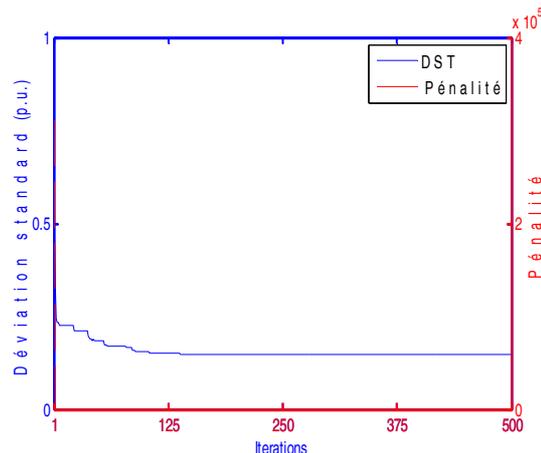


Algorithme MBA

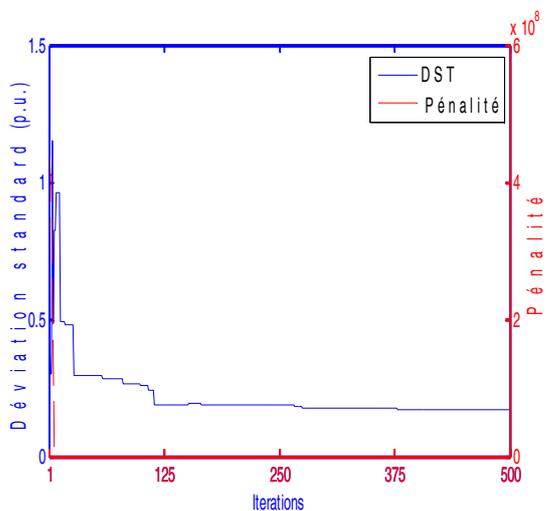
Fig. V.2.d Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30



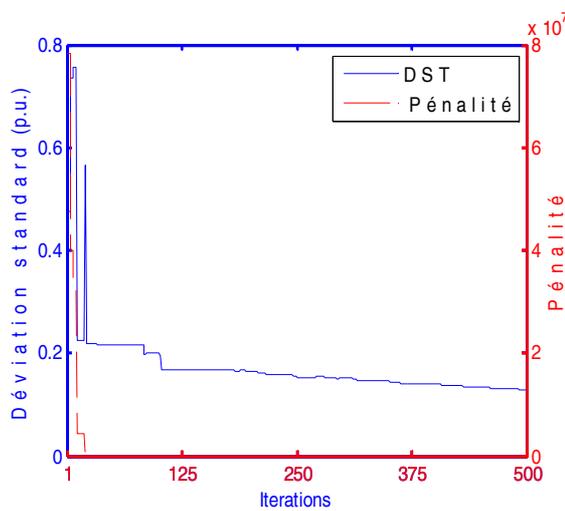
Algorithme BSA



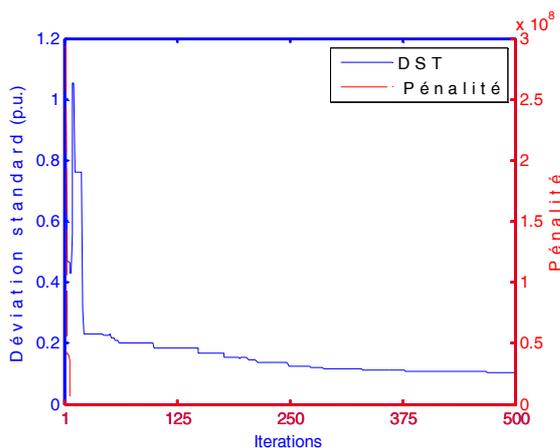
Algorithme BHBO



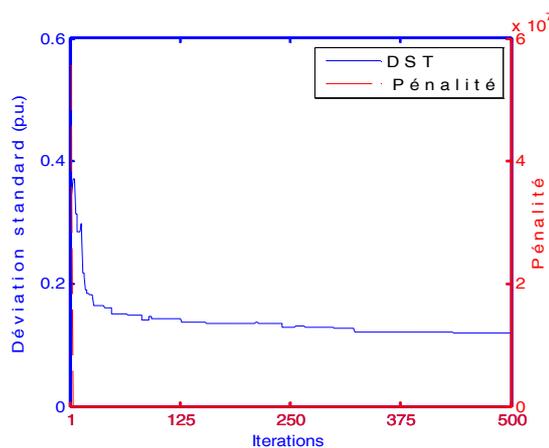
Algorithme GWO



Algorithme CSO



Algorithme JAYA



Algorithme CBO

Fig. V.2.e Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30

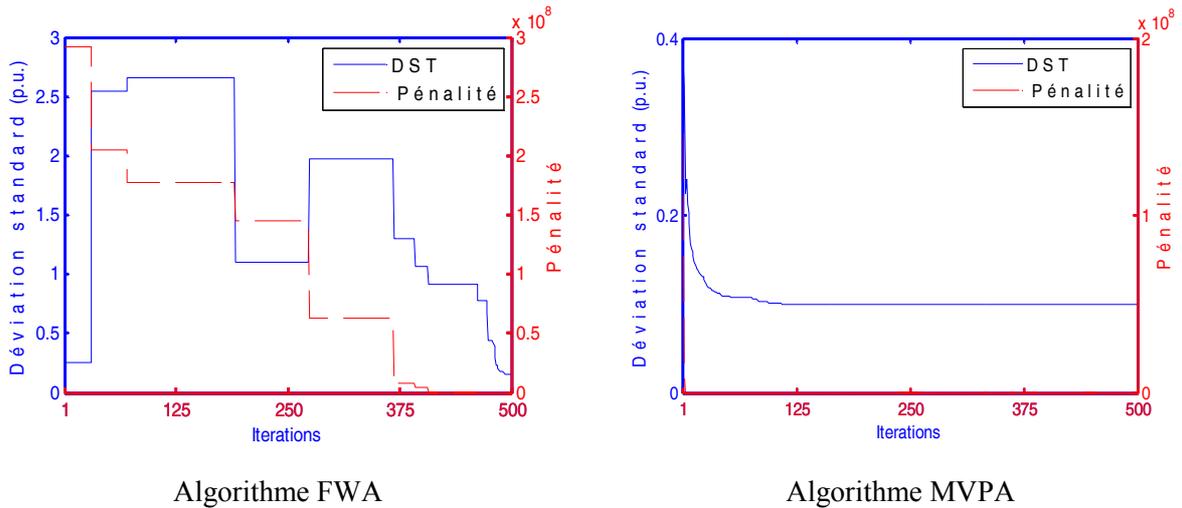


Fig. V.2.f Évolution de la déviation standard et de pénalité au cours des itérations pour le réseau IEEE30

Concernant la déviation standard de tension, nous avons obtenu les résultats de la Figures V.3.

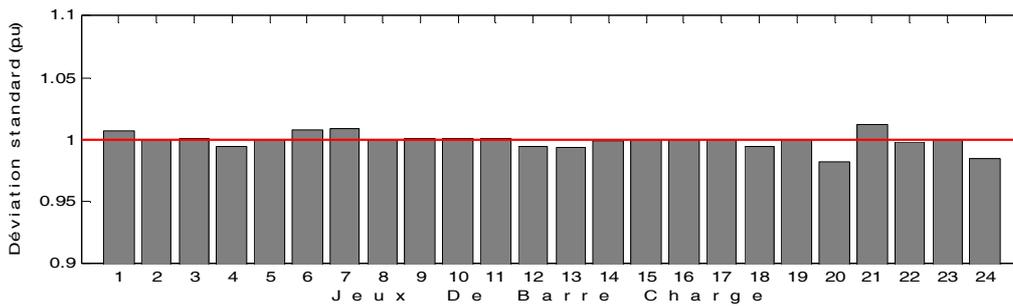


Fig. V.3 Profil de tension après optimisation de la déviation standard

En comparaison avec le profil obtenu en considérant seulement la minimisation du coût (Fig. V.4), on voit une très grande amélioration de la tension accompagnée d'une légère augmentation du coût de production (de 799.03\$/h à 800.84\$/h).

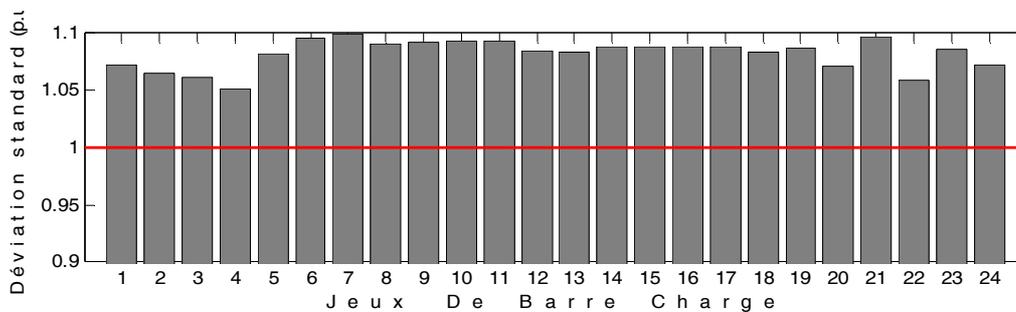


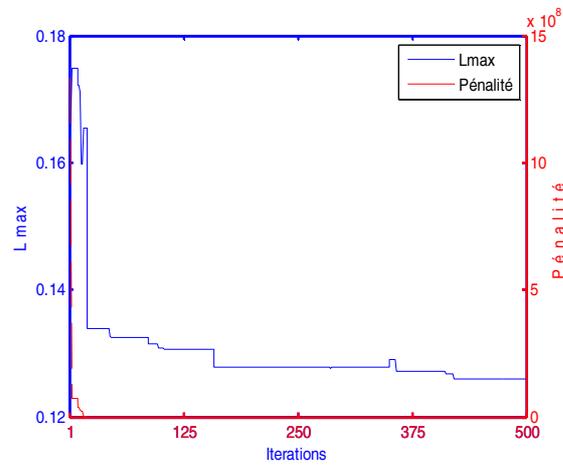
Fig. V.4 Profil de tension avant optimisation de la déviation standard

### V.1.1.3. Amélioration de la stabilité de tension (cas 3)

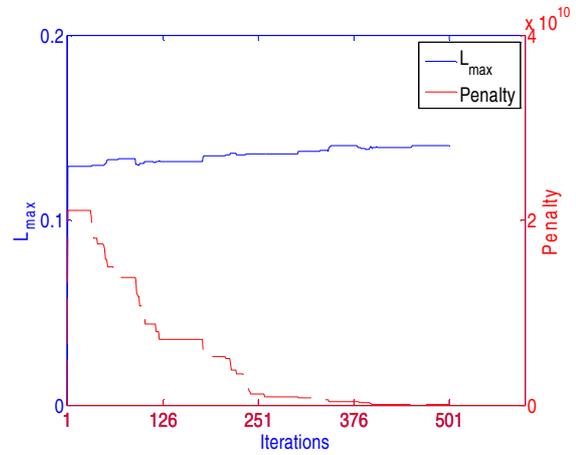
Dans ce cas, il s'agit d'une minimisation double objectif, indice de stabilité de tension du réseau et du coût (I.16 et I.24).

Les résultats obtenus par les 30 algorithmes appliqués sont montrés dans le groupe de figures V.5. a, b, c, d, e.

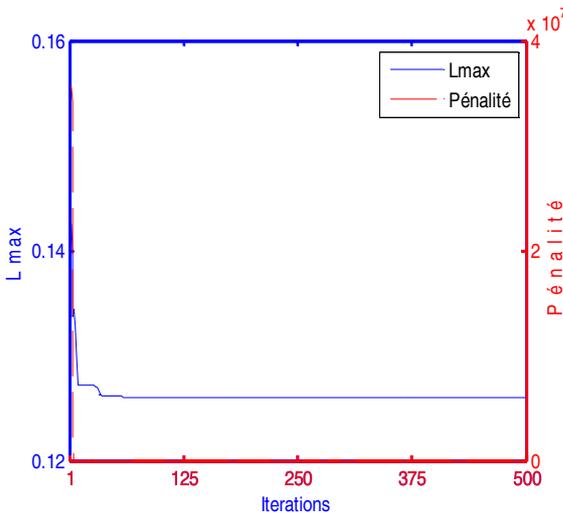
D'après ces résultats, on remarque que les algorithmes GA-MPC, ABC, BBO, PSO convergent dans les 100 premières itérations, tandis que les autres nécessitent un nombre d'itération plus élevée. Les valeurs optimales de l'indice de stabilité sont très proches. La meilleure valeur ( $L=0.1249$ ) est obtenue par l'algorithme ABA alors que la mauvaise ( $L=0.1489$ ) est obtenue par l'algorithme FA.



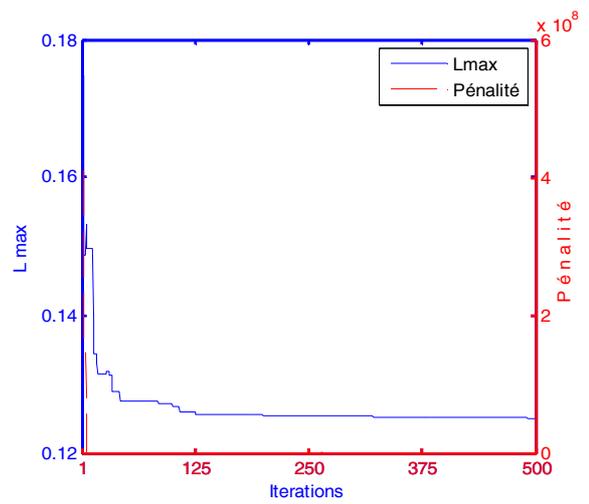
Algorithme GA



Algorithme SA

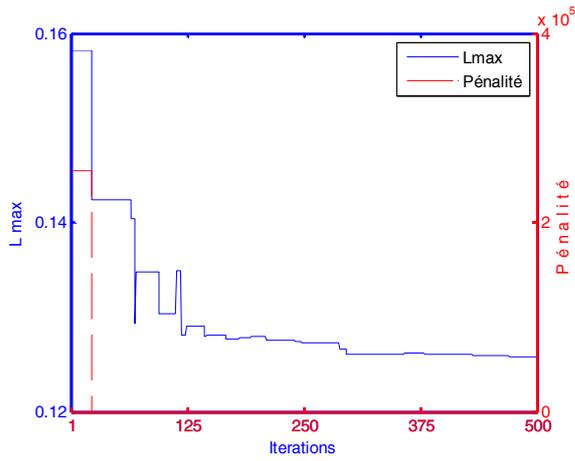


Algorithme PSO

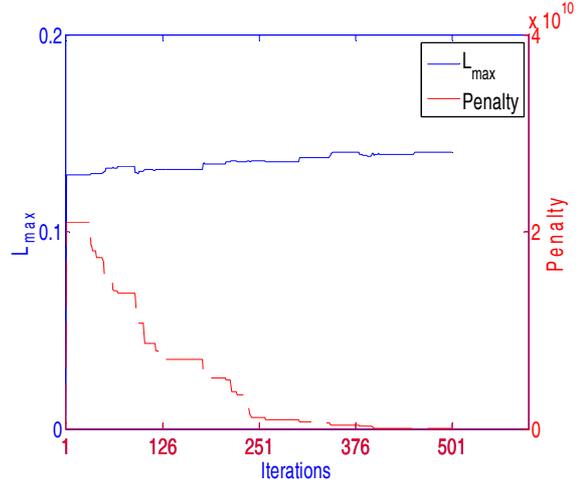


Algorithme DE

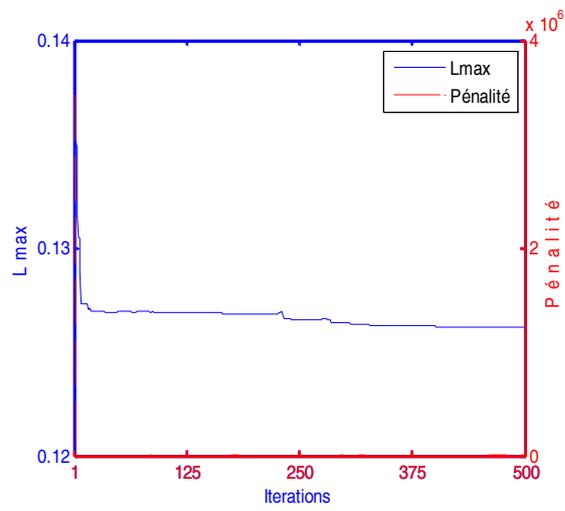
Fig. V.5.a Évolution du  $L_{\max}$  et de pénalité au cours des itérations pour le réseau IEEE30



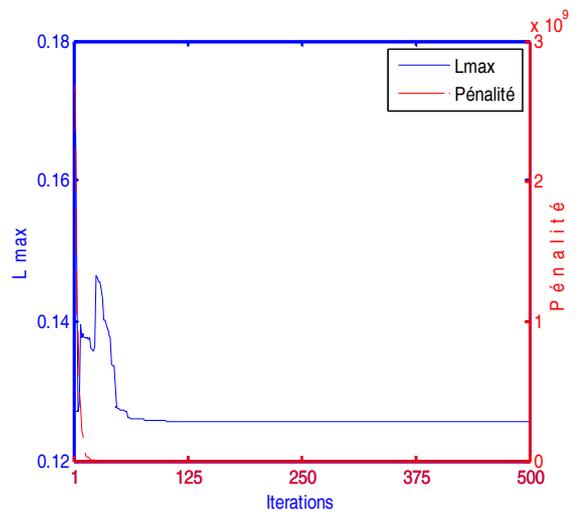
Algorithme ACO



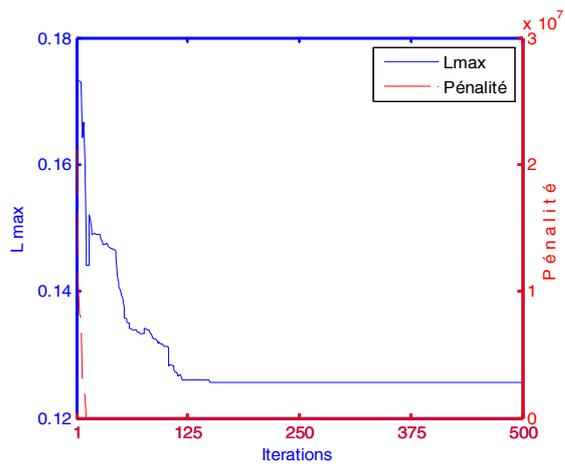
Algorithme HS



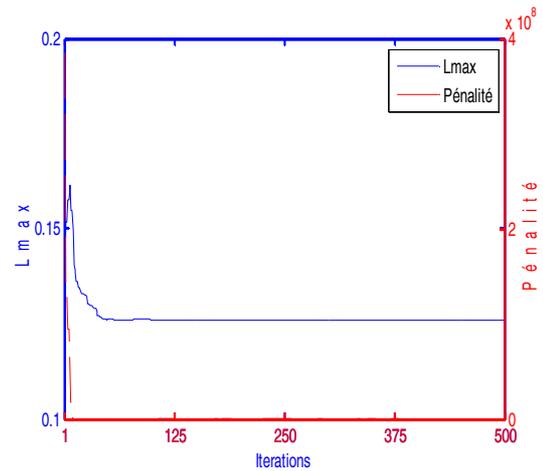
Algorithme EM



Algorithme ABC

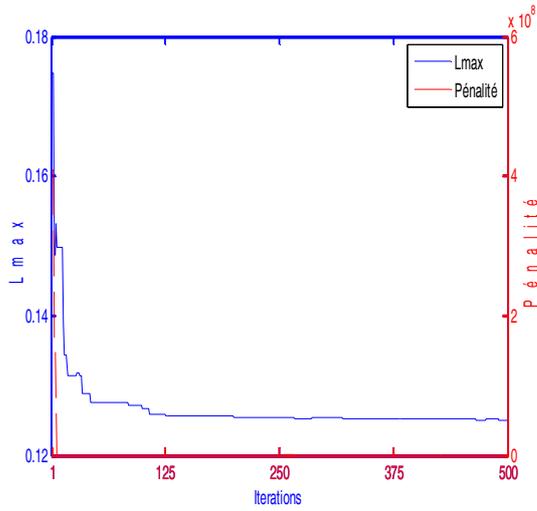


Algorithme ICA

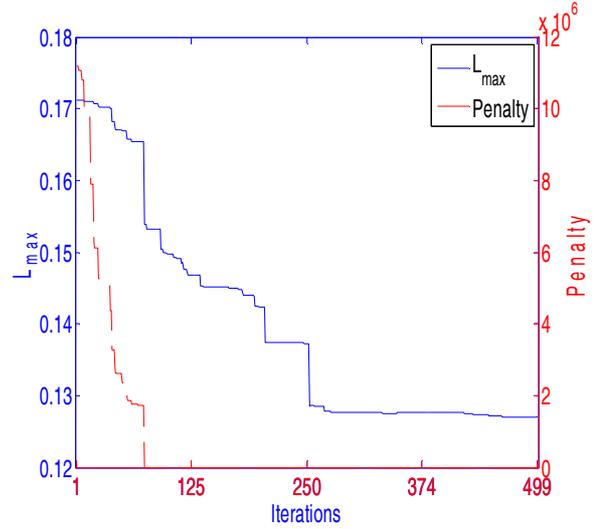


Algorithme BBO

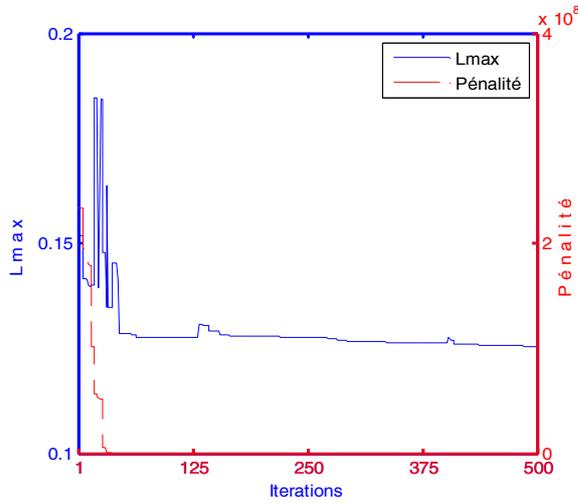
Fig. V.5.b Évolution du  $L_{max}$  et de pénalité au cours des itérations pour le réseau IEEE30



