

ECOLE NATIONALE POLYTECHNIQUE D'ALGER



DEPARTEMENT de GENIE ELECTRIQUE

MEMOIRE

Présenté pour obtenir

LE GRADE DE MAGISTER EN ELECTROTECHNIQUE

Option : **Réseaux Electriques et Haute Tension**

Par

LADJICI Ahmed Amine

Ingénieur d'Etat en Electrotechnique ENP- Alger

Calcul Evolutionnaire Application sur l'Optimisation de la Planification de la Puissance Réactive

Soutenu le 04 juin 2005 devant la commission d'examen :

Président : Mr : A. BOUBAKEUR, Professeur (ENP)

Rapporteur : Mr : A. HELLAL, Professeur (UATLaghouat)

Examineurs : Mr : F. BOUDJEMA, Professeur (ENP)

Mr : A. BOUDOUR, Maitre de conference (USTHB)

Mr : S. ARIF, Chargé de Cours (UATL)

Invitée Mme : M. AMOROUAYECHE, (Dr SONELGAZ)

ملخص

في إطار هذا العمل، قدمنا مفهوم الحساب التطوري و اللوغاريتمات التطورية، و استعمالهم في إيجاد الحل الأمثل للمسائل المتعددة الأهداف أو ذات قواعد إجبارية.

كما قدمنا مسألة تخطيط الطاقة التفاعلية و كيفية حلها باستعمال الحساب التطوري الذي يمتاز في هذا الميدان، ولملاحظة تصرف اللوغاريتمات المستعملة، استعملنا شبكات كهربائية نموذجية 16 و 57 عقدة، كما قدمنا تطبيق على الشبكة الكهربائية الجزائرية

الكلمات المفتاحية الحساب التطوري، اللوغاريتمات التطورية، تخطيط الطاقة التفاعلية، مسائل متعددة الأهداف

Résumé

Dans ce travail, nous avons présenté le calcul Evolutionnaire et les différents Algorithmes Evolutionnaires, ainsi que leur utilisation dans le domaine de l'optimisation continue, spécialement aux problèmes d'optimisation avec contraintes et l'optimisation multi-objectifs.

Nous avons présenté le problème d'optimisation de la planification de la puissance réactive, qui est un problème d'optimisation multi-objectifs avec contraintes, un domaine où le Calcul Evolutionnaire excelle.

Pour observer le comportement des Algorithmes Evolutionnaire face à ce problème, nous avons appliqué nos Algorithmes sur des réseaux modèles : 16 noeuds et IEEE 57 noeuds, ainsi que sur le réseau algérien.

Mots clés : Calcul Evolutionnaire, Algorithmes Evolutionnaire, Contraintes, Multi objectifs, Optimisation de la Planification de la Puissance réactive

Abstract

In our study, the Evolutionary Computation and the Evolutionary Algorithms are presented, and their using in resolution of continuous optimisation problems especially constrained optimisation and multiobjective optimisation.

We have also presented the Optimal Reactive Power Planning (ORPP); witch is a Constrained Multiobjective optimisation problem, a suitable application for Evolutionary Computation.

In order to analyse Evolutionary Algorithms behaviour facing this problem we have made applications to various model power systems: 16 bus, IEEE 57 bus, and the Algerian Network.

Key words: Evolutionary Computation, Evolutionary Algorithms, Constraints, Multiobjective, Optimal Reactive Power Planning.

Dédicaces

A ma mère.

A mon frère Adel.

A ma sœur Ikrame.

Rappelés à Dieu un Mercredi 21 mai 2003.

Que Dieu les Accueil dans son vaste paradis.

A mon père et mes deux sœurs.

Et à tous mes amis ...

Je dédie ce modeste travail

Remerciements

Ce travail a été réalisé sous la direction Scientifique de Monsieur le Professeur **A. HELLAL**. Qu'il trouve ici l'expression de ma sincère reconnaissance pour ces précieuses directives, sa disponibilité et ses conseils.

J'adresse mes vifs remerciements au président du jury : Mr le Professeur **A. BOUBAKEUR**, ainsi qu'aux membres de jury qui ont accepté de juger ce travail.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail, et tout spécialement Mr : **A. LOUCIF**, qui par ces conseils, m'a aidé à la mise en forme de ce document.

SOMMAIRE

INTRODUCTION GENERALE	1
CHAPITRE I : INTRODUCTION AU CALCUL EVOLUTIONNAIRE	
I.1. INTRODUCTION	3
I.2. PRINCIPES DES PROCESSUS EVOLUTIONNAIRES	4
I.3. ALGORITHMES EVOLUTIONNAIRES	5
I.3.1. Algorithme Evolutionnaire de base	6
I.3.2. Différentes classes d'Algorithmes Evolutionnaires	7
I.3.2.1. Algorithmes génétiques	7
I.3.2.2. La programmation Evolutionnaire	8
I.3.2.3. Les Stratégies d'Evolution	10
I.4. LES OPERATEURS D'UN ALGORITHME EVOLUTIONNAIRE	11
I.4.1. La représentation d'individus	12
I.4.2. La reproduction	13
I.4.2.1. L'opérateur de croisement	13
I.4.2.2. L'opérateur de mutation	15
I.4.3. La sélection	18
I.4.3.1. La Sélection déterministe	19
I.4.3.2. Sélection stochastique	19
CHAPITRE II : LE CALCUL EVOLUTIONNAIRE DANS L'OPTIMISATION CONTINUE	
II.1. INTRODUCTION	23
II.2. OPTIMISATION MONO-OBJECTIVE SANS CONTRAINTES	24
II.2.1. Teste 1	24
II.2.1.1. Stratégie d'Evolution	25
II.2.1.2. Programmation Evolutionnaire	26
II.2.2. Teste 2	27
II.2.2.1. Stratégies d'Evolution	28
II.2.2.2. Programmation Evolutionnaire	29
II.2.3. Remarques	29
II.3. L'OPTIMISATION AVEC CONTRAINTES	30
II.3.1. Les fonctions de pénalités	30
II.3.1.1. Les fonctions de pénalités statiques	31
II.3.1.2. Les fonctions de pénalités dynamiques	32
II.3.1.3. Fonctions de pénalités adaptatives	32
II.3.2. Test 1	33
II.3.2.1. Les Stratégies d'Evolution	34
II.3.2.2. Programmation Evolutionnaire	35
II.3.3. Test 2	36
II.3.3.1. Stratégie d'Evolution	37
II.3.3.2. La Programmation Evolutionnaire	38
II.3.4. Remarques	39
II.4. OPTIMISATION MULTI-OBJECTIFS	39
II.4.1. Approche pleine agrégation	40
II.4.1.1. Somme pondérée	40
II.4.1.2. Approche min max	41
II.4.1.3. Approche du vecteur cible	41
II.4.2. Optimisation vectorielle non Pareto	41
II.4.2.1. Approche lexicographique	41
II.4.3. Optimisation vectorielle au sens de Pareto	42
II.4.4. Le Crowding	43
II.4.5. Test	44
II.4.5.1. Approche pleine agrégation	45
II.4.5.2. Optimisation vectorielle suivant au sens de Pareto	48
II.4.6. Remarques	49

CHAPITRE III : OPTIMISATION DE LA PLANIFICATION DE LA PUISSANCE REACTIVE

III.1. INTRODUCTION	50
III.1.1. Contraintes majeurs dans le transport en courant alternative	50
III.1.1.1. Les Limites de stabilité.....	50
III.1.1.2. Profil des tensions.....	51
III.1.2. Compensation de puissance réactive	51
III.1.2.1. Contrôle de la tension.....	52
III.1.2.2. Amélioration de la stabilité.....	52
III.2. MANAGEMENT DE PUISSANCE REACTIVE	53
III.3. PLANIFICATION DE LA PUISSANCE REACTIVE	53
III.3.1. Formulation mathématique	54
III.3.1.1. Coût de fonctionnement.....	54
III.3.1.2. Coût d'investissement :.....	54
III.3.1.3. Les contraintes.....	55
III.3.2. Formulation du problème	56
III.4. RESOLUTION DU PROBLEME	57
III.4.1. Représentation	58
III.4.2. Reproduction	59
III.4.3. Evaluation	59
III.4.3.1. Méthode du vecteur cible.....	60
III.4.3.2. Optimisation vectorielle non Pareto.....	60
III.4.3.3. Optimisation vectorielle au sens de Pareto.....	61
III.4.4. Sélection	61
 CHAPITRE IV : APPLICATIONS	
IV.1. APPLICATION SUR LE RESEAU 16 NOEUDS	62
IV.1.1. Détermination de la puissance réactive à investir	63
IV.1.1.1. Méthode de vecteur cible.....	63
IV.1.1.2. Optimisation vectorielle non Pareto.....	64
IV.1.1.3. Optimisation vectorielle au sens de Pareto.....	65
IV.1.2. Conclusions	67
IV.2. RESEAU IEEE 57 NŒUDS :	67
IV.2.1. Détermination de la puissance réactive à investir	69
IV.2.1.1. Approche du vecteur cible.....	69
IV.2.1.2. Optimisation vectorielle non Pareto.....	71
IV.2.1.3. Optimisation vectorielle au sens de Pareto.....	73
IV.2.2. Conclusions	74
IV.3. APPLICATION AU RESEAU ALGERIEN	75
IV.3.1. Détermination de la configuration optimale de compensateur	76
IV.3.2. Conclusion	78
CONCLUSION GENERALE	80
BIBLIOGRAPHIE	82
ANNEXE	85

INTRODUCTION GENERALE

La conduite des Réseaux Electriques, vu la taille d'un réseau électrique et la complexité des phénomènes dont il est siège, a toujours fait appel aux outils de calcul numériques les plus perfectionnés aidant à l'amélioration des contraintes techniques, économiques et de sécurité, auxquelles elle fait face.

L'optimisation de la planification de puissance réactive (ORPP), est parmi les problèmes les plus classiques dans les réseaux électriques, mais aussi des plus ardues. En plus de la taille des variables qu'il implique, son aspect multi-objectifs et le nombre de contraintes, constituent un véritable défi pour les méthodes d'optimisation classiques. Ces méthodes font souvent appel à des analyses mathématiques très poussées, et exigent des formulations qui s'éloignent de la formulation naturelle du problème.

Cette dernière décennie a vu l'émergence du Calcul Evolutionnaire comme méthode de prédilection dans le domaine de l'optimisation, en général, et de l'optimisation multi-objectifs, en particulier, dû à l'efficacité et la simplicité qu'il offre pour la résolution des problèmes qui ont été considérés, longtemps, comme un véritable challenge pour les méthodes de mathématique appliquée et de recherche opérationnelle.

Il devient, ainsi, naturel d'appliquer le Calcul Evolutionnaire au problème d'optimisation de puissance réactive qui est par définition un problème multi-objectifs avec contraintes, un domaine où le Calcul Evolutionnaire excelle. Et essayer d'obtenir de (meilleures) solutions, en s'appuyant sur les avantages qu'offrent les Algorithmes Evolutionnaires, dont la convergence globale et la non restriction sur les formes des objectifs et des contraintes.

Bien que certains travaux, (à l'ENP ou ailleurs), ont été orientés vers les Algorithmes Génétiques, les Stratégies d'Evolution et la Programmation Evolutionnaire restent méconnus, malgré leur longue implication dans le domaine de l'optimisation et leur efficacité qui n'est plus à démontrer.

C'est dans la perspective de mettre plus de lumière et démontrer l'efficacité de l'application de ces techniques, que nous nous sommes intéressés au domaine du Calcul Evolutionnaire et les aspects fondamentaux liés aux Stratégies d'Evolution et à la Programmation Evolutionnaire, ainsi que leur utilisation pour résoudre un problème de planification de la puissance réactive afin de voir les avantages qu'elles offrent.

Pour cela, nous avons structuré notre travail comme suit :

Le premier Chapitre présente une introduction au calcul évolutionnaire et aux algorithmes évolutionnaires, leurs notions les plus fondamentales liées à leur origine comme simulation de l'évolution basée sur les théories d'évolution naturelle et de la génétique. Les algorithmes de base et les particularités de chaque algorithme y sont présentés et les différents opérateurs d'un Algorithme Evolutionnaire sont discutés, en mettant l'accent sur les opérateurs liés aux Stratégies d'Evolution et à la Programmation Evolutionnaire.

Le Chapitre II concerne le Calcul Evolutionnaire appliqué aux problèmes d'optimisation continue avec différents aspects : mono objectif, contraintes et multi objectifs. Des exemples sur des fonctions test donneront un aperçu sur la convergence vers l'optimum global face à des fonctions multi modales, et des problèmes multi objectifs.

Le Chapitre III présente le problème d'optimisation de la planification de puissance réactive, ainsi que sa formulation.

Dans le quatrième Chapitre, nous avons présenté les résultats de simulation suite à l'application de nos algorithmes sur des réseaux modèles (Réseau à 16 nœuds et IEEE 57 nœuds), et pour donner un aspect plus pratique à notre travail, une application a été effectuée sur le Réseau Algérien.

Une Conclusion vient finir cette étude en présentant une synthèse des différentes méthodes utilisées ainsi que les perspectives futures.

Chapitre I :

***INTRODUCTION AU CALCUL
EVOLUTIONNAIRE***

I.1. Introduction

Le Calcul Evolutionnaire est un domaine jeune, le terme lui-même n'est introduit qu'en 1991. Ce domaine est le résultat d'un effort rassemblant les recherches effectuées suivant différentes approches pour la simulation de l'évolution naturelle. Ces approches sont : les *Algorithmes Génétiques*, les *Stratégies d'Evolution*, et la *Programmation Evolutionnaire*, rassemblées sous l'appellation d'*Algorithmes Evolutionnaires* [01] [02].

Les approches citées ont des aspects fondamentaux communs : chacune utilise la reproduction, le croisement et/ou la mutation pour créer de nouveaux individus à partir d'une population existante, la compétition et la sélection d'individus en compétition pour créer une nouvelle population plus adaptée. Ce qui est l'essence même de la théorie d'évolution [01].

Le Calcul Evolutionnaire essaye de trouver de meilleures solutions aux problèmes où les Algorithmes Evolutionnaires peuvent être appliqués et trouve son intérêt dans les domaines :

L'Optimisation

Les théories d'évolution naturelle modernes, la décrivent comme un processus d'optimisation. L'adaptation des êtres vivants à leurs milieux naturels, suggère que l'évolution peut trouver des solutions optimales aux contraintes posées par l'environnement aux organismes vivants.

Chercher à décrire l'évolution en terme d'un algorithme peut aider à résoudre des problèmes d'optimisation délicats auxquels les méthodes traditionnelles ne donnent pas de résultats satisfaisants, spécialement les problèmes multi modaux ou multi objectifs [01].

L'Adaptation robuste

Les phénomènes réels ne sont jamais statiques. Ils présentent souvent des aspects dynamiques qui requièrent des stratégies robustes pour préserver leurs performances dans un environnement changeant.

Holland, dans le cadre des Algorithmes Génétiques, a décrit une procédure qui peut faire évoluer des stratégies codées sous la forme de chaînes binaires ou sous la forme de lois de comportement explicite, en exploitant le potentiel de recombinaison des stratégies en compétition, et en partageant les connaissances acquises par chaque une d'elles. Le résultat est une procédure robuste qui ajuste les performances des stratégies dans un environnement variable [01].

L'Intelligence Artificielle

Un comportement intelligent peut être défini comme : « la capacité d'un système à adapter son comportement pour atteindre des buts désirés dans un environnement changeant en prédisant les circonstances futures et en prenant les actions adéquates suivant ces prédictions ». Les méthodes d'intelligence artificielle tentent souvent d'imiter l'intelligence humaine en reproduisant les règles qu'elle suit (systèmes experts) ou en simulant les réseaux de neurones.

Les Algorithmes Evolutionnaires offrent une alternative qui ne se base pas sur l'heuristique. En partant du principe que, dans la nature, c'est l'évolution qui a donné naissance à l'intelligence, la simulation de l'évolution naturelle dans une population d'algorithmes prédictifs engendre des individus avec une intelligence croissante.

Dans les premiers travaux sur la Programmation Evolutionnaire, J. Fogel propose un algorithme qui fait évoluer une population de machines à états finis. Pour pouvoir prédire les conditions futures dans un environnement, un processus d'apprentissage similaire à celui des réseaux de neurones et une sélection darwiniste sont appliqués pour ne faire évoluer que les individus dont le comportement est le plus optimal [01].

I.2. Principes des processus évolutionnaires

Les Algorithmes Evolutionnaires se basent sur la simulation du processus d'évolution naturelle. Parmi les théories d'évolution les plus admises on trouve le paradigme néo darwiniste, qui est une reformulation de la théorie d'évolution de Darwin sur la lumière de la génétique moderne. Cette théorie affirme qu'une grande partie de l'histoire de la vie naturelle peut être expliquée par des processus physiques opérant sur et dans les populations et les espèces. Ces processus sont : la reproduction, la compétition et la sélection [03].

Une population est un ensemble d'individus. Cette théorie considère un individu comme une dualité entre son patrimoine génétique "le *génotype*" et ses traits caractéristiques " le *phénotype*". Le génotype est un moyen de stockage des informations génétiques acquises, alors que le phénotype est la traduction de ces informations à l'échelle de l'individu [03].

La reproduction, est une spécificité des organismes vivants, opérant sur le patrimoine génétique d'une population. Elle enveloppe deux processus : le croisement et la mutation. Le croisement combine les génotypes de deux individus pour engendrer une descendance, la mutation introduit une altération aléatoire des génotypes engendrés au cours du croisement pour garantir plus de diversité. La reproduction permet de créer de nouveaux individus et tend à accroître les tailles des populations existantes ou à créer d'autres populations [03].

L'accroissement des tailles des populations dans un environnement à ressources limitées conduit les individus et les populations à entrer en compétition pour acquérir ces ressources. La sélection naturelle est un résultat inévitable de cette compétition, les individus les plus adaptés à leur environnement, les plus forts, ont plus de chance de survivre.

Cette théorie décrit l'évolution naturelle, comme la conséquence de l'interaction entre ces processus physiques : la reproduction crée de nouveaux individus et la sélection choisit les individus les plus optimaux. Vue de cette façon, l'évolution est un processus d'optimisation. La sélection conduit les phénotypes le plus près possible de leurs optimums, suivant les conditions et les contraintes posées par l'environnement. Et comme l'environnement est en changement perpétuel, les individus évoluent continuellement vers de nouveaux optimums. La sélection n'arrête jamais d'opérer indépendamment de la qualité des individus actuels [03].

Les caractéristiques les plus significatives du néo darwinisme sont:

- l'individu est la cible première de la sélection.
- Les variations génétiques sont des phénomènes aléatoires.
- La variation génotypique est le produit du croisement et de la mutation.
- Une évolution graduelle peut incorporer des discontinuités phénotypiques.
- Les changements phénotypiques ne sont pas tous dus à la sélection naturelle.
- L'évolution est un changement d'adaptation et de diversité.
- La sélection est un processus stochastique non déterministe.

Ces caractéristiques forment des bases communes pour tous les Algorithmes Evolutionnaires.

I.3. Algorithmes Evolutionnaires

Les Algorithmes Evolutionnaires partagent des traits communs du fait qu'ils se basent sur la simulation de l'évolution naturelle, en s'inspirant des principes du néo darwinisme [03].

Un Algorithme Evolutionnaire type présente les caractéristiques suivantes [04]:

- i) L'utilisation d'un processus d'apprentissage collectif d'une population d'individus. Un individu représente un point de l'espace de recherche, donc une solution potentielle au problème en cours.
- ii) Les descendants d'un individu sont générés par des processus aléatoires : le croisement et la mutation. Le croisement combine l'information entre des individus de la population pour en créer d'autres. La mutation crée un descendant en générant une réplique erronée de son parent.
- iii) Le processus de sélection favorise la survie des meilleurs individus en se basant sur leur adaptation. L'adaptation est obtenue par une fonction d'évaluation qui mesure la qualité de chaque individu.

Le fait que les algorithmes évolutionnaires partagent les mêmes bases n'exclut pas l'existence de différences notables entre eux. Trois principales classes d'algorithmes évolutionnaires existent : les Algorithmes Génétiques, la Programmation Evolutionnaire et les Stratégies d'Evolution [04].

- Les Algorithmes Génétiques : développés par J. Holland en 1962, considèrent le croisement comme l'opérateur de recherche le plus important dans l'espace des solutions potentielles, la mutation est considérée comme un opérateur secondaire avec un faible taux d'application. La sélection est stochastique et les individus sont souvent représentés en chaînes binaires, mais d'autres représentations peuvent être utilisées.
- Les Stratégies d'Evolution : développées par Rechenberg et Schwefel en 1965, utilisent une mutation avec une distribution normale pour modifier les individus présentés comme des vecteurs de valeurs réelles. La mutation et le croisement sont les deux opérateurs de recherche dans l'espace des solutions potentielles et l'espace des paramètres de stratégies. La sélection peut être déterministe ou stochastique.
- La Programmation Evolutionnaire : développée par J. Fogel en 1962 utilise la mutation comme seul opérateur de recherche. Appliquée aux problèmes d'optimisation continue la Programmation Evolutionnaire est similaire aux Stratégies d'Evolution, tous deux représentant les individus avec des vecteurs de valeurs réelles incluant les paramètres de stratégie. Elle utilise une mutation de distribution normale et une sélection déterministe ou stochastique.

En plus de ces algorithmes de base, d'autres algorithmes dérivés ont vu le jour, comme la programmation génétique, les Systèmes Classifier et les Algorithmes Memétiques ainsi que plusieurs méthodes hybrides.

I.3.1. Algorithme Evolutionnaire de base

En considérant les principes de l'évolution naturelle et les traits communs entre les différents Algorithmes Evolutionnaires, un algorithme de base peut être décrit comme suit [04] :

Considérons :

- > I comme une population arbitraire d'individus α .
- > $F : I \rightarrow R$ la fonction d'adaptation de ces individus.
- > μ et λ : les tailles des populations parents et descendants.
- > $P(t) = \{\alpha I(t), \dots, \alpha \mu(t)\} \in I^\mu$ représente une génération, à l'instant t .

Les opérateurs : La sélection $s : I^\lambda \rightarrow I^\mu$, la mutation $m : I^k \rightarrow I^\mu$ ainsi que le croisement $r : I^\mu \rightarrow I^k$, dépendent d'un ensemble de paramètres : Θ_s , Θ_m et Θ_r , respectivement, qui caractérisent l'opérateur concerné.

Un Algorithme Evolutionnaire utilise :

- Une procédure d'initialisation qui génère une population initiale d'individus, bien que la possibilité d'imposer des points initiaux est toujours envisageable.
- Une fonction d'évaluation déterminant l'adaptation ou la qualité des individus.
- Un critère d'arrêt qui détermine la convergence ou non, de l'algorithme, dépendant d'un ensemble de paramètres regroupés sous Θ_l

Un Algorithme Evolutionnaire typique est réduit à une simple boucle combinant la reproduction, l'évaluation et la sélection :

Input : $\mu, \lambda, \Theta_s, \Theta_m, \Theta_r$

```

1    $t = 0$  ;
2    $P(t) = \text{initialisation}(\mu)$  ;
3    $F(t) = \text{évaluation}(P(t), \mu)$  ;
4   while ( $!P(t), \Theta_l \neq \text{true}$ ) do
5        $P'(t) = \text{reproduction}(P(t), \Theta_r)$  ;
6        $F(t) = \text{évaluation}(P'(t), \lambda)$  ;
7        $P(t+1) = \text{sélection}(P'(t), F(t), \mu, \Theta_s)$  ;
8        $t = t + 1$  ;
9   end do
```

Output : α^* le meilleur individu trouvé, ou
 P^* la meilleure population trouvée.

Après la création de la population initiale et l'évaluation de ses individus la boucle est entamée :

- La population $P(t)$, dite population des parents, est utilisée par la reproduction pour créer la population des descendants $P'(t)$.
- L'ensemble des individus est évalué pour déterminer leur adaptation.
- Pour former la nouvelle génération, l'opérateur de sélection choisit μ individus de la population $P(t) \cup P'(t)$ ou de $P''(t)$ suivant les valeurs d'adaptation des individus $F(t)$.

La population des parents $P(t)$ est de taille μ , les populations $P'(t)$ et $P''(t)$ sont de tailles k et λ respectivement. Avoir $\mu = k = \lambda$ est souvent le cas dans les Algorithmes Génétiques. Avoir $k = \lambda$ est le cas de la Programmation Evolutionnaire.

I.3.2. Différentes classes d'Algorithmes Evolutionnaires

Les Algorithmes Evolutionnaires comptent trois principaux algorithmes : les Algorithmes Génétiques, les Stratégies d'Evolution et la Programmation Evolutionnaire. Comme mentionné précédemment, ces Algorithmes ont été développés séparément par différents chercheurs dans le milieu des années 60 [02].

I.3.2.1. Algorithmes génétiques

Les Algorithmes Génétiques forment une des principales classes des Algorithmes Evolutionnaires, proposés et développés par Holland. Ils se basent sur les théories d'évolution naturelle moderne et utilisent une combinaison de reproduction : croisement et mutation et de sélection, pour générer des individus de plus en plus adaptés à leur environnement, donc des solutions optimales [05].

Les Algorithmes Génétiques se basent sur la théorie des schèmes motivés par la conviction que les problèmes complexes sont plus faciles à traiter en les divisant en un ensemble de sous-problèmes plus simples.

Dans cette théorie une solution est représentée comme un ensemble de gènes (variables) composés d'allèles. Un schème peut être considéré comme une solution partielle dans laquelle seules quelques variables sont définies. Les schèmes fournissent ainsi un moyen de décomposition du problème de recherche en une hiérarchie de sous problèmes plus simples. Le cas le plus simple est d'assigner une valeur à un seul allèle [05].

Un Algorithme Génétique s'attaque en premier aux plus simples de ces sous problèmes, leurs solutions partielles sont combinées par le croisement en solutions plus complètes jusqu'à la résolution total du problème. Pour cela les solutions partielles de ces sous problèmes doivent être indépendantes.

Une extension de la théorie des schèmes est introduite par Goldberg. Elle définit les blocs élémentaires comme étant les schèmes courts de faible ordre hautement efficaces. Ces blocs élémentaires sont identifiés, échantillonnés, recombinaés et rassemblés pour former des chaînes plus longues. Goldberg avance que : « *Comme un enfant construisant des forteresses magnifiques par l'arrangement de simples blocs en bois, un AG cherche l'optimalité des performances à travers la juxtaposition des schèmes courts de faible ordre et très performants* » [05].

Suivant cette théorie, le croisement assemble les blocs élémentaires éparpillés dans la population et les recombine en blocs plus longs et plus adaptés. Pour cela, il est considéré comme l'opérateur de recherche le plus important. La mutation est un opérateur secondaire, utilisé avec un faible taux d'application à cause de sa capacité à détruire les blocs élémentaires. La mutation est utilisée pour assurer la diversité dans la population, par la création de nouveaux schèmes [05].

Algorithme génétique de base

Un algorithme génétique type peut être représenté comme suit [05]:

Input : $\mu, \lambda, \Theta_s, \Theta_m, \Theta_r$

```

1    $t = 0$  ;
2    $P(t) = \text{initialisation}(\mu)$  ;
3    $F(t) = \text{évaluation}(P(t), \mu)$  ;
4   while ( $!P(t), \Theta_l \neq \text{true}$ ) do
5        $C(t) = \text{sélection}(P(t), F(t), \mu, \Theta_r)$  ;
6        $P'(t) = \text{croisement}(C(t), \mu, \Theta_c)$ 
7        $F(t+1) = \text{évaluation}(P'(t), \mu)$  ;
8        $P(t+1) = \text{sélection}(P'(t), F(t), \mu, \Theta_s)$  ;
9        $t = t + 1$  ;
10  end do
```

Output : α^* le meilleur individu trouvé, ou P^* la meilleure population trouvée.

Les paramètres $\mu, \lambda, \Theta_s, \Theta_m$ dépendent des opérateurs utilisés et du problème en cours, l est le critère d'arrêt.

Un Algorithme Génétique procède, généralement, comme suit:

- Une population initiale est générée aléatoirement dans l'espace de recherche, puis chaque individu est évalué.
- Les individus sont soumis à une sélection déterminant les individus aptes à la reproduction (les parents), des copies de chaque parent sont mises dans la sous population $C(t)$. Le nombre de copies qu'un parent reçoit est proportionnel à son adaptation.
- Les parents sont utilisés pour créer des descendants par le croisement et la mutation. Chaque descendant est évalué pour déterminer son adaptation. L'ensemble des individus parents et descendants sont soumis à une sélection pour former la nouvelle génération.

I.3.2.2. La programmation Evolutionnaire

La Programmation Evolutionnaire est une des classes principales des Algorithmes Evolutionnaires. Elle se base sur le concept darwinien de l'évolution pour générer itérativement des solutions plus adaptées à leur environnement [06].

Bien que la Programmation Evolutionnaire a été développée comme une méthode d'intelligence artificielle, son concept était loin des premières recherches en intelligence artificielle qui se basent sur la recherche des heuristiques simples. Au lieu de développer un ensemble complexe de lois de comportements dérivés de l'expertise humaine, la Programmation Evolutionnaire fait évoluer un ensemble d'individus faisant preuve d'un comportement optimal dans un environnement, statique ou dynamique, représenté par une fonction objective [06].

Dans sa forme de base, la Programmation Evolutionnaire utilise les principales procédures d'un Algorithme Evolutionnaire, soit : l'initialisation et la reproduction réduite à la mutation seule, la mutation fournit un moyen d'exploration de l'espace de recherche prévoyant toute convergence vers un optimum local. La fonction d'évaluation mesure la qualité de chaque individu. Enfin, la procédure de sélection favorise la survie des meilleures solutions suivant leurs adaptations [06].

