

Optimisation De L'écoulement De Puissance Dans Les Réseaux Electriques En Utilisant La Méthode AIRS (Artificial Immune Recognition System)

RASOLOFONIRINA Tokiniaina Francky¹, RANDRIAMORA Edmond², ANDRIATSIHOARANA Harlin³,
RAKOTONIAINA Solofohery⁴, RAVONIMANANTSOA Ndaohialy Manda-vy⁵

^{1, 2, 3, 4, 5}Université d'Antananarivo, Ecole Doctorale Science technique de l'Innovation et de l'Ingénierie, Antananarivo
101

Auteur correspondant : RASOLOFONIRINA Tokiniaina Francky. E-mail : rasolofonirinatokiniaina35@gmail.com



Résumé : L'optimisation de l'écoulement de puissance optimal (Optimal Power Flow – OPF) est une étape essentielle pour assurer un fonctionnement fiable, économique et efficace des réseaux électriques. En raison de la complexité non linéaire du problème OPF, les méthodes classiques d'optimisation présentent souvent des limitations en termes de convergence et de performances. Dans cette étude, nous proposons l'application de l'algorithme d'immunité artificielle (AIS), inspiré du système immunitaire biologique, pour résoudre le problème OPF. L'approche est testée sur le réseau standard IEEE 30-bus à l'aide de l'outil MATPOWER sous MATLAB. Les performances de l'AIS sont comparées à d'autres algorithmes métaheuristiques afin d'évaluer sa capacité à minimiser les coûts de production, à améliorer la stabilité du système, et à respecter les contraintes opérationnelles. Les résultats obtenus démontrent l'efficacité de l'approche AIS en termes de précision, de rapidité de convergence et de robustesse, confirmant ainsi son potentiel pour les applications dans la gestion optimale des réseaux électriques.

Mots-Clés : Écoulement de puissance optimal, Algorithme d'immunité artificielle, Réseaux électriques, Optimisation métaheuristique, IEEE 30-bus, MATPOWER

Abstract: Optimal power flow (OPF) optimization is a critical step to ensure reliable, economical, and efficient operation of power grids. Due to the nonlinear complexity of the OPF problem, conventional optimization methods often have limitations in terms of convergence and performance. In this study, we propose the application of the artificial immunity algorithm (AIS), inspired by the biological immune system, to solve the OPF problem. The approach is tested on the standard IEEE 30-bus network using the MATPOWER tool in MATLAB. The performance of AIS is compared with other metaheuristic algorithms to evaluate its ability to minimize production costs, improve system stability, and meet operational constraints. The results obtained demonstrate the effectiveness of the AIS approach in terms of accuracy, convergence speed, and robustness, thus confirming its potential for applications in the optimal management of power grids.

Keywords: Optimal power flow, Artificial immunity algorithm, Power grids, Metaheuristic optimization, IEEE 30-bus, MATPOWER

I INTRODUCTION

L'accroissement de la demande énergétique mondiale, combiné à l'essor des sources d'énergies renouvelables et aux impératifs environnementaux, complexifie la gestion optimale des réseaux électriques. Pour garantir un fonctionnement stable, économique et respectueux des contraintes techniques, les opérateurs doivent continuellement optimiser les flux de puissance à travers le réseau. L'écoulement de puissance optimal (Optimal Power Flow – OPF) est un outil central pour répondre à ces enjeux. Il permet de déterminer la meilleure configuration de fonctionnement du système électrique, en minimisant un objectif (tel que le coût de production ou les pertes) tout en respectant les contraintes physiques des générateurs, des lignes et des charges[1] [2].

Traditionnellement, des méthodes déterministes ont été utilisées pour résoudre le problème OPF. Toutefois, en raison de la nature non linéaire, non convexe et multi-objectifs du problème, ces approches souffrent souvent de lenteur, de manque de précision, ou d'incapacité à sortir des minima locaux. Ces limitations ont conduit à un intérêt croissant pour les algorithmes dits « métaheuristiques », capables d'explorer efficacement l'espace de recherche et d'atteindre de meilleures solutions en un temps raisonnable[3].

Parmi ces approches, l'algorithme d'immunité artificielle (AIS) se distingue par son inspiration biologique issue du fonctionnement du système immunitaire humain. L'AIS combine les concepts de reconnaissance, de sélection clonale, et de mémoire adaptative, offrant une robustesse naturelle face à des environnements complexes et dynamiques comme celui du réseau électrique[4] [5].