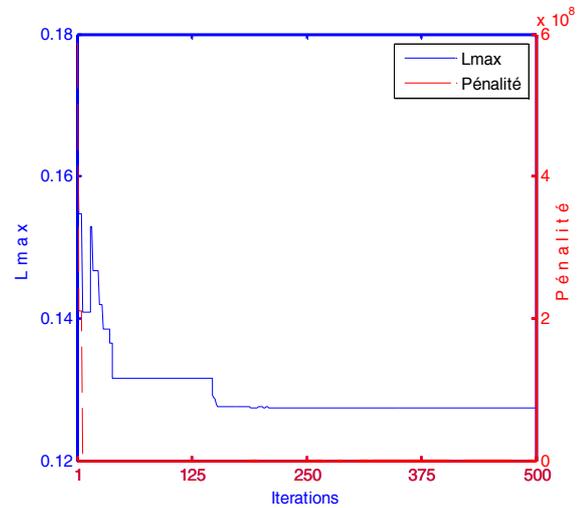
Algorithme FA



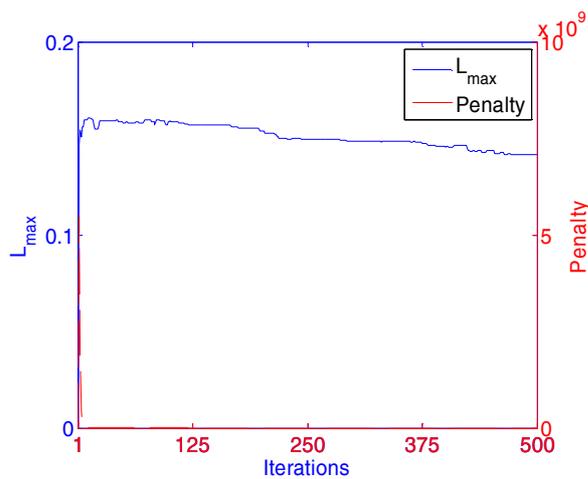
Algorithme LCA



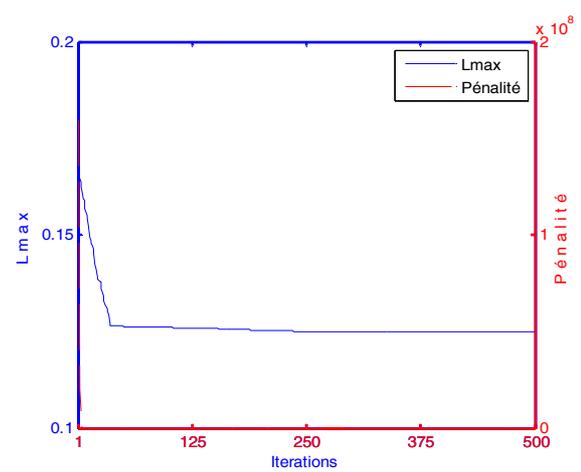
Algorithme GEM



Algorithme GSA

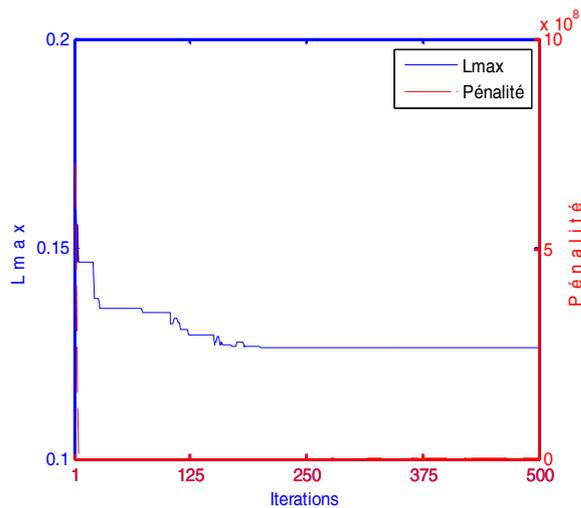


Algorithme WDO

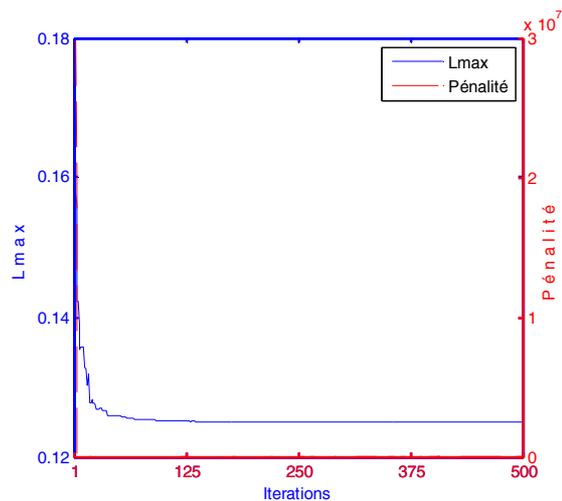


Algorithme ABA

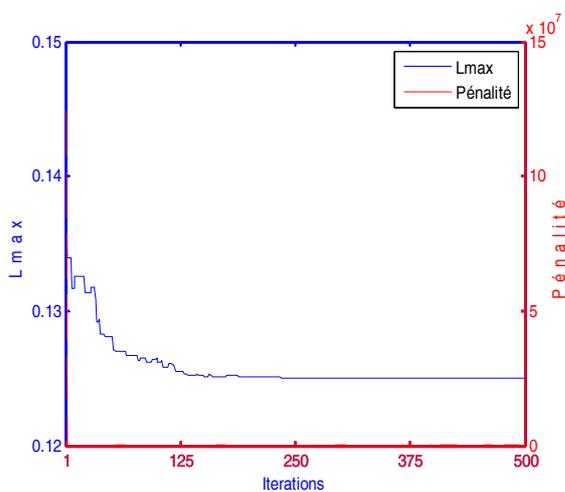
Fig. V.5.c Évolution du  $L_{max}$  et de pénalité au cours des itérations pour le réseau IEEE30



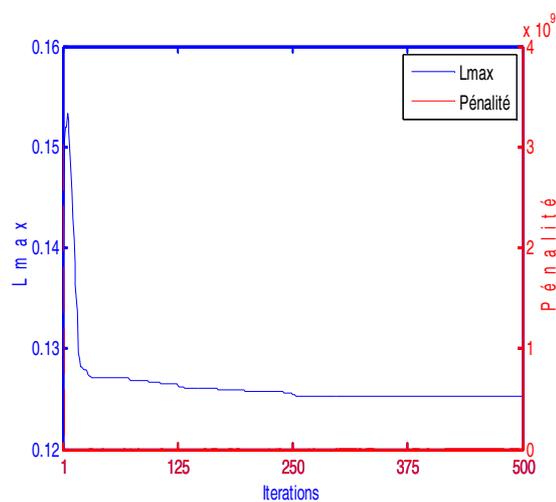
Algorithme CS



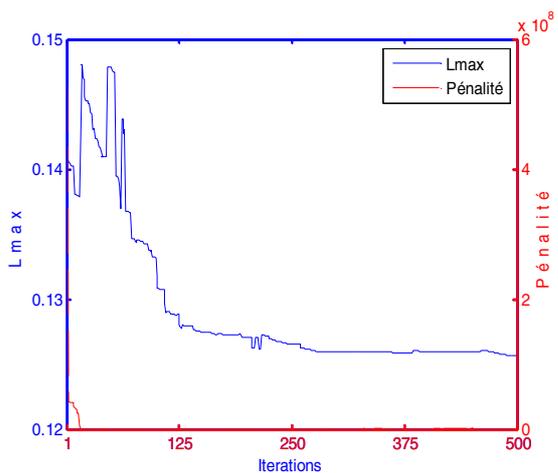
GA-MPC



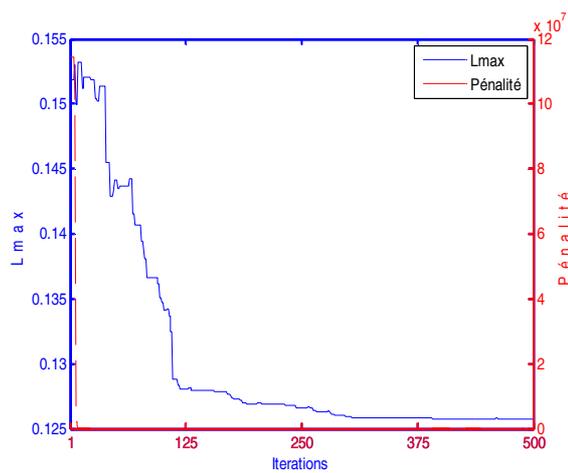
Algorithme TLBO



Algorithme WCA

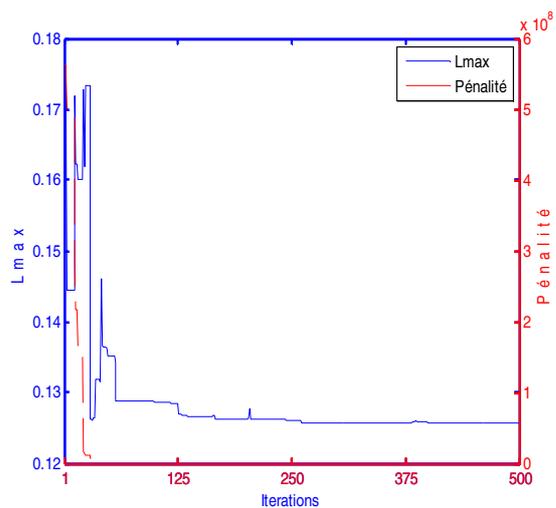


Algorithme DSA

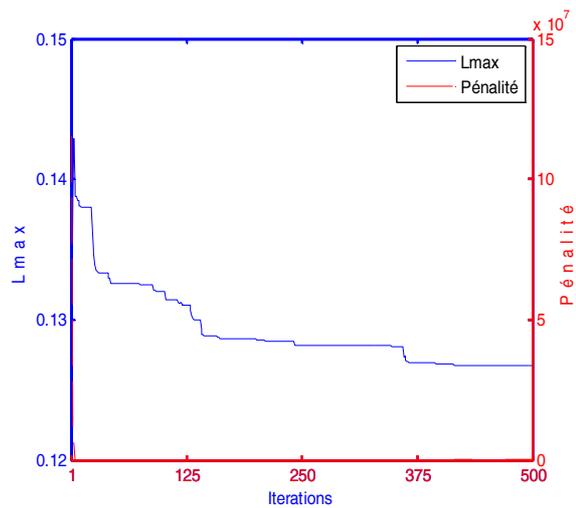


Algorithme MBA

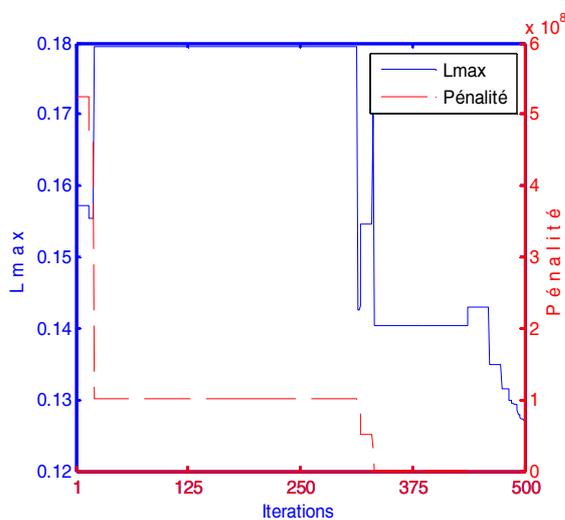
Fig. V.5.d Évolution du L<sub>max</sub> et de pénalité au cours des itérations pour le réseau IEEE30



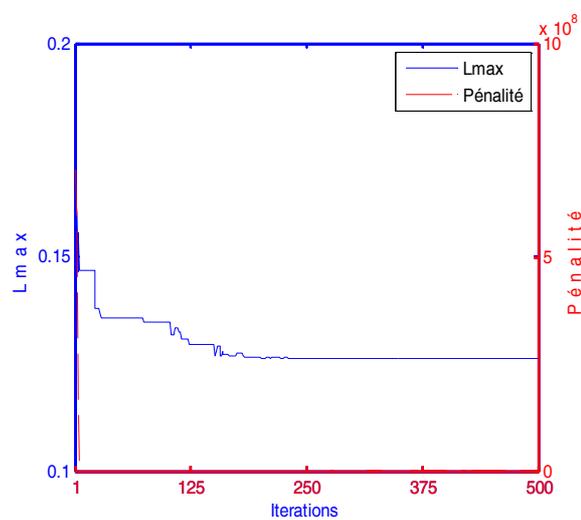
Algorithme BSA



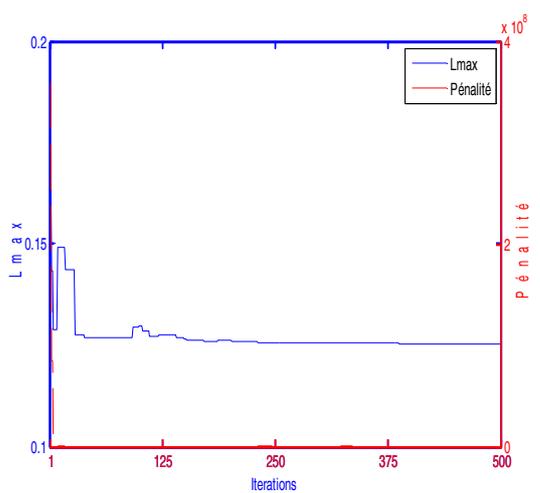
Algorithme BHBO



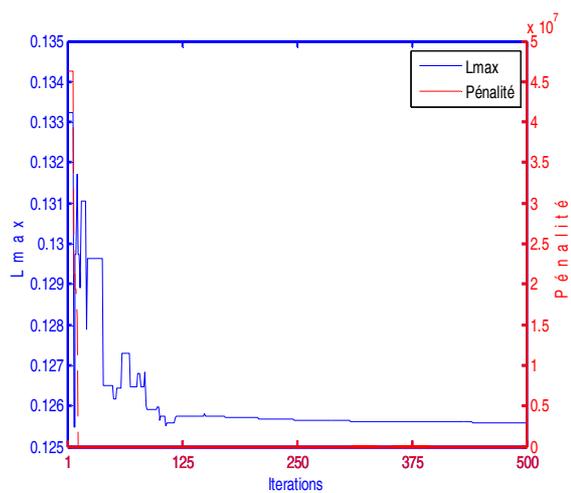
Algorithme GWO



Algorithme CSO

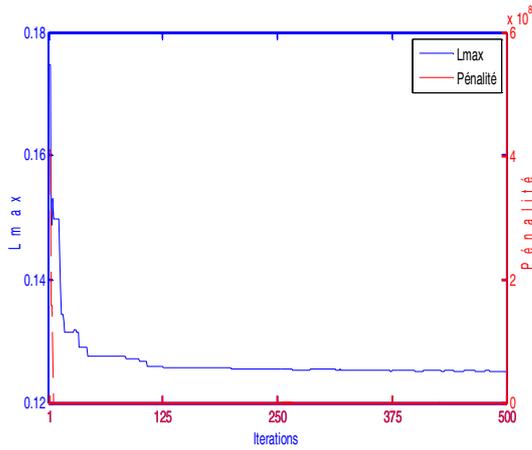


Algorithme JAYA

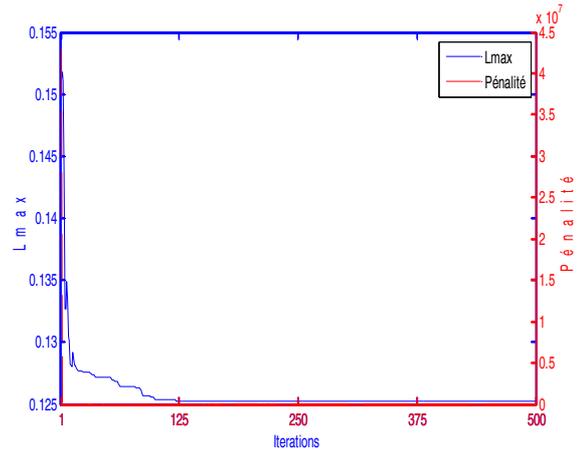


Algorithme CBO

Fig. V.5.e Évolution du L<sub>max</sub> et de pénalité au cours des itérations pour le réseau IEEE30



Algorithme FWA



Algorithme MVPA

Fig. V.5.f Évolution du  $L_{\max}$  et de pénalité au cours des itérations pour le réseau IEEE30

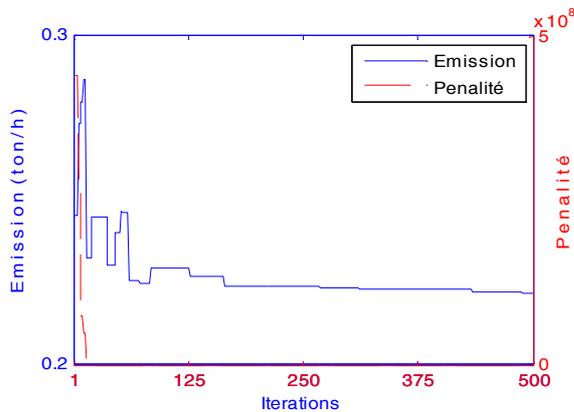
#### V.1.1.4. Minimisation de l'émission gaz (cas 4)

Pour ce cas, nous avons minimisé les deux fonctions objectifs liées au coût de production et l'émission de gaz des unités de production.

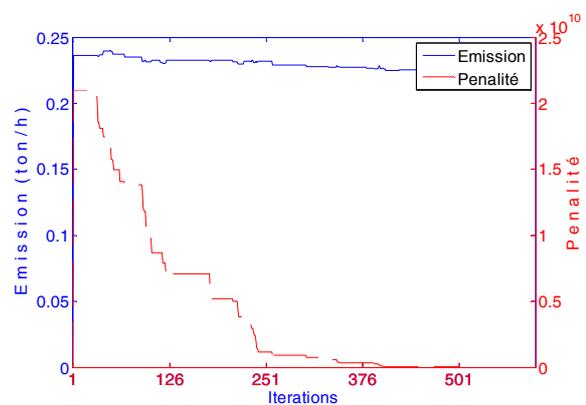
Les résultats d'optimisation sont illustrés dans le groupe des figures V. 6. a, b, c, d, e.

D'après ces résultats, on remarque que les algorithmes GA-MPC, ABC, ABA, GSA, BBO, PSO et MVPA convergent dans les 100 premières itérations, tandis que les autres nécessitent un nombre d'itération plus élevé.

Les valeurs optimales concernant l'émission gaz sont très proches. La meilleure valeur 0.2184 ton/h est obtenue par l'ICA, la deuxième meilleure valeur 0.222 ton/h est obtenue par l'ACO, DE, TLBO, GA-MPC, JAYA, WCA, MVPA, LCA, CBO, GEM, BBO, DSA et FA. La mauvaise valeur 0.3441 ton/h est obtenue par le WDO.



Algorithme GA



Algorithme SA

Fig. V.1.a Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30

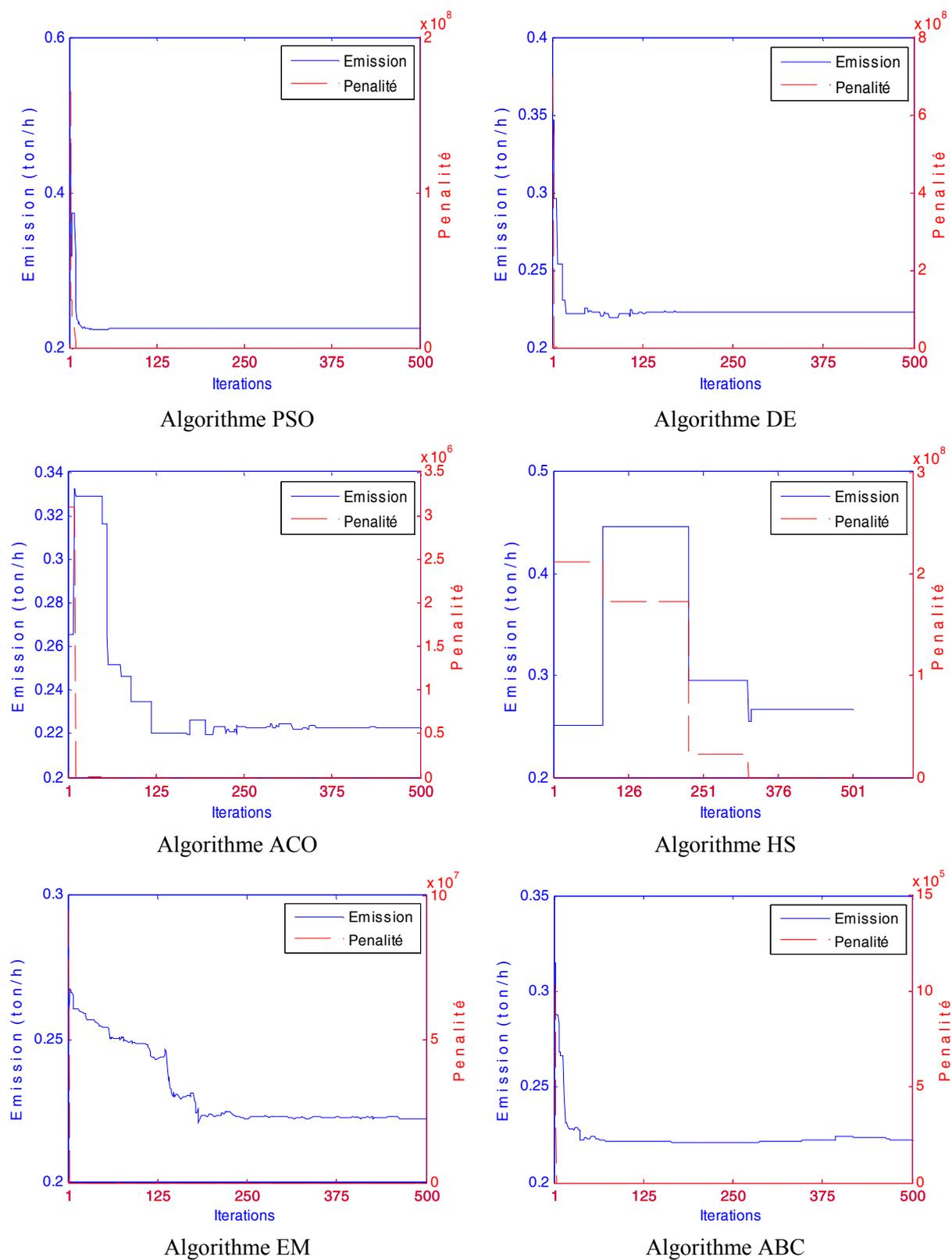
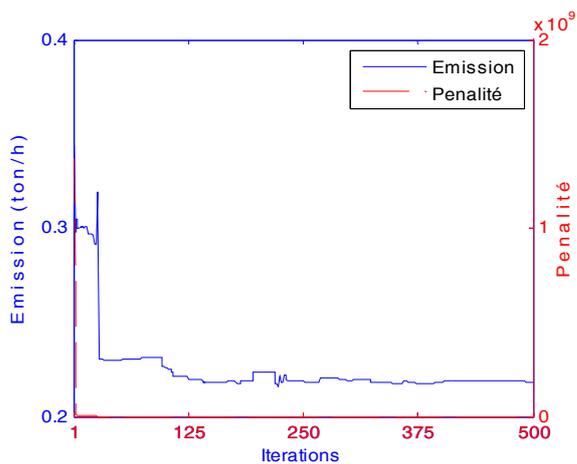
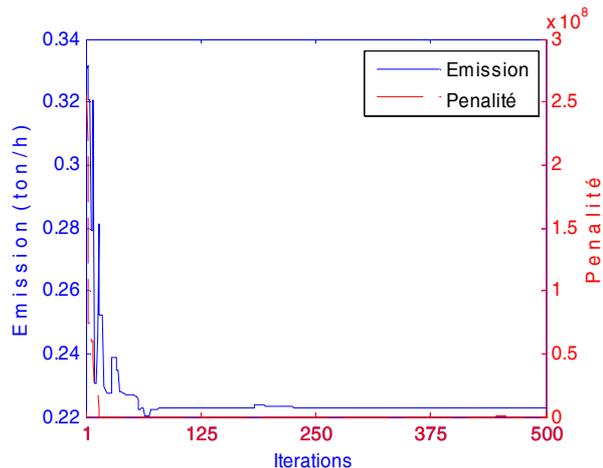


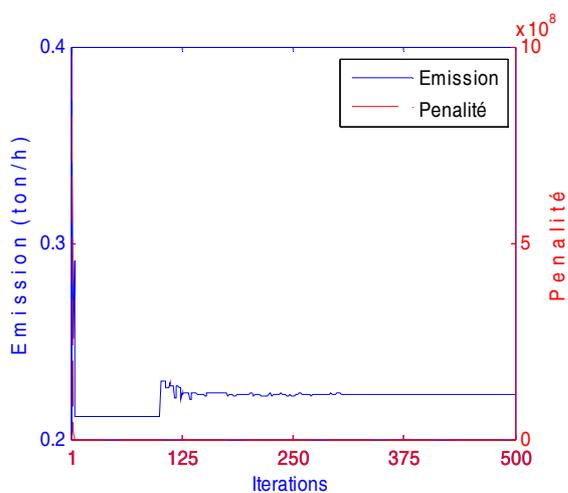
Fig. V.1.b Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30