L'Algorithmique de base

Un algorithme standard d'un Programme Evolutionnaire peut être décrit comme suit [06]:

```

Input :  $\mu, \lambda, \Theta_s, \Theta_m, \Theta_r$ 
1    $t = 0$  ;
2    $P(t) = \text{initialisation}(\mu)$  ;
3   while ( $IP(t), \Theta_l \neq \text{true}$ ) do
4        $F(t) = \text{évaluation}(P(t), \mu)$  ;
5        $P'(t) = \text{mutation}(P'(t), \Theta_m)$  ;
6        $F(t) = \text{évaluation}(P''(t), \lambda)$  ;
7        $P(t+1) = \text{sélection}(P(y), P'(t), F(t), \mu, \Theta_s)$  ;
8        $t = t + 1$  ;
9   end do

```

Output : α^* le meilleur individu trouvé, ou P^* la meilleure population trouvée.

Les paramètres $\mu, \lambda, \Theta_s, \Theta_m$ dépendent des opérateurs utilisés et du problème en cours, l est le critère d'arrêt.

Un Programme Evolutionnaire standard enveloppe:

- La création d'une population initiale aléatoire dans l'espace de recherche, et l'évaluation de ses individus.
- Chaque parent est altéré aléatoirement pour créer k descendants (avec : $\lambda = k * \mu$) par l'opérateur de mutation, ensuite chaque descendant est évalué.
- La population formée par l'ensemble des parents et des descendants est soumise à une sélection qui détermine la nouvelle génération de taille μ .

Les tailles des populations des parents μ et des descendants λ sont généralement constantes, mais rien n'empêche leurs variations.

La Programmation Evolutionnaire diffère philosophiquement des Algorithmes Génétiques. La clé de cette différence est l'observation que l'assemblage de parties optimales ne conduit que rarement à l'optimalité de la solution globale.

Les Algorithmes Génétiques se basent sur l'identification et la combinaison des blocs élémentaires, en les regroupant en des chaînes plus larges et plus adaptées. Dans un Algorithme Génétique, la structure codée (génotype) est de la première importance car elle contient l'ensemble des blocs élémentaires optimaux découverts à travers la recherche, les Algorithmes Génétiques favorisent l'exploitations des schèmes existant sur l'exploration qui est faite à travers la mutation. Cette optimisation locale est différente de celle de la Programmation Evolutionnaire qui est une approche globale à l'optimisation [04] [06].

Un individu, dans un Programme Evolutionnaire, n'est jugé que sur son adaptation, les opérateurs de recherche effectuant simultanément toutes les variables. L'adaptation décrite en terme du comportement de chaque individu est évaluée directement, elle est seule clé de la survie d'un individu dans la population. Cette approche à l'optimisation ne considère aucun partitionnement du problème. Cet Algorithme se base essentiellement sur l'exploration de l'espace de recherche, sans aucune exploitation des caractéristiques des individus trouvés au cours de la recherche. Pour cela la Programmation Evolutionnaire utilise la mutation seule et délaisse le croisement [04] [06].

Comme l'opérateur de mutation est le seul opérateur de recherche, il doit être convenablement défini pour que l'algorithme ne stagne pas dans un optimum local. Dans les premières versions de la Programmation Evolutionnaire appliquées à l'optimisation continue, la mutation est gaussienne avec une moyenne nulle et une variance obtenue comme la racine carrée d'une transformation linéaire de la valeur d'adaptation $\varphi(X)$ [06].

$$X_i(t+1) = X_i(t) + \sqrt{\beta_i(t)\varphi_i(X_i(t) + \gamma_i)} * N_i(0,1) \quad (\text{I-1})$$

Avec : $X(t)$ vecteur de variables objectives. β_i Une constante proportionnelle et γ_i un paramètre de compensation. Les deux constante β_i et γ_i doivent être posées pour chaque problème. $N_i(0,1)$ est une variable aléatoire normale centrée réduite.

Pour venir à bout des difficultés de trouver les paramètres adéquats associés à l'opérateur de mutation la notion d'auto-adaptation de la mutation a vu le jour. Au lieu d'imposer une variance constante, la mutation auto-adaptative fait évoluer sa variance au cours de la recherche ce qui la rend plus efficace. La méthode d'auto-adaptation de la mutation la plus utilisée dans la Programmation Evolutionnaire est proposée par J. Fogel en 1995 [06]:

$$\begin{aligned} X_i(k+1) &= X_i(k) + v_i(k) * N_i(0,1) \\ v_i(k+1) &= v_i(k) + [\sigma v_i(k)]^{1/2} * N_i(0,1). \end{aligned} \quad (\text{I-2})$$

La constante σ assure que la variance ne soit pas négative.

I.3.2.3. Les Stratégies d'Evolution

Les Stratégies d'Evolution constituent une autre classe d'Algorithmes Evolutionnaires. A l'inverse des autres approches, les Stratégies d'Evolution ont été conçues, à l'origine, comme des méthodes d'optimisation numériques. Leurs premières applications étaient l'optimisation d'un corps dans un tunnel de vent [07].

Abordant les problèmes d'optimisation continue, les Stratégies d'Evolution sont très similaires à la Programmation Evolutionnaire. Ils partagent la même philosophie : le problème d'optimisation est considéré dans sa globalité et aucun partitionnement n'est effectué, ils n'obéissent pas à la théorie des schèmes. Les deux méthodes mettent l'accent sur la mutation comme principal opérateur de recherche et utilisent une mutation auto-adaptative [07].

La seule différence notable avec la Programmation Evolutionnaire est l'utilisation du croisement par les Stratégies d'Evolution. Le croisement est utilisé comme second opérateur de recherche, son rôle n'étant pas la manipulation des blocs élémentaires mais d'assurer plus de diversité dans la population.

Algorithme de base

Les différentes Stratégies d'Evolution peuvent être introduites par la notation de Schwefel [07]:

- i) La SE $(\mu + \lambda)$: désigne une Stratégie d'Evolution qui génère λ descendants des μ parents et sélectionne les μ meilleurs individus de l'ensemble des descendants et des parents pour former la nouvelle génération.
- ii) La SE (μ, λ) avec $\lambda = k\mu$ et $k > 1$ ($k \sim 5\div 7$) génère λ descendants des μ parents mais sélectionne les μ meilleurs individus de la population des descendants seulement.
- iii) La SE $(\mu + 1)$ est semblable aux Algorithmes Génétiques à état permanent dans le sens où un seul individu est créé dans chaque génération.

Un algorithme d'une Stratégie d'Evolution standard peut être représenté comme suit [07] [08]:

```

Input :  $\mu, \lambda, \Theta_s, \Theta_m, \Theta_r$ 
1    $t = 0$  ;
2    $P(t) = \text{initialisation}(\mu)$  ;
4   while ( $!P(t), \Theta_l \neq \text{true}$ ) do
5        $P'(t) = \text{croisement}(P(t), \Theta_r)$  ;
6        $P''(t) = \text{mutation}(P'(t), \Theta_m)$  ;
7        $F(t) = \text{évaluation}(P(t), P''(t), \mu, \lambda)$  ;
8        $P(t+1) = \text{sélection}(P(t), P''(t), F(t), \mu, \Theta_s)$  ;
9        $t = t + 1$  ;
10  end do

```

Output : α^* le meilleur individu trouvé, ou P^* la meilleure population trouvée.

Les paramètres $\mu, \lambda, \Theta_s, \Theta_m$ dépendent des opérateurs utilisés et du problème en cours, l est le critère d'arrêt.

Une Stratégie Evolutionnaire procède comme suit :

- l'initialisation aléatoire d'une population dans l'espace de recherche et dans l'espace des paramètres de stratégie.
- la population des parents $P(t)$ est sujet à la reproduction pour créer la population des descendants $P''(t)$.
- la nouvelle génération $P(t+1)$ est sélectionnée soit de la population des descendants seule, ou des populations des descendants et des parents.

I.4. Les opérateurs d'un Algorithme Evolutionnaire

Nous avons vu précédemment qu'un Algorithme Evolutionnaire est, dans sa forme de base, une combinaison de trois processus appliqués à une population d'individus. Ces processus sont : la reproduction, la sélection et l'évaluation. C'est la définition de l'individu et les processus qu'il subit qui définissent un Algorithme Evolutionnaire et les différences entre chaque algorithme.

I.4.1. La représentation d'individus

Dans un Algorithme Evolutionnaire, un individu représente un point de l'espace de recherche, donc une solution potentielle au problème en main. Dans un Programme Génétique, par exemple, un individu représente une portion de code Lisp. Dans un Programme Evolutionnaire traitant l'intelligence artificielle, un individu est un algorithme prédictif.

Lorsqu'on est confronté à des problèmes d'optimisation continue de la forme :

$$\begin{aligned} \min \quad & F(X) : \mathfrak{R}^n \rightarrow \mathfrak{R} \\ \text{st} \quad & X \in A \end{aligned} \tag{I-3}$$

L'espace de recherche est un domaine de \mathfrak{R}^n et le point X de l'espace de recherche est un vecteur de dimension n : $X = \{x_1, \dots, x_n\}$.

Appliqués aux problèmes d'optimisations continues les Algorithmes Evolutionnaires utilisent deux types de codages.

Les Algorithmes Génétiques n'opèrent pas sur les variables réelles mais sur une représentation codée en un alphabet fini (en binaire généralement). L'utilisation du codage par les Algorithmes Génétiques est motivé par la théorie des schèmes où l'utilisateur doit utiliser le codage qui permet la plus grande séparabilité linéaire du problème. En plus, se basant sur la théorie des blocs élémentaires Goldberg recommande [09]:

- l'utilisateur doit choisir un codage tel que les schèmes les plus courts et d'ordre inférieur sont les plus pertinents.
- L'utilisateur doit sélectionner le plus petit alphabet qui permet une expression naturelle du problème.

Les Stratégies d'Evolution et la Programmation Evolutionnaire ne sont pas concernées par la théorie des schèmes, et considèrent le problème dans sa globalité sans aucun partitionnement. Ils considèrent qu'en absence d'évidences théoriques et empiriques sur l'utilité d'un quelconque codage, il n'est pas utile d'utiliser autre que la représentation réelle [09].

Considérant la mutation comme l'opérateur de recherche principal, se présentant sous la forme d'une perturbation gaussienne multi variable, il est possible à chaque individu d'avoir un vecteur auxiliaire déterminant les paramètres de la mutation qui lui est appliquée [09].

En général, un individu est représenté par trois vecteurs (X, α, σ) avec :

- $X : \{x_1, x_2, \dots, x_n\}$ vecteur objectif.
- $\sigma : \{\sigma_1, \sigma_2, \dots, \sigma_n\} \in \mathfrak{R}^n$ vecteur des déviations standard.
- $\alpha : \{\alpha_1, \alpha_2, \dots, \alpha_{n(n-1)/2}\}$ vecteur des angles de rotation définissant la covariance entre les mutations en chaque dimension (il peut être nul dans le cas ou aucune covariance n'est considérée).

I.4.2. La reproduction

Les Algorithmes Evolutionnaires trouvent les solutions optimales d'un problème en explorant l'espace de recherche, cette exploration est accomplie par la création de nouveaux individus par le processus de reproduction qui enveloppe le croisement et la mutation.

I.4.2.1. L'opérateur de croisement

Les opérateurs de croisement introduits dans les Algorithmes Evolutionnaires sont de simples abstractions des processus naturels. Le croisement utilise deux individus ou plus, appelés parents pour créer au moins un nouveau individu appelé descendant.

Le premier opérateur de croisement introduit par HOLLAND, est une procédure en trois étapes [10]:

- Deux parents sont choisis aléatoirement de la population des parents.
- Une ou plusieurs locations (locus) sont choisies comme points de rupture du génome (points de croisement).
- Les segments des chaînes parentes sont échangés et combinés pour produire deux nouveaux individus 'descendants'.

La proportion des parents subissant le croisement durant une génération est contrôlée par le taux de croisement, $pc \in [0,1]$, qui détermine la fréquence d'application de l'opérateur.

Plusieurs formes de croisement existent. Une partie d'entre eux dérivent des efforts entrepris dans la communauté des stratégies d'Evolution due à leur longue implication dans les problèmes d'optimisation continue.

Le Croisement à k points

Le croisement en un seul point est la forme la plus simple, puisqu'il est accompli en sélectionnant aléatoirement un point de croisement $k \in \{1, \dots, n\}$ des deux parents X_1 et X_2 , les segments ainsi constitués sont combinés pour produire deux nouveaux descendant X'_1 et X'_2 , comme suit : [11]

$$\begin{array}{l} X_1: \{x_{11}, \dots, x_{1k}, \dots, x_{1n}\} \\ X_2: \{x_{21}, \dots, x_{2k}, \dots, x_{2n}\} \end{array} \xrightarrow{\text{Croisement}} \begin{array}{l} X'_1: \{x_{11}, \dots, x_{1k}, x_{2k+1}, \dots, x_{2n}\} \\ X'_2: \{x_{21}, \dots, x_{2k}, x_{1k+1}, \dots, x_{1n}\} \end{array} \quad (\text{I-4})$$

Cet opérateur peut être étendu à un croisement en k points en choisissant des positions aléatoires k_i dans les deux parents X_1 et X_2 . Les segments se trouvant entre deux positions k_i, k_{i+1} sont combinés pour créer deux descendants X'_1 et X'_2 , comme suit : [11]

$$\begin{array}{l} X_1: \{x_{11}, \dots, x_{1k_1}, \dots, x_{1k_2}, \dots, x_{1k_i}, \dots, x_{1n}\} \\ X_2: \{x_{21}, \dots, x_{2k_1}, \dots, x_{2k_2}, \dots, x_{2k_i}, \dots, x_{2n}\} \end{array} \xrightarrow{\text{Croisement}} \begin{array}{l} X'_1: \{x_{11}, \dots, x_{1k_1}, x_{2k_1+1}, \dots, x_{2k_2}, x_{1k_2+1}, \dots, x_{1k_i}, x_{2k_i+1}, \dots, x_{2n}\} \\ X'_2: \{x_{21}, \dots, x_{2k_1}, x_{1k_1+1}, \dots, x_{1k_2}, x_{2k_2+1}, \dots, x_{2k_i}, x_{1k_i+1}, \dots, x_{1n}\} \end{array} \quad (\text{I-5})$$

Ce type de croisement est peu utilisé.

Le croisement uniforme

Le croisement uniforme est une extension du croisement à k points. A partir de deux parents X_1 et X_2 , un descendant X' est créé comme suit : [11]

$$X'_i = \alpha x_{1i} + (1 - \alpha)x_{2i}, \quad i = 1, \dots, n \quad (\text{I-6})$$

Où : α est une variable aléatoire uniforme sur $\{0,1\}$.

Le croisement uniforme est le plus utilisée dans les Algorithmes Génétiques.

La recombinaison intermédiaire

Contrairement aux opérateurs de croisement à k points, qui échangent les informations entre les parents, la recombinaison intermédiaire crée les descendants en pondérant les composants de plusieurs parents. La version canonique de cet opérateur combine deux individus X_1 et X_2 pour créer un descendant X' comme suit : [12]

$$X'_i = \alpha x_{1i} + (1 - \alpha)x_{2i} \quad (\text{I-7})$$

Où : α est un nombre entre $[0,1]$.

Cet opérateur peut être étendu à plus de deux parents où un descendant est créé par: [12]

$$X'_i = \sum_{\substack{j=1:k \\ i=1:n}} \alpha_j x_{ji} \quad (\text{I-8})$$

La recombinaison intermédiaire est utilisée essentiellement dans les Stratégies d' Evolution.

Autres opérateurs de croisement

D'autres opérateurs de croisement peuvent être utilisés [12]:

- i) **Le croisement heuristique** : qui peut être formulé comme suit,

$$X' = u(X_2 - X_1) + X_2 \quad (\text{I-9})$$

Où u est une variable aléatoire uniforme entre $[0,1]$ et X_1 et X_2 sont deux parents choisis aléatoirement avec X_2 obligatoirement meilleur que X_1 .

- ii) **Le croisement simplexe** : cet opérateur utilise un groupe de parents de taille supérieure à deux, il détermine l'individu le moins adapté dans ce groupe X_2 , et calcule le centre pondéré C du groupe sans X_2 , le descendant X' étant créé par :

$$X' = C + (C - X_2) \quad (\text{I-10})$$

- iii) **Le croisement géométrique** : dans la version canonique de cet opérateur, le descendant X' est créé en utilisant deux parents comme suit :

$$X' = [(x_{11}x_{21})^{0.5}, \dots, (x_{1n}x_{2n})^{0.5}] \quad (\text{I-11})$$

Cet opérateur peut être généralisé à k parents :

$$X' = [(x_{11}^{\alpha_1} x_{21}^{\alpha_2} \dots x_{k1}^{\alpha_k}), \dots, (x_{1n}^{\alpha_1} x_{2n}^{\alpha_2} \dots x_{kn}^{\alpha_k})] \quad (\text{I-12})$$

- iv) **Le croisement basé sur l'adaptation** : cet opérateur prend k parents et génère un descendant dont chaque élément est sélectionné d'un parent avec une probabilité correspondante à son adaptation relative. Si un parent a une valeur d'adaptation $f(i)$, alors la probabilité qu'un élément du descendant soit choisi de ce parent est :

$$f(i) / \sum f(i) \quad i=1, \dots, k \quad (\text{I-13})$$

I.4.2.2. L'opérateur de mutation

La mutation est définie comme la création d'un individu en produisant une réplique erronée de son parent en introduisant une perturbation aléatoire dans son génotype. Les Algorithmes Evolutionnaires considèrent la mutation de deux points de vue différents :

Cadre des Algorithmes Génétiques

L'opérateur de mutation utilisé dans les Algorithmes Génétiques a été introduit par Holland pour un alphabet fini quelconque $I = A_1x \dots x A_l$, où $A_i = \{\alpha_i1, \dots, \alpha_ik_i\}$.

Cet opérateur procède comme suit :

- i) Déterminer la position où appliquer la mutation : i_1, \dots, i_h ($i_j \in \{1, \dots, l\}$), uniformément aléatoire, chaque position a la même probabilité pm de subir la mutation, indépendamment des autres positions.
- ii) Former le nouveau vecteur $a' = (a_1, \dots, a_{i-1}, a'_i, a_{i+1}, \dots)$ où l'allèle a'_i est choisi aléatoirement uniforme de l'espace de valeurs admissible à la position i .

Dans le cas d'une représentation binaire, on définit par pm la probabilité qu'un allèle $a_i \in \{0, 1\}$ soit inversé, l'opérateur de mutation $m : \{1, 0\}^l \rightarrow \{1, 0\}^l$ produit un nouveau individu $a' = m(a)$ comme suit :

$$a'_i = \begin{cases} a_i & u > p_m \\ 1 - a_i & u \leq p_m \end{cases} \quad (\text{I-14})$$

Où : $u \sim U([0, 1])$ est une variable aléatoire.

Les Algorithmes Génétiques considèrent la mutation comme un opérateur secondaire (background operator), qui ne sert qu'à assurer l'accessibilité de tous les allèles à la recherche, et maintenir ainsi, la diversité dans la population, en créant de nouveaux allèles. Il faut noter que l'opérateur de croisement dans un Algorithme Génétique ne crée pas de nouveaux allèles mais fait propager les meilleurs d'entre eux dans la population. Sans opérateur de mutation, un Algorithme génétique converge vers un optimum local [05].

Les investigations théoriques et empiriques, dans le cadre de l'optimisation des fonctions, montrent les bénéfices qu'apporte la mutation comme opérateur de recherche, plus spécifiquement [13]:

- i) Les résultats empiriques favorisent un large taux initial de mutation, qui décroît exponentiellement à travers le temps.
- ii) La limite inférieure $pm = 1/l$ est considérée comme un taux optimal de mutation.

Ces résultats montrent que, non seulement pour les Stratégies d'Evolution et la Programmation Evolutionnaire, mais aussi pour les algorithmes génétiques, la mutation est un opérateur de recherche important qui ne doit pas être négligée.

Cadre des Stratégies d'Evolution et Programmation Evolutionnaire

Il existe une multitude d'opérateurs de mutation introduits dans le cadre des Stratégies d'Evolution et de la Programmation Evolutionnaire, utilisant une représentation réelle des individus.

Soit : $X' = m(X)$ l'opérateur de mutation, avec X est parent, et X' son descendant. La forme la plus commune de cet opérateur est : $X' = X + M$, où la mutation M est une variable aléatoire de moyenne nulle. Définie de cette manière, la mutation est une procédure de recherche locale [14].

La perturbation M peut prendre différentes formes. Par exemple, M peut être une variable aléatoire suivant une loi de distribution uniforme, donc : $U \rightarrow U(a, b)^n$ où : a et b sont les limites inférieure et supérieure de la variable X , respectivement, (a est souvent égale à $-b$). Le résultat d'un tel opérateur est un descendant qui se situe dans un hypercube défini par $X + U(a, b)^n$. Cette méthode souffre de sa susceptibilité de converger prématurément lorsque le parent se situe sur un optimum local [14].

La forme de mutation la plus utilisée, introduit une perturbation suivant une distribution non bornée, la distribution gaussienne étant souvent choisie. Cette perturbation est définie par :

$$g(x) = [\sigma(2\pi)^{1/2}]^{-1} \exp[-0.5(x - \mu)^2 / \sigma^2] \quad (\text{I-15})$$

La mutation est de moyenne nulle, donc $\mu = 0$. La déviation standard σ fournit, dans ce cas, un paramètre de contrôle sur la distance entre un descendant et son parent ΔX .

Dans les premières méthodes de mutation gaussienne, une seule variance était utilisée pour toutes les dimensions, ce qui manquait d'efficacité et de robustesse. La mutation a été améliorée en utilisant une variance pour chaque dimension indépendamment des autres. La détermination d'un paramètre de contrôle pour chaque dimension est une tâche difficile pour un opérateur humain, ce qui a permis aux opérateurs de mutation auto-adaptatifs de voir le jour.

La mutation auto-adaptative

L'auto-adaptation de la mutation permet de régler la variance au cours du processus de recherche. Dans la première phase de recherche, une variance élevée est exigée pour pouvoir parcourir tout l'espace de recherche. Cette variance doit décroître au fur et à mesure que l'Algorithme converge pour permettre une meilleure recherche locale et converger plus rapidement vers l'optimum.

Les deux méthodes d'auto-adaptation les plus utilisées étaient développées indépendamment par Schwefel dans le cadre des Stratégie d'Evolution et par Fogel dans le cadre de la Programmation Evolutionnaire [14].

Dans ces deux méthodes, un individu est représenté par un vecteur de variables objectifs X et un vecteur de paramètres de stratégie : σ où σ_i est la déviation standard à utiliser pour chaque dimension i . Les paramètres de stratégie sont actualisés différemment pour chaque méthode.

Schwefel a développé la méthode suivante : [15] [16]

$$\begin{aligned}\sigma'_i &= \sigma_i \exp(\tau_0 N(0,1) + \tau N_i(0,1)) \\ X'_i &= X_i + N(0, \sigma'_i)\end{aligned}\tag{I-16}$$

Fogel a proposé la méthode suivante : [14]

$$\begin{aligned}X'_i &= X_i + N(0, \sigma_i) \\ \sigma'_j &= \sigma_j + \chi N(0, \sigma_j)\end{aligned}\tag{I-17}$$

Où $\tau_0 \propto 1/[2(n^{1/2})]^{1/2}$, $\tau \propto 1/(2n)^{1/2}$ et χ sont des constantes.
 $N(0,1)$ est une variable aléatoire normale centrée réduite.

Ces deux mutations peuvent être utilisées indifféremment par les Stratégies d'Evolution et par la Programmation Evolutionnaire.

Plusieurs comparaisons ont été faites entre ces deux formulations. Les tests ont montré que la méthode introduite par Schwefel est plus efficace lorsqu'elle est appliquée sur une série de fonctions standard. Celle introduite par Fogel paraît plus efficace lorsque les fonctions sont bruitées.

Au lieu d'utiliser n perturbations gaussiennes indépendantes, la mutation a été étendue vers une perturbation gaussienne multivariable utilisant une covariance arbitraire. Schwefel a décrit une procédure incorporant des angles de rotation α , comme suit : [16]

$$\begin{aligned}\sigma'_i &= \sigma_i \exp(\tau_0 N(0,1) + \tau N(0,1)) \\ \alpha'_j &= \alpha_j + \beta N_j(0,1) \\ X'_i &= X_i + N(0, \sigma'_i, \alpha'_j)\end{aligned}\tag{I-18}$$

Où: $\beta = 0.0873$ (5°), $i = 1, \dots, n$ et $j = 1, \dots, n(n-1)/2$.

Mutation non uniforme

Une autre méthode de mutation utilisant une perturbation de moyenne nulle est la mutation non uniforme proposée par Michalewicz [14]:

$$X'_i(t) = \begin{cases} X_i(t) + \Delta(t, ub_i - X_i(t)) & \text{si } u < 0.5 \\ X_i(t) - \Delta(t, X_i(t) - lb_i) & \text{si } u \geq 0.5 \end{cases} \quad (\text{I-19})$$

Où :

- $X_i(t)$ est le $i^{\text{ème}}$ élément du vecteur X à la génération t , $X_i \in [lb_i, ub_i]$, les limites supérieures et inférieures (respectivement).
- u est une variable uniforme $U(0,1)$.
- la fonction $\Delta(t,y)$ retourne une valeur entre $[0,y]$ avec $\Delta(t,y)$ tend vers zéro lorsque t tend vers l'infinie.
- la fonction $\Delta(t,y) = yu(1 - t/T)^b$ est souvent utilisée.

Cette méthode de mutation est souvent utilisée dans le cadre des Algorithmes Génétiques (non canonique) utilisant une représentation réelle.

I.4.3. La sélection

La sélection est l'opérateur le plus important dans un Algorithme Evolutionnaire. L'objectif principal de la sélection est de faire valoir les meilleurs individus dans la population. Cet opérateur ne crée aucune nouvelle solution, mais, garde les meilleures solutions dans la population et supprime les autres. C'est ce qui rend un algorithme évolutionnaire une méthode d'optimisation.

On peut différencier entre deux concepts de sélection suivant leurs objectifs : [17]

- La sélection pour la reproduction crée des copies des individus de la population et les places dans la population des parents, le nombre de copies que reçoit chaque individu est proportionnel à son adaptation. Cette sélection est propre aux Algorithmes Génétiques.
- La sélection pour la survie qui choisit les meilleurs individus pour générer la nouvelle génération.

Le principe de la sélection est qu'une solution ayant une plus grande adaptation doit avoir une plus grande probabilité de sélection. Mais, les opérateurs de sélection diffèrent dans la manière d'assigner cette probabilité à chaque individu.

Deux principaux types de sélection existent : [17]

- la sélection déterministe qui trie les individus suivant leurs adaptations et crée la nouvelle génération en choisissant les meilleures solutions.
- La sélection stochastique qui assigne une probabilité à la sélection de chaque individu en fonction de son adaptation.

I.4.3.1. La Sélection déterministe

Cette méthode de sélection choisit les meilleurs individus de la population pour former la nouvelle génération. A l'inverse de la sélection stochastique, aucune probabilité de sélection n'est utilisée du moment où les μ meilleurs individus sont automatiquement choisis, avec μ comme taille de la population des parents.

La Sélection N/2 élitiste :

Les individus des populations des parents et des descendants sont triés suivant leurs adaptations et la meilleure moitié est sélectionnée. Cette méthode est généralement utilisée dans les Algorithmes Génétiques.

Dans le cadre des Stratégies d'évolution

Les Stratégies d'Evolution utilisent souvent la sélection déterministe. On peut différencier entre deux types de sélection :

Stratégie d'évolution SE ($\mu+\lambda$) : cet opérateur sélectionne les μ meilleurs individus à partir de l'ensemble des individus des populations des parents et des descendants. Si, une sélection déterministe est utilisée elle préservera les meilleures solutions trouvées par l'algorithme au cours de la recherche, ce qui peut conduire à une convergence prématurée à un optimum local.

Stratégie d'évolution (μ, λ) : avec : $\lambda = k \mu$, l'opérateur de sélection choisit les μ meilleurs individus de la population des descendants seuls. Dans ce cas, la meilleure solution trouvée dans la génération précédente n'est pas préservée dans la génération suivante. Et comme k descendants sont créés de chaque parent ($k=5\div 7$), la recherche est très exhaustive, et peut éviter les problèmes de convergence prématurée à condition qu'une mutation auto-adaptative soit utilisée.

I.4.3.2. Sélection stochastique

Il existe une multitude d'opérateurs de sélection stochastique. On cite : [17]

Sélection proportionnelle

Cet opérateur assigne une probabilité de sélection proportionnelle à l'adaptation de chaque individu, calculée par :

$$\Pr_{pop}(a_i) = \frac{f(a_i)}{\sum_{j=1}^{\lambda} f(a_j)} \quad (\text{I-20})$$

Où : $f(a_i)$ est l'adaptation assignée à l'individu a_i , λ est le nombre d'individus.

Cet opérateur de sélection présente des inconvénients majeurs :

- comme la probabilité de sélection d'un individu est proportionnelle à son adaptation les valeurs négatives d'adaptation ne sont pas permises. Ce processus ne peut pas prendre en considération les problèmes de minimisation directement, qui doivent être transformés en problèmes de maximisation.
- si, la population contient un individu avec une adaptation exceptionnellement élevée par rapport au reste de la population, un 'super individu', il aura une probabilité de sélection tel qu'il occupera la majeure partie des générations suivantes, ce qui conduit à une convergence prématurée de l'algorithme à un optimum local.
- si, les individus ont des valeurs d'adaptations assez proches, cette méthode de sélection se comporte, alors, comme une sélection aléatoire, ce qui conduit à une stagnation du processus de recherche.