Face à telle situation, Comment optimiser l'écoulement de puissance dans un réseau électrique complexe tout en minimisant les coûts et en garantissant le respect des contraintes opérationnelles ? Les méthodes classiques étant limitées dans leur capacité à gérer la non-linéarité et la complexité du problème OPF, il devient nécessaire d'explorer des approches alternatives. L'application de l'algorithme d'immunité artificielle (AIS) peut-elle améliorer l'efficacité et la précision de l'optimisation OPF, en comparaison avec d'autres méthodes métaheuristiques ?

Dans ce travail, nous proposons d'appliquer l'AIS à la résolution du problème OPF sur le réseau test IEEE 30-bus à l'aide de MATPOWER sous MATLAB. L'objectif est de démontrer que cet algorithme peut fournir des solutions optimales ou quasi-optimales avec une bonne convergence, tout en respectant les contraintes du système.

II REVUE DE LA LITTERATURE

II.1 Le problème d'écoulement de puissance optimal (OPF)

L'écoulement de puissance optimal (OPF) est un problème central dans la planification et l'exploitation des réseaux électriques. Introduit par Carpentier en 1962, il consiste à déterminer les réglages optimaux des variables de commande (puissances actives des générateurs, tensions, angles de phase, positions des transformateurs, etc.) afin de minimiser une fonction objective, généralement le coût de génération, tout en respectant un ensemble de contraintes techniques (équilibres de puissance, limites de tension, capacités des lignes, etc.) [35].

Les méthodes de résolution traditionnelles, telles que la méthode de Newton-Raphson, la méthode du Lagrangien augmenté, ou encore les techniques de points intérieurs, ont montré leur efficacité dans certains cas. Toutefois, en raison de la nature non linéaire, non convexe et fortement contrainte du problème OPF, ces approches déterministes souffrent souvent de limitations, notamment leur sensibilité aux conditions initiales et leur propension à rester bloquées dans des minima locaux [34].

Six méthodes de techniques d'optimisations classiques pour résoudre le problème d'OPF sont classifiées par ordre d'introduction[2][17] :

La méthode de Newton ;

La programmation linéaire ;

La programmation quadratique ;

La programmation non linéaire ;

La méthode hybride de programmation linéaire et programmation intégrale ;

La méthode de point intérieur.

II.2 Métaheuristiques appliquées au OPF

Pour pallier les limites des méthodes classiques, de nombreuses recherches se sont tournées vers les algorithmes dits "métaheuristiques", inspirés des comportements biologiques ou physiques [14][33]. Parmi les plus connus, on retrouve :

- L'**algorithme génétique (GA)**, basé sur la sélection naturelle, offre une bonne robustesse, mais peut souffrir de lenteur en convergence [18][23].
- L'**optimisation par essaim de particules (PSO)**, inspirée du comportement social des oiseaux, est appréciée pour sa rapidité, mais reste sensible aux réglages des paramètres.
- D'autres approches comme les algorithmes de colonies de fourmis (ACO), des chauves-souris (BA), ou encore des lucioles (FA), ont également été explorées avec succès dans la résolution du OPF.

Ces méthodes présentent l'avantage de ne pas dépendre du gradient, ce qui les rend adaptées aux problèmes non dérivables ou discontinus. Cependant, leur performance repose fortement sur l'équilibre entre exploration (diversité des solutions) et exploitation (intensification autour des meilleures solutions)[6] [1][14].

II.3 L'algorithme d'immunité artificielle (AIS)

L'algorithme d'immunité artificielle (Artificial Immune System – AIS) est un paradigme d'intelligence computationnelle inspiré des mécanismes du système immunitaire biologique humain. Ce dernier possède une capacité remarquable à détecter, mémoriser et neutraliser les agents pathogènes grâce à des processus comme la sélection clonale, la mémoire immunitaire, et l'hypermutation.

En informatique, plusieurs variantes de l'AIS ont été développées :

- **CLONALG** (Clonal Selection Algorithm), basé sur la reproduction clonale et la mutation adaptative,
- **AIRS** (Artificial Immune Recognition System), qui combine reconnaissance et apprentissage par mémoire,
- **Opt-AIS**, adapté pour les problèmes d'optimisation à contraintes.

Ces algorithmes sont capables d'explorer efficacement de vastes espaces de recherche grâce à leur diversité et à leur capacité d'adaptation. Ils présentent notamment une bonne robustesse face aux fonctions non convexes et discontinues, ce qui en fait des candidats sérieux pour la résolution du OPF[4] [5].