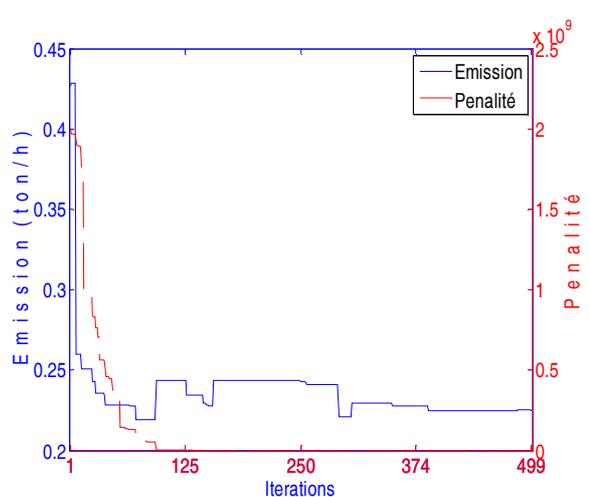
Algorithme ICA



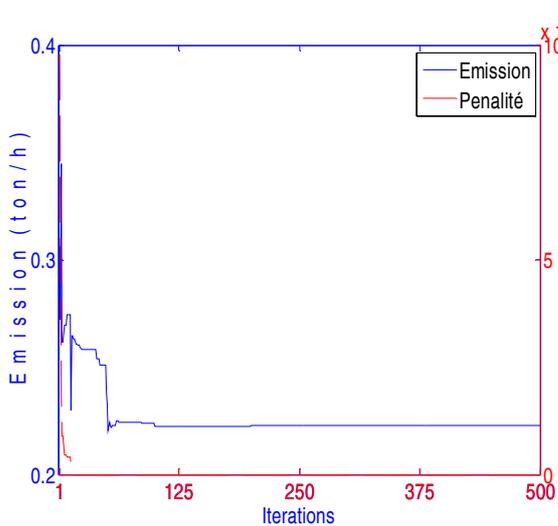
Algorithme BBO



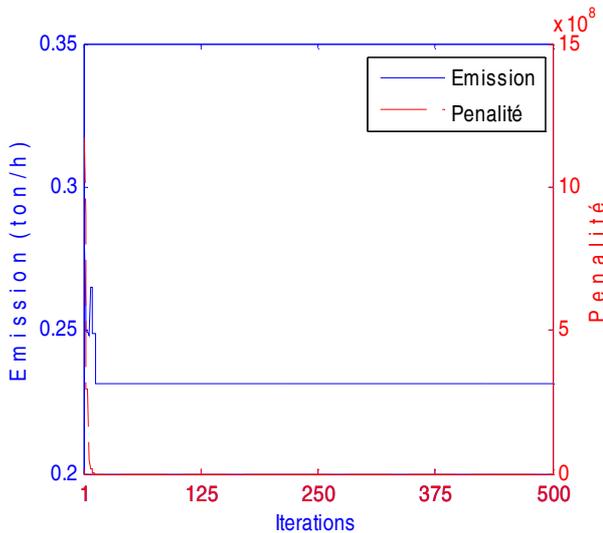
Algorithme FA



Algorithme LCA

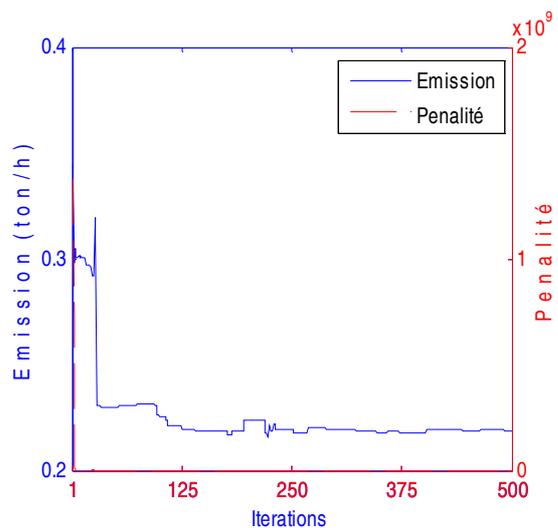


Algorithme GEM

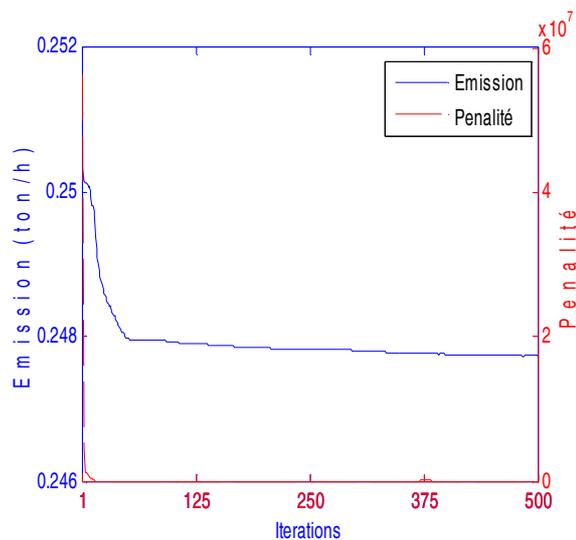


Algorithme GSA

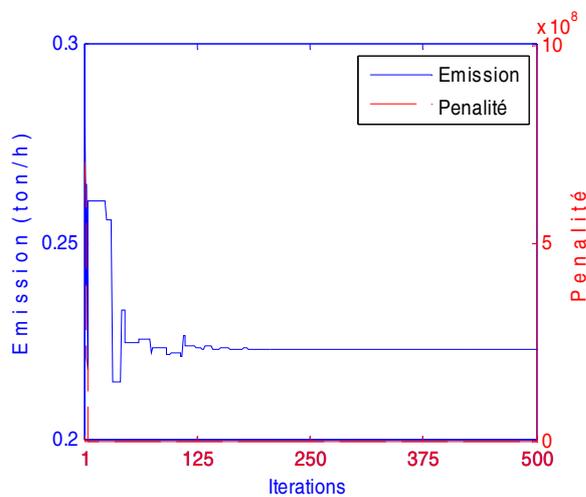
Fig. V.1.c Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30



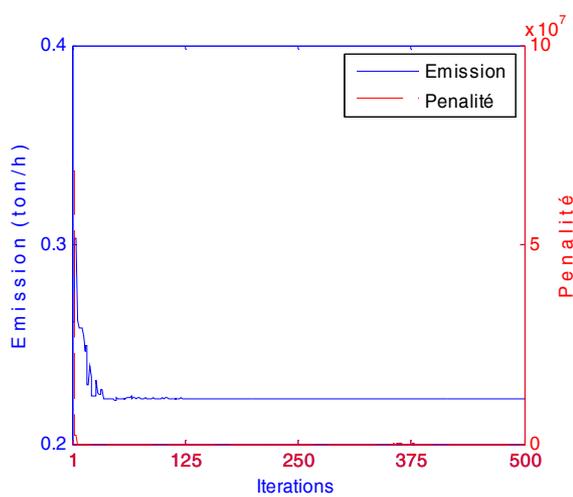
Algorithme WDO



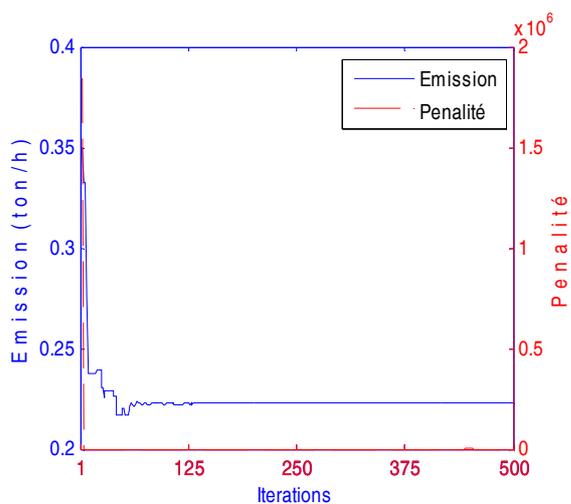
Algorithme ABA



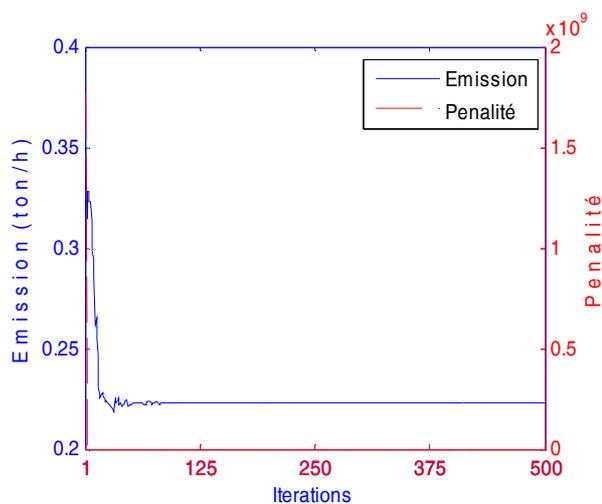
Algorithme CS



GA-MPC

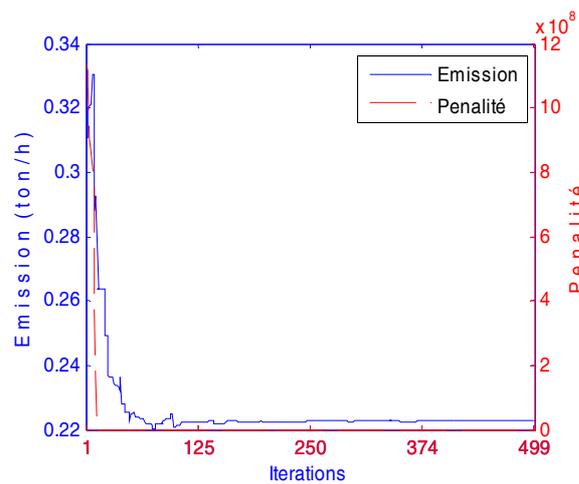


Algorithme TLBO

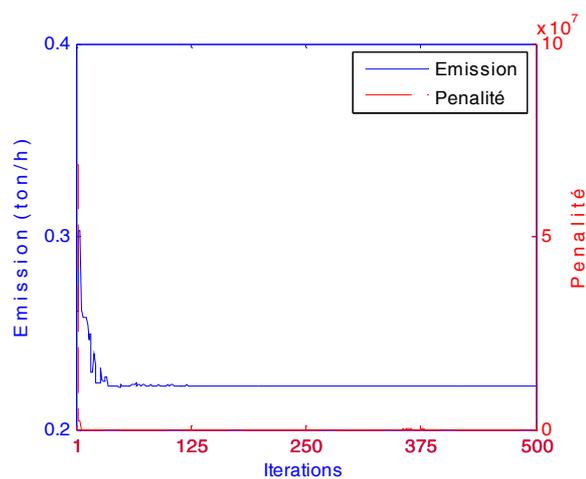


Algorithme WCA

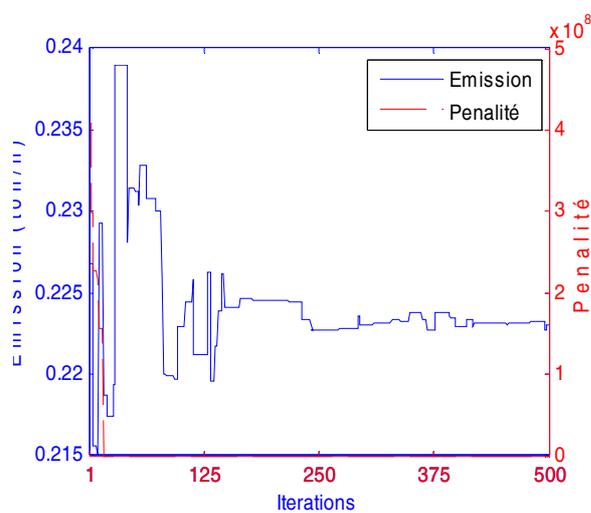
Fig. V.1.d Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30



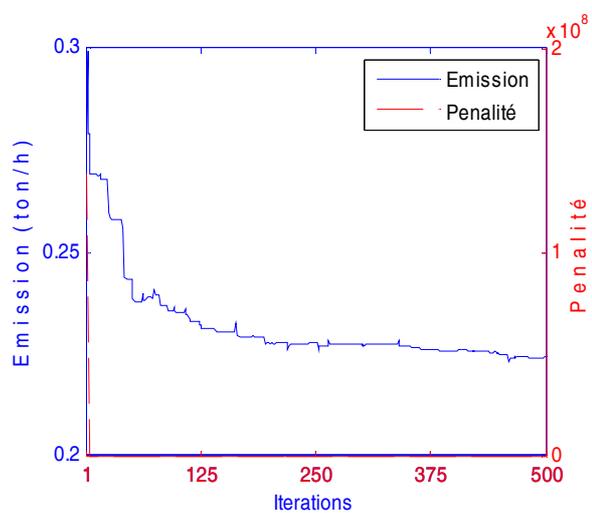
Algorithme DSA



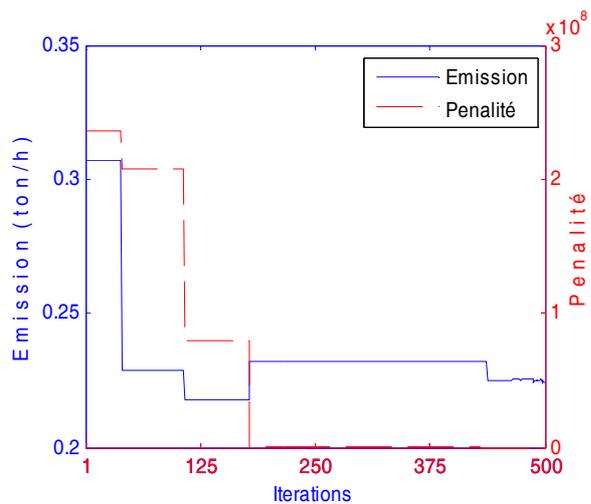
Algorithme MBA



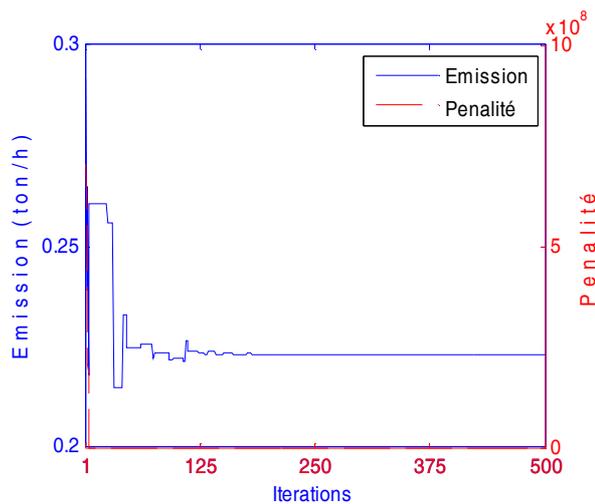
Algorithme BSA



Algorithme BHBO



Algorithme GWO



Algorithme CSO

Fig. V.1.e Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30

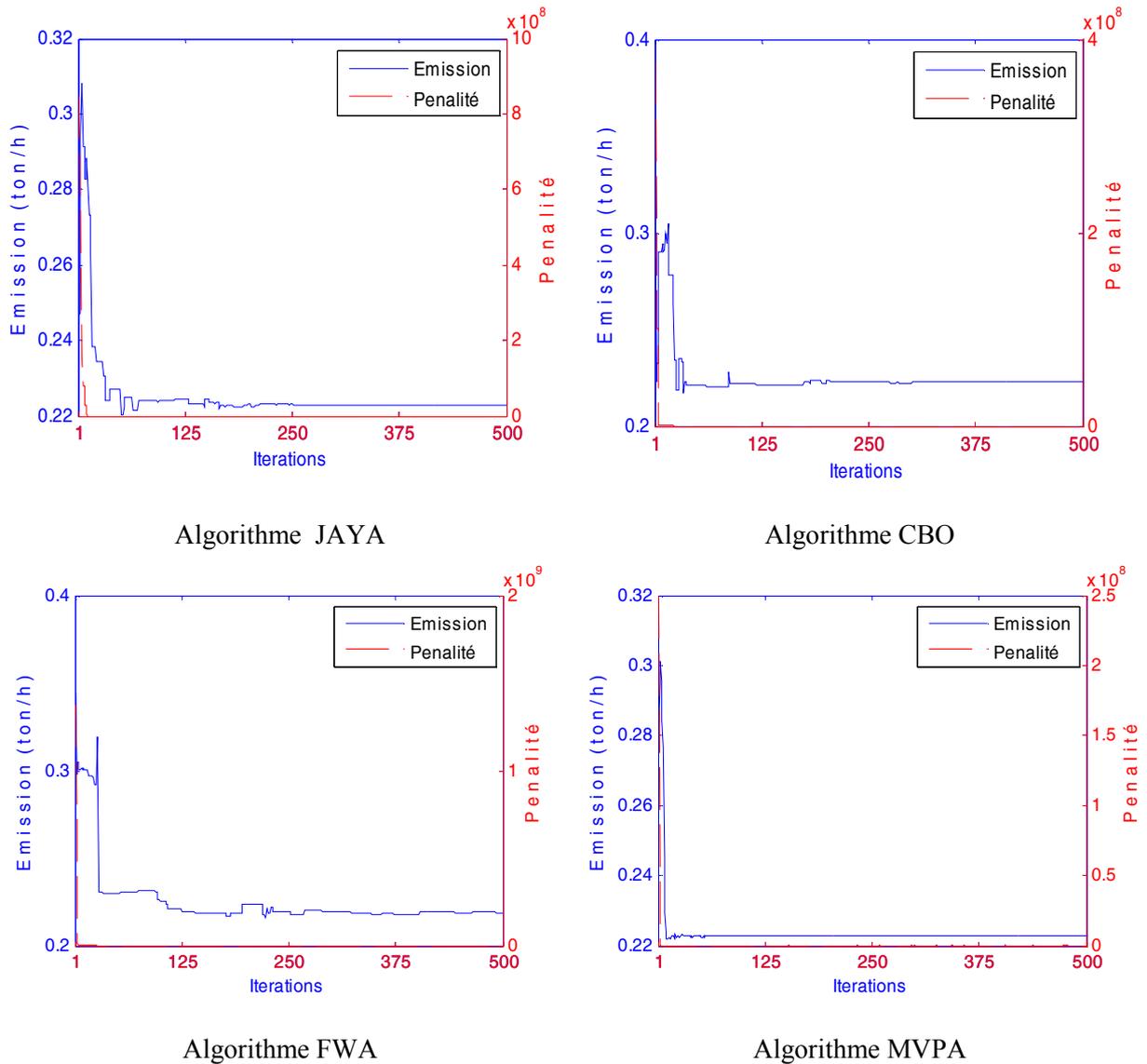


Fig. V.1.f Évolution de l'émission et de pénalité au cours des itérations pour le réseau IEEE30

### V.1.1.5. Minimisation du coût de carburant (cas 5 à 9)

Dans ce cas, on minimise le coût de carburant nécessaire à la production avec la considération de plusieurs phénomènes liés aux unités de production (multitude de combustible et effet de l'ouverture des vannes) et au réseau électrique (introduction des dispositifs FACTS et problème des contraintes de sécurité).

Les résultats obtenus pour toutes les considérations précitées avec un classement des algorithmes en fonction des performances sont illustrés dans le tableau V.5.

**Tableau V.5. Valeurs optimales du coût de carburant des cas traités (cas 5 à 9)**

Cas 5 Multitude de combustibles		Cas 6 Effet de l'ouverture des vannes		Cas 7 Multitude de combustibles et ouverture des vannes		Cas 8 introductions des dispositifs FACTS		Cas 9 problème de contrainte de sécurité SCOPF	
GA-MPC	644,99	ABA	781,25	GA-MPC	681	SA	798,16	SA	798,54
TLBO	645,00	GA-MPC	830,44	DE	681,2	FA	798,99	ICA	798,94
MVPA	645,00	TLBO	830,44	TLBO	681,2	ICA	798,99	TLBO	799,03
DE	645,10	DE	830,44	MVPA	681,5	TLBO	799,00	GA-MPC	799,03
JAYA	645,40	MVPA	830,45	JAYA	681,9	GA-MPC	799,00	MVPA	799,03
FA	645,70	JAYA	830,45	WCA	682,1	MVPA	799,00	JAYA	799,03
DSA	646,00	WCA	830,51	FA	682,3	JAYA	799,00	WCA	799,03
CBO	646,00	DSA	830,66	CBO	682,7	DE	799,00	DE	799,03
WCA	646,03	FA	830,67	ICA	682,9	WCA	799,00	LCA	799,04
BSA	646,30	ICA	830,72	ABC	683,4	LCA	799,00	DSA	799,05
MBA	646,30	ACO	830,73	MBA	683,4	ABC	799,02	ACO	799,07
ICA	646,47	BSA	830,90	FWA	683,5	DSA	799,03	ABC	799,07
FWA	647,17	CBO	830,90	ACO	683,9	BBO	799,06	BSA	799,09
GEM	647,90	GEM	831,12	DSA	684,3	ACO	799,07	BBO	799,11
BBO	647,90	MBA	831,20	CSO	684,3	BSA	799,075	FA	799,11
ABC	648,00	BBO	831,48	EM	685	GEM	799,09	GEM	799,15
LCA	648,40	EM	831,76	BBO	685,1	CBO	799,14	MBA	799,21
CSO	648,50	ABC	831,77	BHBO	685,26	MBA	799,31	CBO	799,26
BHBO	648,80	FWA	832,21	BSA	685,4	FWA	799,32	FWA	799,37
ACO	649,38	LCA	832,58	GSA	686,2	EM	799,47	EM	799,56
EM	649,40	PSO	832,69	CS	686,7	PSO	800,00	PSO	800,20
PSO	649,60	GA	833,13	GEM	687,6	CS	800,35	CSO	800,35
GA	649,80	CSO	833,74	ABA	689,07	GA	800,58	BHBO	800,78
CS	649,90	CS	835,42	GA	689,5	BHBO	800,68	GA	800,86
SA	650,30	BHBO	837,74	GWO	690,49	CSO	801,45	WDO	800,87
GSA	654,40	WDO	842,01	LCA	690,9	WDO	801,66	CS	800,94
WDO	656,80	GWO	843,70	WDO	694,6	ABA	802,59	GWO	803,80
GWO	658,18	GSA	851,99	SA	696,7	HS	806,08	ABA	805,55
ABA	661,50	SA	855,64	HS	739,3	GSA	807,34	HS	808,83
HS	665,10	HS	874,26	PSO	799,99	GWO	809,62	GSA	808,96

Les résultats obtenus ont montré que pour le cas 5 (Tableau V.5) tous les algorithmes ont convergé. Les algorithmes GA-MPC, MVPA, GEM, BHBO, PSO, GSA et ABA ont convergé dans les premières 50 itérations alors que les autres algorithmes ont nécessité plus de 200

itérations. La meilleure valeur 644.99\$/h est obtenue par l'GA-MPC. La deuxième meilleure valeur 645.00\$/h est obtenue par le TLBO et le MVPA. La mauvaise valeur 715.6\$/h (la plus grande) est obtenue par le HS.

Il est à noter que la multitude de types de carburant a considérablement réduit le coût de production.

En ce qui concerne le cas lié à l'ouverture des vannes (cas 6) source de perturbations sur le débit de carburant, on voit une augmentation du coût de carburant par rapport au cas 1. La plus grande augmentation (69\$/h) concerne l'algorithme HS alors que la plus petite 20\$/h concerne l'algorithme EM.

Concernant les performances des algorithmes, la meilleure valeur optimale 781.25 \$/h est obtenue par le ABA, la deuxième meilleure valeur (830,44\$/h) obtenue par GA-MPC, TLBO et le MVPA. Ces derniers ont convergé dans les premières 50 itérations.

Les valeurs les plus mauvaises 874.26\$/h, 830.66\$/h, 830.76\$/h ont été respectivement obtenues par le HS, DSA et le FA.

Pour le cas 7, la meilleure valeur 681.0 \$/h est obtenue par le GA-MPC. Le DE et le TLBO ont donné la deuxième meilleure valeur 681.1 \$/h alors que les plus grandes valeurs 739.3\$/h et 799.9\$/h sont obtenues respectivement par le HS et le PSO.

Pour le cas 8, nous avons introduit le TCSC au niveau de la ligne N°36 avec un degré de compensation de 0.1 et le SVC au niveau de jeu de barre N°22 avec un facteur de compensation de 0.05, les résultats obtenus ont montré une diminution de coût pour certains algorithmes par rapport au premier cas sans FACTS. Une meilleure amélioration est obtenue par le SA (798.16\$/h au lieu de 812.76\$/h). Une mauvaise valeur de 807.34\$/h est obtenue par le GSA.