Pour parer à ces défauts, la fonction d'adaptation doit être mise en l'échelle et bornée entre deux limites, inférieure et supérieure, préétablies. En général, on pose la fonction d'adaptation comme une translation linéaire de la fonction objective, comme suit :

$$\Phi(a_i) = \alpha f(a_i) - \beta(t) \quad (\text{I-21})$$

Avec : Φ fonction d'adaptation, f fonction objective, $\alpha \in [-1,1]$ et $\beta \geq f_{\min}$

Même si une mise en échelle permet de pallier les principaux défauts de la sélection proportionnelle, il reste qu'elle est peu utilisée, car elle exige des calculs supplémentaires qui augmentent le temps d'exécution de l'algorithme.

Sélection basée sur le rang (ranking selection)

La sélection basée sur le rang assigne une probabilité de sélection à chaque individu en fonction de son rang ou de son ordre dans la population.

Cette probabilité est assignée de différentes manières dont : [17]

Rangement linéaire

Le rangement linéaire assigne une probabilité de sélection à chaque individu proportionnellement à son rang dans la population. Cette méthode peut être implémentée en spécifiant un seul paramètre β_{rank} qui définit le nombre de copies assignés au meilleur individu durant chaque génération. La probabilité de sélection pour un individu i peut être définie par :

$$\text{Pr}_{\text{linr_rank}}(i) = \frac{\alpha_{rank} + [\text{rank}(i)/(\mu - 1)](\beta_{rank} - \alpha_{rank})}{\mu} \quad (\text{I-22})$$

Où : α_{rank} est le nombre de copies allouées à l'individu le moins meilleur.

Et comme la somme des probabilités de sélection est égale à 1, i.e. :

$$\sum_{i=0}^{\mu-1} \Pr_{lin_rank}(\alpha_i) = \alpha_{rank} + \frac{\beta_{rank} - \alpha_{rank}}{\mu(\mu-1)} \left(\sum_{i=0}^{\mu-1} i \right) = \frac{1}{2}(\beta_{rank} + \alpha_{rank}) \quad (I-23)$$

Alors,

$$\beta_{rank} = 2 - \alpha_{rank} \quad 1 \leq \beta_{rank} \leq 2$$

La probabilité de la sélection du meilleur individu est deux fois celle de la moyenne de la population, donc le meilleur individu de la population a deux fois plus de chance d'être sélectionné qu'un individu moyen.

Rangement non linéaire

Le rangement non linéaire assigne des probabilités de sélection suivant une fonction non linéaire du rang de l'individu, comme suit :

La probabilité de sélection peut être une fonction quadratique du rang de l'individu :

$$\Pr_{sq_rank}(i) = \frac{\alpha + [rank(i)^2 / (\mu - 1)^2](\beta - \alpha)}{c} \quad (I-24)$$

$$c = \frac{(\beta - \alpha)\mu(2\mu - 1)}{6(\mu - 1)} + \mu\alpha \quad (I-25)$$

Cette version utilise deux paramètres α et β , où $0 < \alpha < \beta$, ce qui donne une probabilité de sélection comprise entre α/c et β/c .

La distribution géométrique est aussi souvent utilisée avec :

$$\Pr_{geom_rank}(i) = \alpha(1 - \alpha)^{\mu-1-rank(i)} \quad (I-26)$$

La sélection basée sur le rang évite le problème de convergence prématurée et de stagnation de l'algorithme, car les probabilités de sélection sont indépendantes de la valeur d'adaptation de l'individu, et peuvent être comprises entre des valeurs bien déterminées pour chaque problème pour améliorer la convergence.

Sélection par tournoi

Dans cette méthode un groupe de q individus est choisi aléatoirement de la population. Ce groupe prend part à un tournoi, l'individu avec la valeur d'adaptation la plus élevée étant choisi, ce processus se répète jusqu'à compléter la nouvelle génération. Les individus participant à un tournoi sont soit éliminés soit remis dans la population.

La forme la plus simple de la sélection par tournoi est la sélection par tournoi binaire. Dans cette méthode, deux individus sont choisis au hasard, et le meilleur d'entre eux est introduit dans la nouvelle génération.

Dans la sélection par tournoi, la sévérité de la sélection est fonction du nombre d'individus participant au tournoi q , elle augmente avec l'augmentation de la taille du groupe q : [18]

- Si, $q = 1$ la sélection est aléatoire.
- Si, $q = \mu$ (taille de la population), la sélection est déterministe.

La sélection par tournoi évite les problèmes de convergence prématurée en fournissant un moyen simple et efficace pour le contrôle de la sévérité de la sélection.

Cette méthode de sélection est la plus utilisée dans les Algorithmes Evolutionnaires, car elle présente de nombreux avantages :

- Simple d'implémentation et assez intuitive.
- Aucun calcul supplémentaire n'est exigé (aucune statistique).
- Aucune mise en échelle de la fonction d'adaptation n'est utilisée et les valeurs d'adaptation négatives sont admises
- Peut être utilisée indifféremment pour la minimisation ou la maximisation.
- Méthode idéale pour une implémentation parallèle, plusieurs sélections peuvent être effectuées en même temps.

La compétition par tournoi :

Cet opérateur est une variante de la sélection par tournoi, elle est souvent utilisée dans le cadre de la Programmation Evolutionnaire. Chaque individu de la population est comparé avec q individus choisis aléatoirement, le score de chaque individu est fonction du nombre d'opposants qu'il domine. Les individus ayant les meilleurs scores sont choisis pour être dans la nouvelle génération [17] [18].

Chapitre II :

***LE CALCUL EVOLUTIONNAIRE
DANS L'OPTIMISATION CONTINUE***

II.1. Introduction

Les problèmes d'optimisation continue sont les plus répandus dans le domaine de l'ingénierie, car on cherche souvent à optimiser un comportement, comme minimiser le coût ou maximiser le rendement. Les problèmes d'optimisation peuvent être assez complexes et enveloppent souvent des contraintes et/ou plusieurs objectifs. La forme générale d'un problème d'optimisation est :

$$\begin{aligned} \min \quad & F(X): \mathfrak{R}^n \rightarrow \mathfrak{R}^k \\ \text{s.t} \quad & \\ & X \in S \\ & X \in B \end{aligned} \quad (\text{II-1})$$

Avec :

- $X = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^n$ vecteur des variables objectives de dimension n
- $F = \{f_1, f_2, \dots, f_k\} \in \mathfrak{R}^k$ vecteur des fonctions objectives. Dans le cas où $k = 1$, le problème est dit mono objectif, si non il est dit multi objectifs.
- $S \subseteq \mathfrak{R}^n$ espace de recherche limité par les contraintes paramétriques : $\underline{x}_i \leq x_i \leq \bar{x}_i$ avec : $i \in \{1, 2, \dots, n\}$
- B région de faisabilité définie par l'ensemble des contraintes.

Nous pouvons différencier entre trois principaux types de problèmes d'optimisation :

- Optimisation mono-objective sans contraintes : c'est le cas le plus simple où le domaine de faisabilité de la fonction objective est \mathfrak{R} .
- Optimisation mono-objective avec contraintes : dans ce cas, le domaine de faisabilité des solutions est défini par l'ensemble des contraintes.
- Optimisation multi-objective : dans ce type de problèmes nous sommes confrontés à plusieurs objectifs qui doivent être optimisés simultanément sujet au contraintes.

II.2. Optimisation mono-objective sans contraintes

Un Algorithme Evolutionnaire est, par définition, une méthode d'optimisation. Un Algorithme Evolutionnaire de base permet de résoudre des problèmes d'optimisation simples (mono-objectif sans contraintes). Il suffit, pour cela, de poser la fonction objective comme fonction d'évaluation.

Pour tester l'efficacité d'un Algorithme Evolutionnaire à résoudre de tels problèmes, nous effectuons un ensemble de tests.

Après la convergence de chaque algorithme, la courbe $f(X_{opt})$ qui donne la convergence de l'algorithme est tracée, avec X_{opt} est le meilleur individu trouvé dans chaque génération n .

II.2.1. Teste 1

La fonction suivante est utilisée comme fonction de test :

$$f(X) = \sum_{i=1}^n x_i^2 \quad (\text{II-2})$$

Avec $-5 \leq x_i \leq 5$

Dans le cas où $n = 2$, la fonction est de la forme suivante :

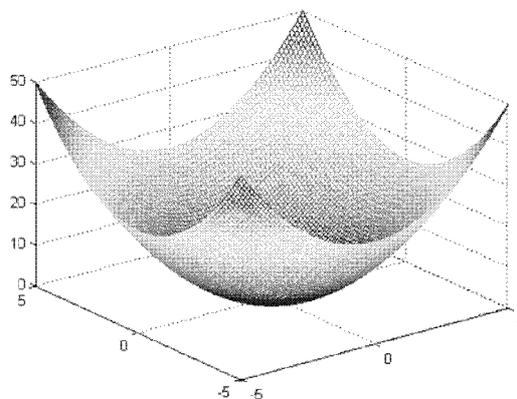


Figure II-1: $F01(X)$ pour $n = 2$

C'est une fonction simple avec un seul optimum : $X^* = [0, 0]$ avec $f(X^*) = 0$.

Pour optimiser cette fonction, nous utilisons les Algorithmes Evolutionnaires suivant :

II.2.1.1. Stratégie d'Evolution

Appliquons les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 20$.
- Taille de la population des descendants : $L = 50$.
- Taille de la population des parents : $m = 10$.
- Une recombinaison intermédiaire.
- Une mutation auto-adaptative suivant la formulation de Schwefel avec $b = 1$; $T = 0.5$
- Une sélection par tournoi avec $q = 5$.

L'évolution de l'algorithme est comme suit :

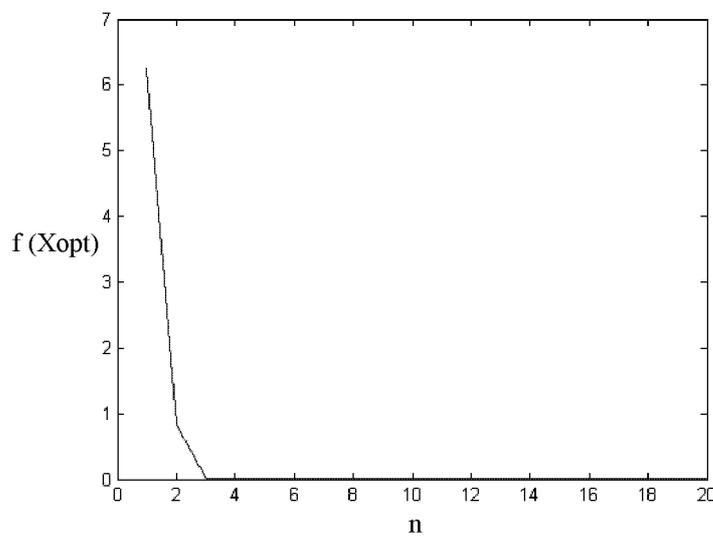


Figure II-2 : Caractéristique de convergence

Après la convergence de l'Algorithme nous obtenons le résultat suivant :

$$X^* = 10^{-3} [0.3265, -0.5215] \quad \text{et} \quad f(X^*) = 1.4460e-007$$

II.2.1.2. Programmation Evolutionnaire

Soient les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 20$.
- Taille de la population des descendants : $L = 40$.
- Taille de la population des parents : $m = 10$.
- Une mutation auto adaptative suivant la formulation de Fogel avec $b = 0.9$.
- Une compétition par tournoi avec $q = 5$.

L'évolution de la convergence et comme suit :

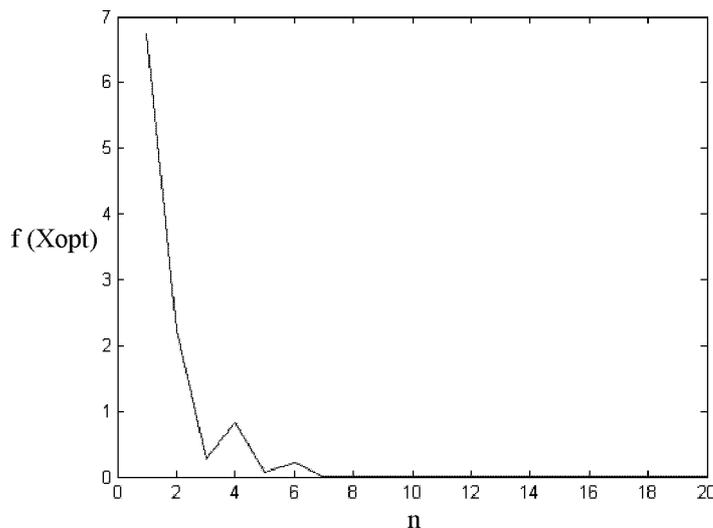


Figure II-3 : Caractéristique de convergence

Le résultat obtenu après l'exécution de l'algorithme est comme suit :

$$X^* = 10^{-3} [-0.2739, 0.3417] \quad \text{et} \quad f(X^*) = 3.3923e-007$$

Cette fonction ne pose aucune difficulté particulière, les deux algorithmes ont convergé vers l'optimum en quelques itérations seulement.

II.2.2. Teste 2

Considérons le problème suivant :

$$\text{Min } F02(X) = 10 * n + \sum_{i=1}^n ((a_i x_i)^2 - 10 \cos(2\pi x_i)) \quad (\text{II-3})$$

Avec : $-5.12 \leq x_i \leq 5.12$ et $a_i = 1$.

Nous voulons trouver le minimum global de cette fonction pour une dimension de $n = 5$. Cette fonction présente un optimum global à $X = 0$ avec $F02(X) = 0$.

Pour avoir une idée sur la forme de cette fonction, traçons la fonction pour $n = 1$ et $n = 2$:

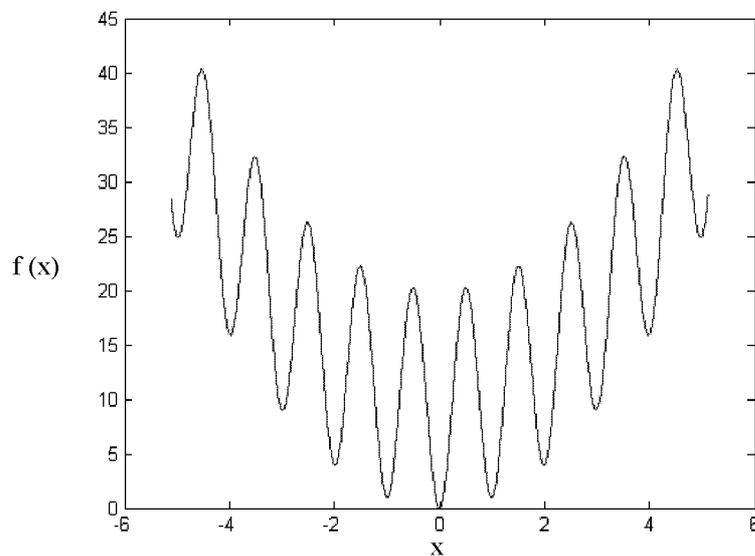


Figure II-4 : $F02(X)$ pour $n = 1$

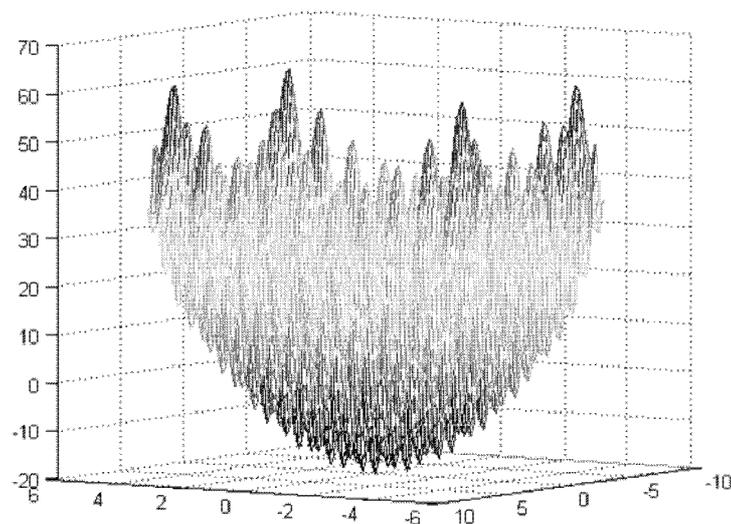


Figure II-5 : $F02(X)$ pour $n = 2$

La particularité de cette fonction est que l'optimum global est entouré d'un ensemble d'optimums locaux. La difficulté réside essentiellement dans le fait que les optimums locaux les plus susceptibles à une convergence prématurée sont les plus proches de l'optimum global. Dans le cas ou $n = 5$, l'optimum global est entouré de 10 optima locaux, ce qui nous oblige de bien calibrer les différents paramètres de nos Algorithmes.

L'optimum global de cette fonction est obtenu pour $X = [0]$ avec $F02(X) = 0$

Pour assurer une bonne convergence il faudrait avoir :

- Une mutation qui peut couvrir l'ensemble de l'espace de recherche.
- Une sévérité de sélection modérée.

II.2.2.1. Stratégies d'Evolution

Appliquons les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 200$.
- Taille de la population des parents : $m = 100$.
- Un croisement uniforme.
- Une mutation auto adaptative suivant la formulation de Schwefel avec $b = 0.9$ et $T = 0.75$.
- Une sélection par tournoi $q = 5$.

La convergence de notre algorithme est comme suit

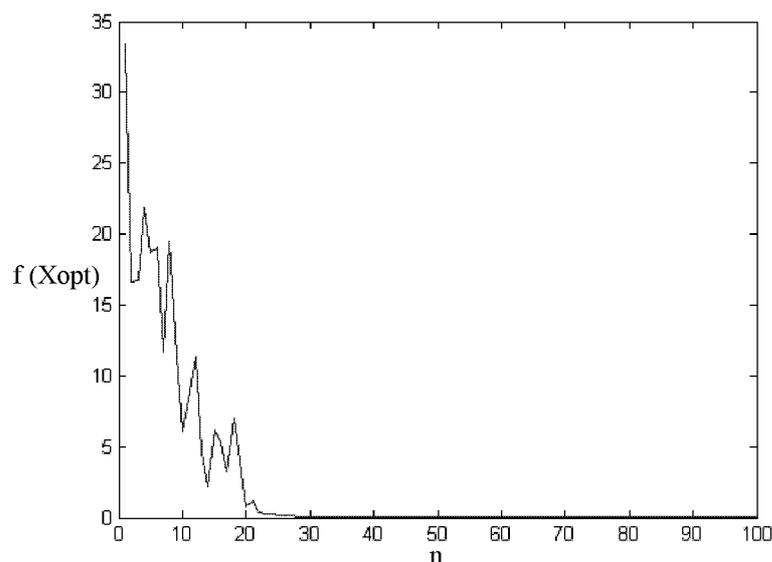


Figure II-6 : Caractéristique de convergence

Nous trouvons les résultats suivants :

$$X^* = 10^{-3} [-0.3086, 0.1629, -0.2136, -0.3997, -0.0398] \text{ et } f(X^*) = 6.5205 \cdot 10^{-5}$$

II.2.2.2. Programmation Evolutionnaire

Nous utilisons un Programme Evolutionnaire avec les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 250$.
- Taille de la population des parents : $m = 100$.
- Une mutation auto-adaptative suivant la formulation de Fogel avec $b = 10$.
- Une sélection déterministe.

L'évolution de l'Algorithme est comme suit :

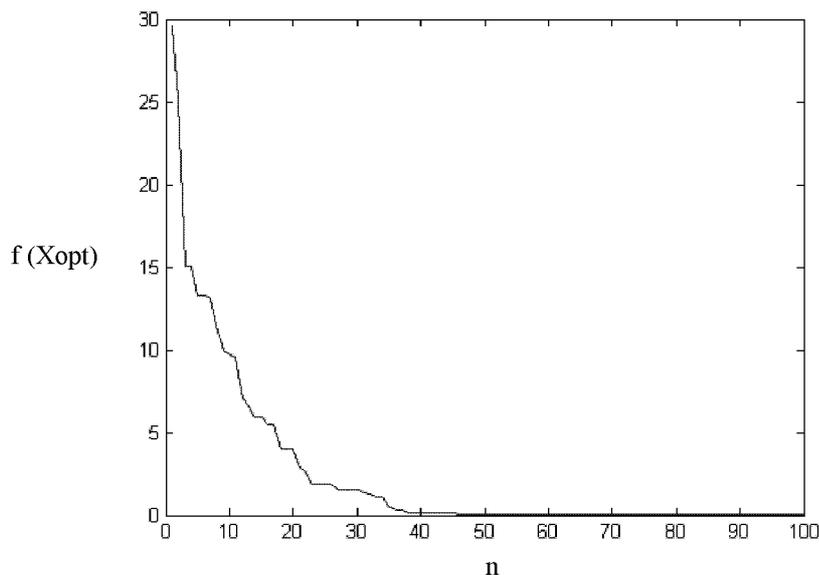


Figure II-7 : Caractéristique de convergence

Les résultats trouvés se traduisent par :

$$X^* = 1.0e-003 * [-0.1744 \quad -0.0259 \quad -0.1661 \quad 0.1682 \quad 0.0001] \text{ et } f(X^*) = 1.7253e-005$$

Nous pouvons constater que les deux algorithmes ont évité les optimums locaux qui entourent l'optimum global, cette caractéristique est due essentiellement à la mutation qui perturbe les solutions obtenues et évite ainsi toute convergence prématurée.

II.2.3. Remarques

Après les tests effectués, nous pouvons conclure que :

Les Algorithmes Evolutionnaires assurent la convergence vers l'optimum global même en présence d'optima locaux, contrairement aux méthodes conventionnelles qui n'assurent la convergence que vers l'optimum local le plus proche.

En comparant les comportements des Stratégies d'Evolution et de la Programmation Evolutionnaire, nous constatons que la Programmation Evolutionnaire, en se basant sur la mutation seule, doit utiliser des populations plus grandes pour bien explorer l'espace de recherche, ce qui la pénalise d'un temps d'exécution plus long.

II.3. L'optimisation avec Contraintes

Les problèmes physiques enveloppent souvent des contraintes qui doivent être respectées par les solutions, l'ensemble des contraintes définissent le domaine de faisabilité, une solution qui viole une ou plusieurs contraintes est dite non faisable et elle ne peut pas être considérée comme une solution adéquate au problème, même si, elle optimise la fonction objective.

Dans le domaine du calcul évolutionnaire les contraintes peuvent être traitées par : [19] [20]

- L'élimination de toutes les solutions qui ne respectent pas les contraintes ou imposer aux opérateurs de ne créer que des solutions faisables.
- L'utilisation des fonctions de pénalités qui pénalisent les solutions qui violent les contraintes.
- La transformation du problème en un problème d'optimisation multi-objective, en reformulant les contraintes les plus prioritaires en des fonctions objectives supplémentaires.

II.3.1. Les fonctions de pénalités

Les fonctions de pénalité introduisent une mesure du degré de satisfaction des contraintes. L'introduction des pénalités permet de transformer le problème d'optimisation avec contraintes en un problème d'optimisation sans contraintes, plus facile à traiter.

La formulation générale d'un problème d'optimisation avec contraintes est comme suit :

$$\begin{aligned} \min \quad & f(x) \\ \text{st} \quad & \\ & x \in A \\ & x \in B \end{aligned} \tag{II-4}$$

Où : x est le vecteur objectif.

Les contraintes $x \in A$ sont considérées comme des contraintes paramétriques qui définissent le domaine de recherche.

Les contraintes $x \in B$ sont des contraintes dures, difficiles à satisfaire, elles déterminent la faisabilité de chaque solution. Ces contraintes sont exprimées en contraintes d'égalités et contraintes d'inégalités.

En introduisant une fonction de pénalité le problème peut être reformulé comme suit :

$$\begin{aligned} \min \quad & f(x) + p(d(x, B)) \\ \text{st} \quad & \\ & x \in A \end{aligned} \tag{II-5}$$

Avec :

- $d(x, B)$ est une fonction décrivant la distance de la solution x du domaine de faisabilité B .
- $p()$ est la fonction de pénalité monotone non décroissante avec $p(0) = 0$.

Si une solution x est optimale pour la formulation (II-4), elle l'est pour la formulation (II-5), réciproquement, si une solution est optimale pour (II-5) elle l'est pour (II-4).

Il existe plusieurs formes possibles de fonctions de pénalités, le choix d'une forme particulière dépend de la nature et de la complexité du problème à traiter.

II.3.1.1. Les fonctions de pénalités statiques

Une fonction de pénalité statique impose une sévérité de pénalité constante sur les solutions qui violent le domaine de faisabilité. Cette pénalité est fonction soit du nombre de contraintes violées, soit en fonction de la distance de la région de faisabilité : [19]

Une fonction de pénalité statique en fonction du nombre de contraintes violées peut être formulée comme suit :

$$p(X) = \sum_{i=1}^m C_i \delta_i \quad \text{où} \begin{cases} \delta_i = 1 & \text{si la contrainte } i \text{ est violée} \\ \delta_i = 0 & \text{si la contrainte } i \text{ est satisfaite} \end{cases} \quad (\text{II-6})$$

Où : C_i est une constante déterminant la sévérité de la pénalité imposée.

La forme de pénalité statique la plus utilisée introduit une pénalité en fonction de la distance de la région de faisabilité :

$$p(X) = \sum_{i=1}^m C_i d_i^k \quad \text{où} \quad d_i = \begin{cases} \delta_i g_i(x) & \text{pour } i = 1, \dots, q \\ |h_i(x)| & \text{pour } i = q+1, \dots, m \end{cases} \quad (\text{II-7})$$

Où :

- C_i est un paramètre déterminant la sévérité de la pénalité.
- $d()$ est la distance euclidienne entre la solution x et de la contrainte i , $k=1 \div 2$.

Les contraintes $1, \dots, q$ sont des contraintes d'inégalités et les contraintes $q+1, \dots, m$ sont des contraintes d'égalité.

Les fonctions de pénalités statiques présentent un inconvénient majeur : la sévérité de la pénalité est introduite par un coefficient constant C_i , qui doit être déterminé pour chaque problème.

La sévérité de la pénalité doit être finement posée car :

- Si C_i est trop élevé les solutions finales seront faisables mais pas optimales.
- Si C_i est trop faible les solutions finales seront optimales mais pas faisables.

II.3.1.2. Les fonctions de pénalités dynamiques

Dans les problèmes d'optimisations avec contraintes, il existe souvent un conflit entre permettre l'exploration des régions infaisables et l'exigence d'une solution finale faisable. Les fonctions de pénalités dynamiques permettent d'augmenter la sévérité de la pénalité imposée avec la progression de la recherche. [19]

L'intérêt de ces fonctions est évident lorsque les régions faisables sont éparpillées et séparées par des régions infaisables. Dans ce cas, imposer une faible sévérité au début de la recherche permet une bonne exploration de l'espace. L'augmentation de la sévérité au cours de la recherche permet de la diriger vers les régions faisables et garantir, ainsi, une pleine satisfaction des contraintes par les solutions finales.

La forme générale d'une fonction de pénalité dynamique basée sur la mesure de la distance de la région de faisabilité peut être décrite comme suit : [19]

$$p(X,t) = \sum_{i=1}^m s_i(t) d_i^k \quad \text{où } d_i = \begin{cases} \delta_i g_i(x) & \text{pour } i = 1, \dots, q \\ |h_i(x)| & \text{pour } i = q+1, \dots, m \end{cases} \quad (\text{II-8})$$

Où : $s_i(t)$ est une fonction monotone non décroissante, t peut être considéré comme le nombre de génération. $s_i()$ peut être comme suit : $s_i(t) = (C_i t)^\alpha$ avec $\alpha = 1 \div 2$.

II.3.1.3. Fonctions de pénalités adaptatives

Les fonctions de pénalités dynamiques sont généralement efficaces, mais la recherche peut être mieux affinée en incorporant un aspect adaptatif à la fonction de la pénalité. Les fonctions de pénalités adaptatives, permettent d'ajuster la sévérité de la pénalité imposée en fonction de la faisabilité des solutions des générations précédentes. [19]

Un exemple d'une fonction de pénalité adaptative : [19]

$$p(X,t) = \sum_{i=1}^m \lambda_i d_i^k \quad (\text{II-9})$$

Avec :

$$\lambda_{t+1} = \begin{cases} \lambda_t \beta_1 & \text{si la meilleure solution des } N_f \text{ générations précédentes est infaisable} \\ \lambda_t / \beta_2 & \text{si la meilleure solution des } N_f \text{ générations précédentes est faisable} \\ \lambda_t & \text{sinon} \end{cases}$$

Cette méthode utilise deux constantes β_1 et β_2 avec $\beta_1 > \beta_2 > 1$ pour adapter la pénalité appliquée, et utilise l'évaluation de la faisabilité des meilleures solutions trouvées dans un intervalle de N_f générations. La pénalité est mise à jour chaque N_f génération.

Un autre exemple de fonction de pénalité adaptative est comme suit : [19]

$$p(X,t) = (F_{\text{fea}}(t) - F_{\text{all}}(t)) \sum_{i=1}^m \left(\frac{d_i}{NFT_i} \right) \quad (\text{II-10})$$

Cette fonction utilise une mesure de la sévérité des contraintes, ainsi qu'une estimation de la proximité de la région faisable.

Avec :

- $F_{all}(t)$ est la valeur non pénalisée de la meilleure solution trouvée, $F_{feas}(t)$ note la valeur de la meilleur solution faisable trouvée, les termes $F_{all}(t)$ et $F_{feas}(t)$ fournissent un étalonnage adaptatif de la pénalité basée sur les résultats de la recherche.
- Le terme NFT_i est une mesure de la proximité de la région faisable, peut être défini comme :

$$NFT_i = \frac{NFT_{0i}}{1 + \Lambda_i} \quad (II-11)$$

- NFT_{0i} est la limite supérieure de NFT_i . Λ_i est un paramètre de recherche dynamique utilisé pour ajuster le terme NFT_i basé sur l'historique de la recherche.

Cette fonction de pénalité incite l'algorithme évolutionnaire à explorer la région faisable et ça périphérie, mais l'interdit de rechercher plus loin.

Pour tester le comportement de nos Algorithmes Evolutionnaire dans le cas de problème d'optimisations avec contraintes, pour cela nous utilisons des fonctions de pénalité dynamiques conjointement avec les Algorithmes Evolutionnaires.

II.3.2. Test 1

La fonction suivante est utilisée comme fonction de test :

$$\begin{aligned} \min \quad & f(x) = (x(1) - 10)^3 + (x(2) - 20)^3 \\ \text{st} \quad & \\ & g1 = -(x(1) - 5)^2 - x(2) - 5)^2 + 100 \leq 0 \\ & g2 = (x(1) - 6)^2 + (x(2) - 5)^2 - 82.81 \leq 0 \end{aligned} \quad (II-12)$$

Avec : $0 \leq x(i) \leq 50$, $i = 1, 2$

La fonction objective est polynomiale mais non quadratique, les contraintes sont des fonctions quadratiques. Cette fonction aura posée de difficulté à la programmation non linéaire vu la forme de la fonction objective et des contraintes.

La solution optimale à ce problème est : [21]

$$X^* = (14.095, 0.842) \quad f(X^*) = -6961.813$$

Pour optimiser cette fonction, nous utilisons une fonction de pénalité dynamique de type :

$$p(t) = s(t) * \sum_{i=1}^m \max(0, g_i)^2 \quad (II-13)$$

Les Algorithmes Evolutionnaires suivants sont utilisés :

II.3.2.1. Les Stratégies d'Evolution

En utilisant les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 200$.
- Taille de la population des parents : $m = 50$.
- Une recombinaison intermédiaire.
- Une mutation auto adaptative suivant la formulation de Schwefel avec $b = 1$; $T = 0.5$.
- Une sélection par tournoi $q = 10$.

Le déroulement de l'algorithme est comme suit :

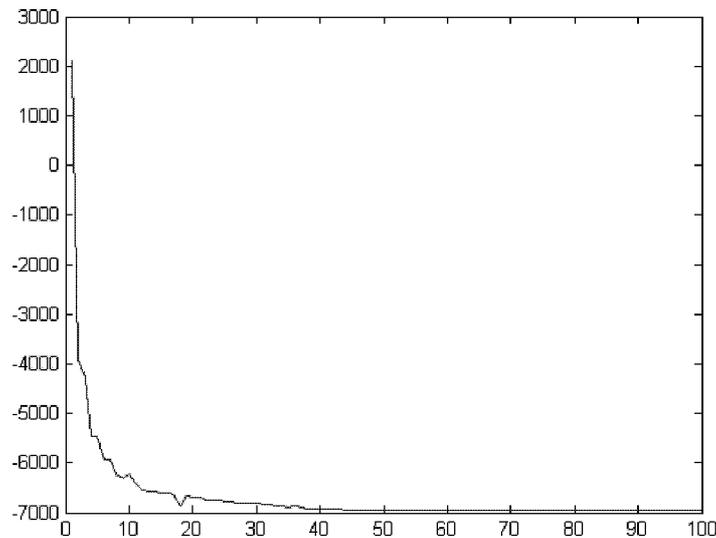


Figure II-8 : *Caractéristique de convergence*

Les résultats obtenus sont comme suit :

$$X^* = 14.0945 \quad 0.8418$$

Avec

$$f(X^*) = -6963.1$$

$$g1(X^*) = 4.9586 \cdot 10^{-5}$$

$$g2(X^*) = 5.9468 \cdot 10^{-5}$$

Nous pouvons remarquer que les deux contraintes sont actives, mais comme leurs valeurs sont proches de zéro, nous pouvons considérer que les contraintes sont satisfaites.

II.3.2.2. Programmation Evolutionnaire

En utilisant les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 200$.
- Taille de la population des parents : $m = 50$.
- Une recombinaison intermédiaire.
- Une mutation auto adaptative suivant la formulation de Schwefel avec $b = 1$; $T = 0.75$.
- Une sélection par tournoi $q = 10$.

Le déroulement de l'algorithme est comme suit :

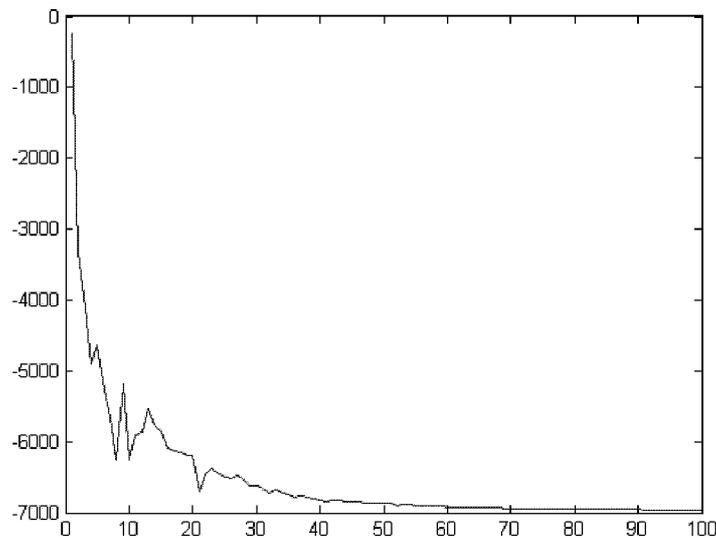


Figure II-9 : Caractéristique de convergence

La solution trouvée est :

$$X = 14.0945 \quad 0.8419$$

Avec :

$$f(X) = -6961.8$$

$$g1(X) = -6.5616 \cdot 10^{-6}$$

$$g2(X) = 6.5616 \cdot 10^{-6}$$

Nous constatons que les deux algorithmes ont convergé vers la solution optimale du problème toute en préservant les contraintes.

II.3.3. Test 2

Considérant le problème suivant : [21]

Minimiser :

$$f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

Sujet à :

$$\begin{aligned} g_1(x) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\ g_2(x) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(x) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_5(x) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_6(x) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(x) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\ g_8(x) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \end{aligned} \tag{II-14}$$

Avec :

$$-10 \leq x_i \leq 10, \quad (i = 1, \dots, 10)$$

La fonction objective est une fonction quadratique ainsi que les contraintes.

Le résultat rapporté par [21] est :

$$X^* = [2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927]$$

$$\text{Avec : } f(X) = 24.3062091$$

Pour ce problème nous utilisons la fonction de pénalité suivante :

$$p(t) = s(t) * \sum_{i=1}^m \max(0, g_i)^2 \tag{II-15}$$

Pour la résolution de ce problème nous utilisons les algorithmes suivants :

II.3.3.1. Stratégie d'Evolution

Nous utilisons cette stratégie avec les paramètres suivant :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 200$.
- Taille de la population des parents : $m = 40$.
- Une recombinaison intermédiaire.
- Une mutation auto adaptative suivant la formulation de Schwefel avec $b = 1$; $T = 0.9$.
- Une sélection par tournoi $q = 10$.

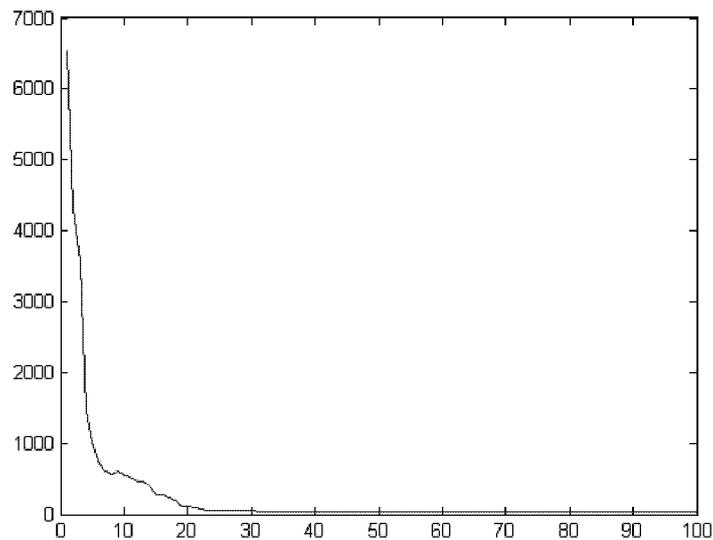


Figure II-10 : Caractéristique de convergence

Après l'exécution de l'algorithme nous obtenons les résultats suivants :

$x = [2.1879, 2.3383, 8.7377, 5.0707, 0.8134, 1.0346, 1.3445, 9.8412, 8.2605, 8.2454]$

Avec :

$f = 24.7686$

Et

$g = [-0.0193, -0.0013, -0.0147, -0.9413, -0.0054, -0.0360, -6.6370, -49.4369]$

II.3.3.2. La Programmation Evolutionnaire

Nous utilisons cette stratégie avec les paramètres suivant

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des descendants : $L = 200$.
- Taille de la population des parents : $m = 100$.
- Une mutation auto adaptative suivant la formulation de Fogel avec $b = 1$.
- Une compétition par tournoi $q = 25$.

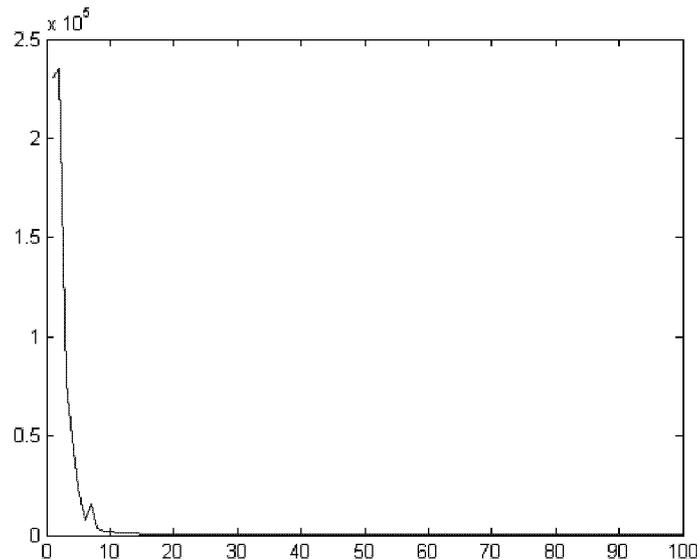


Figure II-11 : Caractéristique de convergence

Les résultats obtenus sont :

$$x = [2.1345, 2.3398, 8.7631, 5.0931, 1.0272, 1.5447, 1.3144, 9.8557, 8.2006, 8.3164].$$

Avec :

$$f = 24.7670$$

$$g = [-0.0045, -0.0059, -0.0264, -0.2686, -1.0521, -0.0896, -5.6649, -50.0967]$$

Les solutions obtenues par les deux algorithmes sont très proches de la solution optimale, et toutes les contraintes sont satisfaites.

II.3.4. Remarques

Suite à ces testes nous pouvons constater que :

Les Algorithmes Evolutionnaires ne se limitent pas à une formulation spécifique. Ils n'ont aucune restriction sur la forme de l'objectif ou des contraintes, ils peuvent être appliqués à tous les problèmes d'optimisation. En plus la convergence vers l'optimum global est généralement assurée. Le Calcul Evolutionnaire constitue, donc, un outil puissant pour l'optimisation des fonctions fortement non-linéaires multimodales.

II.4. Optimisation multi-objectifs

Les problèmes réels invoquent souvent de multiples mesures de performance ou objectifs, qui doivent être optimisés simultanément. En pratique, ceci n'est pas toujours possible car les objectifs peuvent être conflictuels, du fait qu'ils mesurent différents aspects de la qualité de la solution. Dans ce cas, la qualité d'un individu est décrite non pas par un scalaire mais par un vecteur. La performance, la fiabilité et le coût sont des exemples d'objectifs conflictuels [22].

La forme générale d'un problème d'optimisation multi-objectifs est :

$$\begin{aligned} \min \quad & F(X) = \{f_1(X), f_2(X), \dots, f_k(X)\} : \mathbb{R}^n \rightarrow \mathbb{R}^k \\ \text{s.t.} \quad & X \in S \end{aligned} \quad (\text{II-16})$$

Les problèmes multi-objectifs n'ont pas une seule solution mais un ensemble de solutions dites solutions de compromis. Généralement, il n'existe aucune solution pour laquelle tous les objectifs sont optimaux, dû au fait que les objectifs sont, souvent, conflictuels. Un produit, par exemple, ne peut pas être très performant au plus bas prix. Dans ce cas, plusieurs compromis sont possibles et souvent c'est le facteur humain qui décide de l'optimalité d'une solution [22].

Une solution est considérée comme optimale si aucun objectif ne peut être amélioré sans dégrader les autres objectifs, c'est la définition de l'optimalité au sens de Pareto. La notion de Pareto optimalité est définie en terme de dominance : considérant un problème de minimisation couvrant k objectifs, on dit qu'une solution $Y = [y_1, y_2, \dots, y_k]$ domine une solution $Z = [z_1, z_2, \dots, z_k]$ si et seulement si : [22]

$$\forall i \quad y_i \leq z_i \quad \text{et} \quad \exists i \quad y_i < z_i \quad \text{On note} \quad Y \prec Z \quad (\text{II-17})$$

Si, une solution y n'est dominée par aucune autre solution, elle est dite non dominée ou optimale dans le sens de Pareto (Pareto optimale). Deux solutions non dominées sont non comparables (Y ne domine pas Z et Z ne domine pas Y). L'ensemble des solutions non dominées constitue le front de Pareto (l'ensemble des solutions Pareto optimales). Le problème d'optimisation multi-objectifs peut être vu comme la détermination de l'ensemble des solutions Pareto optimales.

En pratique une seule solution de compromis est nécessaire. L'optimisation multi-objectifs est, donc, plus qu'une simple détermination de l'ensemble des solutions Pareto optimales. Même si, une solution Pareto optimale est toujours meilleure que n'importe quelle solution qu'elle domine (non Pareto optimale), il reste que les solutions Pareto optimales ne sont pas toutes de bonnes solutions de compromis.

Les Algorithmes Evolutionnaires se basent sur une formulation scalaire de l'adaptation, cette transformation s'effectue par une transformation monotone des objectifs. Une fois cette mesure scalaire de qualité établie, l'algorithme évolutionnaire procède par la reproduction et la sélection comme de coutume.

Dans le cadre des Algorithmes Evolutionnaires, l'assignement d'une adaptation à une solution suivant un nombre fini d'objectifs, peut se faire selon plusieurs approches : [22] [23] [24]

- Les approches pleine agrégation : les objectifs sont numériquement combinés en une seule fonction à optimiser.
- L'optimisation vectorielle non Pareto : les différents objectifs affectent la sélection des individus une à une.
- L'optimisation vectorielle basée sur la définition de la dominance ou l'optimisation au sens de Pareto.

II.4.1. Approche pleine agrégation

Il existe une multitude de méthodes pour combiner les différents objectifs en une seule fonction à optimiser comme une fonction mono-objectif [22] [23].

II.4.1.1. Somme pondérée

Les n objectifs f_1, \dots, f_n sont pondérés par des coefficients w_1, \dots, w_n définis par l'utilisateur et sommés pour obtenir une mesure scalaire de l'adaptation de chaque individu.

Cette approche est intuitive et facile à implémenter : [22]

$$\begin{aligned} \phi : \mathcal{R}^n &\rightarrow \mathcal{R} \\ f(a_i) &\mapsto \sum w_k f_k(a_i) \end{aligned} \tag{II-18}$$

Les coefficients w_k dépendent du problème en main. La combinaison des poids doit être finement posée pour avoir une solution finale satisfaisante.

II.4.1.2. Approche min max

Cette approche est implémentée comme la minimisation du maximum de la différence entre les objectifs f_1, \dots, f_n et les cibles g_1, \dots, g_n spécifiées par l'utilisateur :

Cette approche peut être formulée comme suit : [22]

$$\begin{aligned} \Phi : \mathfrak{R}^n &\rightarrow \mathfrak{R} \\ f(a_i) &\rightarrow \max_{k=1, \dots, n} \left(\frac{f_k(a_i) - g_k}{w_k} \right) \end{aligned} \quad (\text{II-19})$$

Les paramètres g_k sont les cibles que les objectifs veulent approcher. Les cibles sont soit les valeurs des performances désirées ou les valeurs connues à priori comme inatteignables. Les poids w_k indiquent la direction de recherche dans l'espace des objectifs, souvent posés comme les valeurs absolues des cibles.

II.4.1.3. Approche du vecteur cible

Cette approche consiste à la minimisation de la distance entre le vecteur objectif $f = (f_1, \dots, f_n)$ et un vecteur cible prédéfini $g = (g_1, \dots, g_n)$ suivant une mesure adéquate de distance : [22] [23]

$$\begin{aligned} \Phi : \mathfrak{R}^n &\rightarrow \mathfrak{R} \\ f(a_i) &\mapsto \|[f(a_i) - g]W^{-1}\|_{\alpha} \end{aligned} \quad (\text{II-20})$$

II.4.2. Optimisation vectorielle non Pareto

Dans cette approche, les objectifs ne sont pas combinés en un seul, mais la sélection se fait sur la base du vecteur comprenant tous les objectifs en même temps, l'évaluation de l'adaptation de chaque vecteur peut se faire suivant plusieurs approches, nous citons :

II.4.2.1. Approche lexicographique

Différentes priorités sont assignées aux objectifs suivant leurs importances. Les objectifs les plus prioritaires sont utilisés en premier lieu lorsque les individus sont comparés. Les litiges sont résolus en comparant le deuxième objectif le plus prioritaire et ainsi de suite [22] [23].

$$\begin{aligned} \Phi : \mathfrak{R}^n &\rightarrow \{0, 1, \dots, \mu - 1\} \\ f(a_i) &\mapsto \sum_{j=1}^n I(f(a_j)^\ell < f(a_i)) \end{aligned} \quad (\text{II-21})$$

où I désigne la condition

$$f(a_j)^\ell < f(a_i) \Leftrightarrow \exists p \in \{1, \dots, n\} : \forall k \in \{p, \dots, n\} \quad f_k(a_j) \leq f_k(a_i) \wedge f_p(a_j) < f_p(a_i) \quad (\text{II-22})$$

Les objectifs f_1, \dots, f_n sont triés en priorité croissante. La solution finale dépend essentiellement de la manière dont les priorités sont assignées. Dans le cas où chaque objectif admet un optimum, la solution finale tend vers l'optimum de l'objectif le plus prioritaire.

II.4.3. Optimisation vectorielle au sens de Pareto

Le concept de dominance de Pareto est utilisé pour ranger la population de telle façon que toutes les solutions non dominées auront les valeurs d'adaptation les plus élevées.

Les deux approches les plus connues, sont : [22]

- i) Dans l'approche proposée par Goldberg, toutes les solutions non dominées se voient assigner la même valeur d'adaptation.

$$\begin{aligned} \Phi : \mathcal{R}^n &\rightarrow \{1, 2, \dots, \mu\} \\ f(a_i) &\mapsto \begin{cases} 1 \leftarrow \neg [f(a_j)p < f(a_i)] \forall j \in \{1, \dots, \mu\} \\ \phi \leftarrow \neg [f(a_j)p < f(a_i)] \forall j \in \{1, \dots, \mu\} \end{cases} \end{aligned} \quad (\text{II-22})$$

\neg Not la négation logique

- ii) une autre approche proposée par Fensoca et Flaming où les individus se voient assigner un coût suivant le nombre d'individus qu'il dominant.

$$\begin{aligned} \Phi : \mathcal{R}^n &\rightarrow \{0, 1, \dots, \mu - 1\} \\ f(a_j) &\mapsto \sum_{j=1}^{\mu} I(f(a_j)p < f(a_i)) \end{aligned} \quad (\text{II-23})$$

Avec :

$$f(a_j)p < f(a_i) \Leftrightarrow \forall k \in \{1, \dots, n\} \quad f_k(a_j) \leq f_k(a_i) \wedge \exists k \in \{1, \dots, n\} : f_k(a_j) < f_k(a_i) \quad (\text{II-24})$$

Goldberg note l'utilité des méthodes des niches dans le cadre de l'optimisation multi-objectifs par leurs capacités à maintenir les solutions optimales sur le front de Pareto, ce qui permet d'avoir l'ensemble des solutions non dominées.

Les méthodes des niches étendent l'Algorithme Evolutionnaire aux domaines qui requièrent la localisation et le maintien de solutions multiples, utilisées essentiellement dans les problèmes d'optimisation multi-objectifs, d'apprentissage et de simulation de systèmes adaptatifs. Les deux méthodes de niches les plus connues sont : le partage de l'adaptation (sharing) et le crowding.

II.4.4. Le Crowding

La technique du crowding introduite par De-Jong insère de nouveaux éléments dans la population en remplaçant les éléments similaires. Cette similitude est exprimée en terme de distance Euclidienne. En remplaçant les éléments similaires, le crowding s'efforce de maintenir la diversité préexistante dans la population. Ce remplacement peut être déterministe ou stochastique. Pour minimiser les erreurs de remplacement et garantir ainsi la convergence de l'algorithme le crowding déterministe est utilisé [25] [26].

Un Algorithme Evolutionnaire utilisant un crowding déterministe, peut être décrit comme suit :

Input : le nombre de génération g la taille de la population μ
 $t = 0$;
 $P(t) = \text{initialise}(\mu)$;
for $t = 1$ **to** g **do**
 $P(t) = \text{Shuffle}(P(t-1))$
 for $i = 0$ **to** $\mu/2 - 1$ **do**
 $P_1 = a_{2i+1}(t)$;
 $P_2 = a_{2i+2}(t)$;
 $\{C_1, C_2\} = \text{crossover}(P_1, P_2)$;
 $C'_1 = \text{mutation}(C_1)$;
 $C'_2 = \text{mutation}(C_2)$;
 if $[d(P_1, C'_1) + d(P_2, C'_2)] \leq [d(P_1, C'_2) + d(P_2, C'_1)]$ **then**
 if $F(C'_1) > F(P_1)$ **then** $a_{2i+1}(t) = C'_1$ **end**
 if $F(C'_2) > F(P_2)$ **then** $a_{2i+2}(t) = C'_2$ **end**
 else
 if $F(C'_1) > F(P_1)$ **then** $a_{2i+1}(t) = C'_1$ **end**
 if $F(C'_2) > F(P_2)$ **then** $a_{2i+2}(t) = C'_2$ **end**
 end
 end
end
Output: $P(g)$ la population finale

Cet algorithme procède comme suit :

- En premier lieu, tous les éléments de la population sont groupés en $\mu/2$ paires.
- Tous les paires sont croisées et les descendants mutés.
- chaque descendant entre en compétition avec un de ses parents, pour chaque paire de descendants, deux ensembles de tournois parents – descendants sont possibles
- dans le crowding déterministe, un descendant remplace son parent le plus proche en terme de distance Euclidienne s'il est meilleur que lui.

Le crowding utilise une sélection locale : chaque descendant est comparé avec son parent et l'individu dominant est considéré comme meilleur. L'utilisation d'une sélection locale combinée au concept de dominance permet de stabiliser les solutions sur le front de Pareto, car arriver à ce front, la recherche stagne du fait que par définition les solutions dominantes sont incomparables entre elles.

II.4.5. Test

Pour pouvoir observer le comportement des Algorithmes Evolutionnaires face à un problème d'optimisation multi-objectifs, l'exemple suivant, est utilisé :

Le problème consiste en l'optimisation de deux objectifs f_1 et f_2 avec

$$\begin{aligned} f_1 &= -5(x-1)^2 + 5 \\ f_2 &= -6(x-1.5)^2 + 4 \end{aligned} \quad (\text{II-25})$$

La figure suivante représente les deux fonctions pour $X = [0, 2.5]$

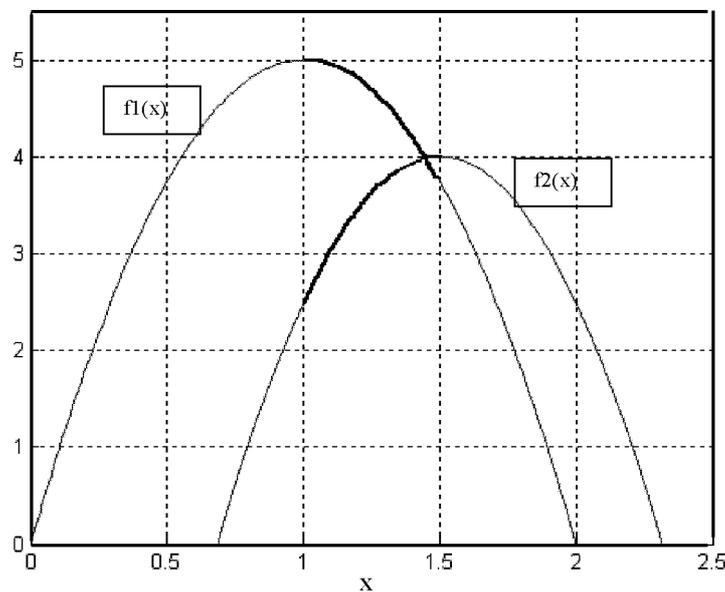


Figure II-12 : L'allure des fonctions utilisées

Suivant la définition de l'optimalité au sens de Pareto, la zone entre $X = 1$ et $X = 1.5$ constitue le front de Pareto. Nous pouvons remarquer que dans cette zone, l'amélioration d'un objectif passe par la détérioration de l'autre. En dehors de cette zone, les deux objectifs peuvent être optimisés simultanément.

Le front de Pareto peut être visualisé en traçant l'ensemble des solutions du problème.

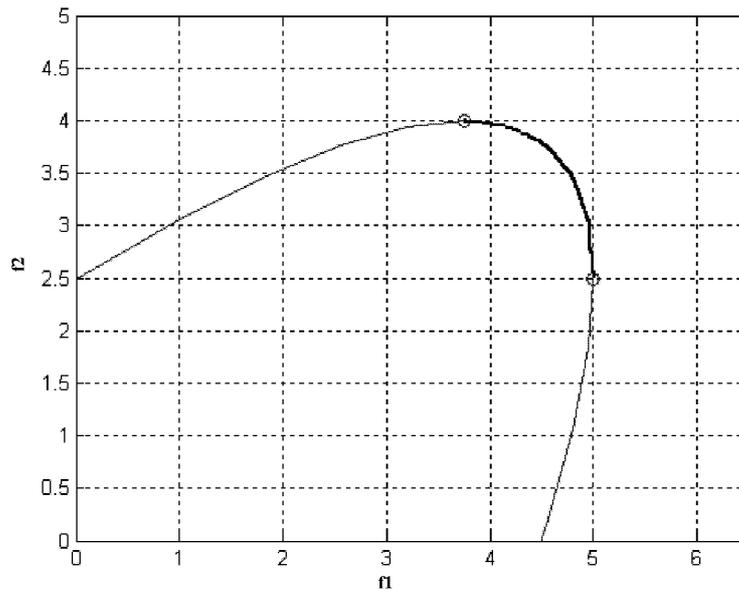


Figure II-13 : Le front de Pareto

Pour résoudre un tel problème, nous utilisons des Algorithmes Evolutionnaires combinés avec deux approches.

- Dans l'approche pleine agrégation, le vecteur cible est utilisé pour trouver une solution de compromis.
- En utilisant le concept d'optimisation au sens de Pareto, le crowding déterministe est associé à l'approche de Goldberg (II-22) pour déterminer le front de Pareto.

II.4.5.1. Approche pleine agrégation

Pour la résolution de notre problème, nous utilisons la méthode du vecteur cible. Nous associons à cette méthode une mesure Euclidienne de distance et un vecteur cible défini par l'optimum de chaque objectif, ce qui donne :

$$fit(a_i) = \sqrt{\sum_{j=1}^k \left(\frac{f_j(a_i) - \bar{f}_j}{\bar{f}_j} \right)^2}, \text{ Avec } \bar{f}_1 = 5 \text{ et } \bar{f}_2 = 4 \quad (\text{II-26})$$

Stratégies d'Evolution

Utilisons une Stratégie d'Evolution SE (m, L) avec les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 50$.
- Taille de la population des descendants : $L = 50$.
- Taille de la population des parents : $m = 10$.
- Une recombinaison intermédiaire comme croisement.
- Une mutation auto adaptative suivant la formulation de Schwefel avec $b = 1$; $T = 0.5$.
- Une sélection par tournoi $q = 5$.