Nous signalons aussi que l'introduction des dispositifs FACTS influe beaucoup plus sur la stabilité de la tension et les pertes de puissance. A titre d'exemple, le SA a amélioré le  $L_{\max}$  d'une valeur 0.0421 pu et les pertes de puissance réactive d'une valeur 3.6742 MVAR.

Pour le cas 9, nous avons considéré des contraintes de sécurité en mode préventif de déclenchement de la ligne 20 du réseau considéré (IEEE30).

Une meilleure valeur de 798.54\$/h est obtenue par le SA. Les algorithmes JAYA, TLBO, GA-MPC, MVPA, WCA et DE ont donné la deuxième meilleure valeur 799.03\$/h alors que la plus grande 895.56\$/h est donnée par le SCA.

### V.1.2. Comparaison des méthodes d'optimisation pour le réseau IEEE57

Dans cette partie, on procède au traitement du cas de réseau IEEE57 ayant une puissance totale générée de 1975.9 MW et caractéristiques mentionnés dans le tableau V.6.

**Tableau V.6. Caractéristiques du réseau IEEE57**

Caractéristiques du système	Nombre	Détails
ligne	57	-
Branches	80	-
Générateur	7	Jeux de barres : 1, 2, 3, 6, 8, 9, 12
Shunts	9	Jeux de barres : 18, 25, 53.
Transformateur	4	Branches:12, 20, 31, 35, 36, 37, 41, 46, 54, 58, 59
Les variables de contrôle	33	-

Les coefficients de coût et d'émission pour ce réseau sont montrés dans le tableau V.7 [66].

**Tableau V.7. Les coefficients de coût et d'émission pour le réseau IEEE57**

Générateur	Jeu de barre	$a$	$b$	$c$	$d$	$e$	$\alpha$	$\beta$	$\gamma$	$\omega$	$\mu$
G <sub>1</sub>	1	0	2	0.00375	18	0.037	4.091	-5.554	6.49	$2 \times 10^{-4}$	0.286
G <sub>2</sub>	2	0	1.75	0.0175	16	0.038	2.543	-6.047	5.638	$5 \times 10^{-4}$	0.333
G <sub>3</sub>	3	0	3	0.025	13.5	0.041	6.131	-5.555	5.151	$1 \times 10^{-5}$	0.667
G <sub>4</sub>	6	0	2	0.00375	18	0.037	3.491	-5.754	6.39	$3 \times 10^{-4}$	0.266
G <sub>5</sub>	8	0	1	0.0625	14	0.04	4.258	-5.094	4.586	$1 \times 10^{-6}$	0.8
G <sub>6</sub>	9	0	1.75	0.0195	15	0.039	2.754	-5.847	5.238	$4 \times 10^{-4}$	0.288
G <sub>7</sub>	12	0	3.25	0.00834	12	0.045	5.326	-3.555	3.38	$2 \times 10^{-3}$	0.2

Les mêmes étapes d'étude appliquées pour le réseau précédent ont été suivies, vu la taille importante de travail à réaliser, l'effet des FACTS et les contraintes de sécurité n'ont pas été pris en compte pour ce cas de réseau. D'autre part, et tenant compte de non disponibilité des données concernant les cas d'émission, la multitude de combustible et l'effet d'ouverture des vannes, l'étude a été limitée aux cas 1, 2 et 3.

Le classement des 30 méthodes appliquées en fonction des performances des résultats obtenus pour les trois cas traités sont illustrés dans le tableau V.8 :

**Tableau V.8 Valeurs optimales des cas traités (cas 1 à 3) et classement des méthodes**

Cas1 (Coût)		Cas 2 (Déviation standard VD)		Cas3 (Indice de stabilité $L_{max}$ )	
Algorithme	Coût	Algorithme	VD	Algorithme	$L_{max}$
GA-MPC	4 805,00	MVPA	0,680	GA-MPC	0,250
DSA	4 813,00	GA-MPC	0,705	ABA	0,254
DE	4 817,40	TLBO	0,709	DSA	0,259
WCA	4 818,00	FA	0,716	JAYA	0,273
GEM	4 832,00	MBA	0,745	MVPA	0,273
LCA	4 834,00	WCA	0,755	WCA	0,274
BBO	4 837,00	DSA	0,760	DE	0,274
TLBO	4 837,00	GEM	0,769	TLBO	0,275
ABC	4 838,00	ICA	0,771	FA	0,276
BSA	4 839,00	ABC	0,789	BBO	0,279
CBO	4 859,00	BBO	0,800	MBA	0,280
FWA	4 867,00	ABA	0,806	BSA	0,282
GA	5 192,90	JAYA	0,818	CBO	0,282
ICA	5 384,00	BSA	0,821	ICA	0,287
ABA	5 427,70	CBO	0,882	GEM	0,290
BHBO	5 603,00	FWA	0,890	LCA	0,291
GWO	5679.10	DE	1.024	ACO	0.293
WDO	5 901,00	ACO	1,026	ABC	0,295
PSO	5 943,00	CSO	1,055	FWA	0,297
MVPA	6 410,00	EM	1,060	GA	0,297
FA	6 415,00	BHBO	1,073	WDO	0,298
JAYA	6 417,00	LCA	1,170	CS	0,300
ACO	6 539,70	CS	1,173	EM	0,301
MBA	6 589,10	GWO	1,470	GWO	0,305
HS	6 626,00	SA	1,542	CSO	0,308
CSO	6 667,20	PSO	1,582	PSO	0,312
EM	6 698,10	GA	1,605	BHBO	0,327
CS	8 010,10	WDO	1,777	SA	0,352
SA	9 355,20	HS	3,096	GSA	0,360
GSA	9 597,50	GSA	4,522	HS	0,365

Le calcul effectué a montré que tous les algorithmes ont convergé sauf le CS, SA et le GSA qui ont, en plus, donné un taux de pénalité trop élevé et résultats mauvais. Malgré leur non convergence, elles ont donnée des valeurs optimales acceptables.

Le meilleur résultat pour le premier et le troisième cas est obtenu par l'algorithme GA-MPC alors que pour le deuxième cas (déviation standard), le meilleur résultat est obtenu par le MVPA et le deuxième meilleur par le GA-MPC.

### V.1.3. Comparaison des méthodes d'optimisation pour le réseau IEEE 118

Le réseau IEEE 118 est un réseau à grande échelle, il comprend 54 générateurs, 186 branches, 9 transformateurs et 12 condensateurs. Les données de ligne, des jeux de barres et de générateur ainsi que les limites minimales et maximales pour les variables de contrôle sont données dans [273].

Pour ce réseau, et comme le précédent, sauf les cas 1 (coût), 2 (déviatation standard) et 3 (indice de stabilité) sont considérés. Les résultats de calcul obtenus sont illustrés dans le tableau V.9.

**Tableau V.9. Valeurs optimales des cas traités (cas 1 à 3)**

Cas 1 (Coût)		Cas 2 (Déviatation standard VD)		Cas 3 (Indice de stabilité $L_{max}$ )	
Algorithme	Coût	Algorithme	VD	Algorithme	$L_{max}$
GA-MPC	135 760,00	GA-MPC	0,6955	GA-MPC	0,0682
ABC	136 610,00	DSA	0,8844	DSA	0,0687
MVPA	136 780,00	BSA	0,903	WCA	0,0689
BBO	137 690,00	ABC	0,9278	MVPA	0,0689
ICA	138 450,00	MVPA	0,9719	ICA	0,0693
FA	143 770,00	ICA	0,9755	BBO	0,0694
WCA	144 980,00	FA	1,1938	ABC	0,0696
DSA	145 320,00	BBO	1,2092	FA	0,0703
GEM	148 620,00	WCA	1,2099	GEM	0,0717
BSA	148 840,00	GEM	2,7313	BSA	0,0726
TLBO	243 640,00	GSA	10,7987	JAYA	10,0636
JAYA	246 450,00	BHBO	10,8524	HS	10,0641
WDO	247 430,00	CSO	10,9184	LCA	10,0678
FWA	253 170,00	CS	11,183	CSO	10,0685
MBA	254 990,00	GA	11,3363	CBO	10,0701
LCA	257 080,00	EM	11,3456	TLBO	10,0705
GWO	257 342,53	CBO	11,5186	BHBO	10,0705
CS	260 500,00	LCA	11,5914	GA	10,0709
CBO	263 280,00	SA	11,6313	EM	10,0712
CSO	264 360,00	ACO	11,6573	GSA	10,0712
GA	265 890,00	HS	12,1049	CS	10,0724
BHBO	265 940,00	JAYA	12,1525	ACO	10,0742
EM	271 880,00	GWO	12,4258	SA	10,0749
ACO	273 900,00	ABA	12,4715	GWO	10,0749
ABA	277 660,00	DE	12,5102	ABA	10,0749
DE	280 690,00	FWA	12,7313	PSO	10,0754
GSA	284 570,00	PSO	12,756	DE	10,076
HS	284 930,00	TLBO	12,8633	FWA	10,0761
PSO	293 190,00	MBA	13,2061	MBA	10,0764
SA	321 670,00	WDO	14,9096	WDO	10,0824

Pour ce réseau, et d'après les résultats du Tableau VI-9, seuls les algorithmes BSA, GEM, ABC, BBO, GA-MPC, WCA, DSA, FA, ICA, MVPA ont convergé. Malgré que les autres n'ont pas convergé et ont fourni des termes de pénalité très élevés ils ont donné relativement de bonne valeurs optimales par rapport à ceux convergeant. A cet effet, et afin de leur déclasser suite à leur non convergences, la valeur optimale du coût est majorée par le rajout de 100 000€/h et celles de la déviation et l'indice de stabilité sont majorées par le rajout de 10 PU. Pour les trois cas étudiés, l'algorithme GA-MPC nous a donné les meilleurs résultats avec un coût de 135 760.00€/h, une déviation standard de 0.6955 PU et un indice de stabilité de 0.0682 PU.

#### V.1.4. Comparaison des méthodes d'optimisation pour le réseau Algérien 59 jeux de barres

Après avoir testé la fiabilité des méthodes d'optimisation adoptées à des réseaux standards, on présente dans cette partie les résultats d'étude effectuée sur le réseau Algérien à 50 jeux de barres et puissance demandée 684.10 MW. Les caractéristiques de ce réseau sont montrées dans le tableau V.10.

**Tableau V.10. Caractéristiques du réseau Algérien 59 [274].**

Caractéristique du système	nombre	Détaille
ligne	59	-
Branche	83	-
Générateur	10	Jeux de barres : 1, 2, 3, 4, 13, 27, 37, 41, 42, 53
Transformateur	12	Branches:15, 29, 31, 36, 45, 47, 48, 57, 64, 72, 82, 83
Les variables de contrôle	19	

Les coefficients de la fonction coût des dix générateurs de ce réseau sont illustrés dans le tableau V.11.

**Tableau V.11. Les coefficients de coût et d'émission pour le réseau algérien**

Générateur	$a$	$b$	$c$	$d$	$e$	$\alpha$	$\beta$	$\gamma$	$\omega$	$\mu$
G <sub>1</sub>	0	1,5	0,0085	64	0,074	4,09	5,554	6,49	0,0002	2,86
G <sub>2</sub>	0	2,5	0,017	58	0,076	2,54	6,047	5,64	0,0005	3,33
G <sub>3</sub>	0	1,5	0,0085	70	0,08	4,26	5,094	4,59	0,0000	8,00
G <sub>4</sub>	0	1,5	0,0085	82	0,09	5,33	3,550	3,38	0,0020	2,00
G <sub>5</sub>	0	2,5	0,017	49	0,084	4,26	5,094	4,59	0,0000	8,00
G <sub>6</sub>	0	2,5	0,017	54	0,082	6,13	5,555	5,15	0,0000	6,67
G <sub>7</sub>	0	2	0,003	64	0,07	4,09	5,554	6,49	0,0002	2,86
G <sub>8</sub>	0	2	0,003	68	0,076	2,54	6,047	5,64	0,0005	3,33
G <sub>9</sub>	0	2	0,003	52	0,084	4,26	5,094	4,59	0,0000	8,00
G <sub>10</sub>	0	1,5	0,0085	70	0,098	5,33	3,550	3,38	0,0020	2,00

En se limitant aux cas 1 à 4 (contraintes de manque de données), les mêmes conditions de calcul adoptées pour les cas de réseaux précédents ont conduit aux résultats du tableau VI- 12.

**Tableau V.12 Valeurs optimales liées aux cas traités (cas 1 à 4)**

Cas 1 (Coût)		Cas 2 (Déviation standard VD)		Cas 3 (Indice de stabilité $L_{max}$ )		Cas 4 (Emission)	
Algorithme	Coût	Algorithme	Algorithme	Coût	Algorithme	ALGO	émission
ICA	1688,4	ICA	1,8186	MBA	0,21	TLBO	0,46
MVPA	1688,6	MVPA	1,8362	ABA	0,2149	ICA	0,49
DSA	1688,7	GEM	1,894	ICA	0,2152	DE	0,49
ABC	1690	DSA	1,9614	GWO	0,2162	DSA	0,5
LCA	1690,2	DE	1,9663	MVPA	0,2165	MBA	0,52
BSA	1695,5	ABC	1,9849	BSA	0,2166	MVPA	0,53
DE	1695,9	BSA	2,0002	GEM	0,2166	ABC	0,54
GEM	1711,9	ABA	2,094	ABC	0,2166	ACO	0,56
TLBO	1744	MBA	2,107	DSA	0,2166	BSA	0,6
ACO	1842,1	WDO	2,1108	ACO	0,2171	ABA	0,71
MBA	1891,6	TLBO	2,1399	TLBO	0,2172	WDO	1,31
WDO	1923,5	ACO	2,3466	DE	0,2176	GEM	1,9702
ABA	1955,9	LCA	2,361	LCA	0,2198	WCA	1,98
BBO	2639,3	GWO	2,5655	WDO	0,2518	GWO	3
GA-MPC	2639,4	WCA	2,6159	BBO	0,2533	LCA	8,35
GWO	2639,5	BBO	2,6487	GA-MPC	0,2534	BBO	8,36
WCA	2639,5	GA-MPC	2,6488	FA	0,2536	GA-MPC	8,37
FA	2639,6	FA	2,6488	WCA	0,2742	FA	8,35
BHBO	2 809,70	SA	11,5812	EM	10,2181	BHBO	10,48
CSO	2 813,20	HS	11,6389	CSO	10,2203	JAYA	10,49
CS	2 852,70	GA	11,68	PSO	10,2234	CS	10,56
GA	2 857,90	BHBO	11,8301	GA	10,2234	EM	10,58
EM	2 965,60	FWA	12,019	CS	10,2268	GA	10,59
GSA	2 990,40	CS	12,0646	HS	10,2307	CSO	10,61
FWA	3 007,00	EM	12,1112	BHBO	10,2317	HS	10,82
JAYA	3 036,00	CSO	12,2784	SA	10,2492	FWA	11,85
CBO	3 111,80	JAYA	12,5302	JAYA	10,2533	SA	12,19
HS	3 147,00	GSA	12,6486	GSA	10,2561	CBO	13,02
PSO	3 171,00	PSO	12,6731	CBO	10,2604	PSO	13,26
SA	3 463,50	CBO	13,3095	FWA	10,2968	GSA	34,06

Selon le tableau V.12, seuls les algorithmes ICA, MVPA, DSA, ABC, LCA, BSA, DE, GEM, TLBO, ACO, MBA, WDO, ABA, BBO ont convergé. Les autres ont divergé en fournissant un terme de pénalité très élevé. Afin de déclasser les algorithmes divergents, nous avons majoré leurs valeurs optimales par le rajout de la valeur 1000\$/h pour le coût et 10PU pour la déviation standard (DST), indice de stabilité  $L_{\max}$  et émission des gaz.

Pour les cas 1 et 2, l'algorithme ICA a donné le meilleur résultat avec un coût de 1688.4\$/h et une déviation standard de 1.1886 PU. Pour le cas 3, c'est l'algorithme MBA qui a donné le meilleur résultat avec un indice de stabilité  $L_{\max} = 0.21$  PU.

Pour le 4ième cas (émission de gaz), le meilleur résultat est obtenu par l'algorithme TLBO avec une valeur de 0.46 ton/h suivi par l'ICA et le DE.

En conclusion, l'étude paramétrique effectuée par application des 30 algorithmes sur les 4 réseaux a montré que l'algorithme GA-MPC est mieux adapté pour les réseaux IEEE 30, 57 et 118 parce qu'il a fourni le meilleur résultat dans la majorité de cas traités. Malgré ses performances, cet algorithme a reculé devant les algorithmes ICA, MBA et TLBO qui ont donné les meilleures performances. Il a donné une valeur de pénalité très élevée.

Pour juger les performances des algorithmes appliqué et les classer pour l'ensemble des problèmes de réseaux traité, nous effectuée une comparaison sur la base du calcul d'une différence par rapport aux algorithmes qui ont fourni le meilleur résultat pour chaque problème. Le résultat de te telle comparaison est figuré dans le Tableau V.13.

Tableau V.13. Sommes des différences par rapport au meilleur algorithme

Réseau Algérienne		réseau IEEE30, 57, 118		tous les réseaux	
Algorithme	somme des différences	Algorithme	somme des différences	Algorithme	somme des différences
ICA	35,20	GA-MPC	0,05	ABC	1 373,52
MVPA	94,30	ABC	883,07	MVPA	2 818,66
DSA	189,70	BBO	1 962,10	ICA	3 523,36
DE	192,80	MVPA	2 625,05	GA-MPC	9 810,27
ABC	254,50	ICA	3 269,00	DSA	9 907,17
BSA	335,30	WCA	9 233,05	BBO	11 916,62
TLBO	384,10	DSA	9 568,08	WCA	12 725,99
MBA	551,60	FA	9 620,11	BSA	13 693,15
ACO	788,80	GEM	12 887,16	GEM	14 716,22
ABA	797,80	BSA	13 114,11	FA	19 481,64
WDO	1 419,10	TLBO	107 912,05	TLBO	108 423,25
GEM	1 615,70	JAYA	112 302,05	WDO	115 535,49
WCA	3 332,60	WDO	112 767,76	MBA	121 799,28
GWO	4 244,20	FWA	117 507,40	GWO	128 075,83
LCA	8 444,00	MBA	121 014,58	LCA	130 452,69
FA	9 715,00	LCA	121 349,61	ACO	141 181,78
BBO	9 724,30	GWO	122 497,94	ABA	143 543,33
GA-MPC	9 734,60	CBO	127 574,15	JAYA	144 675,94
GA	31 174,30	CS	127 948,11	DE	145 578,30
BHBO	31 174,50	CSO	130 464,36	FWA	151 023,38
CS	31 527,10	GA	130 519,22	CS	160 178,17
HS	31 659,60	BHBO	130 979,61	CSO	162 756,88
EM	31 697,90	EM	138 025,33	BHBO	162 762,99
CSO	31 744,90	ACO	139 874,90	GA	162 826,80
JAYA	32 132,50	ABA	142 529,23	CBO	163 429,13
FWA	32 995,80	DE	144 942,45	EM	170 279,44
SA	33 306,90	HS	150 996,95	HS	185 564,48
PSO	35 149,50	GSA	153 608,76	PSO	194 953,66
CBO	35 524,70	PSO	158 568,95	GSA	213 521,42
GSA	55 778,10	SA	190 473,97	SA	225 076,52

## V.2. Comparaison non paramétrique des méthodes d'optimisation

Dans cette partie, on va effectuer une comparaison non paramétrique des algorithmes d'optimisation adoptés. De telle comparaison basée sur le résultat final tient en compte la vitesse de convergence et le niveau de signification de la différence entre algorithmes définie statistiquement. Les méthodes statistiques adoptées sont le test de signes multiples, le test de Friedman et le profil de performance.

### V.2.1. Application de la méthode de test de signes multiples

L'application de la méthode de test de signes multiples a permis de connaître les algorithmes gagnants et ceux perdants en calculant les différences entre les valeurs des fonctions objectif de chaque algorithme par rapport aux valeurs des fonctions objectifs des autres pour les réseaux traités (en total 19 cas d'objectif). Les résultats de telle comparaison sont illustrés dans les tableaux V.14.a et V.14.b.

**Tableau. V.14.a. Nombre des gagnants (signes positifs) par algorithme de contrôle**

	PSO	GA	DE	EM	LCA	CBO	BSA	GEM	ABC	BBO	TLBO	ACO	GA-MPC	BHBO	HS	SA	JAYA	MBA	FWA	GWO	GSA	CS	WCA	CSO	ABA	DSA	WDO	FA	ICA	MVPA
PSO	0	7	1	3	0	2	0	1	0	0	1	2	0	7	11	9	2	4	5	7	11	8	0	6	4	0	9	2	1	2
GA	9	0	3	9	2	5	0	1	0	0	1	5	0	9	16	13	5	4	9	12	15	10	0	8	8	0	12	2	2	2
DE	17	15	0	15	13	14	12	12	12	14	5	14	4	15	16	14	11	15	16	14	16	15	11	15	10	9	17	13	7	5
EM	15	9	3	0	4	5	1	2	1	1	1	4	0	9	14	14	4	3	9	12	15	11	0	8	7	0	10	1	1	1
LCA	18	16	4	14	0	9	4	4	4	8	4	10	4	15	17	16	6	7	13	15	16	16	3	14	9	2	15	6	2	2
CBO	16	13	4	13	8	0	2	6	5	5	2	11	0	13	14	13	2	10	14	13	14	12	1	12	10	1	13	2	5	2
BSA	18	18	6	17	13	13	0	12	5	10	6	13	4	17	18	16	8	12	17	17	18	18	5	17	13	0	18	8	6	3
GEM	17	17	6	16	14	12	5	0	7	8	7	13	4	17	18	15	9	12	16	17	17	17	4	16	13	1	17	6	3	2
ABC	18	18	6	17	13	12	11	10	0	12	6	15	4	18	18	16	9	11	17	17	18	18	6	18	14	4	18	11	4	4
BBO	18	18	4	17	10	13	8	9	6	0	3	11	4	18	18	15	9	10	16	15	18	18	4	17	11	1	14	8	4	2
TLBO	17	17	9	17	13	16	12	11	12	14	0	16	4	16	16	15	13	16	17	16	17	17	11	16	12	10	17	14	9	6
ACO	16	13	4	14	8	7	5	5	2	7	2	0	4	12	16	14	5	10	12	14	15	14	4	13	11	1	14	6	0	1
GA-MPC	18	18	11	18	13	18	14	14	14	14	9	14	0	18	18	16	16	14	18	15	18	18	14	18	12	14	14	15	11	10
BHBO	11	9	3	9	3	5	1	1	0	0	1	6	0	0	15	14	6	4	9	12	17	12	0	7	8	0	12	2	1	2
HS	7	2	2	4	1	4	0	0	0	0	2	2	0	3	0	8	3	3	8	6	9	4	0	3	3	0	3	1	1	1
SA	9	5	4	4	2	5	2	3	2	3	3	4	2	4	10	0	5	5	8	7	10	4	2	4	4	2	7	3	3	3
JAYA	16	13	5	14	11	16	10	9	9	9	3	13	0	12	15	13	0	13	15	13	16	14	7	14	9	7	13	8	6	1
MBA	14	14	3	15	11	8	5	6	6	8	2	8	4	14	15	13	5	0	13	14	15	15	5	14	10	3	15	5	4	3
FWA	13	8	2	9	5	4	1	2	1	2	1	6	0	9	10	10	3	5	0	12	11	11	0	10	6	1	11	2	2	2
GWO	11	6	4	6	3	5	1	1	1	3	2	4	3	6	12	10	5	4	6	0	10	7	2	7	4	1	7	6	1	3
GSA	7	3	2	2	2	4	0	1	0	0	1	3	0	1	9	8	2	3	7	8	0	4	0	1	5	0	6	1	1	1
CS	10	8	3	7	2	6	0	1	0	0	1	4	0	6	14	14	4	3	7	11	14	0	0	5	8	0	8	1	1	1
WCA	18	18	4	18	14	17	13	14	12	14	5	14	2	18	18	16	9	13	18	15	18	18	0	18	11	8	14	11	9	2
CSO	12	10	3	10	4	6	1	1	0	1	2	5	0	11	15	14	4	4	8	11	17	13	0	0	8	0	10	1	1	1
ABA	14	10	8	11	9	8	5	5	4	7	6	7	6	10	15	13	9	8	12	13	13	10	7	9	0	4	12	8	4	5
DSA	18	18	9	18	16	16	17	16	13	17	8	17	4	18	18	16	11	15	17	17	18	18	10	17	14	0	18	13	9	6
WDO	9	6	1	8	3	5	0	1	0	4	1	4	4	6	15	11	5	3	7	11	12	10	4	8	6	0	0	6	1	2
FA	16	16	5	17	11	16	10	12	7	9	4	12	2	16	17	15	10	13	16	12	17	17	7	17	10	5	12	0	6	1
ICA	17	16	10	17	16	13	12	15	14	14	9	18	7	17	17	15	12	14	16	17	17	17	9	17	14	9	17	11	0	9
MVPA	16	16	11	17	15	16	15	16	14	16	8	17	5	16	17	15	13	15	16	15	17	17	13	17	13	12	16	16	9	0
Nb>=R critique=15	17	13	0	13	3	7	2	3	0	2	0	5	0	13	21	12	1	4	13	11	20	14	0	12	0	0	10	2	0	0