Après exécution nous obtenons le résultat suivant :

$X^* = 1.2999$ avec : $f_1(X^*) = 4.5503$ et $f_2(X^*) = 3.7597$

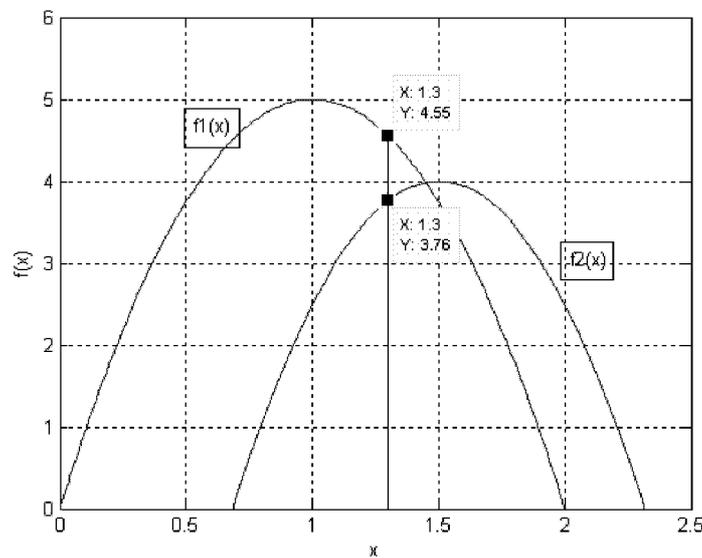


Figure II-14 : Solution obtenue

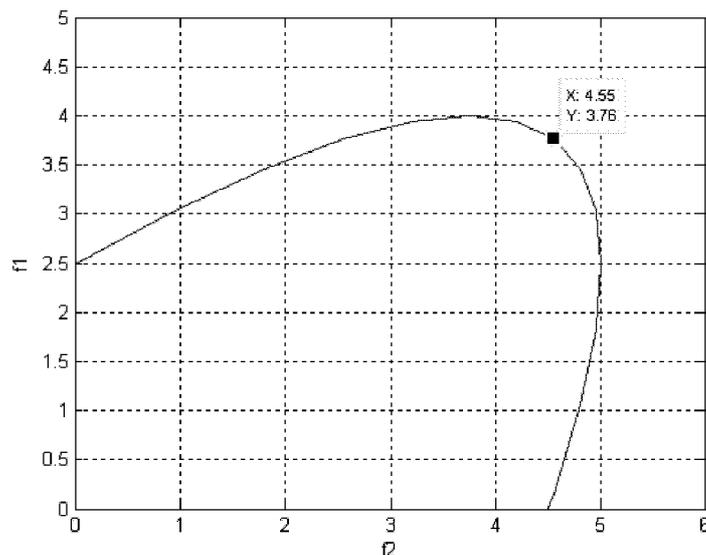


Figure II-15 : Représentation de la solution obtenue sur le front de Pareto

Programmation Evolutionnaire

Utilisons un Programme Evolutionnaire avec les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 20$.
- Taille de la population des descendants : $L = 50$.
- Taille de la population des parents : $m = 10$.
- Une mutation auto adaptative suivant la formulation de Fogel avec $b = 1$; $T = 0.5$.
- Une sélection par tournoi $q = 5$.

Après convergence nous obtenons les résultats suivants :

$X^* = 1.3001$ avec $f_1(X^*) = 4.5498$ et $f_2(X^*) = 3.7602$

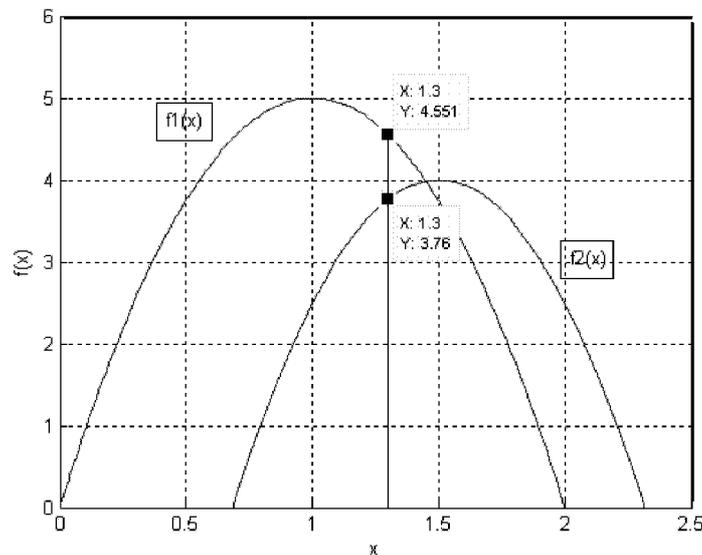


Figure II-16 : Solution obtenue

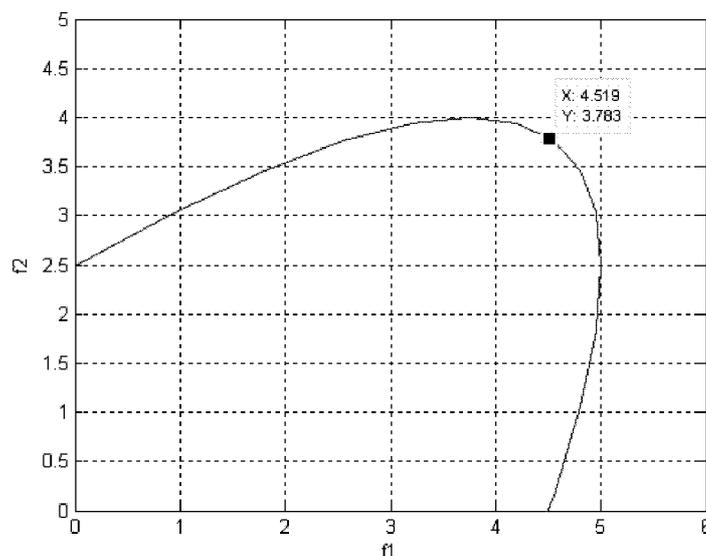


Figure II-17 : Représentation de la solution obtenue sur le front de Pareto

Remarques :

Nous pouvons remarquer que les résultats obtenus par les deux Algorithmes sont identiques et correspondent à l'optimum de la combinaison des deux fonctions. La solution obtenue peut être considérée comme une bonne solution de compromis.

II.4.5.2. Optimisation vectorielle suivant au sens de Pareto

Pour déterminer le front de Pareto nous utilisons les Algorithmes Evolutionnaires combinés à un crowding déterministe.

Stratégie d'Evolution

Une Stratégie Evolutionnaire SE (m, L) combinée à un crowding déterministe est utilisée avec les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des parents : $m = 50$.
- Taille de la population des descendants $L = 50$.
- Une recombinaison intermédiaire comme croisement.
- Une mutation de distribution normale avec $\sigma = 0.01 (X_{max} - X_{min})$.
- Une sélection déterministe de type SE (2+2).

Les solutions obtenues à la convergence de l'algorithme sont représentées sur le graphe suivant :

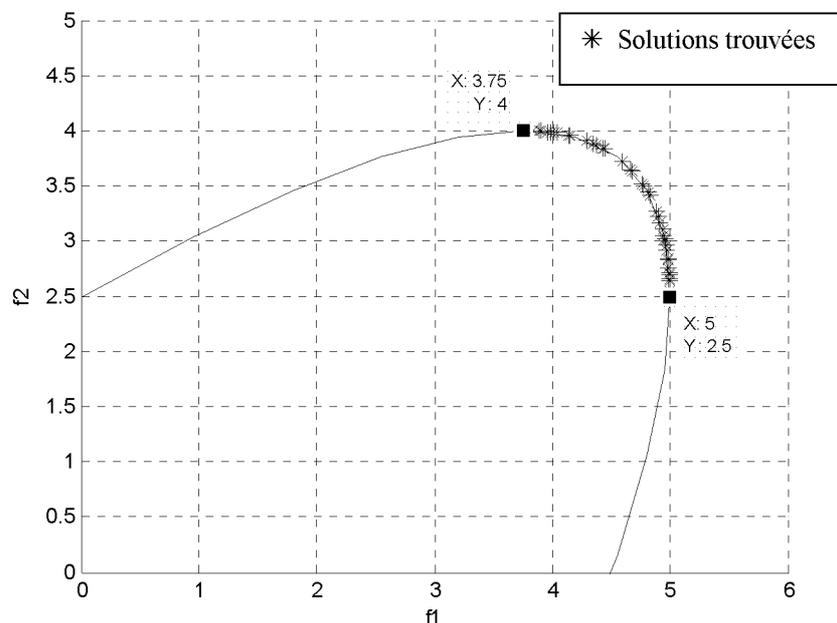


Figure II-18 : Représentation des solutions sur le front de Pareto

Nous pouvons remarquer que toutes les solutions obtenues appartiennent au front de Pareto.

Programmation Evolutionnaire

L'algorithme de crowding défini précédemment ne peut pas être rigoureusement appliqué, car la Programmation Evolutionnaire n'utilise pas le croisement. Pour cela une méthode alternative est utilisée en gardant le même principe, mais, en n'utilisant que la mutation seule.

Notre algorithme est appliqué avec les paramètres suivants :

- Nombre maximal de génération : $n_{max} = 100$.
- Taille de la population des parents : $m = 50$.
- Une recombinaison intermédiaire comme croisement.
- Une mutation de distribution normale avec $\sigma = 0.01 (X_{max} - X_{min})$.

Les résultats trouvés sont représentés dans le graphe suivant :

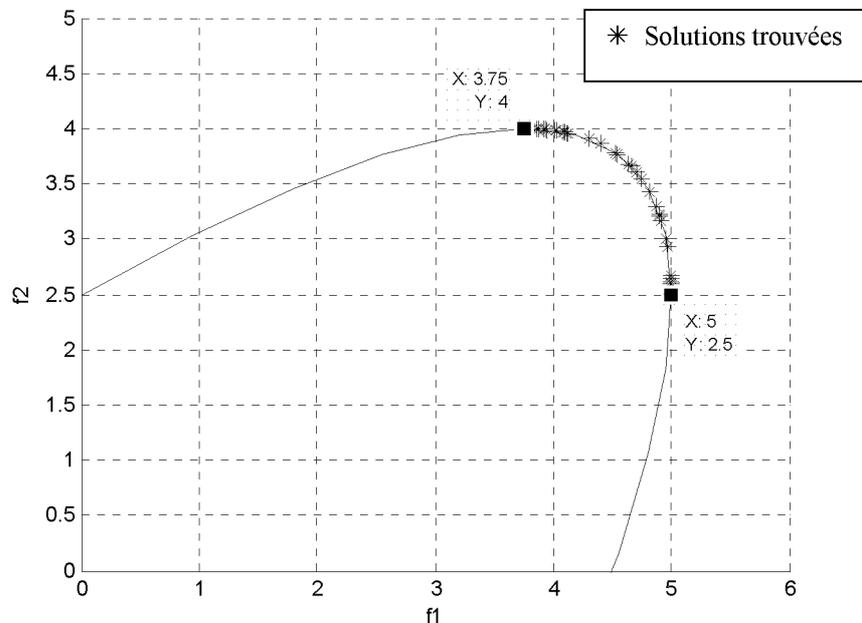


Figure II-19 : Représentation des solutions sur le front de Pareto

Nous pouvons remarquer que toutes les solutions obtenues appartiennent au front de Pareto

II.4.6. Remarques

D'après les résultats obtenus, nous pouvons constater que toutes les solutions trouvées appartiennent au front de Pareto, du moment où les solutions Pareto optimales sont de meilleures solutions de compromis que les solutions non Pareto optimales.

Les Algorithmes Evolutionnaires associés au crowding, permettent de localiser et déterminer le front de Pareto. En pratique la détermination des solutions Pareto optimales est, utile, mais pas suffisant car en générale une seule solution est requise. L'introduction de critères de décision est indispensable.

Chapitre III

***OPTIMISATION
DE LA PLANIFICATION
DE LA PUISSANCE REACTIVE***

III.1. Introduction

Comme tout secteur productif, la production et le transport d'énergie sont sujets aux lois du marché. En plus de la dérégulation, du développement des interconnexions et des fluctuations des prix des combustibles, l'aspect économique force les opérateurs à acheminer le plus d'énergie possible à travers leurs réseaux de la manière la plus rentable possible.

Augmenter la puissance produite et transmise à travers le réseau n'est pas le seul souci des opérateurs, l'amélioration de la qualité et la réduction des coûts de fonctionnement tout en respectant les contraintes de sécurité du réseau, sont considérés comme des objectifs majeurs. Ces objectifs ne peuvent être atteints si l'impact des deux contraintes majeures dans un réseau électrique n'est pas atténué, soient la limite de stabilité et le profil de tension [27].

III.1.1. Contraintes majeurs dans le transport en courant alternative

III.1.1.1. Les Limites de stabilité

Les machines synchrones dans un réseau électrique sont des alternateurs opérant au synchronisme. Le concept de base dans le maintien du synchronisme est la stabilité. La stabilité est la tendance du système à opérer régulièrement en régime ou mode normal. C'est aussi une mesure de la capacité du système à se rétablir après une perturbation extérieure (chocs de foudre, changement de charge ou manœuvres) [27].

La stabilité est liée à la capacité de transfert des lignes de transport. Si la puissance transmise à travers une ligne augmente, graduellement sans aucune perturbation extérieure, une certaine limite est atteinte pour laquelle toutes les machines synchrones connectées à cette ligne perdent leur synchronisme, le système devient instable. Cette limite est connue comme la limite de stabilité en régime permanent, c'est la puissance maximale qui peut être transmise en état permanent [27].

Cette limite dépend de plusieurs facteurs, dont :

- L'excitation des alternateurs et les tensions des lignes.
- Le nombre et les connections des lignes de transport.
- Le nombre et le type des machines synchrones connectées.
- Le cheminement des écoulements des puissances active et réactive dans le réseau.
- La disposition et les caractéristiques des équipements de compensation.

Il n'est pas pratique de faire opérer le système près de sa limite de stabilité, une marge étant toujours gardée pour maintenir la stabilité du système après une perturbation (changement de charge ou manœuvres). Pour déterminer une marge appropriée, la définition de la stabilité dynamique et transitoire est très importante.

Un système de transport est dit dynamiquement stable s'il retrouve un point de fonctionnement stable après une perturbation mineure. Le degré de la stabilité dynamique est exprimé en terme de degré d'amortissement des composantes transitoires des tensions, des courant et de l'angle interne des machines synchrones [27].

Le troisième souci majeur en terme de stabilité est la stabilité transitoire. Un système a une stabilité transitoire s'il peut retrouver un point de fonctionnement stable après une perturbation majeure. Cette perturbation peut être un défaut sévère éliminant un circuit, ou une panne d'un équipement majeur comme les alternateurs, les lignes d'interconnexions ou les transformateurs. Cette possibilité de recouvrir un fonctionnement normal dépend de plusieurs facteurs dont la nature et l'emplacement du défaut et le niveau de charge avant la perturbation [27].

III.1.1.2. Profil des tensions

La deuxième condition fondamentale dans le transport à courant alternatif, est le maintien d'un niveau de tension correcte. Les réseaux de transport ne peuvent pas tolérer des tensions anormales même pour de courtes périodes.

Les sous-tensions, qui sont généralement associées à la surcharge du réseau ou au déficit de la génération, causent des dégradations des performances des charges, en particulier pour les moteurs à induction. Dans les systèmes lourdement chargés, les sous-tensions sont un indicateur de l'approche de la limite de stabilité [27].

Les surtensions sont des conditions dangereuses augmentant le risque de contournement ou de décharge dans les systèmes d'isolation. La saturation des transformateurs sujets à des surtensions peut produire des courants riches en harmoniques, ce qui augmente les risques de ferrorésonance ainsi que des résonances harmoniques, dans le cas où une capacitance suffisante est présente.

Les surtensions ont plusieurs causes, la réduction de la charge durant certaines périodes de la courbe de charge journalière peuvent causer des surtensions graduelles, ce qui peut écourter la vie des isolations même si le niveau de décharge n'est pas atteint. Des surtensions brusques peuvent résulter d'une déconnection de charge ou autre équipement. Des surtensions transitoires sévères peuvent résulter des manœuvres des lignes ou des chocs de foudre. Dans les réseaux avec de longues lignes de transports, l'effet Ferranti (surtensions aux faibles charges) limite le transfert de puissance si aucune compensation n'est prévue [27].

III.1.2. Compensation de puissance réactive

Avec l'augmentation de la demande, l'amélioration du fonctionnement du réseau en terme de stabilité et de profil de tension, devient une nécessité pour répondre aux différentes contraintes imposées par les aspects économiques et techniques.

L'approche la plus naturelle à l'amélioration du fonctionnement d'un réseau électrique, est de le renforcer par la construction de nouvelles lignes ou de nouvelles sous-stations. Mais pour des raisons économiques, une telle approche n'est pas favorable. Même si elle augmente la stabilité du réseau et sa capacité de transfert, elle ne permet pas une exploitation optimale de ses ressources. En plus, elle nécessite des investissements considérables qui ne sont pas rentables à court et à moyen terme.

En revanche, une exploitation optimale des sources de puissance réactive peut aider à améliorer le fonctionnement du réseau, en améliorant le profil des tensions et en augmentant la capacité de transfert des lignes existantes, sans avoir à construire d'autres lignes, ce qui permet de réduire les coûts d'investissement et augmenter la rentabilité.

L'exploitation optimale de la puissance réactive, passe par l'installation de dispositifs de compensation, compensateurs. Un compensateur de puissance réactive peut principalement aider à [28]:

- Contrôler la tension
- Améliorer la stabilité

III.1.2.1. Contrôle de la tension

La fonction principale d'un compensateur réactive est le maintien de la tension en un nœud dans les limites imposées ($\pm 5\%$, par exemple). La tension en un nœud de réseau est fonction de la puissance réactive injectée en ce nœud.

L'installation d'un compensateur maintient la tension dans les limites imposées, s'il est adéquat et suffisant. L'injection de puissance réactive en un nœud fait augmenter sa tension, alors que l'absorption de puissance réactive en ce nœud la fait baisser. C'est la caractéristique principale des compensateurs shunts.

La fonction la plus importante d'un compensateur est de contrecarrer toute éventualité d'effondrement de tension (voltage collapse), par exemple lors des pics de charge, où beaucoup de lignes du réseau sont surchargées. Si, un défaut majeur apparaît en ce moment précis, le risque d'effondrement devient grand et peut conduire à un black-out. Ce risque est efficacement contrecarré par une injection rapide de puissance réactive aux nœuds de charge les plus vulnérables [28].

III.1.2.2. Amélioration de la stabilité

L'installation d'un dispositif de compensation dans une ligne de transport, améliore la capacité de transfert de cette ligne ce qui permet d'améliorer la stabilité permanente du système.

L'amélioration de la stabilité dynamique, se fait par le contrôle de la puissance réactive qu'injecte le compensateur shunt, de façon à faire varier la tension proportionnellement au taux de variation de l'angle rotorique, donc proportionnellement à la fréquence du système. Alors que l'amélioration de la stabilité transitoire peut se faire en augmentant la limite de stabilité permanente.

De plus, la compensation de puissance réactive permet de [28]:

- Contrôler le flux de puissance de sorte qu'il suive des couloirs prescrits de transport.
- Permettre un chargement sécurisé des lignes de transport à un niveau proche des limites thermiques, tout en évitant la surcharge et réduire les marges de génération avec la capacité de transférer plus de puissance entre les zones contrôlées.
- Prévenir les pannes en cascade en limitant l'impact des défauts et les pannes d'équipements.
- Réduire les coûts financiers et/ou l'impact sur l'environnement en évitant éventuellement la construction de nouvelles lignes d'extension.

III.2. Management de puissance réactive

Pour qu'un dispositif de compensation aide à améliorer le fonctionnement du réseau, il doit être correctement dimensionné et correctement installé. La gestion des différentes sources de puissance réactive dans un réseau fait appel au management de puissance réactive.

Le management de la puissance réactive peut être défini comme étant le contrôle des tensions des générateurs, des prises variables des transformateurs, des différents compensateurs existants, ainsi que la disposition de nouvelles sources de puissance réactive de manière à améliorer le fonctionnement du réseau, en termes de coût et de profil de tension [27].

Le management de la puissance réactive en régime permanent et dans les conditions dynamiques, peut être divisé comme suit [27]:

1. **Planification de la puissance réactive** : concerne l'installation et la suppression des équipements de puissance réactive dans le réseau, afin d'améliorer son fonctionnement. Typiquement, cet effort est effectué pour les conditions futures du système dans plusieurs mois ou plusieurs années.
2. **Planification du fonctionnement du système** : concerne l'amélioration du fonctionnement des équipements de puissance réactive existants. Cette planification est exécutée pour les conditions des systèmes anticipés pour plusieurs jours à un an au futur.
3. **Répartition et contrôle de la puissance réactive** : elle détermine le fonctionnement des compensations, et la quantité de puissance réactive à injecter dans le réseau en temps réel.

Le terme équipement fait référence aux compensateurs ainsi qu'aux équipements de contrôle et de communication requises pour permettre un contrôle en temps réel.

Les équipements de compensation de la puissance réactive qui doivent être installés, supprimés ou contrôlés incluent : les condensateurs shunt commutables, les réactances shunt, les compensateurs statiques, les alternateurs ainsi que les prises des transformateurs. Les équipements secondaires incluent : les appareillages de mesure, les relais de protection, le contrôle automatique (sub-station automation), les sectionneurs et les disjoncteurs, ainsi que les équipement de communication.

III.3. Planification de la puissance réactive

La planification de la puissance réactive est une fonction intégrante du management de la puissance réactive. L'objectif de la planification de la puissance réactive est de minimiser le coût d'investissement dans les équipements de puissance réactive nécessaire pour permettre au réseau de fonctionner d'une manière acceptable, dans les conditions futures.

Ce problème nécessite la détermination de l'installation optimale des équipements qui satisferaient les différentes contraintes imposées au réseau. Le problème de planification de puissance réactive est mathématiquement large, vu le nombre de variables et de contraintes qu'il fait intervenir.

Nous sommes concernés par le problème de planification de la puissance réactive à court terme. Dans notre approche au problème, qui est un prolongement aux travaux déjà faits dans le cadre du Laboratoire des Réseaux Electriques à l'Ecole Nationale Polytechnique [29] [30] [31], nous nous basons sur les considérations suivantes :

- Nous considérons un problème de planification à court terme, donc, un seul investissement est effectué sans prendre en considération les investissements futurs
- L'investissement est entrepris pour améliorer le fonctionnement du réseau face à des conditions futures. Nous dimensionnons nos sources réactives suivant une estimation du niveau de charge le plus critique que le réseau subira dans quelques années (par exemple).
- L'amélioration du fonctionnement du réseau est mesurée par les pertes actives, donc le coût de fonctionnement, ainsi que des fonctions de qualité traduisant les violations sur les contraintes de tension.

III.3.1. Formulation mathématique

Suivant ces considérations, la formulation des objectifs et des contraintes sera comme suit :

III.3.1.1. Coût de fonctionnement

Le coût de fonctionnement du réseau est estimé en termes de pertes actives. Les pertes actives dans un réseau électrique, peuvent être déterminées en utilisant le calcul d'écoulement de puissance. La formulation des pertes est comme suit :

$$P_{loss} = \sum \text{real}(V * (\text{conj}(Y_{bus} * V))) \quad (\text{III-1})$$

Il faut noter qu'une formulation quadratique des pertes n'est pas nécessaire, du fait qu'un Algorithme Evolutionnaire n'a aucune restriction sur la forme des fonctions objectives ou des contraintes.

III.3.1.2. Coût d'investissement :

Le coût d'investissement peut être considéré comme le coût des compensateurs installés ainsi que le coût de leur exploitation. En l'absence de données relatives au prix du MVAR ainsi que du coût d'installation d'un compensateur, nous considérerons le coût d'investissement comme étant le volume de la puissance réactive investie, soit [29] à [32]:

$$Q_{inj} = \sum (\delta_c |Q_c| + \delta_i |Q_i|) \quad (\text{III-2})$$

avec : δ coût du MVAR. Nous poserons $\delta = 1$.
 Q_c volume de la puissance réactive capacitive installée.
 Q_i volume de la puissance réactive inductive installée.

III.3.1.3. Les contraintes

Les contraintes du système peuvent être classées en :

Contraintes paramétriques

Les contraintes paramétriques sont définies par l'ensemble des paramètres auxquels un contrôle direct existe, c'est-à-dire les limites de la génération des puissances actives, les limites des volumes des compensateurs, les limites sur les réglages des transformateurs, ainsi que les limites des tensions aux nœuds de génération et aux nœuds de régulation.

Les contraintes paramétriques définissent les limites de l'espace de recherche, une formulation explicite de ces contraintes par une fonction de pénalité n'est pas nécessaire. L'ensemble des contraintes paramétriques est comme suit [29] à [32]:

$$\begin{aligned}
 \underline{P}_{gi} &\leq P_{gi} \leq \overline{P}_{gi} \\
 \underline{Q}_{ci} &\leq Q_{ci} \leq \overline{Q}_{ci} \\
 \underline{Q}_{li} &\leq Q_{li} \leq \overline{Q}_{li} \\
 \underline{T}_i &\leq T_i \leq \overline{T}_i \\
 \underline{V}_{gi} &\leq V_{gi} \leq \overline{V}_{gi}
 \end{aligned}
 \tag{III -3}$$

Contraintes dures

Les contraintes dures enveloppent l'ensemble des paramètres du réseau auxquels aucun contrôle direct n'existe représentées par les tensions des nœuds de charges comme principales contraintes dures dans le réseau, influencées principalement par l'écoulement de puissance réactive dans le réseau et formulées par :

$$\underline{V}_i \leq V_i \leq \overline{V}_i
 \tag{III -4}$$

Ainsi que les contraintes sur les limites thermiques des lignes et des transformateurs

$$S(i, j) \leq \underline{S}(i, j)
 \tag{III -5}$$

Ces contraintes peuvent être formulées comme suit :

$$\begin{aligned}
 g_u(i) &= (V_i - \overline{V}_i) \leq 0 \\
 g_d(i) &= (\underline{V}_i - V_i) \leq 0 \\
 g_L(i) &= (\overline{S}(i, j) - S(i, j)) \leq 0
 \end{aligned}
 \tag{III -6}$$

Avec V_i : Tension au noeud de charge i.
 \underline{V}_i : Limite inférieure.
 \overline{V}_i : Limite supérieure.

Et $S(i, j)$ puissance transmise dans la branche i, j .
 $\underline{S}(i, j)$ limite thermique de la branche i, j .

Et comme il n'existe aucun contrôle directe sur ces paramètres, nous assurerons la pleine satisfaction de ces contraintes soit en introduisant des fonctions de pénalités, soit en formulant des fonctions de qualité introduites comme des fonctions objectives supplémentaires.

Contraintes de fonctionnement

Les contraintes de fonctionnement sont définies par les équations d'écoulement de puissance, correspondant à un point de fonctionnement du réseau, pour une configuration de charge et de génération donnée :

$$\begin{aligned} P_k &= V_k \sum Y_{ij} V_j \cos(\delta_i - \delta_j - \theta_{ij}) \\ Q_k &= V_k \sum Y_{ij} V_j \sin(\delta_i - \delta_j - \theta_{ij}) \end{aligned} \quad (\text{III -7})$$

Les contraintes de fonctionnement sont traitées par la procédure du calcul d'écoulement de puissance, pour un niveau de charge et une configuration de compensateurs données, ce qui nous permet d'avoir l'ensemble des paramètres du réseau.

III.3.2. Formulation du problème

Le problème est considéré comme : la détermination de la configuration optimale des compensateurs pour la minimisation des pertes actives et les violations sur les contraintes, données par la procédure d'écoulement de puissance. Ce qui revient en fin de compte à l'optimisation de l'écoulement de puissance pour un niveau de charge, de génération et de compensation donné, sujet aux différentes contraintes qui régissent le système.

Le problème peut être formulé donc comme suit :

$$\begin{aligned} & \min P_{loss}(Q_{inj}) \\ & \min Q_{inj}(Q_{inj}) \\ \text{Sujet à} & \quad g(Q_{inj}) \leq 0 \\ \text{Données par} & \quad h(P_g, S_D, Q_{inj}) = 0 \\ & \quad \{P_g, S_D, Q_{inj} \in \Omega\} \end{aligned} \quad (\text{III -8})$$

III.4. Résolution du problème

La résolution du problème de l'optimisation de la Planification de la puissance réactive, passe par la détermination de la configuration optimale de dispositifs de compensation, permettant l'amélioration du fonctionnement du réseau. Pour cela l'algorithme suivant est utilisé :

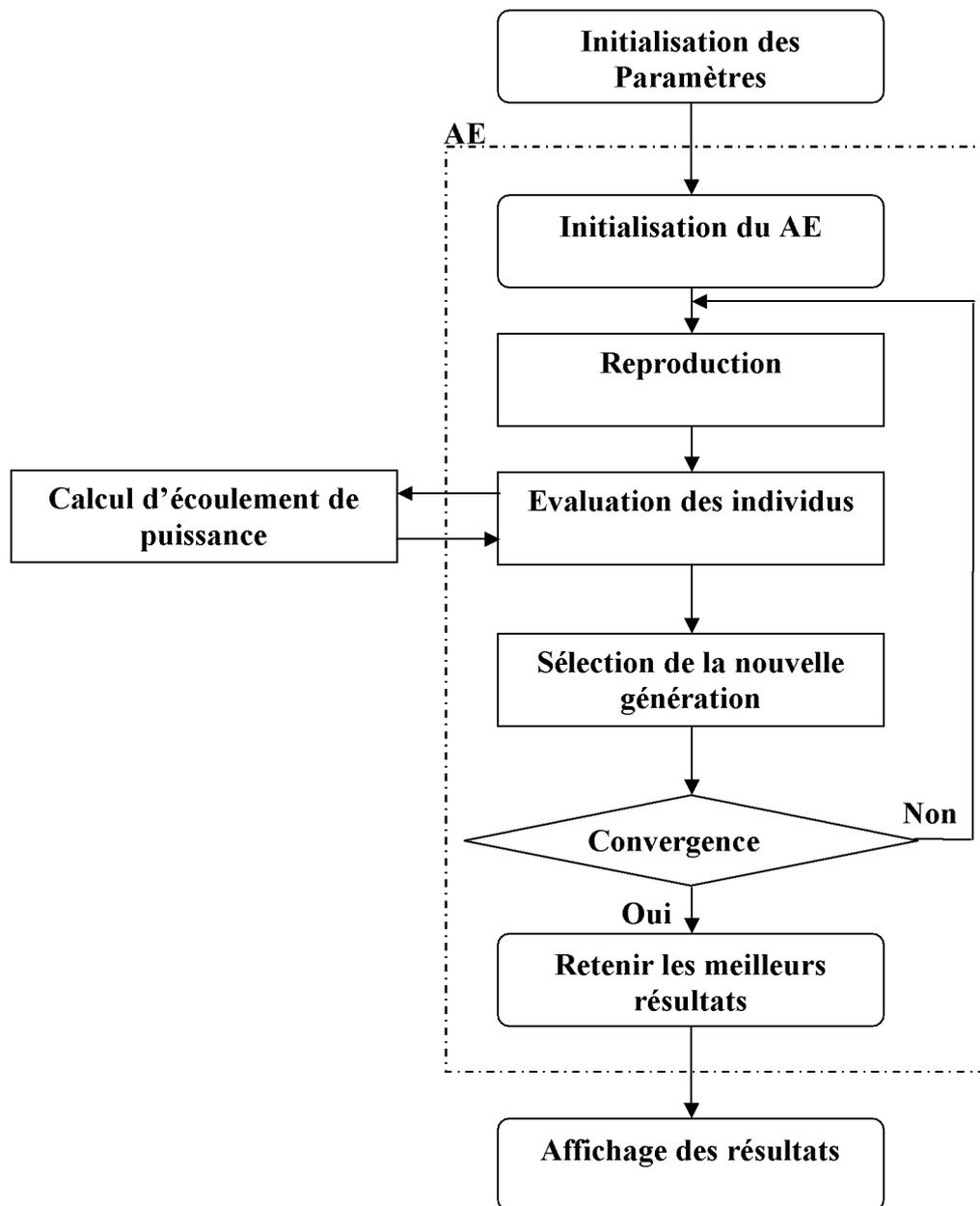


Figure III-1 : Algorithme de résolution

Cet algorithme reprend le schéma de base d'un Algorithme Evolutionnaire, en incorporant les procédures inhérentes à la planification de la puissance réactive. L'algorithme de résolution enveloppe les procédures suivantes :

- Initialisation des paramètres relatifs au calcul d'écoulement de puissance et aux paramètres de l'Algorithme Evolutionnaire.
- Initialisation de l'Algorithme Evolutionnaire par la création de la population initiale
- Reproduction : création de la population des descendants en utilisant le croisement et/ou la mutation.
- Evaluation : l'assignement d'une valeur d'adaptation à chaque individu, suivant les valeurs des fonctions objectives et les contraintes données par le calcul d'écoulement de puissance et la méthode d'évaluation utilisée.
- Test de convergence : teste la convergence de l'algorithme en fonction du nombre de générations maximales imposé.