Tableau V.14.b. Nombre des perdants (signes négatifs) par algorithme de contrôle

	PSO	GA	DE	EM	LCA	CBO	BSA	GEM	ABC	BBO	TLBO	ACO	GA-MPC	BHBO	HS	SA	JAYA	MBA	FWA	GWO	GSA	CS	WCA	CSO	ABA	DSA	WDO	FA	ICA	MVPA
PSO	0	9	17	15	18	16	18	17	18	18	17	16	18	11	7	9	16	14	13	11	7	10	18	12	14	18	9	16	17	16
GA	7	0	15	9	16	13	18	17	18	18	17	13	18	9	2	5	13	14	8	6	3	8	18	10	10	18	6	16	16	16
DE	1	3	0	3	4	4	6	6	6	4	9	4	11	3	2	4	5	3	2	4	2	3	4	3	8	9	1	5	10	11
EM	3	9	15	0	14	13	17	16	17	17	17	14	18	9	4	4	14	15	9	6	2	7	18	10	11	18	8	17	17	17
LCA	0	2	13	4	0	8	13	14	13	10	13	8	13	3	1	2	11	11	5	3	2	2	14	4	9	16	3	11	16	15
CBO	2	5	14	5	9	0	13	12	12	13	16	7	18	5	4	5	16	8	4	5	4	6	17	6	8	16	5	16	13	16
BSA	0	0	12	1	4	2	0	5	11	8	12	5	14	1	0	2	10	5	1	1	0	0	13	1	5	17	0	10	12	15
GEM	1	1	12	2	4	6	12	0	10	9	11	5	14	1	0	3	9	6	2	1	1	1	14	1	5	16	1	12	15	16
ABC	0	0	12	1	4	5	5	7	0	6	12	2	14	0	0	2	9	6	1	1	0	0	12	0	4	13	0	7	14	14
BBO	0	0	14	1	8	5	10	8	12	0	14	7	14	0	0	3	9	8	2	3	0	0	14	1	7	17	4	9	14	16
TLBO	1	1	5	1	4	2	6	7	6	3	0	2	9	1	2	3	3	2	1	2	1	1	5	2	6	8	1	4	9	8
ACO	2	5	14	4	10	11	13	13	15	11	16	0	14	6	2	4	13	8	6	4	3	4	14	5	7	17	4	12	18	17
GA-MPC	0	0	4	0	4	0	4	4	4	4	4	4	0	0	0	2	0	4	0	3	0	0	2	0	6	4	4	2	7	5
BHBO	7	9	15	9	15	13	17	17	18	18	16	12	18	0	3	4	12	14	9	6	1	6	18	11	10	18	6	16	17	16
HS	11	16	16	14	17	14	18	18	18	18	16	16	18	15	0	10	15	15	10	12	9	14	18	15	15	18	15	17	17	17
SA	9	13	14	14	16	13	16	15	16	15	15	14	16	14	8	0	13	13	10	10	8	14	16	14	13	16	11	15	15	15
JAYA	2	5	11	4	6	2	8	9	9	9	13	5	16	6	3	5	0	5	3	5	2	4	9	4	9	11	5	10	12	13
MBA	4	4	15	3	7	10	12	12	11	10	16	10	14	4	3	5	13	0	5	4	3	3	13	4	8	15	3	13	14	15
FWA	5	9	16	9	13	14	17	16	17	16	17	12	18	9	8	8	15	13	0	6	7	7	18	8	12	17	7	16	16	16
GWO	7	12	14	12	15	13	17	17	17	15	16	14	15	12	6	7	13	14	12	0	8	11	15	11	13	17	11	12	17	15
GSA	11	15	16	15	16	14	18	17	18	18	17	15	18	17	9	10	16	15	11	10	0	14	18	17	13	18	12	17	17	17
CS	8	10	15	11	16	12	18	17	18	18	17	14	18	12	4	4	14	15	11	7	4	0	18	13	10	18	10	17	17	17
WCA	0	0	11	0	3	1	5	4	6	4	11	4	14	0	0	2	7	5	0	2	0	0	0	0	7	10	4	7	9	13
CSO	6	8	15	8	14	12	17	16	18	17	16	13	18	7	3	4	14	14	10	7	1	5	18	0	9	17	8	17	17	17
ABA	4	8	10	7	9	10	13	13	14	11	12	11	12	8	3	4	9	10	6	4	5	8	11	8	0	14	6	10	14	13
DSA	0	0	9	0	2	1	0	1	4	1	10	1	14	0	0	2	7	3	1	1	0	0	8	0	4	0	0	5	9	12
WDO	9	12	17	10	15	13	18	17	18	14	17	14	14	12	3	7	13	15	11	7	6	8	14	10	12	18	0	12	17	16
FA	2	2	13	1	6	2	8	6	11	8	14	6	15	2	1	3	8	5	2	6	1	1	11	1	8	13	6	0	11	16
ICA	1	2	7	1	2	5	6	3	4	4	9	0	11	1	1	3	6	4	2	1	1	1	9	1	4	9	1	6	0	9
MVPA	2	2	5	1	2	2	3	2	4	2	6	1	10	2	1	3	1	3	2	3	1	1	2	1	5	6	2	1	9	0
<b>Nb&gt;=R critique =15</b>	0	2	11	2	9	1	12	12	13	11	15	3	14	2	0	0	5	5	0	0	0	0	12	2	1	19	1	11	14	20

Le tableau V.14.a montre le nombre de signes positifs des différences entre l'algorithme de contrôle (positionné horizontalement) et les autres algorithmes (positionnés verticalement), c-à-d, les cas où l'algorithme de contrôle a fourni un résultat mauvais par rapport aux autres algorithmes. Si le nombre de cas est supérieur à 15 (nombre critique), on conclut que l'algorithme de contrôle est significativement mauvais que l'autre.

Selon ce critère, on remarque que l'algorithme GSA et HS sont significativement mauvais par rapport à plus de 20 algorithmes.

Le tableau V.14.b montre le nombre de signes négatifs de différence entre l'algorithme de contrôle et les autres, cas où l'algorithme de contrôle a fourni un résultat meilleur que ceux des autres. Selon ce tableau, on conclut que l'algorithme MVPA est significativement meilleur par rapport à 20 algorithmes avec un degré de signification  $\alpha=0.5$  alors qu'il n'a pas une différence significative par rapport à neuf algorithmes en l'occurrence le DE, ABC, TLBO, GA-MPC, JAYA, WCA, ABA, DSA et l'ICA. Il est alors classé premier. On trouve ainsi, le deuxième l'algorithme DSA qui est significativement meilleur par rapport à 19 algorithmes, le troisième l'algorithme TLBO avec 15 algorithmes, et quatrième le GA-MPC et l'ICA avec 14 algorithmes.

### V.2.2. Application de test de Friedman

L'application de test de Friedman permettant le classement des algorithmes du meilleur au plus mauvais sur la base de la moyenne des rangs a conduit aux résultats du Tableau V.15.

**Tableau V.15 moyenne des rangs des algorithmes**

ALGO	GA_MPC	MVPA	DSA	ICA	TLBO	WCA	DE	ABC	BSA	FA	GEM	BBO	JAYA	LCA
Somme des rangs	98	112	116	133	134	154	166	173	190	210	212	220	230	262
Moyenne des rangs	5	6	6	7	7	9	9	10	11	12	12	12	13	15

MBA	ABA	ACO	CBO	CSO	BHBO	EM	GA	FWA	WDO	GWO	CS	SA	PSO	GSA	HS
272	287	291	301	366	370	375	377	381	387	398	401	410	434	458	460
15	16	16	17	20	21	21	21	21	22	22	22	23	24	25	26

Les résultats du tableau V.16 ont donné une statistique de Friedman  $F_d^2=276.64$  qui est supérieure à  $R_{critique}=42$  ce qui signifie que l'hypothèse nulle (pas de différence significative entre les algorithmes) est rejetée et donc il y a au moins un algorithme ayant une différence significative avec les autres. Aussi, on remarque que les algorithmes GA-MPC, MVPA et

DSA ont les meilleurs rangs par rapport aux autres algorithmes alors que les algorithmes PSO, GSA et HS ont les plus mauvais.

L'évaluation de la probabilité  $p$  (calculé pour chaque paire d'algorithme) qui représente la signification de la différence de chaque algorithme par rapport aux autres a donné les résultats du tableau V.16.

**Tableau V.16. Valeur de probabilité  $P$  par paire d'algorithme**

	PSO	GA	DE	EM	LCA	CBO	BSA	GEM	ABC	BBO	TLBO	ACO	GA-MPC	BHBO	HS
PSO	1,000	1,000	0,010	0,071	0,005	0,039	0,002	0,005	0,002	0,002	0,010	0,019	0,002	0,796	0,439
GA	1,000	1,000	0,039	0,796	0,020	0,197	0,002	0,005	0,002	0,002	0,010	0,119	0,002	1,000	0,039
DE	0,010	0,039	1,000	0,039	0,243	0,071	0,439	0,439	0,606	0,197	0,597	0,119	0,237	0,039	0,020
EM	0,071	0,796	0,039	1,000	0,071	0,197	0,005	0,010	0,005	0,005	0,010	0,069	0,002	0,796	0,121
LCA	0,005	0,020	0,243	0,071	1,000	0,516	0,243	0,197	0,153	0,796	0,243	0,604	0,153	0,039	0,010
CBO	0,039	0,197	0,071	0,197	0,516	1,000	0,026	0,197	0,091	0,071	0,020	1,000	0,002	0,197	0,121
BSA	0,002	0,002	0,439	0,005	0,243	0,026	1,000	0,296	0,296	0,796	0,439	0,299	0,121	0,005	0,002
GEM	0,005	0,005	0,439	0,010	0,197	0,197	0,296	1,000	0,516	0,897	0,606	0,299	0,121	0,005	0,002
ABC	0,002	0,002	0,606	0,005	0,153	0,091	0,296	0,516	1,000	0,439	0,606	0,035	0,197	0,002	0,002
BBO	0,002	0,002	0,197	0,005	0,796	0,071	0,796	0,897	0,439	1,000	0,153	0,436	0,121	0,002	0,002
TLBO	0,010	0,010	0,597	0,010	0,243	0,020	0,439	0,606	0,606	0,153	1,000	0,038	0,352	0,014	0,020
ACO	0,019	0,119	0,119	0,069	0,604	1,000	0,299	0,299	0,035	0,436	0,038	1,000	0,069	0,194	0,019
GA-MPC	0,002	0,002	0,237	0,002	0,153	0,002	0,121	0,121	0,197	0,121	0,352	0,069	1,000	0,002	0,002
BHBO	0,796	1,000	0,039	0,796	0,039	0,197	0,005	0,005	0,002	0,002	0,014	0,194	0,002	1,000	0,071
HS	0,439	0,039	0,020	0,121	0,010	0,121	0,002	0,002	0,002	0,002	0,020	0,019	0,002	0,071	1,000
SA	0,796	0,439	0,197	0,302	0,071	0,439	0,039	0,071	0,039	0,071	0,121	0,194	0,039	0,302	0,606
JAYA	0,039	0,197	0,296	0,121	0,697	0,039	1,000	0,796	0,606	0,796	0,117	0,299	0,009	0,302	0,071
MBA	0,071	0,071	0,071	0,039	0,796	0,796	0,153	0,197	0,243	0,439	0,039	0,604	0,071	0,071	0,039
FWA	0,197	0,897	0,020	0,796	0,121	0,121	0,005	0,010	0,005	0,010	0,010	0,194	0,002	0,796	0,606
GWO	0,606	0,439	0,121	0,439	0,071	0,302	0,010	0,010	0,010	0,039	0,039	0,119	0,039	0,439	0,302
GSA	0,439	0,071	0,020	0,051	0,020	0,121	0,002	0,005	0,002	0,002	0,010	0,038	0,002	0,020	1,000
CS	1,000	0,796	0,039	0,439	0,020	0,302	0,002	0,005	0,002	0,002	0,010	0,069	0,002	0,439	0,121
WCA	0,002	0,002	0,511	0,002	0,091	0,005	0,197	0,121	0,439	0,121	0,602	0,069	0,117	0,002	0,002
CSO	0,606	0,796	0,039	1,000	0,071	0,302	0,005	0,006	0,002	0,005	0,020	0,119	0,002	0,606	0,071
ABA	0,121	0,796	0,606	0,606	0,796	0,796	0,121	0,121	0,071	0,302	0,302	0,436	0,197	0,796	0,039
DSA	0,002	0,002	0,897	0,002	0,039	0,006	0,014	0,027	0,243	0,02	0,897	0,009	0,153	0,002	0,002
WDO	1	0,439	0,02	0,796	0,071	0,302	0,005	0,01	0,005	0,071	0,02	0,119	0,071	0,439	0,039
FA	0,01	0,01	0,439	0,005	0,363	0,01	0,796	0,439	0,606	0,897	0,302	0,299	0,091	0,01	0,005
ICA	0,005	0,01	0,363	0,005	0,02	0,071	0,302	0,071	0,121	0,121	0,606	0,004	0,796	0,005	0,005
MVPA	0,01	0,01	0,296	0,005	0,051	0,01	0,071	0,039	0,197	0,039	0,597	0,009	0,511	0,01	0,005
<b>Nombre des algorithmes (<math>P &lt; 0.01</math>)</b>	<b>13</b>	<b>11</b>	<b>1</b>	<b>11</b>	<b>2</b>	<b>5</b>	<b>11</b>	<b>11</b>	<b>11</b>	<b>9</b>	<b>6</b>	<b>3</b>	<b>11</b>	<b>10</b>	<b>11</b>

Tableau V.16 (la suite). Valeur de probabilité  $P$  par paire d'algorithme

	SA	JAYA	MBA	FWA	GWO	GSA	CS	WCA	CSO	ABA	DSA	WDO	FA	ICA	MVPA
PSO	0,796	0,039	0,071	0,197	0,606	0,439	1,000	0,002	0,606	0,121	0,002	1,000	0,010	0,005	0,010
GA	0,439	0,197	0,071	0,897	0,439	0,071	0,796	0,002	0,796	0,796	0,002	0,439	0,010	0,010	0,010
DE	0,197	0,296	0,071	0,020	0,121	0,020	0,039	0,511	0,039	0,606	0,897	0,020	0,439	0,363	0,296
EM	0,302	0,121	0,039	0,796	0,439	0,051	0,439	0,002	1,000	0,606	0,002	0,796	0,005	0,005	0,005
LCA	0,071	0,697	0,796	0,121	0,071	0,020	0,020	0,091	0,071	0,796	0,039	0,071	0,363	0,020	0,051
CBO	0,439	0,039	0,796	0,121	0,302	0,121	0,302	0,005	0,302	0,796	0,006	0,302	0,010	0,071	0,010
BSA	0,039	1,000	0,153	0,005	0,010	0,002	0,002	0,197	0,005	0,121	0,014	0,005	0,796	0,302	0,071
GEM	0,071	0,796	0,197	0,010	0,010	0,005	0,005	0,121	0,006	0,121	0,027	0,010	0,439	0,071	0,039
ABC	0,039	0,606	0,243	0,005	0,010	0,002	0,002	0,439	0,002	0,071	0,243	0,005	0,606	0,121	0,197
BBO	0,071	0,796	0,439	0,010	0,039	0,002	0,002	0,121	0,005	0,302	0,020	0,071	0,897	0,121	0,039
TLBO	0,121	0,117	0,039	0,010	0,039	0,010	0,010	0,602	0,020	0,302	0,897	0,020	0,302	0,606	0,597
ACO	0,194	0,299	0,604	0,194	0,119	0,038	0,069	0,069	0,119	0,436	0,009	0,119	0,299	0,004	0,009
GA-MPC	0,039	0,009	0,071	0,002	0,039	0,002	0,002	0,117	0,002	0,197	0,153	0,071	0,091	0,796	0,511
BHBO	0,302	0,302	0,071	0,796	0,439	0,020	0,439	0,002	0,606	0,796	0,002	0,439	0,010	0,005	0,010
HS	0,606	0,071	0,039	0,606	0,302	1,000	0,121	0,002	0,071	0,039	0,002	0,039	0,005	0,005	0,005
SA	1,000	0,439	0,302	1,000	0,697	0,606	0,302	0,039	0,302	0,243	0,039	0,606	0,071	0,071	0,071
JAYA	0,439	1,000	0,302	0,071	0,302	0,039	0,121	0,433	0,121	1,000	0,363	0,302	0,439	0,121	0,035
MBA	0,302	0,302	1,000	0,121	0,121	0,039	0,039	0,121	0,071	0,606	0,039	0,071	0,121	0,071	0,039
FWA	1,000	0,071	0,121	1,000	0,439	0,439	0,439	0,002	0,606	0,439	0,005	0,606	0,010	0,010	0,010
GWO	0,697	0,302	0,121	0,439	1,000	0,606	0,606	0,027	0,606	0,153	0,010	0,439	0,197	0,010	0,039
GSA	0,606	0,039	0,039	0,439	0,606	1,000	0,121	0,002	0,020	0,121	0,002	0,197	0,005	0,005	0,005
CS	0,302	0,121	0,039	0,439	0,606	0,121	1,000	0,002	0,363	0,796	0,002	0,796	0,005	0,005	0,005
WCA	0,039	0,433	0,121	0,002	0,027	0,002	0,002	1,000	0,002	0,302	0,897	0,071	0,606	0,796	0,149
CSO	0,302	0,121	0,071	0,606	0,606	0,020	0,363	0,002	1,000	0,897	0,003	1,000	0,005	0,005	0,005
ABA	0,243	1,000	0,606	0,439	0,153	0,121	0,796	0,302	0,897	1,000	0,071	0,606	0,439	0,071	0,121
DSA	0,039	0,363	0,039	0,005	0,01	0,002	0,002	0,897	0,003	0,071	1	0,005	0,302	0,796	0,363
WDO	0,606	0,302	0,071	0,606	0,439	0,197	0,796	0,071	1	0,606	0,005	1	0,197	0,01	0,02
FA	0,071	0,439	0,121	0,01	0,197	0,005	0,005	0,606	0,005	0,439	0,302	0,197	1	0,363	0,051
ICA	0,071	0,121	0,071	0,01	0,01	0,005	0,005	0,796	0,005	0,071	0,796	0,01	0,363	1	0,796
MVPA	0,071	0,035	0,039	0,01	0,039	0,005	0,005	0,149	0,005	0,121	0,363	0,02	0,051	0,796	1
<b>Nombre des algorithmes (P&lt;0.01)</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>11</b>	<b>5</b>	<b>11</b>	<b>11</b>	<b>10</b>	<b>10</b>	<b>0</b>	<b>13</b>	<b>5</b>	<b>10</b>	<b>12</b>	<b>11</b>

Selon ce tableau, on voit que l'algorithme PSO est significativement différent par rapport à 13 algorithmes alors que le CS et le GSA sont significativement différents par rapport à 11 algorithmes (marqués en rouge).

Tenant compte des valeurs de rangs de l'algorithme PSO (plus grande valeur des sommes des rangs selon le Tableau V.15), on conclut qu'il est le plus mauvais par rapport à 13 algorithmes du point de vue signification.

Selon ce tableau, on constate que l'algorithme GA-MPC est meilleur du point de vue signification par rapport à 11 algorithmes suivi par les algorithmes MVPA et DSA qui sont respectivement meilleurs par rapport à 13 et 11 algorithmes.

### V.2.3. Application de la méthode de profil de performance

Une autre méthode permettant de distinguer entre les algorithmes du point de vue performance est celle dite de profil de performance. De telle méthode est basée sur le calcul de trois facteurs  $m_{(p,s)}$ ,  $r_{(p,s)}$ , et  $\rho_s(\tau)$ .

Le  $m_{(p,s)}$  est défini comme le rapport de deux différences. La première différence entre la moyenne des valeurs optimales de 30 exécutions de la fonction objectif du problème  $p$  obtenue par l'algorithme  $s$  et la meilleure valeur obtenue pour ce problème avec tous les algorithmes. La deuxième différence entre les valeurs la plus mauvaise et la meilleure. Plus  $m_{(p,s)}$  est grand plus l'algorithme est mauvais. Un algorithme soit le meilleur pour ce problème si  $m_{(p,s)} \approx 0$ .

$r_{(p,s)}$  est le rapport entre la valeur  $m_{p,s}$  de l'algorithme  $s$  et la plus petite valeur de  $m_{(p,s)}$  pour le problème  $p$ . Si  $r_{(p,s)} = 1$ , l'algorithme  $s$  est le meilleur pour le problème  $p$ . Si  $r=2$ , l'algorithme  $s$  a deux fois le résultat de meilleur algorithme.

Le facteur  $\rho_s(\tau)$  est le produit de l'inverse du nombre total des problèmes et le nombre des cas où les valeurs de  $r_{(p,s)}$  sont égale ou inférieure à  $\tau$  (valeur prédéfinie). Si  $\tau = 1$  et  $\rho_s(\tau) = 1$ , on conclut que l'algorithme  $s$  a obtenu la meilleur valeur pour tous les problèmes. Si  $\rho_s(\tau) = 0.5$ , on conclut que cet algorithme a obtenu le meilleur résultat pour la moitié des problèmes.

La figure V.7 montre les variations des valeurs de profil de performance avec la valeur de  $\tau$  et cela pour les 30 algorithmes testés pour tous les réseaux traités et cas considérés (19 cas).