La procédure d'écoulement de puissance est appelée à chaque évaluation, par souci d'efficacité, nous utilisons une procédure de calcul d'écoulement de puissance découlé rapide avec ajustement de paramètres [33]. L'ajustement de paramètres permet de traiter les contraintes paramétriques par la procédure d'écoulement de puissance. En plus, de l'ajustement des paramètres, la formulation vectorielle de jacobéen permet de rendre la procédure de calcul plus rapide [34], en exploitant la puissance du MATLAB dans le calcul matriciel.

III.4.1. Représentation

La représentation est la première procédure à définir. Elle est considérée comme l'interface entre l'Algorithme Evolutionnaire et le problème à résoudre. Un individu représente une solution potentielle au problème en cours, il est primordial que sa représentation permette une expression naturelle du problème.

Nous voulons trouver la meilleure disposition, possible, de compensateurs pour améliorer le fonctionnement du réseau en termes de pertes actives et de profil des tensions, tout en minimisant le coût d'investissement. Naturellement, un individu doit représenter une configuration donnée de compensateurs. Cette configuration est déterminée par l'ensemble des nœuds candidats à la compensation et la quantité de puissance réactive à disposer en chaque nœud candidat.

La façon la plus naturelle de représenter un individu, peut être présentée sous la forme suivante :

$$ind(i) = \{ Q_{inj} \quad R_{cand} \} \quad (\text{III -9})$$

Avec : Q_{inj} : quantité de puissance réactive à disposer en chaque nœud candidat.

R_{cand} : Ensemble des nœuds candidats à la compensation.

Notons que le vecteur Q_{inj} est le seul sujet à la reproduction et l'évaluation. Donc c'est le seul paramètre à optimiser. Le vecteur des nœuds candidats R_{cand} doit être choisi avant le déroulement de l'algorithme de calcul.

Choix des nœuds candidats

La fonction première d'un dispositif de compensation est de réduire l'écoulement réactif, afin de diminuer les pertes et d'améliorer le profil des tensions. Pour cela, les compensateurs doivent être installés le plus près possible des concentrations des nœuds de charge. Un tel emplacement présente les avantages suivants :

- Permet de fournir un support de la tension, car ces nœuds sont souvent assez éloignés des centres de production.
- Permet d'isoler les charges des perturbations du réseau, ceci peut être grandement amélioré par l'utilisation des systèmes de compensation de troisième génération (FACTS) ainsi que les filtres d'harmoniques.

Mais, d'une façon générale, les nœuds candidats peuvent être déterminés en observant le comportement du réseau face aux conditions les plus défavorables qu'il peut subir, en effectuant un calcul d'écoulement de puissance.

III.4.2. *Reproduction*

La reproduction est une des procédures essentielles des Algorithmes Evolutionnaires. Elle permet de créer de nouveaux individus et explorer, ainsi, l'espace de recherche.

La méthode d'optimisation dépend, essentiellement, du type du problème à traiter. Pour les problèmes d'optimisation, la mutation auto adaptative est incontournable pour les avantages qu'elle offre. Nous avons choisi d'utiliser :

- Une recombinaison intermédiaire et une mutation auto-adaptative suivant la formulation de SCHWEFEL dans le cadre des Stratégies d'Evolution.
- Une mutation auto adaptative suivant la formulation de FOGEL dans le cadre de la Programmation Evolutionnaire.

III.4.3. *Evaluation*

L'évaluation des individus est une étape clé dans le calcul évolutionnaire, car c'est sur la base de l'évaluation de chaque individu que se joue la décision de le maintenir ou de l'éliminer de la génération. L'évaluation dépend principalement du problème en cours. Dans notre cas, la qualité d'un individu est jugée sur la base des fonctions objectives et des violations sur les contraintes.

Les valeurs des fonctions objectives et des contraintes sont données par la procédure d'écoulement de puissance, basée sur la configuration de compensation introduite par chaque individu. Par souci d'efficacité, nous utilisons une procédure d'écoulement de puissance avec ajustement de paramètres. Cette méthode permet d'éviter l'introduction de contraintes paramétriques comme contraintes supplémentaires.

Nous avons choisi trois méthodes d'évaluation les plus utilisées dans les problèmes d'optimisation multi-objectifs :

III.4.3.1. Méthode du vecteur cible

Dans cette méthode les objectifs et les contraintes sont combinés en une seule fonction objective à optimiser, l'évaluation de l'adaptation en utilisant la fonction suivante :

$$f(ind(i)) = \sqrt{((Q_{inj} - Q_g)^2 + (P_{loss} - P_g)^2)} + Ct^2 H \quad (\text{III -10})$$

$$\text{Avec : } H = \sum_{i=1}^n (\max(g_u(i), 0) + \max(g_d(i), 0))$$

Avec P_{loss} : somme des pertes actives du réseau.

P_g : valeur des pertes actives a la quelle nous voulons aboutir.

Q_{inj} : puissance réactive à investir aux nœuds candidats R_{cand} .

Q_g : la puissance réactive que nous voulons investir.

C : constante

III.4.3.2. Optimisation vectorielle non Pareto

Dans l'optimisation vectorielle, la fonction d'évaluation ne retourne pas un scalaire mais un vecteur, composé des fonctions objectives et des fonctions de qualité. Les fonctions de qualité sont introduites pour assurer la faisabilité des solutions, en introduisant les contraintes comme fonctions objectives supplémentaires [35] [36].

$$\min F = (f_1, f_2, f_3, f_4, f_5, f_6)$$

$$f_1 = Q_{inj}$$

$$f_2 = P_{loss}$$

$$f_3 = \sum_{i=1}^n (V(i) - V(i))^* (\max(g_u, 0)) \quad (\text{III -11})$$

$$f_4 = \sum_{i=1}^n (V(i) - V(i))^* (\max(g_d, 0))$$

$$f_5 = (V_{\min} - \min(V(i)))^2$$

$$f_6 = (\max(V(i)) - V_{\max})^2$$

Avec :

Comme la sélection se base sur une mesure scalaire de l'adaptation, plusieurs méthodes peuvent être utilisées pour affecter une valeur d'adaptation aux individus. Une des méthodes les plus utilisées est l'approche lexicographique, cette approche se basant sur la définition d'une priorité pour chaque objectif.

Nous proposons une approche alternative, basée sur la notion de priorité et un système de score. Lors de la sélection, chaque individu est comparé avec ses concurrents. Au terme de chaque comparaison, l'individu se voit affecter un score déterminé suivant :

$$S(k, i) = \sum_{j=1}^n I(F_k(j) < F_i(j)) * P(j) \quad (\text{III -12})$$

Avec : $P(j)$: définie suivant la priorité de chaque objectif, I note la condition.

Pour accélérer le processus de sélection nous utilisons une compétition par tournoi, pour limiter le nombre de comparaisons. En utilisant cette méthode de sélection, chaque individu se voit attribuer un score : $scr_k = \sum S(k, j)$, les meilleurs individus seront choisis pour la génération suivante.

Dans ce cas une définition judicieuse des fonctions de qualité et des priorités est primordiale, car, comme dans le cas de l'approche lexicographique, l'optimisation va toujours dans le sens des objectifs les plus prioritaires.

Les objectifs et les priorités ainsi définis, nous permettent d'avoir une seule solution optimale, et non un ensemble de solutions optimales. Nous pouvons constater que la solution optimale, du problème ainsi défini, est celle qui optimise les fonctions f_5 et f_6 . Avoir $f_5=0$ et $f_6=0$ n'est possible que pour une seule valeur de $Qinj$, qui est la valeur minimale pour laquelle toutes les contraintes sont pleinement satisfaites.

III.4.3.3. Optimisation vectorielle au sens de Pareto

Dans cette méthode la notion de dominance est utilisée pour pouvoir comparer les individus. Cette méthode suppose qu'il n'y a pas une seule solution optimale mais un ensemble de solutions définissant le front de Pareto ou l'ensemble des solutions non dominées. Pour pouvoir stabiliser les solutions sur le front de Pareto, le concept des niches est introduit par l'utilisation du Crowding.

Pour assurer la satisfaction des contraintes, nous utilisons, en plus des fonctions objectives, des fonctions de qualité considérées comme objectifs supplémentaires. Ces fonctions ont pour but de limiter le front de Pareto aux seules solutions faisables et aux régions jugées intéressantes.

III.4.4. Sélection

Le choix de la méthode de sélection à utiliser dépend essentiellement de la nature du problème à résoudre et du type de l'Algorithme Evolutionnaire choisi. Généralement, ce choix, ainsi que les paramètres de la méthode choisie, sont basés sur le comportement de l'Algorithme et sa convergence. Une convergence rapide peut être synonyme d'une convergence prématurée.

Nous utilisons différentes méthodes d'optimisation, suivant l'Algorithme Evolutionnaire à utiliser, en particulier :

- Une sélection par tournoi, est utilisée, dans le cadre des Stratégies d'Evolution.
- Une compétition par tournoi, est utilisée, dans le cadre de la Programmation Evolutionnaire.
- Une Sélection déterministe, est utilisée, dans le cadre du Crowding.

Les paramètres de chaque sélection sont choisis d'une manière à assurer un comportement optimal à l'Algorithme.

Chapitre IV

APPLICATIONS

Pour valider les algorithmes développés, nous exécuterons des tests sur des réseaux standard : le réseau 16 nœuds et le réseau IEEE 57 nœuds. Pour chaque réseau nous basons notre planification sur une estimation de la pointe de charge sur une période donnée (cinq ans par exemple). Enfin, et pour donner un aspect plus pratique à notre travail nous appliquerons cette planification sur le Réseau Electrique Algérien.

IV.1. Application sur le réseau 16 nœuds

Une fois que la procédure de calcul est développée, nous pouvons la tester sur un exemple simple d'un réseau à 16 nœuds [37].

Nous planifions l'installation de dispositifs de compensation, en prévention de l'augmentation de la charge du réseau dans cinq ans. Pour étudier l'évolution du comportement du réseau, le tableau suivant nous présente les résultats du calcul d'écoulement de puissance pour les niveaux de charges suivant :

- **Niveau 1** : Le niveau nominal actuel : la charge nominale du réseau.
- **Niveau 2** : La pointe de charge actuelle estimée à 1,25 de la charge nominale du réseau, ce niveau est confondu avec le niveau de charge nominal estimé dans cinq ans.
- **Niveau 3** : La pointe de charge planifiée estimée à 1,50 de la charge nominale du réseau.

Nous pouvons remarquer d'après ces résultats que le nœud 16, du fait qu'il est un nœud de charge pour lequel aucun support de tension n'est disponible, subit des chutes de tension qui dépassent les limites acceptables (pour un nœud de charge).

noeuds	Tensions (pu)		
	Niveau 1	Niveau 2	Niveau 3
1	1.0000	1.0000	1.0000
2	1.0152	1.0063	0.9924
3	1.0446	1.0405	1.0319
4	1.0384	1.0312	1.0195
5	1.0334	1.0220	1.0066
6	1.0359	1.0263	1.0123
7	1.0348	1.0238	1.0095
8	1.0406	1.0325	1.0218
9	1.0500	1.0500	1.0500
10	1.0470	1.0430	1.0374
11	1.0387	1.0300	1.0178
12	1.0360	1.0248	1.0095
13	1.0086	0.9961	0.9793
14	0.9967	0.9774	0.9529
15	1.0074	0.9907	0.9644
16	0.9656	0.9174	0.8454

Tableau IV-1 *Profile des tension pour
Les différents niveaux de charge*

	Niveau 1	Niveau 2	Niveau 3
Pertes actives (pu)	3.3228	7.5403	14.8958

Tableau IV-2 *Pertes actives en fonction du niveau de charge*

IV.1.1. Détermination de la puissance réactive à investir

Pour déterminer la quantité de puissance réactive à investir, nous utilisons les Algorithmes Evolutionnaires suivant : Stratégie d'Evolution et Programmation Evolutionnaire, combinés au trois méthodes d'évaluation retenues : la méthode de vecteur cible, l'optimisation vectorielle non Pareto et l'optimisation vectorielle non Pareto.

Choix des nœuds candidats

Nous pouvons remarquer d'après les résultats du calcul d'écoulement de puissance, que le nœud 16, qui est un nœud de charge pour lequel aucun support de tension n'est disponible, subit une chute de tension considérable. Ce nœud peut être considéré comme nœud candidat à la compensation.

IV.1.1.1. Méthode de vecteur cible

Nous utilisons les deux Algorithmes Evolutionnaires suivant pour optimiser le système (III-10) :

- une Stratégie d'Evolution SE (m, L) :
 - taille de la population des parents : $m = 10$.
 - taille de la population des descendants : $L = 50$.
 - une recombinaison intermédiaire comme croisement.
 - une mutation auto adaptative avec : $b = 1.0$ et $T = 0.8$.
 - une sélection par tournoi avec : $q = 5$.

- une Programmation Evolutionnaire PE :
 - taille de la population des parents : $m = 20$.
 - taille de la population des descendants : $L = 50$.
 - une mutation auto adaptative, avec : $b = 0.75$.
 - une sélection de type Compétition par tournoi, avec : $q = 5$.

Le critère d'arrêt est pris en terme de nombre de générations. Nous considérons que la convergence est atteinte après 50 générations. Les résultats obtenus, sont donnés dans le tableau suivant :

Algorithme Evolutionnaire	Stratégie d'Evolution	Programmation Evolutionnaire
Noeuds candidats	16	16
Tension du nœud 16 (pu)	0.9500	0.9499
puissance réactive installée (pu)	0.3741	0.3737
pertes actives (pu)	0.1285	0.1285

Tableau IV- 3 : Résultat d'écoulement de puissance pour les solutions obtenues
Par l'approche du vecteur cible

Les résultats obtenus par les différents algorithmes sont identiques. L'installation d'un dispositif de compensation de 37 MVars au nœud 16 permet d'améliorer le profil de tension et permet de réduire les pertes actives de 2 MW. Donc, les objectifs fixés par la planification de la puissance réactive sont atteints.

Nous pouvons remarquer aussi, que le résultat obtenu, correspond à la plus petite quantité de puissance réactive nécessaire pour assurer la satisfaction des contraintes, ce qui s'accorde le mieux avec l'aspect économique du problème, ceci est dû au fait que le gain de 0,1 MW correspond à l'investissement de 1 MVar.

IV.1.1.2. Optimisation vectorielle non Pareto

La formulation (IV-11) est utilisée, avec les Algorithmes Evolutionnaires suivants :

- une Stratégie d'Evolution SE (m, L)
 - taille de la population des parents : $m = 10$.
 - taille de la population des descendants : $L = 50$.
 - une recombinaison intermédiaire comme croisement.
 - une mutation auto adaptative avec : $b = 1.0$ et $T = 0.8$.
 - une sélection de type Compétition par tournoi, avec : $q = 5$.

- une Programmation Evolutionnaire PE
 - taille de la population des parents : $m = 20$.
 - taille de la population des descendants : $L = 50$.
 - une mutation auto adaptative, avec : $b = 0.75$.
 - une sélection de type Compétition par tournoi, avec : $q = 5$.

Après convergence, nous obtenons les résultats suivants :

Algorithme Evolutionnaire	Stratégie d'Evolution	Programmation Evolutionnaire
Noeuds candidats	16	16
Tension du nœud 16 (pu)	0.9501	0.9500
puissance réactive installée (pu)	0.3745	0.3741
pertes actives (pu)	0.1248	0.1248

Tableau IV- 4 : Résultat d'écoulement de puissance pour les solutions trouvées
Par l'optimisation vectorielle non Pareto

Nous pouvons constater que les résultats donnés par l'utilisation de l'optimisation vectorielle, sont identiques à ceux donnés par la méthode du vecteur cible. Ceci est dû au choix des fonctions de qualité et des priorités, ce qui nous a permis de diriger la recherche vers les solutions qui correspondent le plus à nos attentes.

IV.1.1.3. Optimisation vectorielle au sens de Pareto

Cette optimisation se base sur la définition de la dominance pour déterminer l'ensemble des solutions non dominées qui correspondent aux solutions optimales (suivant Pareto). Pour déterminer ce front, donc les solutions non dominées faisables, nous avons utilisé, en plus des fonctions objectives, des fonctions de qualité qui assurent la faisabilité des solutions Pareto optimale. Le problème sera formulé comme suit :

Pour stabiliser les solutions sur le front de Pareto, un Crowding déterministe est utilisé. Le crowding utilise une sélection locale, un descendant remplace son parent s'il le domine, l'évaluation de la dominance est faite suivant un ordre de priorité : les fonctions de qualité sont prioritaires sur les fonctions objectives, donc elles sont comparées en premier lieu. Si ces fonctions sont incomparables, les fonctions objectives sont comparées.

Nous appliquons une Stratégie d'Evolution utilisant un crowding déterministe, avec une sélection de type SE (2+2), et les paramètres suivants :

- Taille de la population $M = 50$.
- Croisement uniforme.
- Mutation normale avec une variance $\sigma = 0.03 * (q_{\max} - q_{\min})$.
- Nombre de génération max = 100.

La convergence du programme rapporte les résultats représentés par la figure 3.

Notons que le front de Pareto est limité entre : $[0,37 \ 0,1285]$ et $[0,69 \ 0,1242]$. Toutes les solutions se trouvant entre ces deux points sont faisables et non dominées. L'analyse de ces résultats nous montre que les solutions non dominées ne sont pas toutes des solutions optimales. Considérons par exemple les deux cas extrêmes suivants : les points $[0,37 \ 0,1285]$ et $[0,69 \ 0,1242]$. Ils sont tous les deux des solutions Pareto optimales, mais le gain de 0,43 MW ne justifie pas l'investissement de 32 MVars supplémentaires. La solution ne peut pas donc être considérée comme optimale, vu les contraintes économiques formulées par la minimisation de la puissance réactive investie.

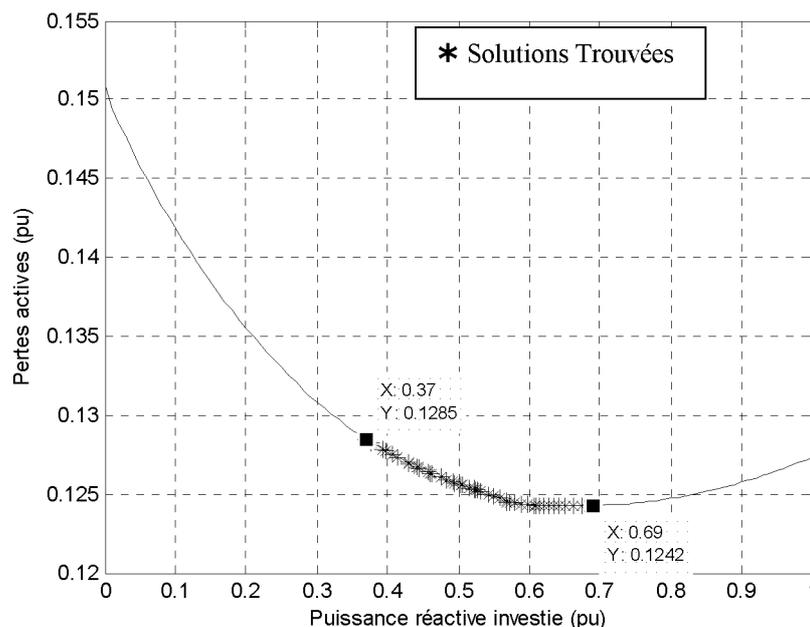


Figure IV-1 : Représentation des solutions sur Front de Pareto

Pour avoir des résultats qui concordent avec l'aspect économique du problème, i.e. minimisation de l'investissement, nous utilisons la même formulation que pour l'optimisation vectorielle non Pareto (IV-1).

$$\begin{aligned}
 \min F &= \{f_1, f_2, f_3, f_4, f_5, f_6\} \\
 f_1 &= Q_{inj} \\
 f_2 &= P_{loss} \\
 f_3 &= I(\min(V) < \underline{V}) \\
 \text{avec : } f_4 &= \sum_{i=1}^n I(\underline{V}(i) > V(i)) \\
 f_5 &= I(\bar{V} < \min(V)) \\
 f_6 &= \sum_{i=1}^n I(V(i) > \bar{V}(i))
 \end{aligned} \tag{IV-1}$$

Pour la résolution du problème ainsi formulé, nous utilisons la même stratégie d'Evolution que précédemment, ainsi qu'un Programme Evolutionnaire utilisant un crowding avec une sélection déterministe de type PE (1+1).

Après convergence des deux programmes et en considérant la solution optimale comme celle qui minimise le plus la puissance réactive investie, nous obtenons les résultats suivants :

Algorithme Evolutionnaire	Stratégie d'Evolution	Programmation Evolutionnaire
Noeuds candidats	16	16
Tension du nœud 16 (pu)	0.9500	0.9499
puissance réactive installée (pu)	0.3738	0.3736
pertes actives (pu)	0.1248	0.1248

Tableau IV-5 : Résultats d'écoulement de puissance des solutions trouvées
Par l'optimisation vectorielle au sens de Pareto

En utilisant les objectifs ainsi formulés, nous limitons le front de Pareto à la région faisable correspondant au minimum de puissance réactive nécessaire pour assurer la satisfaction des contraintes. Les résultats obtenus sont proches des résultats donnés par les autres approches.

IV.1.2. Conclusions

L'analyse des résultats obtenus montre que :

- La préservation des contraintes peut être obtenue, soit par des fonctions de pénalité, soit par l'introduction de fonctions de qualité. Dans ce dernier cas introduire la notion priorité est nécessaire.
- La définition des fonctions de qualité est très importante dans le cadre de l'optimisation vectorielle, car elles permettent de diriger la recherche pour pouvoir atteindre des solutions satisfaisantes.

IV.2. Réseau IEEE 57 nœuds :

Les données de ce réseau se trouvent en annexe. Ce réseau est constitué de 63 branches, 17 transformateurs et 6 nœuds contrôlés.

Nous voulons planifier l'installation de dispositifs de compensation, en prévention d'une augmentation de la charge du réseau sur une période de cinq ans. Le niveau de charge nominal actuel, le niveau de charge nominal dans cinq ans, ainsi que la pointe de charge dans cinq ans sont définie comme suit :

Niveau 1: niveau de charge nominal actuel.

Niveau 2: Le niveau de charge nominal dans cinq ans, estimé à 125% de la charge nominale actuelle.

Niveau 3: La pointe de charge dans cinq ans, posée comme 150% de la charge nominale actuelle.

Le calcul d'écoulement de puissance pour chaque niveau de charge donne les pertes actives et les tensions du réseau en fonction du niveau de charge, les résultats obtenus sont présentés dans le tableau V-5 et la figure V-8, respectivement.

	Niveau 1	Niveau 2	Niveau 3
Pertes Actives (pu)	0,2787	0,4437	0,6879

Tableau IV-6 : Pertes actives en fonction du niveau de charge

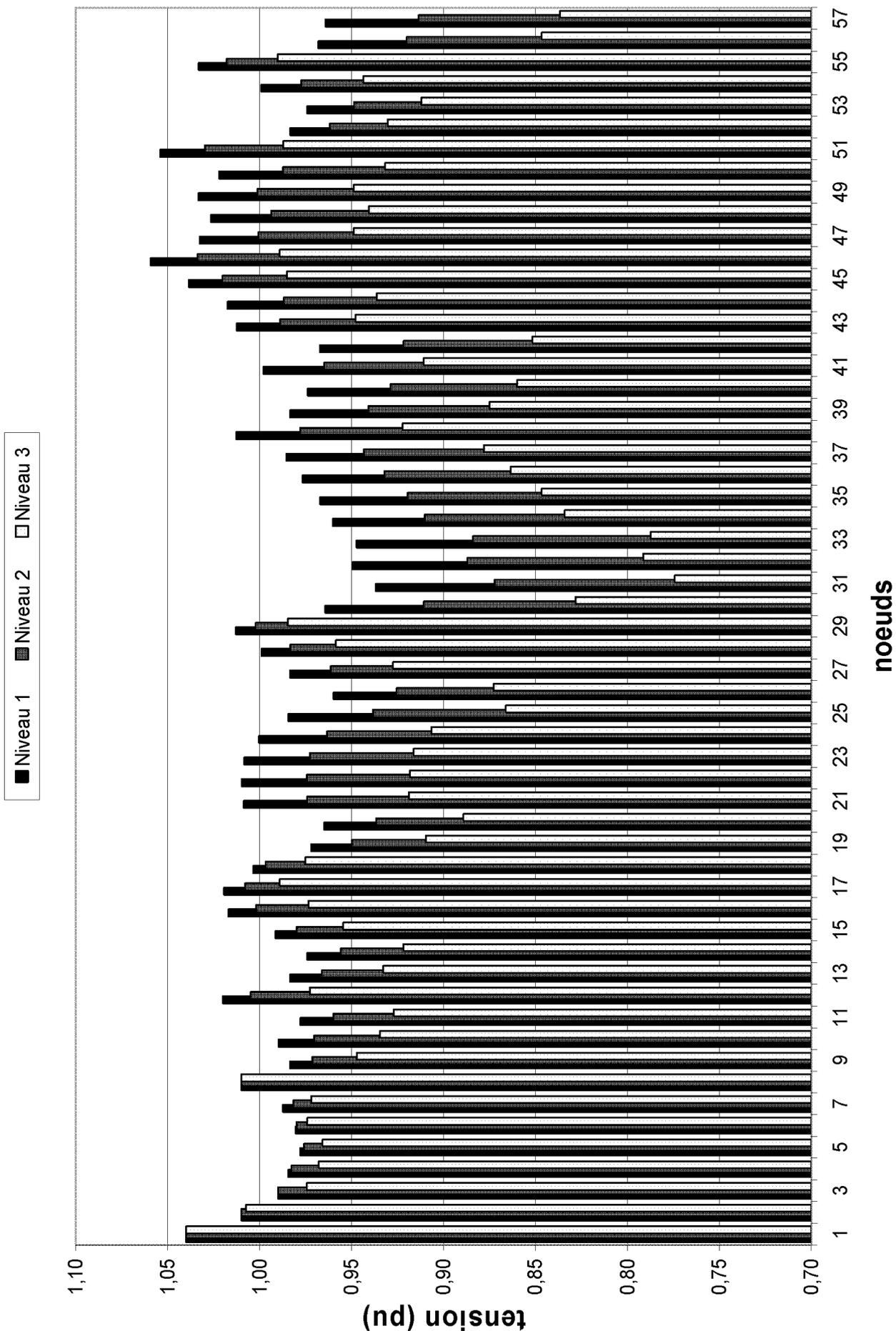


Figure IV-2 : profil de tension pour les différents niveaux de charge

IV.2.1. Détermination de la puissance réactive à investir

L'analyse des résultats du calcul d'écoulement de puissance effectué pour les différents niveaux de charge, montre que pour fonctionner correctement avec une charge correspondant au niveau 3, le réseau doit être renforcé par l'installation de dispositifs de compensation, pour diminuer les pertes actives et corriger le profil de tension.

Les contraintes sur les tensions des nœuds de charge (contraintes dures) de ce système sont définies comme suit :

$$\underline{V} = 0.95 \quad \text{et} \quad \bar{V} = 1.05 \quad (\text{IV-2})$$

Choix des nœuds candidats

Les nœuds candidats peuvent être déterminés en analysant les résultats du calcul d'écoulement de puissance pour les différents niveaux de charge. Nous pouvons remarquer que les nœuds 11, 24 et 34 sont des nœuds de charge pour lesquelles l'installation de compensateurs est recommandée, du fait des chutes de tension subies par ces nœuds.

Une fois les nœuds candidats définis, il reste à déterminer la quantité optimale de puissance réactive à investir en chaque nœud candidat. Pour cela, nous utilisons notre algorithme de calcul, combiné aux différentes méthodes d'évaluations retenues :

IV.2.1.1. Approche du vecteur cible

En utilisant cette méthode, la détermination de la quantité optimale de puissance réactive à investir, passe par la minimisation de la fonction objective définie dans (III-10).