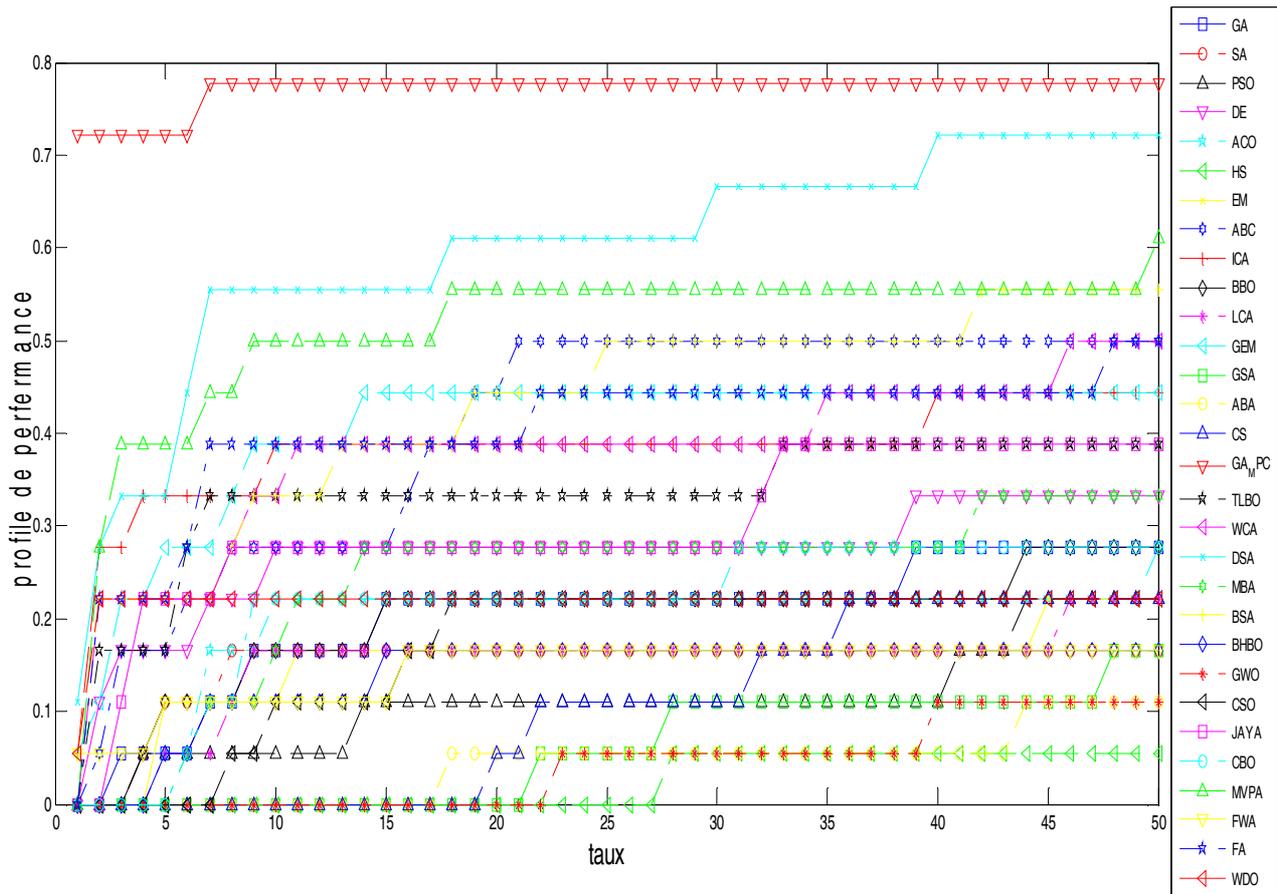


Fig. V.7. Profil de performance pour tous les cas

Le trace du profil de performance compare comment les algorithmes peuvent bien estimer l'optimum global lié de l'un à l'autre.

On remarque que pour  $\tau = 1$ , l'algorithme GA-MPC a trouvé les meilleures solutions (première place) dans plus de 73% des cas alors que pour  $\tau = 10$ , il a trouvé des solutions égales ou inférieures dix fois la meilleure solution trouvée dans environ 78% des 19 problèmes étudiés.

L'algorithme DSA n'a pas trouvé de meilleure solution dans aucun cas. Par contre, il a trouvé des valeurs inférieures ou égales le double du meilleur résultat obtenu pour 22% des problèmes étudiés et des valeurs inférieures ou égales à 25 fois le meilleur résultat pour 38% des cas.

En troisième position, on trouve l'algorithme MVPA avec 27% des cas où ses résultats sont inférieurs ou égales à deux fois le meilleur résultat obtenu. Cet algorithme n'a pas obtenu de meilleur résultat pour aucun cas.

## **Conclusion**

Dans ce chapitre, nous avons analysé les résultats de simulation obtenus par les 30 algorithmes pour les 19 cas traités (09 cas pour le réseau IEEE30, 03 cas pour IEEE 57, 03 cas pour IEEE118, et 04 cas pour le réseau algérien).

Les figures illustrant l'évolution de fonction objectif pour les trois premiers problèmes qui sont le coût, la déviation standard et l'index de stabilité pour le réseau IEEE30 ont été présentées et examinées.

La valeur de la fonction objectif pour les autres cas pour les différents réseaux étudiés ont été organisées sous forme de tableaux et classées de meilleur au pire, et des constatations ont été conclues de ces résultats.

Deux méthodes de comparaison ont été appliquées « paramétrique et non paramétrique », d'après les méthodes de comparaison paramétrique nous avons conclu que l'algorithme ABC est le meilleur par rapport aux autres ensuite c'est les algorithmes MVPA, ICA, GA-MPC et DSA. Cette étude est effectuée en se basant sur les valeurs finales de fonction objectif classées de meilleur au plus mauvais sans la considération de aucun critère non paramétrique dans l'étude.

L'application des méthodes non paramétriques est basée sur la valeur finale des résultats avec l'introduction des critères non paramétriques et statistiques qui définissent le niveau de signification de la différence entre les algorithmes.

Trois tests non paramétriques ont été appliqués dans ce travail qui sont le test de signes multiples, le test de Friedman et le profil de performance, d'après les résultats de simulation ont conclu que les algorithmes MVPA, GA-MPC, DSA sont les plus adéquats pour les problèmes d'optimisation de l'OPF.

# **Conclusion Générale**

## Conclusion Générale

Dans notre travail de thèse, nous avons traité un problème lié à l'écoulement de puissance optimal (OPF) dont l'objectif est l'identification de ou des algorithmes les plus adéquats pour résoudre les problèmes de l'OPF en tenant en compte des contraintes d'égalité et d'inégalité. Pour cela, une étude comparative des performances de trente algorithmes mis en œuvre pour différents cas de réseaux a été réalisée. Pour permettre la comparaison de ces algorithmes, ils sont testés pour les mêmes conditions d'exécution notamment le nombre d'itération fixée à 500 itérations.

Lors de la résolution du problème de l'OPF, les variables des fonctions objectif ont été calculés par résolution du problème de l'écoulement de puissance et cela par application de la méthode de Newton-Raphson. Dans ce travail, neuf cas de fonctions objectif ont été traités pour quatre types de réseaux électriques, IEEE30, IEEE57, IEEE 118 et le réseau Algérien réel à 59 jeux de barres. A cause d'un manque de données pour certains réseaux, sauf pour le réseau IEEE30, les trente algorithmes ont été testés pour les neuf cas d'objectif.

Deux types de comparaison ont été effectuées. Une dite paramétrique basée sur l'optimum global, la moyenne et la déviation standard et l'autre dite non paramétriques basée sur le test de signe multiple, test de Friedman et le, profil de performance.

Pour le réseau IEEE30, et pour la minimisation de l'objectif coût de carburant, les trente (30) exécutions de chaque algorithme ont montré la robustesse de tous les algorithmes pour résoudre ce type de problème sauf HS et le SA. Les algorithmes PSO, GA, DE, CBO, TLBO, GA-MPC, GSA, GWO, GSA, WCA, ICA, MVPA, WDO, FA ont convergé dans les 20 premières itérations, tandis que les algorithmes CSO, ABA, SA ont atteint leurs minimums après 50 itérations. L'algorithme ICA a fourni le meilleur minimum 798.98\$/h.

Toujours pour le même réseau IEEE30, la minimisation multi objectif déviation du profil de tension et coût de carburant a montré aussi la robustesse de tous les algorithmes (tous les algorithmes ont convergé). Les algorithmes GA-MPC et TLBO ont données le meilleur optimum pour la déviation standard 0.0986PU. Les algorithmes MVPA, DE, DSA et JAYA ont fourni la deuxième meilleure valeur plus proche de l'optimum global.

Pour l'amélioration de la stabilité de tension des jeux de barres charge, tous les algorithmes ont convergé. Sauf le GA-MPC, ABC, BBO et le PSO ont convergé dans les 100 premières itérations. Les optimums de l'index de sécurité de tous les algorithmes sont très proches du meilleur optimum 0.1249 fourni par l'algorithme ABA. Le FA a fourni le mauvais résultat 0.1489.

Pour l'objectif émissions de gaz polluant, par rapport aux autres, seuls les algorithmes GA-MPC, ABC, ABA, GSA, BBO, PSO et MVPA ont convergé dans les 100 premières itérations. L'algorithme ICA a fourni le meilleur optimum 0,2184 ton/h alors que l'ACO, DE, TLBO, GA-MPC, JAYA, WCA, MVPA, LCA, CBO, GEM, BBO, DSA et le FA ont donné un optimum plus proche 0,222 ton/h.

Pour la minimisation du coût de production avec considération de plusieurs types de combustibles, tous les algorithmes ont convergé. Le GA-MPC, MVPA, GEM, BHBO, PSO, GSA et le ABA ont convergé dans les premières 50 itérations alors que les autres ont nécessité plus de 200 itérations. Le GA-MPC a fourni le meilleur optimum 644,99\$/h suivi par les TLBO et le MVPA avec un optimum 645,00\$/h. Les résultats obtenus ont montré que la multiplication de types de combustible peut considérablement réduire le coût de production.

Les résultats liés à la minimisation du coût de carburant avec la considération de l'effet d'ouverture des vannes ont montré que le GA-MPC, TLBO et le MVPA ont convergé dans les premier 50 itérations et fourni le meilleur optimum 830,43\$/h.

Pour la minimisation du coût avec la considération de l'option de plusieurs combustibles et l'effet d'ouverture des vannes simultanément, le GA-MPC a fourni le meilleur optimum 681.0 \$/h suivi par le DE et le TLBO avec un optimum 681,1 \$/h.

Pour la minimisation du coût de carburant avec introduction des dispositifs FACTS (TCSC au niveau de la ligne N°36 avec un degré de compensation égale à 0.1 et un SVC au niveau du jeu de barre N°22 avec un facteur de compensation égale à 0,05), le SA a donné le meilleur optimum 798,16\$/h. Comparativement à l'optimum fourni pour le cas sans FACTS 812,76\$/h, on constate que l'introduction des FACTS ont considérablement diminué le coût. Aussi, les FACTS ont amélioré la stabilité des tensions ( $L_{max} = 0,1201$  pu avec FACTS et  $L_{max} = 0,1622$  pu sans FACTS) et minimisé les pertes réactives (-8,0727 MVAR sans FACTS et -4,3528 MVAR avec FACTS).

Pour la minimisation du coût de production avec la considération des contraintes de sécurité en mode préventif en cas de déclenchement de la ligne 20 de réseau, le SA a fourni le meilleur optimum 798.54\$/h suivi par les algorithmes JAYA, TLBO, GA-MPC, MVPA, WCA et le DE avec un optimum 799,03\$/h.

Pour le réseau IEEE57, tous les algorithmes ont convergé sauf le CS, SA et le GSA. Le GA-MPC a fourni le meilleur optimum pour la minimisation simultanée des objectifs coûts de carburant et amélioration de la stabilité de tension des jeux de barres charge. Pour la minimisation de la déviation standard, le meilleur optimum est obtenu par le DSA alors que l'algorithme GA-MPC est classé deuxième.

Pour le réseau IEEE 118, les objectifs minimisés sont le coût de carburant, l'amélioration de la dévidassions standard et la stabilité de tension. Parmi les 30 algorithmes testés, seuls le BSA, GEM, ABC, BBO, GA\_MPC, WCA, DSA, FA, ICA et le MVPA ont convergé. Le GA-MPC a donné les meilleurs optimums, un cout de 135 760,00\$/h, une déviation standard de 0,6955 PU, et un indice de stabilité de 0,0682 PU.

Pour le réseau Algérien à 59 jeux de barres, et pour les cinq objectifs minimisés, seuls l'ICA, MVPA DSA, ABC, LCA, BSA, DE, GEM, TLBO, ACO, MBA, WDO, ABA et le BBO ont convergé.

Pour le coût de carburant, l'ICA a fourni le meilleur optimum 1688,4\$/h suivi par le MVPA et le DSA. Pour l'amélioration de la dévidassions standard, l'ICA a fourni le meilleur optimum 1,1886 PU suivi par le MVPA et le GEM. Pour la stabilité de tension, le MBA a donné le meilleur optimum 0.21 PU. Pour l'objectif émissions de gaz, le meilleur optimum 0.46 ton/h est obtenu par le TLBO suivi par l'ICA et le DE.

La comparaison paramétrique a ainsi montré que l'algorithme ABC est le plus adéquat pour la résolution de tous les problèmes pour les 04 réseaux étudiés suivi par le MVPA, ICA, GA-MPC et le DSA.

Selon la comparaison non paramétrique basée sur les tests de signes multiples, de Friedman et le profil de performance et qui tiennent en compte la valeur finale de l'objectif et la vitesse de convergence de l'algorithme et l'introduction des critères non paramétriques et statistique qui définit le niveau de signification de la différence entre les algorithmes, on a trouvé que les algorithmes MVPA, GA-MPC et le DSA sont les plus adéquats pour les problèmes d'optimisation de l'OPF.

Tenant compte des résultats obtenus par les deux types de comparaison, on voit que les algorithmes MVPA, GA-MPC et le DSA sont les plus performants pour la résolution des problèmes de l'OPF.

Comme perspective à ce travail, on envisage compléter l'étude par la résolution de tous les objectifs pour tous les réseaux traité et réalisé un test complet des 30 algorithmes adoptés. De même, on envisage développe d'autres algorithmes sur la base de couplage des algorithmes parmi ceux adoptés et testé dans ce travail pour améliorer les performances que donne chacun d'eux tout seul.

# **References Bibliographiques**

## **References bibliographiques**

- [1] T. Niknam , M.R. Narimani "A modified shuffle frog leaping algorithm for multi-objective optimal power flow", *Energy Syst* ,pp. 6420–6432 (2011).
- [2] M. AlRashidi, M.El-Hawary "Applications of computational intelligence techniques for solving the revived optimal power flow problem", *Electr. Power Syst. Res.* 79 (4) ,pp. 694-702, (2009).
- [3] Frank, S., Steponavice, I. Rebennack, S. Optimal power flow: a bibliographic survey I. *Energy Syst* 3, pp.221–258 (2012).
- [4] H. Dommel , W. Tinney "Optimal power flow solutions", *IEEE Trans. Power Appar. Syst.* 87(10), pp. 1866–1876 (1968).
- [5] O. Alsac , B. Stott "Optimal load flow with steady-state security", *IEEE Trans. Power Appar. Syst.* BPAS-93(3), 7, pp. 45–751 (1974).
- [6] J. A. Momoh, "A generalized quadratic-based model for optimal power flow," *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*, vol.1,pp. 261-271 (1989).
- [7] G. F. Reid , L. Hasdorff "Economic dispatch using quadratic programming", *IEEE Transactions on Apparatus and Power Systems*, Vol. 92, Issue: 6, Nov, pp.2015- 2023.(1973)
- [8] N. Grudinin "Reactive power optimization using successive quadratic programming method", *IEEE Trans. Power Syst*, vol. 13, no. 4, page. 1219-1225, Nov. 1998.
- [9] S. Granville "Optimal reactive dispatch through interior point methods", *IEEE Trans. Power Syst.*, vol. 9, no. 1, pp. 136-146, (1994).
- [10] J. A. Momoh and J. Z. Zhu, "Improved interior point method for OPF problems," in *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1114-1120, Aug. 1999.
- [11] Castronuovo , D. Edgardo "New versions of interior point methods applied to the optimal power flow problem" , *IEEE Transactions on Power Systems* 16,pp.1-6 (2001).
- [12] GL. Torres "An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates" , *Power Systems, IEEE Transaction* , 13 , pp 1211 – 1218 (1998 )
- [13] J.H. Holland "Genetic algorithms and the optimal allocation of trials", *SIAM J. Comput.* 2 pp.88–105 (1973).
- [14] A. Bakritzs, V. Perirtridis "Genetic Algorithm Solution to the Economic Dispatch Problem" , *IEE Proc,-Generation Transmission Distribution*, vol. 141, no. 4, pp. 377-382, (1994)
- [15] M.Younes, M. Rahli "Optimal Power based on Hybrid Genetic Algorithm", *Journal of*

Information Science and Engineering, vol. 23, pp.1801-1816, Jan 2007.

- [16] M. Sailaja Kumari, S. Maheswarapu " Enhanced Genetic Algorithm based computation technique for multi-objective Optimal Power Flow solution", *Electrical Power and Energy Systems* 32, pp.736–742, (2010)
- [17] J. Kennedy, R. Eberhart " Particle swarm optimization" , *Proceedings of IEEE International Conference on Neural Networks IV*, pp. 1942–1948 (1995).
- [18] Z.L. Gaing , X.H Liu "New constriction particle swarm optimization for Security-Constrained optimal power flow solution", *International Conference on Intelligent Systems Applications to Power Systems (ISAP 2007)*, pp. 1–6 (2007).
- [19] P. Yumbla, J. Ramirez "Optimal power flow subject to security constraints solved with a particle swarm optimizer", *IEEE Trans. Power Syst.* 23(1), pp.33–40 (2008).
- [20] Cui-Ru Wang, He-Jin Yuan, Zhi-Qiang Huang, Jiang-Wei Zhang and Chen-Jun Sun, "A modified particle swarm optimization algorithm and its application in optimal power flow problem," *2005 International Conference on Machine Learning and Cybernetics*, Vol. 5,2005, pp. 2885-2889 (2005)
- [21] K. Swapur " Swarm intelligence approach to the solution of optimal power flow", *Indian Inst. Sci.* 86, pp.439–455 (2006).
- [22] J.Y Kim , M.M Jeong " PC cluster based parallel PSO algorithm for optimal power flow", *International Conference on Intelligent Systems Applications to Power Systems (ISAP 2007)*, pp. 1–6 (2007).
- [23] H. Hoseinabadi , H. Hosseini "Optimal power flow solution by a modified particle swarm optimization algorithm", *43rd International Universities Power Engineering Conference (UPEC 2008)*, pp. 1–4 (2008).
- [24] W. Zhang , Y. Liu "Reactive power optimization based on PSO in a practical power system" , *IEEE Power Engineering Society General Meeting*, vol. 1, pp. 239–243 (2004).
- [25] A. Coath , G. Al-Dabbagh "Particle swarm optimisation for reactive power and voltage control with grid-integrated wind farms", *IEEE Power Engineering Society General Meeting*, vol. 1, pp. 303–308 (2004).
- [26] B. Zhao , C. Guo, " Improved particle swam optimization , algorithm for OPF problems", *IEEE PES Power Systems Conference and Exposition*, vol. 1, pp. 233–238 (2004).
- [27] V. Iachogiannis , J. Lee " A comparative study on particle swarm optimization for optimal

- steady state performance of power systems", IEEE Trans. Power Syst.21(4), pp.1718–1728 (2006).
- [28] Li. D , L. Gao "Adaptive particle swarm optimization algorithm for power system reactive power optimization", American Control Conference (ACC '07), pp. 4733–4737 (2007).
- [29] K. Pouya , Lesani " An angle-based PSO approach for reactive power management problem", Power Systems Conference and Exposition (PSCE '09), pp. 1–6. IEEE/PES, Piscataway (2009).
- [30] S. Kirkpatrick, C. Gelatt " Optimization by simulated annealing", Science220, Volume220,Number4598, pp.671–680 (1983).
- [31] Y. Hsiao, T. Liu, C. Chiang " A new approach for optimal VAR sources planning in large scale electric power systems", IEEE Trans. Power Syst.8(3), pp.988–996 (1993).
- [32] K. Wong , C. Fung "Simulated annealing based economic dispatch algorithm", IEE Proceedings, vol. 140, pp. 509–515 (1993).
- [33] Roa-Sepulveda, C. Pavez-Lazo " A solution to the optimal power flow using simulated annealing", International Journal of Electrical Power & Energy Systems Volume 25, Issue 1, Pages 47-57 (2003)
- [34] F. Glover "Tabu search—part II", ORSA J. Comput.1(2), 4–32 (1990).
- [35] J. Bland , G. Dawson " Tabu search and design optimization", Comput. Aided Des.23(3), pp.195–201 (1991) .
- [36] H. Mori , T. Hayashi "New parallel tabu search for voltage and reactive power control in power systems", IEEE International Symposium on Circuit and Systems (ISACAS'98), pp. 431–434 (1998).
- [37] M. Abido "Optimal power flow using tabu search algorithm", Electr. Power Compon. Syst.30, pp.469– 483 (2002).
- [38] T. Kulworawanichpong , S. Sujitjorn "Optimal power flow using tabu search", IEEE Power Eng. Rev. 6, pp.37–40 (2002).
- [39] Altun , Yalcinoz "Implementing soft computing techniques to solve economic dispatch problem in power systems", Expert Syst. Appl.35, pp.1668–1678 (2008).
- [40] N. Muthuselvan , P. Somasundaram "Application of tabu search algorithm to security constrained economic dispatch" , J. Theor. Appl. Inf. Technol.5, pp.602–608 (2009).
- [41] K. Lee , J.V Iachogiannis "Optimization of power systems based on ant colony system

- algorithms: an overview", Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, pp. 22–35 (2005).
- [42] K. Swarup "Ant colony optimization for economic generator scheduling and load dispatch", Proceedings of the 6th WSEAS International Conference on Evolutionary Computing, Portugal, pp. 167–175 (2005).
- [43] M. Kalil , I. Musirin "Ant colony based optimization technique for voltage stability control", Proceedings of the 6th WSEAS International Conference on Power Systems, pp.149-154 (2006).
- [44] B. Allaoua, A. Laoufi "Collective intelligence for optimal power flow solution using ant colony optimization", Leonardo Electron. J. Pract. Technol.13, pp.88–105 (2008).
- [45] B. Gasbaoui , B. Allaoua "Ant colony optimization applied on combinatorial problem for optimal power flow solution", Leonardo J. Sci.14, pp.1–17 (2009).
- [46] M. Tripathy and S. Mishra, "Bacteria Foraging-Based Solution to Optimize Both Real Power Loss and Voltage Stability Limit," in IEEE Transactions on Power Systems, vol. 22, no. 1, pp. 240-248( 2007)
- [47] M. Tripathy , S. Mishra "Bacteria foraging-based solution to optimize both real power loss and voltage stability limit", IEEE Trans. Power Syst.22(1), 240–248 (2007).
- [48] M. Li, M. Tang "Bacterial foraging algorithm with varying population for optimal power flow", Applications of Evolutionary Computing. Lectures Notes In Computer Science, pp. 32–41. Springer, Berlin (2007).
- [49] K. Passino "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Syst. Mag.22 (3),pp. 52–67 (2002).
- [50] W. J. Tang, M. S. Li, S. He, Q. H. Wu and J. R. Saunders, "Optimal Power Flow With Dynamic Loads Using Bacterial Foraging Algorithm," 2006 International Conference on Power System Technology, pp. pp.1-5 (2006)
- [51] W. Tang , M. Li " Bacterial foraging algorithm for optimal power flow in dynamic environments", IEEE Trans. Circuits Syst. I, Regul..55(8), pp.2433–2443 (2008).
- [52] Storn, R., Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997).
- [53] Lampinen , J. Zelinka " Mixed integer-discrete-continuous optimization by differential evolution", 5th International Mendel Conference on Soft Computing, Brno, Czech Republic, pp. 77–81 (1999).
- [54] J. Chiou , F. Wang "A hybrid method of differential evolution with application to optimal control problems of a bioprocess system", Proceeding 1998 IEEE on Evolutionary

Computation Conference, vol. 1, pp. 627–632 (1998).

- [55] K. Price , R. Storn "Differential Evolution: a Practical Approach to Global Optimization", Birkhäuser, book Berlin (2005).
- [56] G. Onwubolu , D. Davendra " Differential Evolution: A Handbook for Global Permutation Based Combinatorial Optimization", Studies in Computational Intelligence. Springer, Berlin (2009).
- [57] L. Coelho , V. Mariani " Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints", *Energy Convers. Manag.*48, pp.1631–1639 (2007).
- [58] K. Vaisakh , LR. Srinivas "Differential evolution based OPF with conventional and non conventional cost characteristics", *International Conference on Power System Technology and IEEE Power India Conference (POWERCON)*, pp. 1–9 (2008).
- [59] S. Sayah , K. Zehar "Modified differential evolution algorithm for optimal power flow with non smooth cost functions", *Energy Convers. Manag.*49, pp.3036–3042 (2008).
- [60] G. Bakare , G. Krost " Differential evolution approach for reactive power optimization of Nigerian grid system", *IEEE Power Engineering Society General Meeting*, pp. 1–6 (2007).
- [61] M. Basu "Optimal power flow with FACTS devices using differential evolution", *Int. J. Electr. Power Energy Syst.*30(2), pp.150–156 (2008).
- [62] C. Sumpavakup , L. Srikun " A solution to the optimal power flow using artificial bee colony algorithm", *international conference on power system technology (POWERCON)*, pp 1–5, 2010 .
- [63] A. Khorsandi , SH. Hosseinian " Modified artificial bee colony algorithm based on fuzzy multi-objective technique for optimal power flow problem", *Electr Power Syst Res* 2013, pp.206–236 (2013)
- [64] A.N Afandi , H. Miyauchi "Improved artificial bee colony algorithm considering harvest season for computing economic dispatch on power system" *IEEJ Transactions on Electrical and Electronic Engineering*, 9(3), pp.251-257 (2014).
- [65] H.R.E Bouchekara, M.Abido , A.E Chaib " Optimal Power Flow Using an Improved Electromagnetism-like Mechanism Method", *Electric Power Components and Systems*, 44:4, pp.434-449 (2016)
- [66] B. Jeddi, AH. Einaddin " A novel multi-objective approach based on improved electromagnetism-like algorithm to solve optimal power flow problem considering the

- detailed model of thermal generators", *Int Trans Electr Energ Syst* Volume 27, Issue 4, pp. 1-20 (2017).
- [67] A. Shabanpour , H. Reza Seifia "Modified teaching–learning based optimization for multi-objective optimal power flow problem" , *Energy Conversion and Management* Volume 77, January 2014, pp.597-607. (2014)
- [68] M. Ghasemia , S.Ghavidel "An improved teaching–learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow", *International Journal of Electrical Power & Energy Systems* Volume 65, February 2015, pp. 375-384.(2015)
- [69] J.Andrés Martín , G. Antonio " Optimal distributed generation location and size using a modified teaching–learning based optimization algorithm" "*International Journal of Electrical Power & Energy Systems*" Volume 50, September 2013, pp. 65-75. (2013)
- Mandal , Barun "Multi-objective optimal power flow using quasi-oppositional teaching learning based optimization". *Applied Soft Computing*. Pp.590-606 (2014).
- [70] S. Sultana. P. KumarRoy "Optimal capacitor placement in radial distribution systems using teaching learning based optimization", *International Journal of Electrical Power & Energy Systems* Volume 54, January 2014, pp. 387-398. (2014)
- [71] H. R. E. H. Boucekara, A. E. Chaib "Multiobjective optimal power flow using a fuzzy based grenade explosion method", *Energy Systems* November 2016, Volume 7, Issue 4, pp 699–721 (2016).
- [72] S. Surender Reddy "Congestion Management Using Multi-Objective Grenade Explosion Method", *WSEAS TRANSACTIONS on POWER SYSTEMS* Volume 11, pp. 81-89 (2016).
- [73] A.E.Chaib, H.R.E.H.Boucekara, "Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm", *International Journal of Electrical Power & Energy Systems* Volume 81, pp. 64-77 (2016).
- [74] Attia El-Fergany "Optimal allocation of multi-type distributed generators using backtracking search optimization algorithm", *Electrical Power and Energy Systems* 64,pp. 1197–1205 (2015).
- [75] H.R.E.H. Boucekara , M.A. Abido "Optimal power flow using the league championship algorithm: A case study of the Algerian power system", *Energy Conversion and Management* 87,pp. 58–70 (2014).
- [76] A. Salhi , D. Naimi, T. bouktir " Optimal power flow resolution using artificial bee colony algorithm based grenade explosion method " , *Electrical Systems* 12-4,pp. 734-756 (2016).
- [77]

- [78] A. Barzegar, A. Sadollah, L. Rajabpour and R. Su, "Optimal power flow solution using water cycle algorithm," 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1-4 (2016).
- [79] K. Lenin , B.Ravindranath "Water Cycle Algorithm for Solving Optimal Reactive Power Dispatch Problem" , Journal of Engineering and Technology Research, 2 (2), pp.1-11 (2014).
- [80] Mani Ashouri "Application of Krill Herd and Water Cycle Algorithms on Dynamic Economic Load Dispatch Problem", I.J. Information Engineering and Electronic Business, 4, pp. 12-19 (2014).
- [81] Ali Asghar Heidaria , Rahim Ali Abbaspour "Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems", Applied Soft Computing Volume 57, , pp.657-671 ( 2017).
- [82] A. Bhattacharya , P.K. Roy "Solution of multi-objective optimal power flow using gravitational search algorithm " , IET Generation, Transmission & Distribution Volume 6, Issue 8, pp. 751 – 763 (2012).
- [83] S. Duman , Ug̃ur G̃üvenç "Optimal power flow using gravitational search algorithm", Energy Conversion and Management 59,pp. 86–95 (2012).
- [84] J. Radosavljević , M. Jevtić "Optimal power flow for distribution networks using gravitational search algorithm", Electrical Engineering, Volume 96, Issue 4, pp 335–345 (2014).
- [85] Y. Sonmez, U. Guvenc, S. Duman and N. Yorukeren, "Optimal power flow incorporating FACTS devices using Gravitational Search Algorithm," 2012 International Symposium on Innovations in Intelligent Systems and Applications, pp. 1-5 (2012).
- [86] J. Radosavljević, N. Arsić and M. Jevtić, "Optimal power flow using hybrid PSO-GSA algorithm," 2014 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTU-CON), pp. 136-140 (2014).
- [87] P.K.Roy , S .P.Ghoshal "Biogeography based optimization for multi-constraint optimal power flow with emission and non-smooth cost function", Volume 37, Issue 12, pp 8221-8228 (2010).
- [88] A. Bhattacharya , P.K. Chattopadhyay "Application of biogeography-based optimisation to solve different optimal power flow problems", Volume 5, Issue 1, pp. 70 – 80 (2011).
- [89] P. K. Roy, S. P. Ghoshal & S. S. Thakur (2010) Multi-objective Optimal Power Flow Using Biogeography-based Optimization, Electric Power Components and Systems, Volume 38, - Issue 12, pp.1406-1426 (2010).

- [90] R. Rarick, D. Simon, F. E. Villaseca and B. Vyakaranam, "Biogeography-based optimization and the solution of the power flow problem," 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 1003-1008 (2009).
- [91] H. Ouafa , T. Bouktir "Biogeography Based Optimization Approach for Solving Optimal Power Flow Problem", Mediamira Science Publisher. Volume 54, pp. 3-4 ( 2013).
- [92] A.Ramesh Kumara , L.Premalatha "Optimal power flow for a deregulated power system using adaptive real coded biogeography-based optimization", International Journal of Electrical Power & Energy Systems Volume 73, pp. 393-399 (2015).
- [93] A. Chand Bansal , P. Farswan "A novel disruption in biogeography-based optimization with application to optimal power flow problem ", Volume 46, Issue 3, pp 590–615 (2017).
- [94] P. K. Roy , D. Mandal "Quasi-oppositional Biogeography-based Optimization for Multi-objective Optimal Power Flow", Electric Power Components and Systems Vol. 40 , Iss. 2, pp. 236-256 ( 2011).
- [95] K. Joshi , Vi. Kumar "Application of FACTS Device in Reactive power flow Solution using Biogeography -based optimization", International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering Vol. 3, Issue 2014.
- [96] H. Ouafa , S. Linda, T.Bouktir "Solving Bi-Objective Optimal Power Flow using Hybrid method of Biogeography-Based Optimization and Differential Evolution Algorithm: A case study of the Algerian Electrical Network.", Journal of Electrical Systems 12(1),pp.197-215 (2016)
- [97] N. Sinsupan , U. Leeton "Application of harmony search to optimal power flow problems" , International Conference on Advances in Energy Engineering, Beijing, , pp. 219-222 (2010).
- [98] W. Pei Sung , J. Chih-Ming "Optimal Power Flow Solution Using the Harmony Search Algorithm", Applied Mechanics and Materials (Volumes 599-601) ,pp.1938-1941 (2014).
- [99] A.H.Khazali , M.Kalantar "Optimal reactive power dispatch based on harmony search algorithm", International Journal of Electrical Power & Energy Systems Volume 33, Issue 3 , pp.684-692 (2011).
- [100] S.Sivasubramani , K.S.Swarup "Multi-objective harmony search algorithm for optimal power flow problem", International Journal of Electrical Power & Energy Systems Volume 33, Issue 3, pp.745-752 (2011).
- [101] K. Pandiarajana , C.K. Babulal "Fuzzy harmony search algorithm based optimal power flow for power system security enhancement", Electrical Power and Energy Systems Volume 78, , pp. 72-79 (2016).

- [102] R. H. Bhesdadiya, I. N. Trivedi, M. H. Pandya, D. P. Ladumor, P. Jangir and A. Parmar, "Solution of Optimal Power Flow with voltage stability enhancement using Grey Wolf Optimization," 2016 International Conference on Electrical Power and Energy Systems (ICEPES), pp. 232-238 (2016).
- [103] M. Siavash, C. Pfeifer, A. Rahiminejad and B. Vahidi, "An application of grey wolf optimizer for optimal power flow of wind integrated power systems," 2017 18th International Scientific Conference on Electric Power Engineering (EPE), pp. 1-6 (2017).
- [104] Attia A. El-Fergany , Hany M. Hasanien " Single and Multi-objective Optimal Power Flow Using Grey Wolf Optimizer and Differential Evolution Algorithms", Electric Power Components and Systems, 43:13, 1548-1559 (2015).
- [105] A. A. A. Mohamed, A. A. M. El-Gaafary "Multi-objective Modified Grey Wolf Optimizer for Optimal Power Flow" , 2016 Eighteenth International Middle East Power Systems Conference (MEPCON), pp. 982-990 (2016).
- [106] M. H. Sulaimana , Zuriani Mustaffa "Using the gray wolf optimizer for solving optimal reactive power dispatch problem" , Applied Soft Computing Volume 32, pp.286-292 (2015).
- [107] A. Deep , N. Hena "Security Constrained Optimal Power Flow using Grey Wolf Optimization Technique", International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering Vol. 4, Issue 5, pp 196- 206 (2016) .
- [108] S.Sivasakthi , N.Muralikrishnan " Chicken Swarm Optimization for Economic Dispatch with Disjoint Prohibited Zones Considering Network Losses", Journal of Applied Science and Engineering Methodologies Volume.2, No.2, pp.2-259 (2016).
- [109] N. Irsalinda , A. Thobirin "Chicken Swarm as a Multi-Step Algorithm for Global Optimization" , International Journal of Engineering Science Invention Volume 6, Issue 1, pp.08-14 (2017).
- [110] H. Delkhosh Abatari , M. S. S. Abad "Application of bat optimization algorithm in optimal power flow" , 2016 24th Iranian Conference on Electrical Engineering (ICEE), Shiraz, pp. 793-798 (2016).
- [111] S. Mishra, K. Shaw "A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data" Procedia Technology 4, pp. 802 – 806 ( 2012 ).
- [112] O. Herbadji , K. Nadhir "Optimal power flow with emission controlled using firefly algorithm" , 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO), Hammamet, pp. 1-6 (2013).
- [113] T. Govindaraj , V. Tamildurai "Firefly Algorithm For Optimal Power Flow Considering

- Control Variables" , International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering Vol. 2, Issue 2, pp. 1116-1121 ( 2011).
- [114] T. Hariharan , M. S. Kuppusamy "Optimal Power Flow Using Firefly Algorithm with Unified Power Flow Controller", Circuits and Systems 07(08),pp.1934-1942 (2016).
- [115] H.Deenadhayalan, R.suresh "A Novel Firefly Programming Method for Function Optimization" , International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 3, Special Issue 4,pp.210-215 (2014).
- [116] Abhishek Rajan , T. Malakar "Optimal reactive power dispatch using hybrid Nelder–Mead simplex based firefly algorithm ", Electrical Power and Energy Systems 66,pp. 9–24 (2015).
- [117] O. Herbadji , T. Bouktir "Optimal Power Flow Using Firefly Algorithm with Consideration of FACTS Devices UPFC" , International Journal on Electrical Engineering and Informatics - Volume7, Number1, pp.1934-1942 (2015).
- [118] V. Rao Bathina , V. Nagesh Gundavarapu "Optimal Location of Thyristor-controlled Series Capacitor to Enhance Power Transfer Capability Using Firefly Algorithm", Electric Power Components and Systems Vol. 42 , Iss. 14, pp.1541-1553 (2014) .
- [119] E. Manoranjitham , G.Shunmugalatha "Application of firefly algorithm on optimal power flow control incorporating simplified impedance UPFC model", International Journal of Electrical Power & Energy Syste- Volume 71, October 2015, pp. 358-363 (2015).
- [120] M.G. Sugirtha , p. Latha "Firefly Algorithm Based Multi-Objective Optimal Power Flow in the Presence of Wild Power" Asian Journal of Information Technology 1 5 (4),pp. 703-711, (2016).
- [121] B. Mahdad , K. Srairi "Security optimal power flow considering loading margin stability using hybrid FFA-PS assisted with brainstorming rules", Applied Soft Computing archive Volume 35 Issue C, pp. 291-309 (2015).
- [122] A. Mishra , G.V. Nagesh Kumar "Firefly Algorithm Based Generator Contingency Management Using an Interline Power Flow Controller", I J C T A, 9(32), pp. 29-37 (2016).
- [123] R. Selvarasu, M. Surya "TCSC PLACEMENT FOR LOSS MINIMISATION USING SELF ADAPTIVE FIREFLY ALGORITHM", Journal of Engineering Science and Technology Vol. 10, No. 3, 291 – 306 (2015) .
- [124] H. Ranjbar , S. F. Zarei "Imperialist competitive algorithm based optimal power flow" , 2014 22nd Iranian Conference on Electrical Engineering (ICEE), Tehran, pp. 746-750( 2014).
- [125] M. Ghasemia , S. Ghavidel "Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm" , Energy Volume 78, 15, Pages 276-289 (2014).

- [126] M. Ghasemi , S. Ghavidel "A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non-smooth cost functions", Engineering Applications of Artificial Intelligence , Volume 29, 54-69 (2014)
- [127] Anh T.N.L., Vo D.N, Ongsakul W, "Cuckoo Optimization Algorithm for Optimal Power Flow", Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1, vol 1. Springer, Cham,pp.479-493 (2015).
- [128] A. Ponnusamy , N. Rengarajan "OPTIMAL POWER FLOW SOLUTION USING CUCKOO SEARCH ALGORITHM" , ARPN Journal of Engineering and Applied Sciences VOL. 9, NO. 12, pp.2687-2691(2014).
- [129] A. Chen , S. Qiu "Optimal Power Flow Using Gbest-Guided Cuckoo Search Algorithm with Feedback Control Strategy and Constraint Domination Rule", Mathematical Problems in Engineering Volume 2017, pp.1-14 (2017).
- [130] L. A. Dung , V. N. Dieu "Application of Cuckoo Search Algorithm for Optimal Power Flow in Power System" , GMSARN International Journal 9,pp. 45 – 50 (2015).
- [131] K. Lenin , B.Ravindranath Reddy "Improved Cuckoo Search Algorithm for Solving Optimal Reactive Power Dispatch Problem " , International Journal of Research in Electronics and Communication Technology IJRECT 2014 Vol. 1 Issue 1 (2014).
- [132] T. Wang, M. Meskin and I. Grinberg, "Comparison between particle swarm optimization and Cuckoo Search method for optimization in unbalanced active distribution system," 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE), pp. 14-19 (2017)
- [133] T.Govindaraj , S.Udayakumar "Optimal Reactive Power Planning and Real Power Loss Minimization Using Cuckoo Search Algorithm", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol. 2, Issue 2,pp.1-5 (2014).
- [134] A. Mukherjee, V. Mukherjee "Solution of optimal power flow using chaotic krill herd algorithm" Chaos, Solitons & Fractals, vol. 78, pp. 10–21( 2015).
- [135] ALLEN, W.stagg "Computer methods in power system analysis (chapter 08)" , International Student Edition Book.
- [136] H Saadat " Computational aids in control systems using MATLAB". Book Electrical Engineering Series. McGraw-Hill, 1993.
- [137] A. El, M. A. Abido "Optimal power flow using differential evolution algorithm", Electr. Power Syst. Res., vol. 80, no. 7, pp. 878–885( 2010).
- [138] M. Rezaei Adaryani , a. Karami "Artificial bee colony algorithm for solving multi-objective optimal power flow problem", Int. J. Electr. Power Energy Syst., vol. 53, pp. 219–230,(2013).

- [139] Y. Sönmez and N. Yörükeren "Optimal power flow using gravitational search algorithm", *Energy Convers. Manag.*, vol. 59, pp. 86–95 (2012).
- [140] Y. Z. Li, M. S. Li, and Q. H. Wu "Electrical Power and Energy Systems Energy saving dispatch with complex constraints : Prohibited zones , valve point effect and carbon tax", *Electrical Power and Energy Syst.*, vol. 63, pp. 657–666 (2014).
- [141] Paul Denholm, Robert Margolis "History of Optimal Power Flow and Formulations", *Cite SEER*. December, pp. 1–36 (2012).
- [142] H.R.E.H. Boucekara, A.E. Chaib "Optimal power flow using an Improved Colliding Bodies Optimization algorithm", *Applied Soft Computing*, 42 119–131(2016).
- [143] J. Park, K. Lee, J. Shin, and K. Y. Lee, "A Particle Swarm Optimization for Economic Dispatch With Non smooth Cost Functions", *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42( 2005).
- [144] T. Niknam, M. R. Narimani, and R. Azizipanah-abarghooee "A new hybrid algorithm for optimal power flow considering prohibited zones and valve point effect", *Energy Convers. Manag.*, vol. 58, pp. 197–206 (2012).
- [145] M. Ghasemi, S. Ghavidel, S. Rahmani, A. Roosta "A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non- smooth cost functions", *Engineering Applications of Artificial Intelligence Volume 29* , pp. 54-69 (2014).
- [146] A. H. M. S.A.H. Soliman, "Modern Optimization Techniques with Applications in Electric Power Systems",book: *Energy Syst.* Springer, New York/ Heidelberg/Dordrecht/London, 2012.
- [147] M. A. Abido "Optimal power flow using particle swarm optimization", *International Journal of Electrical Power & Energy Systems*, vol. 24. pp. 563–571 (2002).
- [148] K. Vaisakh , L. R. Srinivas "Engineering Applications of Artificial Intelligence Evolving ant direction differential evolution for OPF with non-smooth cost functions", *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 426–436 ( 2011).
- [149] J. Hazra and A. K. Sinha "A multi-objective optimal power flow using particle swarm optimization"Volume21, Issue1, Pages 1028-1045 (2011)
- [150] M. A. Abido "Power System Stability Enhancement Using Facts Controllers : A Review Power System Stability Enhancement Using Facts Controllers ", *Arab. J. Sci. and Engineering*, vol. 34, no. 1, pp. 153–172 (2008)
- [151] J. Lemay, A. S. Mehraban, A. K. J. Reeve "Proposed Terms And Definitions For Flexible Ac Transmission System (facts) ", *Power Delivery, IEEE Transactions on*, vol. 12, no. 4, pp.

1848–1853 (1997).

- [152] S. Y. Ge, T. S. Chung "A New Method To Incorporate Facts Devices In Optimal Power Flow", IEE Cat. NO 98,EX137 1998, pp. 122–127 (1998).
- [153] C. R. F. Enrique Acha "FACTS Modelling and Simulation in Power Networks", JOHN WILEY & SONS, LTD, 2004.
- [154] AL-HULAIL, M. M. et ABIDO, M. A. "Optimal power flow incorporating FACTS devices using particle swarm optimization". In : The first GCC industrial electrical & electronics conference. PP.1-4 (2003).
- [155] M. Basu "Multi-objective optimal power flow with FACTS devices», Energy Convers. Manag., vol. 52, no. 2, pp. 903–910 (2011).
- [156] D. Conforti, D. Menniti "Optimal load-flow with N-1 steady-state security via high performance computing", Proceedings of 8th Mediterranean Electrotechnical Conference on Industrial Applications in Power Systems, Computer Science and Telecommunications (MELECON 96), Bari, Italy, vol.2, pp. 873-879 (1996).
- [157] A. Monticelli and S. Granville "Security-constrained optimal power flow with post-contingency corrective rescheduling", IEEE Trans. Power Syst., vol. 00, no. 1, pp. 175–180, (1987).
- [158] T. Duong, Y. Jiangang "A new method for secured optimal power flow under normal and network contingencies via optimal location of TCSC", Int. J. Electr. Power Energy Syst., vol. 52, no. 1, pp. 68–80 (2013).
- [159] Wei Qiu, A. J. Flueck and Feng Tu, "A new parallel algorithm for security constrained optimal power flow with a nonlinear interior point method," IEEE Power Engineering Society General Meeting, Vol. 1-2005, pp. 447-453 (2005)
- [160] F. Capitanescu, M. Glavic, D. Ernst, "Applications of security-constrained optimal power flows", in Proceedings of Modern Electric Power Systems Symposium (MEPS), Wroclaw (Poland), pp.1-8 (2006).
- [161] T. A. Mikolinnas, "Contingency Selection Of Branch Ouwaes Causing Voltage Problems and / or reactive sources and loads," Power Appar. Syst., vol. 00, no. 12, pp. 3899–3904( 1983).
- [162] T.A. Mikolinnas B. F. Wollenberg, "An Advanced Contingency Selection Algorithm," IEEE Trans. Power Appar. Syst., vol. Vol. PAS-1, no. 2, pp. 608–617 (1981).
- [163] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, "Contingency Filtering Techniques for Preventive Security-Constrained Optimal Power Flow," IEEE Trans. Power Syst., vol. 22, no. 4, pp. 1690–1697 (2007).

- [164] T. J. Bertram, K. D. Demaree and L. C. Dangelmaier, "An integrated package for real-time security enhancement," in IEEE Transactions on Power Systems, vol. 5, no. 2, pp. 592-600 (1990).
- [165] J. Mohammadi, G. Hug and S. Kar, "A benders decomposition approach to corrective security constrained OPF with power flow control devices," 2013 IEEE Power & Energy Society General Meeting, 2013, pp. 1-5 (2013).
- [166] A. O. Ekwue, "A review of automatic contingency selection algorithms for online security analysis," 1991 Third International Conference on Power System Monitoring and Control, 1991, pp. 152-155. (1991)
- [167] Singiresu S. Ra "Engineering Optimization Theory and Practice" Fourth Edition Copyright .
- [168] Ilhem Boussaid " Perfectionnement de métaheuristiques pour l'optimisation continue" Thèse doctorat Université Paris-Est, 2013.
- [169] Xin-She Yang (2011) " Review of metaheuristics and generalized evolutionary walk algorithm", Int. J. Bio-Inspired Computation, Vol. 3, No. 2, pp. 77-84 (2011).
- [170] Xin-She Yang" Nature-Inspired Metaheuristic Algorithms" Second Edition 2010 Luniver Press.
- [171] S. Binitha , S Siva Sathya, "A Survey of Bio inspired Optimization Algorithms", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, pp.137- 151(2012).
- [172] Jason Brownlee "Clever Algorithms Nature-Inspired Programming Recipes" First Edition. LuLu. January 2011 ISBN: 978-1-4467-8506-5.
- [173] Andre yPopov Hamburg "Genetic Algorithms for Optimization" these Andrey Popov TU-Sofia 2003
- [174] C D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," in IEEE Transactions on Power Systems, vol. 8, no. 3, pp. 1325-1332 (1993)
- [175] SHARMA, Bhavna et PANDIT, Manjaree. Security constrained optimal power flow employing particle swarm optimization. In : 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science. IEEE, 2012. p. 1-4 (2012).
- [176] R. Storn , K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous space", Journal of Global Optimization, 11, pp.341-359 (1997).
- [177] V. Ecaterina, M. Elisa "Differential Evolution in Parameters Estimation", Journal of Electrical and Electronics Engineering, Volume4, Numbrel,pp. 251-254( 2011).