Les Algorithmes Evolutionnaires utilisés sont les suivants :

- **Stratégie d'Evolution** : paramètres utilisés

- Taille de la population parente : $M = 20$
- Taille de la population descendante : $L = 100$
- Nombre de générations maximum: $NMAX = 100$
- Croisement : recombinaison intermédiaire
- Paramètre lié à la mutation : $b = 0.9, T = 0.75$
- Paramètre lié à la sélection par tournoi: $q = 10$

La solution obtenue après convergence de cet algorithme, ainsi que le profil de tension correspondant, sont présentés dans le tableau IV-6, et la figure V-7 respectivement.

- **Programme Evolutionnaire** : paramètres utilisés

- Taille de la population parente : $M = 20$
- Taille de la population descendante : $L = 100$
- Nombre de générations maximum: $NMAX = 150$
- Paramètre lié à la mutation : $b = 0.75$
- Paramètre lié à la sélection par tournoi: $q = 10$

La solution obtenue après la convergence de cet algorithme, ainsi que les résultats d'écoulement de puissance, sont donnés par le tableau IV-6, et la figure IV-8, respectivement.

Les résultats obtenus par cette méthode, sont comme suit :

Algorithme Evolutionnaire	Stratégie d'Evolution			Programmation Evolutionnaire		
Noeuds candidats	11	24	34	11	24	34
puissance réactive installée (pu)	0,4821	0,1274	0,4985	0,4792	0,1259	0,5013
Total (pu)	1,1080			1,1064		
pertes actives (pu)	0,6402			0,6405		

Tableau IV-7 : Solutions optimales obtenus par L'approche vecteur cible

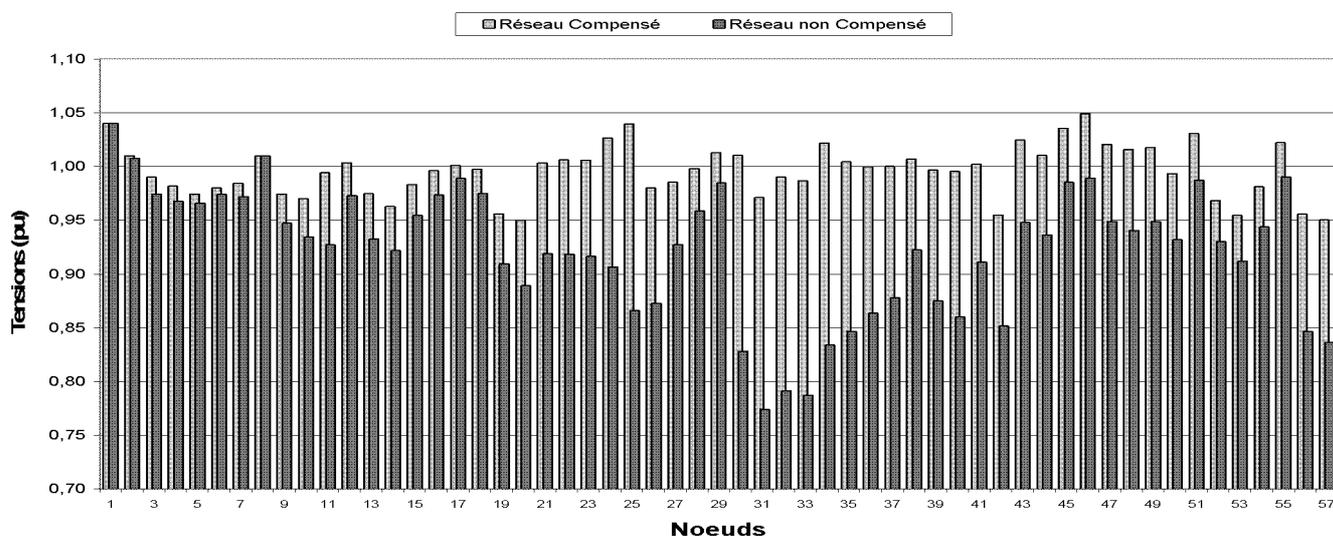


Figure IV-3: Profil de tensions obtenues avec une Stratégie d'Evolution Approche du vecteur cible

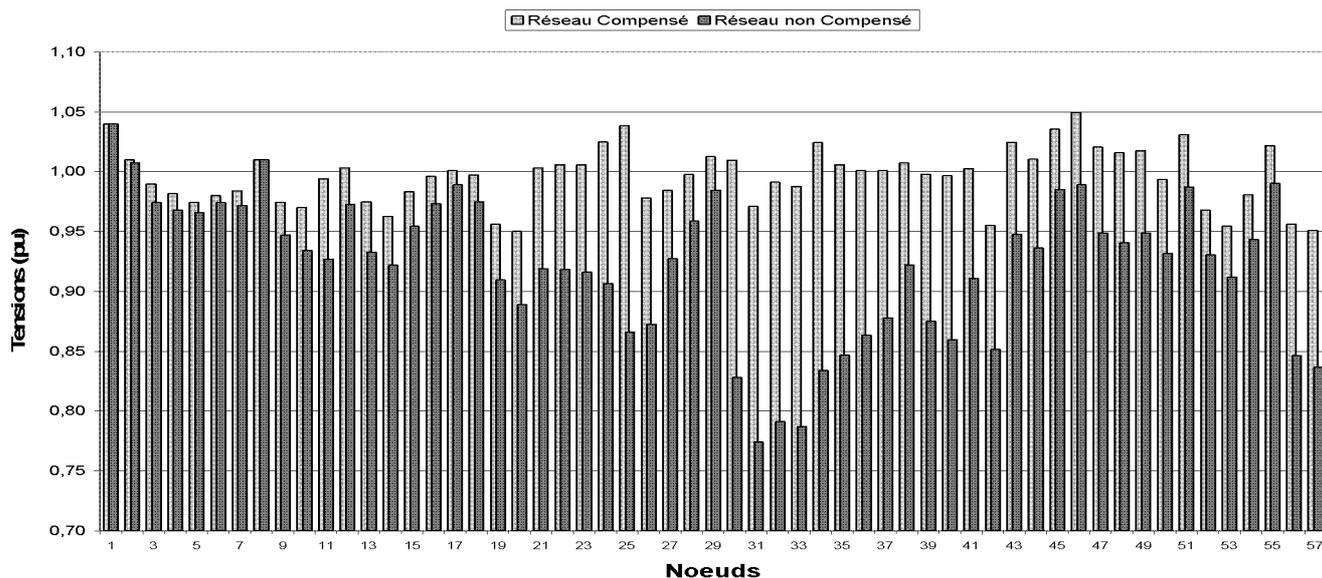


Figure IV-4 : Profil de tensions obtenues avec un Programme Evolutionnaire Approche du vecteur cible

D'après les résultats obtenus, nous constatons qu'un investissement de 110 MVars permet d'améliorer le fonctionnement du réseau en terme de profil de tension, puisque toutes les tensions sont dans leurs limites imposées, en plus d'un gain de 4 MW en réduisant les pertes actives de près de 7%.

IV.2.1.2. Optimisation vectorielle non Pareto

La quantité optimale de puissance réactive à investir est déterminée en minimisant le système (III-11), en utilisant les Algorithmes Evolutionnaires suivants :

- **Stratégie d'Evolution** : paramètres utilisés

- Taille de la population parente : $M = 20$
- Taille de la population descendante : $L = 50$
- Nombre de générations maximum : $NMAX = 50$
- Paramètres liés à la mutation : $b = 0.75, T = 0.5$
- Sélection par compétition par tournoi: $q = 5$

La solution obtenue après la convergence de cet Algorithme, ainsi que les résultats d'écoulement de puissance, sont donnée par le tableau IV-7, et la figure IV-9, respectivement.

- **Programme Evolutionnaire** : paramètres utilisés

- Taille de la population parente : $M = 20$
- Taille de la population descendante : $L = 50$
- Nombre de générations maximum: $NMAX = 50$
- Mutation auto adaptative, avec: $b = 0.5$
- Compétition par tournoi, avec: $q = 5$

La solution obtenue après convergence de cet algorithme, ainsi que les résultats d'écoulement de puissance, sont donnés par le tableau IV-7, et la figure IV-10, respectivement.

Les résultats obtenus sont comme suit :

Méthode	Stratégie d'Evolution			Programmation Evolutionnaire		
	11	24	34	11	24	34
puissance réactive installée (pu)	0.4615	0.1524	0.4916	0.4776	0.1241	0.5036
Total (pu)	1.1054			1.1052		
pertes actives (pu)	0.6405			0.6404		

Tableau IV-8 : Solutions optimales obtenues par
L'optimisation vectorielle non Pareto

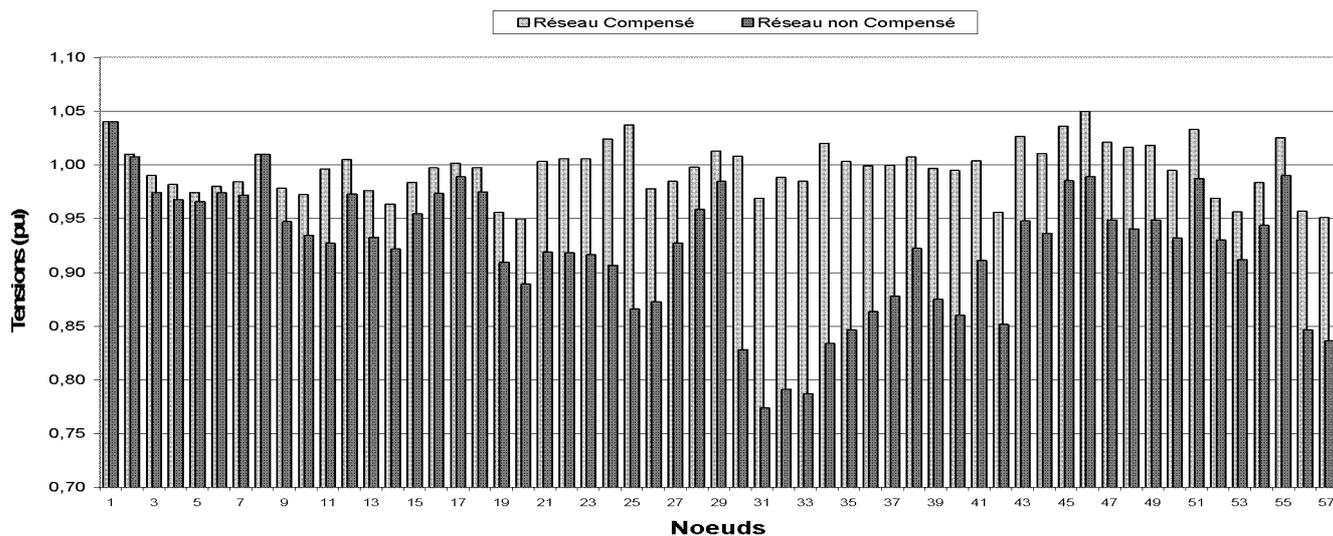


Figure IV-5 : Profil de tensions obtenues avec une Stratégie d'Evolution
Optimisation vectorielle non Pareto

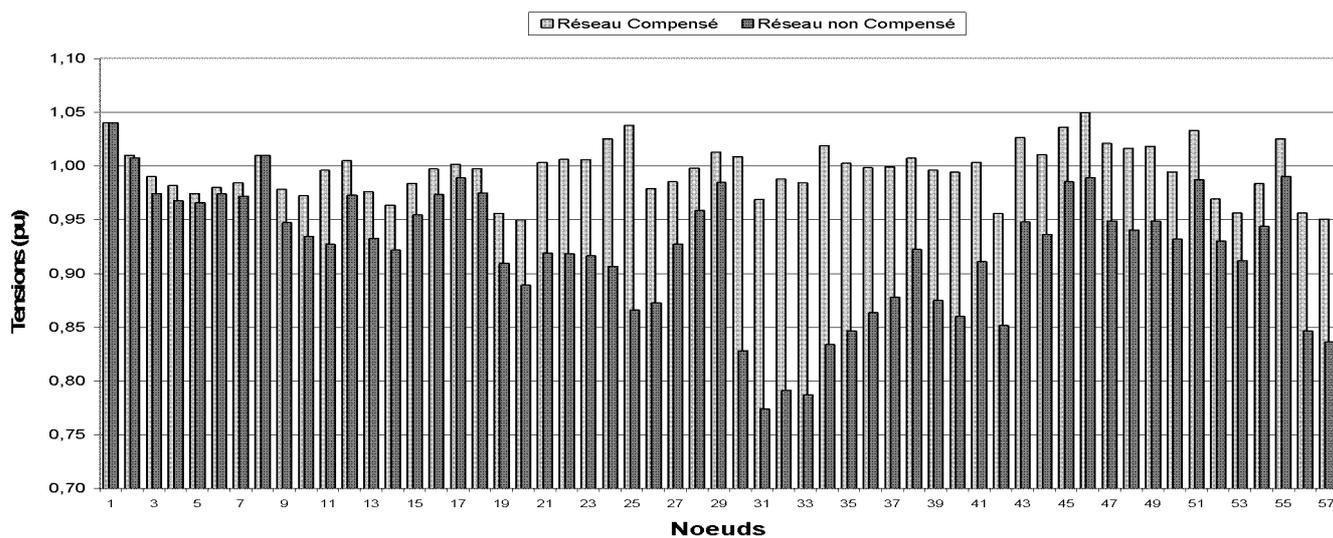


Figure IV-6 : Profil de tensions obtenues avec un Programme Evolutionnaire
Optimisation vectorielle non Pareto

En utilisant cette méthode, nous trouvons que l'installation de 110 MVars permet de réduire les pertes actives de tout en assurant la satisfaction des contraintes, du moment où toutes les tensions sont dans leurs limites imposées. Les résultats obtenus par cette méthode sont très proches de ceux trouver par la méthode du vecteur cible.

IV.2.1.3. Optimisation vectorielle au sens de Pareto

La détermination de la configuration optimale est déterminée par la minimisation du système (IV-1), pour cela nous utilisons les Algorithmes suivants :

- **Stratégie d'Evolution** incluant un crowding déterministe, les paramètres de cette méthode sont comme suit :
 - Taille de la population parente : $M = 25$
 - Taille de la population descendante : $L = 50$
 - Nombre de max génération : $NMAX = 100$
 - Recombinaison intermédiaire.
 - Mutation gaussienne, avec : $\sigma = 0.03 (q_{max} - q_{min})$
 - Sélection SE (2+2) déterministe.

La solution obtenue après la convergence de cet Algorithme, ainsi que les résultats d'écoulement de puissance, sont données par le tableau IV-8, et la figure IV-11, respectivement.

- **Programme Evolutionnaire** incluant un crowding déterministe, les paramètres de cette méthode sont comme suit :
 - Taille de la population parente : $M = 25$
 - Taille de la population descendante : $L = 50$
 - Nombre de max génération : $NMAX = 150$
 - Mutation gaussienne, avec : $\sigma = 0.03 (q_{max} - q_{min})$
 - Sélection PE (1+1) déterministe.

La solution obtenue après convergence de cet algorithme, ainsi que les résultats d'écoulement de puissance, sont donnés par le tableau IV-8 et la figure IV-12, respectivement.

Nous considérons comme solution optimale, la solution qui minimise le plus la puissance réactive à investir. En se basant sur ces considérations les résultats trouvés sont comme suit :

Méthode	SE avec crowding			PE avec crowding		
	11	24	34	11	24	34
Noeuds candidats						
puissance réactive installée (pu)	0,4821	0,1374	0,4985	0,4792	0,1259	0,5013
Total (pu)	1,1180			1,1064		
pertes actives (pu)	0,6412			0,6405		

Tableau IV-9 : Solutions optimales obtenus par
L'optimisation vectorielle Pareto

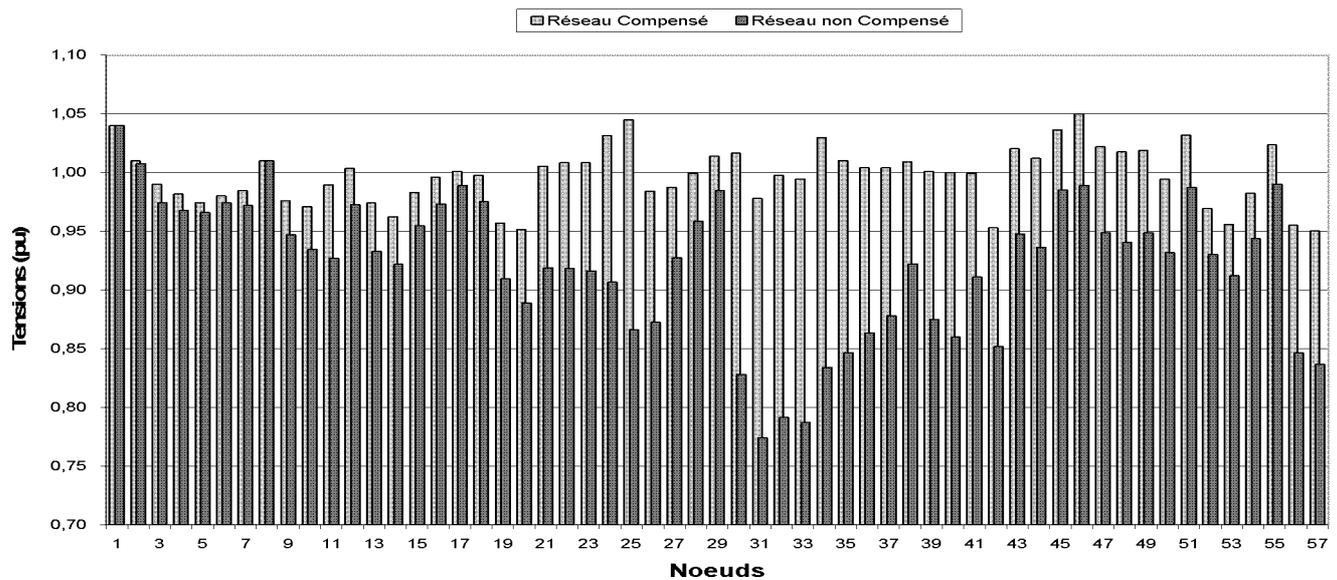


Figure IV-7 : Profil de tensions obtenues avec une Stratégie d'Evolutionnaire Optimisation Vectorielle au sens de Pareto

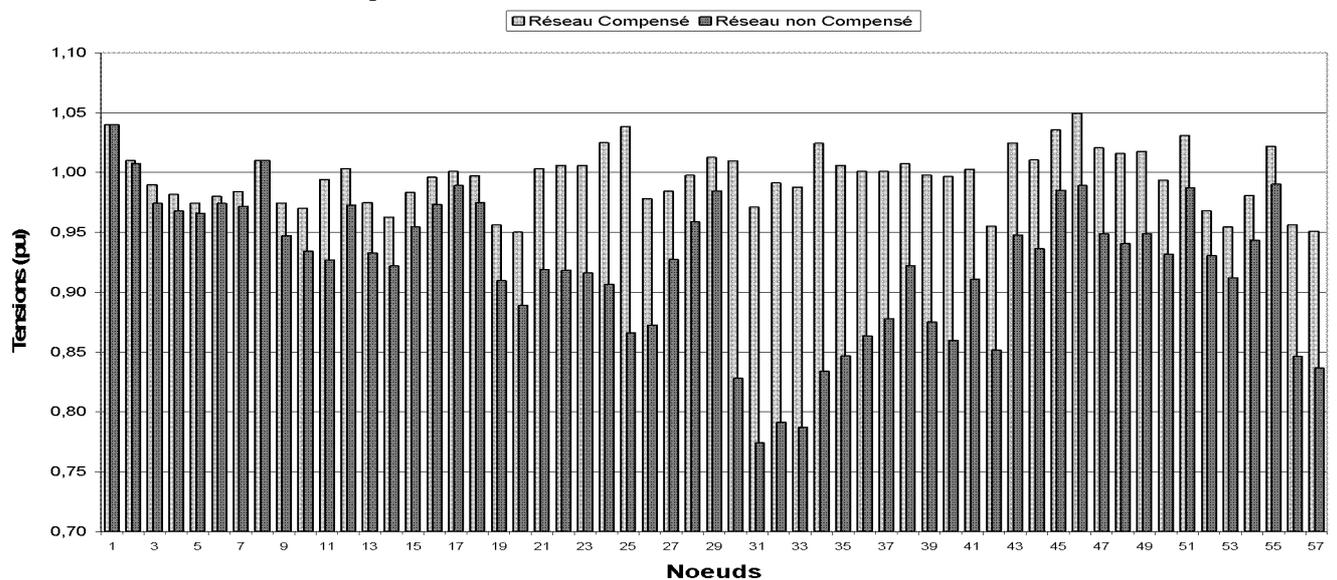


Figure IV-8 : Profil de tensions obtenues avec le Programme Evolutionnaire Optimisation Vectorielle au sens de Pareto

La solution trouvée par cette méthode suggère que l'installation de 110 MVars améliore le fonctionnement du réseau en terme de coût de fonctionnement, les pertes actives sont réduites de, ainsi qu'en terme de satisfaction de contrainte du moment ou toutes les contraintes sont respectées.

IV.2.2. Conclusions

Malgré le fait que le réseau IEEE 57 nœuds présente plus de difficultés vu sa taille, ce qui se répercute sur le nombre et la taille des dispositifs de compensation nécessaires à l'amélioration de son fonctionnement, les résultats obtenus par les différentes méthodes sont assez proches, les écarts ne dépassent pas quelques fractions de %.

Nous pouvons considérer, dans ce cas, que les différents algorithmes ont convergé vers la même solution. Cette solution permet de satisfaire les différents objectifs attendus de cette planification :

- L'amélioration du profil de tension : la configuration de compensateur ainsi trouvée permet d'éliminer les chutes de tension, toutes les tensions étant dans leurs limites inférieures.
- La réduction du coût de fonctionnement par la diminution des pertes actives, l'investissement effectué permet de réduire les pertes actives de 7 %.

Cet investissement permet de surcharger le réseau à 150 % de sa charge nominale, ce qui permet d'augmenter la capacité de transfert du réseau sans avoir à construire d'autre ligne.

IV.3. Application au Réseau Algérien

Pour donner un aspect pratique à notre travail, nous appliquerons les Algorithmes développés pour planifier l'installation d'un compensateur de puissance réactive, afin d'améliorer le fonctionnement du tronçon Saida - Bechar du réseau HT-THT Algérien [31].

Vu les grandes distances entre les charges (de 200 à 1000 km) et leur augmentation constante, la connexion des régions de sud constitue un défi technique et économique. A moyen et long terme, les charges arriveront à des niveaux élevés. Pour affronter ce problème, des solutions envisagées par SONELGAZ à savoir [31]:

- L'interconnexion de ces charges au réseau national.
- Trouver des sources locales de combustible quand c'est possible.

Dans le cas de la région de Bechar, aucune source locale de combustible n'est disponible, une étude technico-économique faite par la SONELGAZ montre la nécessité d'interconnecter cette région par le biais d'une ligne Saida Bechar via Ain-Sefra d'une longueur de 520 km sous une tension de 220 kV [31].

Cette ligne, qui prend départ de la sous-station de Saida, est constituée d'un tronçon de 270 km avec une simple ligne de section 411 mm² et un tronçon avec une ligne double de 250 km reliant Ain-Sefra et Bechar. Les données relatives à ce réseau sont disposées dans l'annexe.

A cause de sa longueur, chacune des sections de cette ligne donne lieu à des surtensions aux différents nœuds de cette ligne. Pour montrer l'influence du niveau de charge sur les tensions des nœuds, nous faisons varier les niveaux de charges à Ain-Sefra et à Bechar, en considérant les heures de pointe et les heures creuses, suivant le tableau IV-9, les différents cas suivant sont obtenus [31] :

Nœud	Heure de pointe	Heure creuse
14	0.40	0.20
17	0.70	0.40

Tableau IV-10 : Puissance active installée (pu) pour Différents points de la courbe de charge

- **Cas 1** : Heure de pointe à Ain-Sefra et à Bechar.
- **Cas 2** : Heure creuse à Ain-Sefra et Heure de pointe à Bechar.
- **Cas 3** : Heure de pointe à Ain-Sefra et Heure creuse à Bechar.
- **Cas 4** : Heure creuse à Ain-Sefra et à Bechar.

Le tableau suivant présente les résultats du calcul d'écoulement de puissance pour les différents cas de charge :

Nœud	Tension (pu)			
	cas1	cas2	cas3	cas4
7	1.0668	1.0944	1.1120	1.1271
14	1.0445	1.1875	1.2556	1.3371
15	1.0458	1.1651	1.2446	1.3087
16	1.0196	1.1550	1.2747	1.3429
17	1.0139	1.1671	1.3281	1.4017

Tableau IV-11 : Profil de tensions en fonction du niveau de charge

	cas1	cas2	cas3	cas4
Pertes actives	0.7548	0.7302	0.7317	0.7324

Tableau IV-12 : Pertes actives en fonction du niveau de charge

Nous pouvons constater que la sévérité des sursensions augmente avec la diminution de charge. Le niveau de charge déterminé par le cas 4 donne lieu à des sursensions très élevées qui doivent être corrigées.

Nous voulons, par l'installation d'un dispositif de compensation, réduire ces sursensions et faire diminuer les pertes actives. Pour dimensionner le dispositif à installer nous considérons le cas 4 qui est le plus sévère.

IV.3.1. Détermination de la configuration optimale de compensateur

Par souci de comparaison nous rapportons les résultats présentés par Mr Arif dans sa thèse de Magister [31], développée dans le cadre du Laboratoire des Réseaux Electriques à l'Ecole Nationale Polytechnique. Pour cela le nœud 17 est considéré comme nœud candidat à la compensation.

Les limites de tensions de charge (contraintes dures) sont comme suit :

$$\underline{V} = 0.95 \quad \text{et} \quad \bar{V} = 1.05 \quad (\text{V-3})$$

Les tableaux suivants présentent les résultats obtenus par [31] et par notre algorithme de calcul.

Tensions (p.u)	7	1,0800
	14	1,1220
	15	1,1160
	16	1,0950
	17	1,0640
Puissance réactive investie (p.u)		0,4267
Pertes Actives (p.u)		0,6922

Tableau IV-13 : Solution rapportée par [31]

Les résultats proposés par [31] montrent que l'investissement de 42.67 MVars permet de diminuer les pertes actives de 7%, mais n'assure pas la satisfaction des contraintes, les surtensions restant assez élevées.

Algorithme Evolutionnaire		Approche Vecteur cible		Optimisation Vectorielle non Pareto		Optimisation Vectorielle Pareto	
		SE	PE	SE	PE	SE	PE
Tensions (pu)	7	1,0738	1,0737	1,0738	1,0738	1,0738	1,0738
	14	1,0782	1,0775	1,0782	1,0782	1,0782	1,0782
	15	1,0604	1,0598	1,0605	1,0605	1,0604	1,0605
	16	1,0068	1,0060	1,0068	1,0068	1,0068	1,0068
	17	0,9500	0,9489	0,9501	0,9501	0,9500	0,9501
Puissance réactive investie (pu)		0,4809	0,4814	0,4808	0,4809	0,4809	0,4808
Pertes Actives (pu)		0,6947	0,6948	0,6947	0,6947	0,6947	0,6947

Tableau IV-14 : Solutions obtenues en utilisant
Les différents Algorithmes Evolutionnaires

Les résultats obtenus par les différents Algorithmes Evolutionnaires sont identiques et nous pouvons considérer que les algorithmes ont convergé vers la même solution. Cette solution suggère que l'investissement de 48,1 MVars permet de réduire les pertes actives de 6% et de réduire les surtensions. Les tensions des nœuds 15, 16, et 17 sont dans leurs limites, mais les contraintes sur les tensions des nœuds 7 et 14 restent violées.

Les résultats présentés dans les tableaux IV-12 et IV-13, montrent que la solution proposée par [31] et celle obtenue par l'application des différents Algorithmes Evolutionnaires, ne remplissent pas les objectifs de la planification, du moment où les contraintes dures du système, les contraintes sur les tensions, ne sont pas totalement satisfaites. Mais, il reste que la solution obtenue par l'application des Algorithmes Evolutionnaires offre de meilleures performances pour un investissement plus conséquent.

En analysant les résultats obtenus par les Algorithmes Evolutionnaires, nous pouvons constater que la solution trouvée est la meilleure configuration possible, en posant le nœud 17 comme nœud candidat, du moment que la tension du nœud candidat a atteint sa limite inférieure. Donc, l'installation de plus de puissance réactive en ce nœud n'est possible qu'en violant la contrainte sur la tension.

Donc, même si elles ne sont pas totalement faisables, les solutions obtenues par l'application des Algorithmes Evolutionnaires peuvent être considérées comme les solutions les plus optimales possible au problème ainsi formulé.

L'écart entre les résultats obtenus peut s'expliquer par le fait que la méthode utilisée par [31] est une méthode d'optimisation locale, qui n'assure la convergence que vers l'optimum locale le plus proche. Les Algorithmes Evolutionnaire sont connus pour être des méthodes d'optimisation globale. Par l'exploration parallèle de l'espace de recherche et en évitant toute convergence prématurée vers les optimums locaux, ces Algorithmes assurent la convergence vers l'optimum global, donc, vers de meilleurs résultats.

Les résultats obtenus par les Algorithmes Evolutionnaires restent valables du moment que le niveau de charge, pour lequel le calcul est effectué représente le cas le plus extrême. De plus, l'utilisation de transformateurs à réglage en charge peut limiter les surtensions par la variation du rapport de transformation.

Nous voulons à présent, voir le comportement du réseau en considérant le nœud 15 comme nœud candidat.