- [178] A. Abou El Ela, M. A. Abido "Optimal power flow using differential evolution algorithm" , Electr Eng 91,69–78 (2009).
- [179] C.Kumar, T.Alwarsamy "Solution of Economic Dispatch Problem using Differential Evolution Algorithm", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-6, pp.236-241(2012).
- [180] Dorigo , L. M. Gambardella " A study of some properties of ant-Q." In H-M. Voigt, W. Ebeling, I. Rechenberg, and H-P. Schwefel, editors, Proceedings of PPSN IVFourth International Conference on Parallel Problem Solving From Nature, pp.656–665 (1996).
- [181] L. Gambardella , M. Dorigo."Ant-Q: A reinforcement learning approach to the traveling salesman problems", Twelfth International Conference on Machine Learning, pp.252–260 (1995).
- [182] M. Dorigo and L. M. Gambardella" Ant colony system: A cooperative learning approach to the traveling salesman problems", Technical Report TR/IRIDIA/1996-5, IRIDIA, Université Libre de Bruxelles, 1997.
- [183] M. Dorigo and L. M. Gambardella" Ant colony system : A cooperative learning approach to the traveling salesman problem" . IEEE Transactions on Evolutionary Computation, 1(1),pp.53–66 (1997).
- [184] T. BOUKTIR , L. SLIMANI "Optimal Power Flow of the Algerian Electrical Network using an Ant Colony Optimization Method", Leonardo Journal of Sciences ISSN 1583-0233, issue 7, pp. 43-57 (2005).
- [185] Woo Z, Hoon J, Loganathan GV " A New Heuristic Optimization Algorithm: Harmony Search". SIMULATION. 76(2),pp. 60-68 (2001).
- [186] M.YOUNES , F. KHODJA "A Hybrid Harmony Search Algorithm Approach for Optimal Power Flow", Leonardo Journal of Sciences, Issue 20, pp. 125-144 (2012).
- [187] X.-S. Yang, "Harmony Search as a Metaheuristic Algorithm : Music-Inspired Harmony Search Algorithm Theory and Applications" , Computational Intelligence, Springer Berlin, vol. 191, pp. 1-14 (2009).
- [188] Z.W. Geem " Recent Advances in Harmony Search Algorithm", Springer-Verlag Berlin Heidelberg , SCI 270, pp. 65–75 (2010).
- [189] S.I. Birbil , S. Fang " an Electromagnetism like mechanism for global optimization", J. of Global Optimization 25, pp. 263–282 (2003).

- [190] Shih-Hsin Chen, Pei-Chann Chang "A Hybrid Electromagnetism-Like Algorithm for Single Machine Scheduling Problem" , ICIC 2007, LNAI 4682, pp. 543–552 (2007).
- [191] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.(2005)
- [192] Y Zhang, Y Wang " Improved Imperialist Competitive Algorithm for Constrained Optimization", International Forum on Computer Science-Technology and Applications",pp. 204-207 (2009).
- [193] E. Atashpaz-Gargari , C. Lucas " Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", Proceedings of the IEEE Congress on Evolutionary Computation, pp. 4661-4667 (2007).
- [194] A. Kaveh, S. Talataharib "Imperialist Competitive Algorithm For Engineering Design Problems", Asian Journal Of Civil Engineering (Building And Housing) Vol. 11, No. 6 ,pp.675-697(2010).
- [195] E. Rasul, J. Sadaei " Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (RHWFTS–ICA) for short term load forecasting", Energy Convers, 76, pp.1104–1116 (2013).
- [196] D. Simon, "Biogeography-based optimization", IEEE Transactions on Evolutionary Computation, vol. 12,pp. 702 – 713 (2008).
- [197] Y. Song, M. Liu and Z. Wang, "Biogeography-Based Optimization for the Traveling Salesman Problems," 2010 Third International Joint Conference on Computational Science and Optimization, 2010, pp. 295-299 (2010).
- [198] T. Theofanis "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem "Hindawi Publishing Corporation International Journal of Combinatorics Volume 2011, Page: 1- 32 (2011).
- [199] L. DEKHICI, K; BELKADI "Firefly Algorithm for Economic Power Dispatching With Pollutants Emission", Informatica Economică vol. 16, no 2, pp. 45-75 (2012).
- [200] P. Balachennaiah "Firefly algorithm based solution to minimize the real power loss in a power system", Ain Shams Engineering Journal, Volume 9, Issue 1, March 2018, pp. 89-100 (2018)
- [201] H. Ali"League Championship Algorithm: A New Algorithm for Numerical Function Optimization". In: International Conference of Soft Computing and Pattern Recognition, pp. 43-48 (2009).

- [202] S.Muhammad, A.bMuhammad "A check pointed league championship algorithm-based cloud scheduling scheme with secure fault tolerance responsiveness", *Applied Soft Computing*-Volume 61, pp. 670-68 (2017).
- [203] A. Ahrari, M. Shariat-Panahi " GEM: A novel evolutionary optimization method with improved neighborhood search" *Applied Mathematics and Computation*, pp. 376–386 (2009).
- [204] M. Pallantla , A. Singh " Grenade Explosion Method for Maximum Weight Clique Problem"*Springer-Verlag Berlin Heidelberg* 2012, pp. 20–27 (2012).
- [205] E. Rashedi, H. Nezamabadi "GSA: A Gravitational Search Algorithm" , *Information Sciences* 179,pp. 2232–2248 (2009).
- [206] N. M. Sabri, M. Puteh and M. R. Mahmood, "An overview of Gravitational Search Algorithm utilization in optimization problems," 2013 IEEE 3rd International Conference on System Engineering and Technology, 2013, pp. 61-66 (2013).
- [207] Z. Bayraktar, M. Komurcu, Z. H. Jiang, D. H. Werner and P. L. Werner, "Stub-loaded inverted-F antenna synthesis via Wind Driven Optimization," 2011 IEEE International Symposium on Antennas and Propagation (APSURSI), 2011, pp. 2920-2923 (2011).
- [208] Z. Bayraktar, M. Komurcu "stub-loaded inverted-f antenna synthesis via wind driven optimization", *IEEE*, pp. 2920- 2923 (2011) .
- [209] Z. Bayraktar, M. Komurcu, J. A. Bossard and D. H. Werner, "The Wind Driven Optimization Technique and its Application in Electromagnetics," in *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 5, pp. 2745-2757 (2013).
- [210] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm, in: *Nature Inspired Cooperative Strategies for Optimization*" , *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, pp.65-74 (2010).
- [211] C. Teodoro Bora, S. Coelho "Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem", *IEEE TRANSACTIONS ON MAGNETICS*, VOL. 48, NO. 2, P 9 ,pp.74-950 (2012).
- [212] X-S Yang, S. Deb "Engineering optimisation by cuckoo search", *Int. J. Mathematical Modelling and Numerical Optimisation*. Vol. 1, N° 4, pp. 330-343 (2010).
- [213] A. Akbarzadeh , E. Shadkam " The Study Of Cuckoo Optimization Algorithm For Production Planning Problem " , *International Journal of Computer-Aided Technologies (IJCAx)* Vol.2, No.3, pp 1-9 (2015).
- [214] X-S Yang, S. Deb " Cuckoo search via Lévy flights" , *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210-214 (2009).

- [215] M. aber, A. Ruhul "GA with a New Multi-Parent Crossover for Solving IEEE-30 bus system ", CEC2011 Competition Problems" IEEE, pp. 1034-1040 (2011)
- [216] A. E. EibenP, E. Raué "Genetic algorithms with multi-parent recombination ", International Conference on Parallel Problem Solving from Nature PPSN: Parallel Problem Solving from Nature — PPSN III, pp 78-8 (1994).
- [217] h.Tung Khuat , M. Hanh "A genetic algorithm with multi-parent crossover using quaternion representation for numerical function optimization", Applied Intelligence volume 46, pp. 810–826 (2017)
- [218] C .repinšek, Shih-Hsi Liu "A note on teaching–learning-based optimization algorithm", Information Sciences 212, pp. 79–93 (2012).
- [219] R.V. Rao, V.J. Savsani, D.P. Vakharia " Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", Computer-Aided Design 43 (3), pp. 303–315 (2011).
- [220] R.V. Rao, V.J. Savsani, D.P. Vakharia" Teaching–learning-based optimization: an optimization method for continuous non-linear Large-scale problems", Information Sciences 183 (1) (2012) pp.1–15 (2012).
- [221] Babak Amiri "Application of Teaching-Learning-Based Optimization Algorithm on Cluster Analysis", J. Basic. Appl. Sci. Res., 2(11),pp. 11795-11802(2012).
- [222] H. Eskandar , A. Sadollah "Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems" , Computers and Structures 110–111, pp.151–166 (2012).
- [223] A. Barzegar, Ali Sadollah "A Novel Fully Informed Water Cycle Algorithm for Solving Optimal Power Flow Problems in Electric Grids", Applied Soft Computing 30, pp. 58–71 (2015)
- [224] Hegazy Rezk, Ahmed Fathy "The Application of Water Cycle Optimization Algorithm for Optimal Placement of Wind Turbines in Wind Farms " , Journals Energies Volume 12 Issue 22, pp.1-17 (2019).
- [225] P. Civicioglu, " Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm", Computers and Geosciences, vol. 46, pp. 229–247 (2012).
- [226] Bo Liu "Composite Differential Search Algorithm", Journal of Applied Mathematics Volume 2014 (2014), PP.1-15 (2014).

- [227] D.Goswami, S.Chakraborty "Differential search algorithm-based parametric optimization of electrochemical micromachining processes" *International Journal of Industrial Engineering Computations* 5,pp. 41–54 (2014).
- [228] Ali Sadollah, Ardeshir Bahreininejad " Mine blast algorithm for optimization of truss structures with discrete variables", *Computers & Structures*, 102-103,pp. 49-63 (2012).
- [229] A. Sadollah " Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems", *Applied Soft Computing* Volume 13, Issue 5, May 2013, pp. 2592-2612 (2013).
- [230] Pinar Civicioglu "Backtracking Search Optimization Algorithm for numerical optimization problems" , *Applied Mathematics and Computation* 219 ,pp. 8121–8144 (2013).
- [231] S. O. Kolawole and H. Duan, "Backtracking search algorithm for non-aligned thrust optimization for satellite formation," *11th IEEE International Conference on Control & Automation (ICCA)*, 2014, pp. 738-743 (2014).
- [232] K. Guney, A. Durmus "Backtracking Search Optimization Algorithm for Synthesis of Concentric Circular Antenna Arrays" , *International Journal of Antennas and Propagation*, vol. 2014,pp.1-14 (2014).
- [233] A. Hatamlou " Black hole: A new heuristic optimization approach for data clustering", *Information Sciences* 222,pp. 175–184 (2013).
- [234] H.R.E.H. Boucheka "Optimal power flow using black-hole-based optimization approach", *Applied Soft Computing* 24 , pp. 879–888 (2014).
- [235] H. Deeb, A. Sarangi, D. Mishra et al., Improved Black Hole optimization algorithm for data clustering, *Journal of King Saud University –Computer and Information Sciences*, pp.1-10 (2020).
- [236] I. Soto , B. Crawford "Adaptive Black Hole Algorithm for Solving the Set Covering Problem", *Recent Advances on Swarm Intelligence for Solving Complex Engineering Problems* Volume 2018, pp1-23 (2018).
- [237] S. M. Mirjalili , A. Lewis "Grey wolf optimizer", *Advances in Engineering Software*, pp. 46-61(2014).
- [238] N.S. Tung , S. Chakravorty "Grey Wolf Optimization for Active Power Dispatch Planning Problem Considering Generator Constraints and Valve Point Effect", *International Journal of Hybrid Information Technology* Vol.8, No.12, pp. 117-13 (2015).

- [239] A.M Ahmed , S.M Yahia "Grey Wolf Optimization for Multi Input Multi Output System", Universal Journal of Communications and Network 3(1), pp.1-6 ( 2015).
- [240] X. Meng, , Yu Liu "A New Bio-inspired Algorithm: Chicken Swarm Optimization" , Springer International Publishing Switzerland , pp. 86–94 (2014).
- [241] S.Sivasakthi,N.Muralikrishnan "Chicken Swarm Optimization for Economic Dispatch with Disjoint Prohibited Zones Considering Network Losses" , Journal of Applied Science and Engineering Methodologies Volume.2, No.2, pp. 255-259 (2016).
- [242] WU. DINGHUI , XU. SHIPENG "Convergence Analysis and Improvement of the Chicken Swarm Optimization Algorithm", IEEE, pp.9400-9412 (2016).
- [243] R. Rao " Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems" . Int. J. Ind. Eng. Comput. 7, pp.19–34 (2016).
- [244] A. Wadood, S.Gholami Farkoush "Application of the JAYA Algorithm in Solving the Problem of the Optimal Coordination of Overcurrent Relays in Single- and Multi-Loop Distribution Systems" , journal Complexity-Volume 2019-pp. 1-13 (2019).
- [245] A. Kaveh, V.R. Mahdavi "Colliding bodies optimization: a novel meta-heuristic method" , Comput. Struct. 139,pp. 18–27 (2014).
- [246] A. Kaveh, M. Ilchi Ghazaan "Enhanced colliding bodies optimization for design problems with continuous and discrete variables" , Adv. Eng. Softw. 77,pp. 66–75 (2014).
- [247] Ying Tan, Yuanchun Zhu "Fireworks Algorithm for Optimization", Springer-Verlag Berlin Heidelberg 2010, pp. 355–364 (2010).
- [248] V. Kumara, J. KumarChhabra "Optimal Choice of Parameters for Fireworks Algorithm", Procedia Computer ScienceVolume 70, 2015, Pages 334-340 (2015).
- [249] Zhiwei Xu, Kai Zhang, "A Fireworks Algorithm Based on Transfer Spark for Evolutionary Multitasking" , Front. Volume 13,pp.1-14 (2020) .
- [250] H.R.E.H. Boucekara, "Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport". Oper Res Int J 20, pp.139–195 (2020).
- [251] H. Khattab, A. Sharieh "Most Valuable Player Algorithm for Solving Minimum Vertex Cover Problem", International Journal of Advanced Computer Science and Applications, Vol. 10, No.8, -pp.159-167 (2020).
- [252] M.R. AlRashidi, M.E. El-Hawary "Applications of computational intelligence techniques for solving the revived optimal power flow problem" Electr. Power Syst. Res. 79 (4),pp. 694–702 (2009)

- [253] S. Frank, I. Steponavice, S. Rebennack "Optimal power flow: a bibliographic survey II. Non-deterministic and hybrid methods", *Energy Syst.* 3 (3) ,pp.259–289 (2012).
- [254] S. Garcia, F.Herrera "An Extension on Statistical Comparisons of Classifiers over Multiple Data Setsfor all Pairwise Comparisons", *Journal of Machine Learning Research* 9 pp. 2677-2694 (2008).
- [255] S. García, D. Molina "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC 2005 Special Session on Real Parameter Optimization", *CEC 2005 Special Session on Real Parameter Optimization by using non-parametric test procedures*, *J Heuristics*: pp. 617–644 (2005).
- [256] H.SSANI, M. EHGHOSSI "Teaching Mathematics and Its Applications", *Advance Access publication* 12, 108-112 (2010).
- [257] J. Derrac, S. García "Analyzing convergence performance of evolutionary algorithms: A statistical approach", *Information Sciences* 289,pp. 41–58 (2014).
- [258] Demšar, Janez "Statistical comparisons of classifiers over multiple data sets", *The Journal of Machine Learning Research* 7,pp. 1-30 (2006).
- [259] J. Salvador " A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", *Swarm and Evolutionary Computation* 1,pp. 3–18, (2011).
- [260] A. Rhyne, R. Steel " Tables for a treatments versus control multiple comparisons sign test", *Technometrics* 7,pp. 293–306 (1965).
- [261] S.García, A. Fernández "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power" , *Information Sciences* 180 ,pp. 2044–2064 (2010).
- [262] Multi-Agency Radiation Survey and Site Investigation Manual (MARSSIM). Revision 1 , Appendix I Statistical Tables and Procedures.
- [263] Rosie Shier. 2004 "Statistics: 2.2 The Wilcoxon signed rank sum test" the mathematic learning Support center.
- [264] F. Wilcoxon and R. A. Wilcox "Some Rapid Approximate Statistical Procedures" (Pearl River, NY: Leder le Laboratories, 1964), with permission of the American Cyanamid Company.
- [265] M. Friedman " The use of ranks to avoid the assumption of normality implicit in the analysis of variance" , *Journal of the American Statistical Association* 32, pp. 674–701 (1937).
- [266] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings", *Annals of Mathematical Statistics* 11 ,pp. 86–92 (1940).

- [267] D.W Zimmerman, Zumbo "Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks". *Journal of Experimental Education*, 62, pp.75-86 (1993).
- [268] D. PEREIRA, A. AFONSO " Overview of Friedman's Test and Post-hoc Analysis", *Communications in Statistics—Simulation and Computation*, 44,pp. 2636–2653 (2015).
- [269] Siegel, S. Castellan "Nonparametric Statistics for the Behavioral Sciences", 2nd ed.New York: McGraw-Hill (1988).
- [270] E. Batten Page "Ordered Hypotheses for Multiple Treatments: A Significance Test for Linear Ranks" *Journal of the American Statistical Association*, Vol. 58, No. 301, pp. 216-230, (1963).
- [271] M. MONTAZ , C. KHOMPATRAPORN " A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems", *Journal of Global Optimization*, pp.635–672 (2005)
- [272] E. Dolan, J. Moré " Benchmarking optimization software with performance profiles" , *Math. Program.* 91, pp. 201–213 (2002).
- [273] 2014 OPF PROBLEMS. <<https://www.uni-due.de/ieee-wgmho/competition2014n.d>>
- [274] L. Slimani , T. Bouktir " Economic power dispatch of power system with pollution control using artificial bee colony optimization", *Turkish Journal of Electrical Engineering & Computer Sciences* vol-21,pp.1515 -1527 (2013).

في هذه الأطروحة، عالجتنا مشكلة التدفق الأمثل للطاقة الكهربائية الهدف منها تحديد الخوارزمية أو الخوارزميات الأكثر ملائمة لحل مشاكل التدفق الأمثل للطاقة مع مراعاة شروط التكافؤ و عدم التكافؤ. من ذلك قمنا بمقارنة خصائص ثلاثين خوارزمية تم تطبيقها على عدة شبكات كهربائية. تم اختبار ثلاثين خوارزمية على العديد من الدوال المعرفة للأهداف المراد تحسينها ( تخفيض تكلفة الوقود، إحراف التوتر الكهربائي، تحسين معامل ثبات التوتر، تخفيض إبعث الغازات الملوثة، تخفيض تكلفة الإنتاج مع استعمال عدة أنواع من الوقود، تخفيض تكلفة الوقود مع مراعاة تأثير عملية فتح الصمامات، تخفيض التكلفة مع استعمال أنواع متعددة من الوقود وتأثير فتح الصمامات، تخفيض التكلفة مع إدخال أجهزة FACTS ومراعاة قيود السلامة). هذه الدوال قد تم اختبارها على أربع شبكات كهربائية (IEEE30، IEEE57، IEEE 118، و الشبكة الجزائرية لنقل الكهرباء). بسبب قص البيانات لبعض الشبكات، قمنا باختبار ميع الدوال المعرفة فقط من الشبكة IEEE30

عند حل مشكلة التدفق الأمثل للطاقة، تم حساب متغيرات الدوال المعرفة للأهداف المراد تحسينها عن طريق حل مشكلة تدفق الطاقة بتطبيق طريقة بوتن رافسون. من الحصول على مقارنة صحيحة، تم اختبار ميع الخوارزميات في نفس الشروط خاصة عدد التكرارات والتي تم تحديدها 500 تكرار. وعين من المقارنة تم استعمالهما الأولى والمسماة المقارنة المعيارية التي تعتمد على حساب القيمة الصغرى، الوسطى والإحراف المعياري والثابتة والتي تسمى الإحصائية الغير معيارية تعتمد على اختبار الإشارات المتعددة، واختبار فريدمان، وفعالية المنحنى.

أظهرت المقارنة المعيارية أن خوارزمية ABC هي الأكثر ملائمة لحل ميع المشكلات للشبكات الأربعة التي تمت دراستها تليها MVPA ثم ICA ثم GA-MPC وأخيرا DSA. أظهرت المقارنة الغير معيارية أن خوارزميات MVPA، GA-MPC و DSA هي الأكثر ملائمة لحل كل مشاكل التدفق الأمثل للطاقة.

مع الأخذ في الحسبان النتائج التي تم الحصول عليها من خلال وعي المقارنة، سنتنتج أن خوارزميات MVPA و GA-MPC و DSA هي الأكثر كفاءة لحل مشاكل التدفق الأمثل للطاقة.

الكلمات المفتاحية: تدفق الأمثل لطاقة (OPF)، واربزميات تحسين الأمثل، مقارنة الإحصائية غير معيارية، تدفق الأمثل لطاقة مع مراعاة قيود سلامة SCOPF.

## Résumé

Dans notre travail de thèse, nous avons traité un problème lié à l'écoulement de puissance optimal (OPF) dont l'objectif est l'identification de ou des algorithmes les plus adéquats pour résoudre les problèmes de l'OPF en tenant en compte des contraintes d'égalité et d'inégalité. Pour cela, une étude comparative des performances de trente algorithmes mis en œuvre pour différents cas de réseaux a été réalisée. Trente algorithmes ont été testés pour neuf cas de fonctions objectif (coût de carburant, déviation du profil de la tension, perfectionnement de la stabilité de la tension, émissions du gaz polluant, coût de carburant avec la considération de plusieurs types de combustibles, coût de carburant avec introduction de l'effet d'ouverture des vannes, coût de carburant avec la considération de plusieurs types de combustibles et l'effet d'ouverture des vannes, coût avec introduction des dispositifs FACTS et la prise en compte des contraintes de sécurité (SCOPF) et cela pour quatre réseaux électriques (IEEE30, IEEE57, IEEE 118 et le réseau Algérien réel à 59 jeux de barres). A cause d'un manque de données pour certain réseaux, sauf pour le réseau IEEE30, les neuf cas d'objectif ont été testés.

Lors de la résolution du problème de l'OPF, les variables des fonctions objectif ont été calculés par résolution du problème de l'écoulement de puissance et cela par application de la méthode de Newton-Raphson. Pour permettre une comparaison valable, tous les algorithmes ont été exécutés pour les mêmes conditions d'application notamment le nombre d'itération qui a été fixé à 500 itérations. Deux types de comparaison ont été effectués. Une dite paramétrique basée sur l'optimum global, la moyenne et la déviation standard et l'autre dite non paramétriques basée sur le test de signe multiple, test de Friedman et le, profile de performance. La comparaison paramétrique a montré que l'algorithme ABC est le plus adéquat pour la résolution de tous les problèmes pour les 04 réseaux étudiés suivi par le MVPA, ICA, GA-MPC et le DSA. La comparaison non paramétrique a montré que les algorithmes MVPA, GA-MPC et le DSA sont les plus adéquats pour les problèmes d'optimisation de l'OPF.

Tenant compte des résultats obtenus par les deux types de comparaison, on voit que les algorithmes MVPA, GA-MPC et le DSA sont les plus performants pour la résolution des problèmes de l'OPF.

**Mots clés :** Ecoulement de puissance optimal (OPF), Méthodes d'optimisation métaheuristiques, Méthodes de Comparaison statistique non paramétrique, FACTS, SCOPF.

## Abstract

In this work, an optimal power flow problem (OPF) has been treated where the aim is the identification of the powerful algorithm that can take into account both equality and inequality constraints for solving the OPF problems. For this, a comparative study of thirty algorithms implemented on four cases of bus systems has been achieved.

The considered algorithms have been tested on nine objective functions which are the fuel cost, voltage profile improvement, stability index enhancement, gas emission reduction, cost with multi-fuels option, cost with valve-point effect, cost with valve-point effect and multi-fuels option, cost with FACTS devices and security constraints optimal power flow (SCOPF) for the IEEE30, IEEE57, IEEE 118 and the Algerian 59 bus test system.

When solving the OPF problem, the control variables have been computed by solving the power flow problem using the Newton-Raphson method. To allow a valid comparison between the algorithms, these last have been performed under the same computing conditions notably the number of iterations fixed in our case as 500.

Two kinds of comparison have been done; parametric and non parametric. The parametric one is based on the consideration of the global optimum, the mean value and the standard deviation while in non parametric comparison we consider the multiple sign test, the Friedman test and the profile performance.

The parametric comparison has shown that the ABC algorithm is the best to use for all the considered objective functions for all the treated bus systems followed by the MVPA, ICA, GA-MPC and the DSA algorithms. The non parametric comparison has showed that the MVPA, GA-MPC and the DSA are the best for solving the OPF problems.

Taking into account the results of the two comparisons, we conclude that the MVPA, GA-MPC and the DSA are the suitable algorithms for solving the OPF problems.

**Key words:** Optimal power flow (OPF), Metaheuristic Optimization Algorithm, parametric and non parametric comparison, FACTS, SCOPF.