Le tableau suivant récapitule les résultats donnés par les différentes méthodes :

Algorithme Evolutionnaire		Approche Vecteur cible		Optimisation Vectorielle non Pareto		Optimisation Vectorielle Pareto	
		SE	PE	SE	PE	SE	PE
Tensions (pu)	7	1,0600	1,0600	1,0600	1,0599	1,0600	1,0600
	14	1,0088	1,0090	1,0089	1,0087	1,0090	1,0092
	15	0,9942	0,9944	0,9943	0,9941	0,9945	0,9946
	16	1,0056	1,0058	1,0057	1,0055	1,0059	1,0060
	17	1,0359	1,0361	1,0361	1,0358	1,0362	1,0364
Puissance réactive investie (pu)		0,7380	0,7378	0,7378	0,7382	0,7376	0,7374
Pertes Actives (pu)		0,6952	0,6952	0,6952	0,6952	0,6952	0,6951

Tableau IV-15 : Solutions obtenues avec les différents Algorithmes Evolutionnaires

Les solutions obtenues par les différents Algorithmes Evolutionnaires sont, dans ce cas aussi, identiques. Les différents algorithmes ont convergé donc vers la même solution. Ces résultats montrent que l'installation d'un compensateur de 74 MVars au nœud 15, permet de préserver toutes les contraintes puisque toutes les tensions sont dans leurs limites imposées, avec des pertes actives réduites de 5%.

La solution obtenue peut être considérée comme l'optimum global du fait que la configuration obtenue correspond à la plus petite quantité de puissance réactive nécessaire pour assurer la satisfaction des contraintes sur les tensions. Ceci est dû au fait que la tension du nœud 7 est dans sa limite supérieure, diminuer la quantité de puissance réactive investie conduit à la violation de cette contrainte. Au contraire, augmenter l'investissement n'aura aucun avantage apparent du moment où toutes les tensions sont faisables.

IV.3.2. Conclusions

En comparant les résultats donnés par les deux configurations nous pouvons noter que :

- Investir plus peut améliorer le fonctionnement du réseau, seul l'opérateur peut fixer le volume de l'investissement qu'il peut effectuer en faisant un compromis entre les performances qu'il veut atteindre et le prix à payer.
- Le choix des nœuds candidats à la compensation est un problème critique, du moment où la configuration de compensateurs à installer est définie à la fois par le volume du compensateur et son emplacement.

CONCLUSION GENERALE

CONCLUSION GENERALE

Dans ce travail, nous avons présenté les aspects les plus fondamentaux du Calcul Evolutionnaire liés, principalement, aux Stratégies d'Evolution et à la Programmation Evolutionnaire. Ainsi, les Algorithmes de base et les différents opérateurs en se basant, toujours, sur les méthodes utilisées dans l'optimisation continue.

L'application des Algorithmes Evolutionnaires sur différents types de problèmes d'optimisation continue, ainsi que les différentes approches utilisées pour contourner les différentes difficultés, nous a permis de constater que :

- La convergence vers l'optimum global est assurée, même en présence de multitude d'optima locaux.
- Les Algorithmes Evolutionnaires, permettent un traitement efficace des contraintes sans aucune restriction sur leurs formes ou celle des objectifs.
- Ils fournissent un moyen simple et efficace pour la résolution des problèmes multi-objectifs.

En plus, et en appliquant les différents Algorithmes Evolutionnaires et les méthodes d'évaluation, pour la résolution du problème de l'optimisation de puissance réactives, les différents résultats obtenus nous permettent de constater que le Calcul Evolutionnaire offre des avantages considérables en :

- Permettant une expression naturelle au problème en cours, du moment où aucune analyse n'est effectuée (gradient, lagrangien)
- Fournissant un moyen de contrôle sur les solutions obtenues par l'introduction des butes, dans l'approche du vecteur cible, ou par les fonctions de qualité dans l'optimisation vectorielle. Ce qui permet de diriger la recherche vers les solutions les plus intéressantes.
- L'aspect de convergence globale, permet d'obtenir les solutions les plus optimales.
- La possibilité d'avoir un ensemble de solutions, et d'appliquer des critères de préférences, durant ou après la recherche permet de faciliter la prise de décision.

L'avantage majeur des Algorithmes Evolutionnaires reste leur robustesse, du moment où les mêmes algorithmes peuvent être utilisés pour différents problèmes. Leur inconvénient majeur est leur incapacité à opérer en temps réel, dû au temps de calcul qu'il nécessite. Mais il reste qu'ils sont parfaits dans les applications off-line comme pour les problèmes de planification et de paramétrage.

Notons que les Algorithmes utilisés sont des Algorithmes de base, plusieurs améliorations sont introduites dans le déroulement de l'algorithme ou de ces opérateurs, ces améliorations contribuent à la diminution du temps de calcul, et accélèrent la convergence des Algorithmes Evolutionnaires, sans conduire, pour autant à une convergence prématurée :

- L'utilisation d'une mutation avec covariance non nulle.
- L'utilisation de la sélection continue et la possibilité de rendre l'algorithme plus rapide.
- L'utilisation des méthodes de niches facilitant les implémentations parallèles des Algorithmes Evolutionnaires.

Dans ce travail, nous avons mis l'accent sur l'optimisation de l'écoulement réactif, pour cela, nous nous sommes limités aux seuls effets des compensateurs de puissance réactive, ce travail peut être complété en introduisant :

- L'effet des prises réglables des transformateurs.
- L'impacte des contraintes de sécurité et les modes préventifs et correctifs.
- L'introduction de l'optimisation de l'écoulement de puissance active par une répartition optimale des marges de générations.

Comme perspectives à ce travail, des études peuvent être axées sur deux aspects très importants dans l'optimisation de la planification de la puissance réactive : le choix des nœuds candidats, et l'amélioration du fonctionnement dans les modes dynamiques et transitoires :

Nous avons vu que le choix des nœuds candidats est un élément décisif dans l'optimisation de la configuration des dispositifs de compensation. Les nœuds candidats peuvent être introduits comme paramètre à optimiser en utilisant une hybridation d'un Algorithme Evolutionnaire avec la méthode de recherche tabou (tabou search) par exemple, ou en utilisant un algorithme méta-évolutionnaire.

Dans notre travail, nous nous sommes limités à l'amélioration du fonctionnement du système dans le régime permanent (statique), mais l'impacte des dispositifs de compensation ne se limite pas à ce régime. Les améliorations de la stabilité dynamique et transitoire et l'effet des dispositifs FACTS (SVC, STATCOM, UPFC) peuvent être étudiées.

BIBLIOGRAPHIE

- [01] **David B Fogel**, “*Why Evolutionary Computation*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap A1.1 Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [02] **F. Streichert**, “*Introduction to Evolutionary Algorithms*“, University of Tuebingen, 2002.
- [03] **David B Fogel**, “*Principles of evolutionary processes*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap A2.1. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [04] **Thomas Bäck**, “*Evolutionary Algorithms and Their Standard Instances*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap B1.1. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [05] **Larry J Eshelman**, “*Genetic algorithms*“, *Handbook of Evolutionary Computation*, Oxford University Press, 1997. chap B1.2
- [06] **William Porto**, “*Evolutionary programming*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap B1.4. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [07] **Günter Rudolph**, “*Evolution strategies*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap B1.3. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [08] **F. Kursawe**, “*Towards Self-adapting Evolution Strategies*“, IEEE International Conference on Evolutionary Computation, 1995. vol. I, pp 238-288
- [09] **David B Fogel**, “*Representations: Real-valued vectors*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, Chap C1.3. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [10] **Lashon B Booker**, “*Recombination: Binary strings*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C3.3.1. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [11] **H. Pohlheim** , “*GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB (user guide)*“, 1997
- [12] **David B Fogel**, “*Recombination: Real-valued vectors*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C3.3.2. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [13] **Thomas Bäck**, “*Mutation: Binary strings*“, In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C3.2.1. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.

- [14] **David B Fogel**, “*Mutation: Real-valued vectors*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C3.2.2. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [15] **T.P. Runarsson and X. Yao**, “*Continuous Selection and Self-adaptive Evolution Strategies*”, IEEE Proc of the 2002 Congress on Evolutionary Computation, , 2002.
- [16] **N.Hansen A.Ostermeier**, “*Completely Derandomized Self-Adaptation in Evolution Strategies*”, Massachusetts Institute of Technology, 2001.
- [17] **David B Fogel**, “*selection methods*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C2.6. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [18] **Tobias Blickle**, “*Tournament Selection*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C2.3. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [19] **Alice E Smith and David W Coit**, “*Penalty functions*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chap C5.2. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [20] **P.D Surry, N.J Radcliff, I.D Boyd**, “*A Multi-Objective Approach to Constrained Optimisation: The COMOGA Method*”, Evolution Computing: AISB Workshop, 1995.
- [21] **T.P Runarsson and X. Yao**, “*Stochastic Ranking for Constrained Evolutionary Optimisation*”, IEEE Transactions on Evolutionary Computation, pp 1-17, 2000.
- [22] **C M Fonseca and P J Fleming**, “*Multi-objective optimization*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pp C4.5. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [23] **Carlos A**, “*Evolutionary Multiobjective Optimization: Past, Present and Future*” National Polytechnic Institute, Mexico, 2001.
- [24] **Günter Rudolph**, “*On a Multi-Objective Evolutionary Algorithm and its convergence to the Pareto set*”, IEEE World Congress on Computational Intelligence, pp 512-518, 1998.
- [25] **Samir W Mahfoud**, “*Niching methods*”, In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pp C6.1. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [26] **B. Porter and F. Xue**, “*Niche Evolution Strategy for Global Optimisation*”, IEEE Proc of the 2001 Congress on Evolutionary Computing, pp 1086-1092, 2001
- [27] **T.J.E MILLER**, “*Reactive Power Control in Electric System*”, Ed John Wiley & sons, 1982. pp 353-355.
- [28] **A.A. Ladjici, A. Barar**, “*Introduction aux Systèmes Flexible de Transmission en Courant Alternatif: FACTS*”, PFE, Ecole National Polytechnique, Alger, 2002.

- [29] **S. Arif A. Hellal A. Bensnoui**, “*Méthode de décomposition de type Benders pour une localisation optimale de la compensation dans un réseau électrique*”, Proc of the 2nd CEA, Alger, 1994.
- [30] **M.BELAZZOUG**, “*Répartition Optimale des Sources de Puissance Réactive dans un Réseau Electrique*”, thèse de magistère ENP 2001.
- [31] **S.Arif**, “*Planification de l'énergie réactive dans les Réseaux Electriques. Application au Réseau Algérien* “. Thèse de Magister, Ecole Nationale Polytechnique, Alger, 1995
- [32] **K.Y. Lee and F. F. Yang**, “*Optimal Reactive Power Planning using Evolutionary Algorithms: a comparative study for Evolution Programming, Evolutionary Strategy, Genetic Algorithm and Linear Programming*”, IEEE Transaction on Power System, vol. 13 pp 101-108, 1998.
- [33] **F. L. Alvarado**, “*Solving Power Flow Problems with a MATLAB Implementation of the Power System Application Data Dictionary*”, IEEE Proc of the 32nd Hawaii International Conference on System Sciences, 1999.
- [34] **S. K. Chang and V. Brandwajn**, “*Adjust Solution in Fast Decoupled Load Flow*”, IEEE Transaction on Power Systems, vol.3 pp 726-733, 1988.
- [35] **K.C. Tan, T.H. Lee and E.F. Khor**, “*Evolutionary Algorithms with Goal and Priority Information for Multi-Objective Optimisation*”, IEEE Proceedings of the 1999 Congress on Evolutionary Computation, pp 106-113.. 1999.
- [36] **M.M. Begovic, B. Radibratovic, F.C. Lambert**, “*On Multiobjective Volt-VAR Optimisation in Power Systems*”, IEEE Proc of the 37th Hawaii Conference on System Sciences, 2004.
- [37] **CHARLES A. GROSS**, “*Power System Analysis*”, ed Jhon Wiley & Sons, New York, 1986, pp282-289

ANNEXE

Données du réseau IEEE 57 noeuds

Données des Nœuds

Nœud	Tension (pu)	Zone	Type	PG (MW)	QG (MVAr)	PL (MW)	QL (MVAr)
1	1.0400	0	1	128.90	-16.10	55.00	17.00
2	1.0100	0	2	0.00	-0.80	3.00	88.00
3	0.9900	0	2	40.00	-1.00	41.00	21.00
4	1.0000	0	3	0.00	0.00	0.00	0.00
5	1.0000	0	3	0.00	0.00	13.00	4.00
6	0.9800	0	2	0.00	0.80	75.00	2.00
7	1.0000	0	3	0.00	0.00	0.00	0.00
8	1.0100	0	2	450.00	62.10	150.00	22.00
9	0.9800	0	2	0.00	2.20	121.00	26.00
10	1.0000	0	3	0.00	0.00	5.00	2.00
11	1.0000	0	3	0.00	0.00	0.00	0.00
12	1.0200	0	2	310.00	128.50	377.00	24.00
13	1.0000	0	3	0.00	0.00	18.00	2.30
14	1.0000	0	3	0.00	0.00	10.50	5.30
15	1.0000	0	3	0.00	0.00	22.00	5.00
16	1.0000	0	3	0.00	0.00	43.00	3.00
17	1.0000	0	3	0.00	0.00	42.00	8.00
18	1.0000	0	3	0.00	0.00	27.20	9.80
19	1.0000	0	3	0.00	0.00	3.30	0.60
20	1.0000	0	3	0.00	0.00	2.30	1.00
21	1.0000	0	3	0.00	0.00	0.00	0.00
22	1.0000	0	3	0.00	0.00	0.00	0.00
23	1.0000	0	3	0.00	0.00	6.30	2.10
24	1.0000	0	3	0.00	0.00	0.00	0.00
25	1.0000	0	3	0.00	0.00	6.30	3.20
26	1.0000	0	3	0.00	0.00	0.00	0.00
27	1.0000	0	3	0.00	0.00	9.30	0.50
28	1.0000	0	3	0.00	0.00	4.60	2.30
29	1.0000	0	3	0.00	0.00	17.00	2.60
30	1.0000	0	3	0.00	0.00	3.60	1.80
31	1.0000	0	3	0.00	0.00	5.80	2.90
32	1.0000	0	3	0.00	0.00	1.60	0.80
33	1.0000	0	3	0.00	0.00	3.80	1.90
34	1.0000	0	3	0.00	0.00	0.00	0.00
35	1.0000	0	3	0.00	0.00	6.00	3.00
36	1.0000	0	3	0.00	0.00	0.00	0.00

Nœud	Tension (pu)	Zone	Type	PG (MW)	QG (MVar)	PL (MW)	QL (MVar)
37	1.0000	0	3	0.00	0.00	0.00	0.00
38	1.0000	0	3	0.00	0.00	14.00	7.00
39	1.0000	0	3	0.00	0.00	0.00	0.00
40	1.0000	0	3	0.00	0.00	0.00	0.00
41	1.0000	0	3	0.00	0.00	6.320	3.00
42	1.0000	0	3	0.00	0.00	7.10	4.40
43	1.0000	0	3	0.00	0.00	2.00	1.00
44	1.0000	0	3	0.00	0.00	12.00	1.80
45	1.0000	0	3	0.00	0.00	0.00	0.00
46	1.0000	0	3	0.00	0.00	0.00	0.00
47	1.0000	0	3	0.00	0.00	29.70	11.60
48	1.0000	0	3	0.00	0.00	0.00	0.00
49	1.0000	0	3	0.00	0.00	18.00	8.50
50	1.0000	0	3	0.00	0.00	21.00	10.50
51	1.0000	0	3	0.00	0.00	18.00	5.30
52	1.0000	0	3	0.00	0.00	4.90	2.20
53	1.0000	0	3	0.00	0.00	20.00	10.00
54	1.0000	0	3	0.00	0.00	4.10	1.40
55	1.0000	0	3	0.00	0.00	6.80	3.40
56	1.0000	0	3	0.00	0.00	7.60	2.20
57	1.0000	0	3	0.00	0.00	6.70	2.00

Données des lignes :

From	To	R	X	B
1	2	0.0083	0.0280	0.0645
2	3	0.0298	0.0850	0.0409
3	4	0.0112	0.0366	0.0190
4	5	0.0625	0.1320	0.0129
4	6	0.0430	0.1480	0.0174
6	7	0.0200	0.1020	0.0138
6	8	0.0339	0.1730	0.0235
8	9	0.0099	0.0505	0.0274
9	10	0.0369	0.1679	0.0220
9	11	0.0258	0.0848	0.0109
9	12	0.0648	0.2950	0.0386
9	13	0.0481	0.1580	0.0203
13	14	0.0132	0.0434	0.0055
13	15	0.0269	0.0869	0.0115
1	15	0.0178	0.0910	0.0494
1	16	0.0454	0.2060	0.0273
1	17	0.0238	0.1080	0.0143
3	15	0.0162	0.0530	0.0272

From	To	R	X	B
5	6	0.0302	0.0641	0.0062
7	8	0.0139	0.0712	0.0097
10	12	0.0277	0.1262	0.0164
11	13	0.0223	0.0732	0.0094
12	13	0.0178	0.0580	0.0302
12	16	0.0180	0.0813	0.0108
12	17	0.0397	0.1790	0.0238
14	15	0.0171	0.0547	0.0074
18	19	0.4610	0.6850	0.0000
19	20	0.2830	0.4340	0.0000
21	22	0.0736	0.1170	0.0000
22	23	0.0099	0.0152	0.0000
23	24	0.1660	0.2560	0.0042
26	27	0.1650	0.2540	0.0000
27	28	0.0618	0.0954	0.0000
28	29	0.0418	0.0587	0.0000
25	30	0.1350	0.2020	0.0000
30	31	0.3260	0.4970	0.0000
31	32	0.5070	0.7550	0.0000
32	33	0.0392	0.0360	0.0000
34	35	0.0520	0.0780	0.0016
35	36	0.0430	0.0537	0.0008
36	37	0.0290	0.0366	0.0000
37	38	0.0651	0.1009	0.0010
37	39	0.0239	0.0379	0.0000
36	40	0.0300	0.0466	0.0000
22	38	0.0192	0.0295	0.0000
41	42	0.2070	0.3520	0.0000
41	43	0.0000	0.4120	0.0000
38	44	0.0289	0.0585	0.0010
46	47	0.0230	0.0680	0.0016
47	48	0.0182	0.0233	0.0000
48	49	0.0834	0.1290	0.0024
49	50	0.0801	0.1280	0.0000
50	51	0.1386	0.2200	0.0000
29	52	0.1442	0.1870	0.0000
52	53	0.0762	0.0984	0.0000
53	54	0.1878	0.2320	0.0000
54	55	0.1732	0.2265	0.0000
44	45	0.0624	0.1242	0.0020
56	41	0.5530	0.5490	0.0000
56	42	0.2125	0.3540	0.0000

From	To	R	X	B
57	56	0.1740	0.2600	0.0000
38	49	0.1150	0.1770	0.0015
38	48	0.0312	0.0482	0.0000

Données des Transformateurs

Données des Nœuds de Régulation

From	To	X	Tap
4	18	0.5550	0.9700
4	18	0.4300	0.9780
21	20	0.7767	1.0430
24	25	1.1820	1.0000
24	25	1.2300	1.0000
24	26	0.0473	1.0430
7	29	0.0648	0.9670
34	32	0.9530	0.9750
11	41	0.7490	0.9550
15	45	0.1042	0.9550
14	46	0.0735	0.9000
10	51	0.0712	0.9300
13	49	0.1910	0.8950
11	43	0.1530	0.9580
40	56	1.1950	0.9580
39	57	1.3550	0.9800
9	55	0.1205	0.9400

Nœud	Tension	Qmax	Qmin
2	1.0100	50.00	-17.00
3	0.9900	60.00	-10.00
6	0.9800	25.00	-8.00
8	1.0100	200.00	-140.00
9	0.9800	9.00	-3.00
12	1.0200	150.00	-150.00

Données des Condensateurs Statique

Nœud	Qc
18	0.10
25	0.10
53	0.10

Données du réseau Algérien

IMPEDANCE AND LINE CHARGING DATA

Line designation		Resistance p.u	Reactance p.u	Line charging p.u
14	15	0.00360	0.16320	0.00000
15	16	0.05010	0.23650	0.17840
15	16	0.05010	0.23650	0.17840
15	07	0.04640	0.21900	0.16520
17	16	0.00240	0.14840	0.00000
02	04	0.00740	0.05060	0.07860
02	11	0.00880	0.05690	0.08840
04	03	0.00850	0.04030	0.03030
04	10	0.01400	0.04980	0.03550
01	18	0.02740	0.12950	0.09760
18	01	0.01390	0.09480	0.14740
01	06	0.01840	0.08710	0.06570
01	08	0.01510	0.07140	0.05380
01	11	0.00610	0.02890	0.05180
01	11	0.00740	0.00350	0.02640
01	11	0.00310	0.00209	0.00324
18	26	0.02500	0.09040	0.06490
18	26	0.01830	0.08630	0.06510
18	29	0.00730	0.05060	0.10550
18	06	0.01710	0.06290	0.04540
18	09	0.01700	0.08060	0.06080
12	13	0.00130	0.00450	0.00320
13	05	0.00380	0.01350	0.00970
13	11	0.00510	0.01800	0.01290
07	08	0.01500	0.07090	0.05350
07	09	0.02280	0.10760	0.08110

Line designation		Resistance p.u [*]	Reactance p.u [*]	Line charging p.u [*]
08	10	0.01570	0.07400	0.05580
10	11	0.02880	0.10200	0.07290
30	23	0.01480	0.07010	0.05280
30	46	0.01230	0.08760	0.06600
19	20	0.00650	0.02440	0.01760
19	20	0.00730	0.02780	0.02020
19	24	0.01970	0.07320	0.05300
19	25	0.00470	0.02310	0.15580
19	27	0.00600	0.02960	0.06050
19	28	0.01980	0.07270	0.05240
20	22	0.00560	0.02630	0.01980
20	26	0.02400	0.08610	0.06180
21	29	0.00730	0.05060	0.10550
21	27	0.0093	0.06310	0.13210
21	27	0.0093	0.06310	0.13210
22	26	0.01220	0.05780	0.04360
23	24	0.01950	0.09200	0.06890
23	26	0.01220	0.05780	0.04360
24	38	0.01520	0.04370	0.03860
24	51	0.02130	0.10070	0.07600
25	27	0.00180	0.00860	0.12610
26	29	0.00450	0.03160	0.06580
28	49	0.01780	0.06550	0.04720
32	40	0.01580	0.07440	0.05610
32	31	0.00990	0.04670	0.03520
32	37	0.01670	0.07880	0.05950
31	34	0.03030	0.10750	0.07680
33	34	0.03180	0.11290	0.08080
33	35	0.02240	0.07950	0.05700
33	36	0.00650	0.05030	0.09250
34	40	0.00560	0.02660	0.02000
34	44	0.01110	0.05260	0.03960
34	36	0.01900	0.06750	0.04840

Line designation		Resistance p.u [*]	Reactance p.u [*]	Line charging p.u [*]
35	36	0.00510	0.01800	0.01290
36	50	0.01110	0.08620	0.15850
39	38	0.00150	0.00200	0.00530
38	45	0.02030	0.07000	0.05160
40	41	0.01050	0.04950	0.03730
40	48	0.03090	0.11960	0.08720
40	50	0.01000	0.06950	0.14530
42	43	0.01380	0.06570	0.04950
42	51	0.03000	0.14200	0.10620
41	43	0.01850	0.08760	0.06600
43	57	0.03520	0.16640	0.12550
43	63	0.03710	0.17520	0.13210
44	45	0.02980	0.10780	0.07760
45	48	0.01430	0.05270	0.03800
45	49	0.01230	0.04480	0.03220
45	50	0.00570	0.03980	0.08320
47	46	0.00110	0.06710	0.00000
47	55	0.38440	1.36280	0.00540
46	51	0.03340	0.15770	0.11890
46	53	0.03880	0.18330	0.13820
48	50	0.02220	0.10510	0.07930
48	51	0.02280	0.10760	0.08110
48	51	0.01160	0.08980	0.16500
49	50	0.00930	0.07180	0.13210
52	62	0.03710	0.17520	0.13210
52	53	0.01480	0.07000	0.05280
54	56	0.08540	0.28540	0.00120
54	56	0.08540	0.28540	0.00120
55	54	0.55000	1.04800	0.00420
54	53	0.00120	0.07420	0.00000
57	63	0.01900	0.08960	0.06750
61	58	0.04920	0.16450	0.00070
58	59	0.00180	0.07800	0.00000

Line designation		Resistance p.u [*]	Reactance p.u [*]	Line charging p.u [*]
58	60	0.06090	0.15670	0.00080
58	60	0.06090	0.15670	0.00080
59	62	0.01670	0.07880	0.05950
59	63	0.02970	0.14010	0.10570

* Impedance and line-charging susceptance in p.u on a 100.000 kVA base Line charging one-half of total charging of line

OPERATING CONDITIONS

N°	Bus Name	Voltage	Generation		load	
		pu	MW	MVAr	MW	MVAr
1	Mersat *	1,10	750,00	180,00	124,30	62,10
2	Beni-Saf	1,00	0,00	0,00	54,60	26,10
3	Oujda	1,00	0,00	0,00	0,00	0,00
4	Ghazaouet	1,00	0,00	0,00	34,00	25,00
5	Ravin Blanc	1,00	60,00	40,00	0,00	0,00
6	Relizane	1,00	0,00	0,00	51,10	25,60
7	Saida	1,00	0,00	0,00	74,90	37,50
8	Sidi-Belabès	1,00	0,00	0,00	66,10	33,00
9	Tiaret	1,11	300,00	100,00	119,00	59,50
10	Tlemcen	1,00	0,00	0,00	91,70	45,80
11	Zahana	1,00	0,00	0,00	62,60	31,30
12	Petit lac 1	1,00	60,00	40,00	142,00	71,00
13	Petit lac 2	1,00	0,00	0,00	0,00	0,00
14	Ain Sefra 1	1,00	0,00	0,00	39,00	19,50
15	Ain Sefra 2	1,00	0,00	0,00	0,00	0,00
16	Béchar 1	1,00	0,00	0,00	0,00	0,00
17	Béchar 2	1,00	10,00	0,00	70,00	35,00
18	Oued-Sly	1,00	0,00	84,00	42,00	0,00
19	Alger Est	1,09	490,00	15,00	100,00	60,00
20	Larbaa	1,00	120,00	26,00	252,00	126,00
21	Alger Ouest	1,00	0,00	0,00	0,00	0,00
22	Beni Merad	1,00	0,00	0,00	172,00	86,00
23	Berrouaghia	1,00	0,00	0,00	97,00	48,00
24	Bouira	1,00	0,00	0,00	87,00	43,00
25	Hamma	1,00	110,00	70,00	234,00	117,00
26	Khemis	1,00	0,00	0,00	60,00	30,00
27	Ouled Fayet	1,00	0,00	0,00	68,00	34,00
28	Tizi-ouzou	1,00	0,00	0,00	104,00	52,00
29	Kherba	1,00	0,00	0,00	27,00	14,00

N°	Bus Name	Voltage	Generation		load	
		pu	MW	MVAr	MW	MVAr
30	Ain-Oussera	1,00	0,00	0,00	26,00	12,00
31	El-Aouinet	1,00	0,00	0,00	48,00	24,00
32	Ain Beida	1,00	0,00	0,00	68,00	32,00
33	EL Hadjar	1,00	110,00	69,00	322,00	161,00
34	El Khroub	1,00	0,00	0,00	93,00	46,00
35	Skikda 1	1,00	200,00	50,00	133,00	83,00
36	Skikda 2	1,08	600,00	79,00	0,00	0,00
37	Tebessa	1,00	0,00	0,00	64,00	32,00
38	Akbou 1	1,00	0,00	0,00	0,00	0,00
39	Akbou 2	1,00	0,00	0,00	35,00	17,00
40	Ain M'lila	1,00	0,00	0,00	19,00	45,00
42	Barika	1,00	0,00	0,00	32,00	16,00
43	Biskra	1,00	0,00	0,00	82,00	41,00
44	C.Laid	1,00	0,00	0,00	26,00	12,00
45	Darguinah	1,00	0,00	0,00	36,00	18,00
46	Djelfa 1	1,00	0,00	0,00	0,00	0,00
47	Djelfa 2	1,00	0,00	0,00	56,00	28,00
48	El Hassi	1,00	0,00	0,00	141,00	50,00
49	El Kseur	1,00	0,00	0,00	54,00	26,00
50	Jijel	1,00	500,00	0,00	159,00	79,00
51	M'sila	1,09	550,00	10,00	89,00	44,00
52	Ghardaia	1,00	0,00	0,00	23,00	12,00
53	Tilghemt 1	1,00	100,00	3,00	61,00	30,00
54	Tilghemt 2	1,00	0,00	0,00	0,00	0,00
55	Laghouat	1,00	0,00	0,00	0,00	0,00
56	Hassi-Rmel	1,00	80,00	0,00	32,00	16,00
57	El-Oued	1,00	0,00	0,00	30,00	15,00
58	H.Messaoud nord 1	1,00	80,00	27,00	148,00	74,00
59	H.Messaoud nord 2	1,09	100,00	155,00	0,00	0,00
60	H.Messaoud sud	1,00	28,00	8,00	51,00	76,00
61	H.Messaoud	1,00	16,00	4,00	0,00	0,00
62	Ouargla	1,00	0,00	0,00	25,00	12,00
63	Touggourt	1,00	0,00	0,00	26,00	13,00

* : Swing machine

REGULATED BUS DATA

Bus Number	Voltage magnitude en p.u	Minimum MVar capability	Maximum MVar capability
01	1.100	80	200
09	1.114	00	100
19	1.086	00	200
36	1.082	-50.	200
51	1.091	-50.	450
59	1.091	-50.	155

TRANSFORMER DATA

Transformer designation	Tap setting
14-15	1.020
16-17	1.060
46-47	0.980
53-54	0.980
58-59	0.980