II.4 Applications de l'AIS à l'OPF

Malgré le potentiel théorique de l'AIS, son application dans le domaine du OPF reste encore limitée. Quelques travaux notables méritent d'être mentionnés :

- Zhang et al. (2014) ont appliqué l'algorithme CLONALG à un réseau IEEE 14-bus et ont obtenu des résultats compétitifs par rapport aux autres méthodes métaheuristiques.
- H. Gupta et al. (2016) ont utilisé AIS pour la minimisation simultanée du coût et des émissions dans le cadre d'un OPF multiobjectif.
- Dans d'autres domaines connexes comme la classification ou la détection de fautes, AIS a déjà démontré une grande capacité de généralisation.

Cependant, dans la littérature actuelle, très peu de recherches ont exploré l'application de l'AIS aux grands systèmes de puissance (comme les réseaux IEEE 30, 57 ou 118-bus), ni son intégration avec des scénarios réalistes comprenant des incertitudes de charge ou des sources d'énergie renouvelable. Cette lacune scientifique ouvre la voie à de nouvelles recherches.

Ainsi, ce travail propose d'exploiter le potentiel de l'AIS pour résoudre le problème OPF, en visant une amélioration de la qualité des solutions, de la vitesse de convergence et de la stabilité, tout en comparant ses performances à celles des métaheuristiques établies.

III CONTEXTE THEORIQUE ET METHODOLOGIQUE

III.1 Modélisation du problème de l'Écoulement de puissance optimal OPF

III.1.1 Introduction

L'objectif est d'optimiser la distribution de l'énergie entre les différentes centrales électriques, en minimisant les coûts tout en tenant compte des limites de production et de la capacité de transport. Ce chapitre vise à démontrer comment les méthodes d'optimisation traditionnelles peuvent être utilisées pour atteindre des coûts de production minimaux.

III.1.2 Caractéristiques des systèmes électriques

La fonction coût a une forme non linéaire qui peut être approximée à une courbe quadratique du type $C_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2$ ou P_{Gi} est la quantité produite.

La constante a_i est appelée coût de marche à vide, elle représente le coût pour maintenir la marche d'une unité de production à production nulle. Le coût incrémental (ou marginal) de production est le coût pour produire une unité supplémentaire d'énergie. Ce coût est important pour prendre les décisions d'exploitation à court terme ($\lambda = \frac{dC_i}{dP_{Gi}} = b_i + 2c_i P_{Gi}$) [7][8].

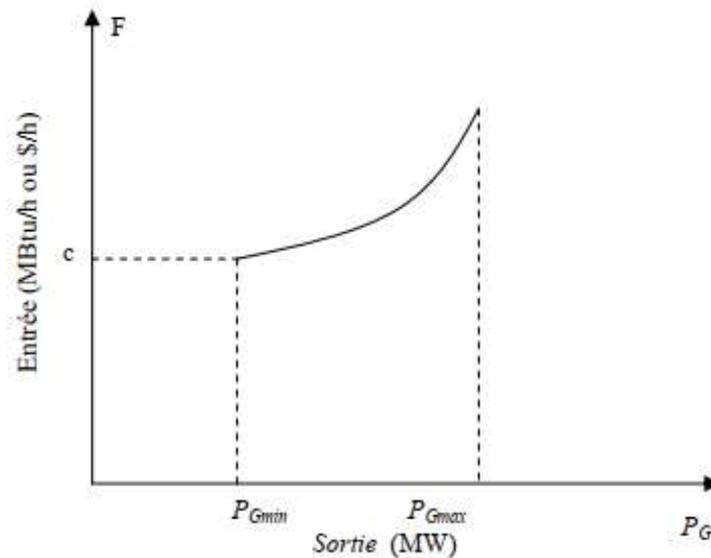


Figure 1: Caractéristique entrée-sortie d'une unité de production

Outre le coût variable à court terme, d'autres aspects sont importants pour la production d'électricité. Cela inclut le coût spécifique de démarrage et d'arrêt des unités de production. Le coût de démarrage est l'énergie nécessaire pour mettre en marche les installations de production. Ce coût dépend de l'état de l'unité (démarrage à froid ou à chaud). Il y a aussi des contraintes techniques à considérer. L'unité ne peut fonctionner de manière stable qu'entre un niveau de production minimal et maximal. L'inertie des moyens de production limite la vitesse de changement de production, appelée contrainte de rampe. Un temps minimal de démarrage est également nécessaire.

Les unités de production varient en fiabilité et en incertitude. La fiabilité indique la précision de la prévision de capacité, qui peut être affectée par des facteurs comme les conditions climatiques, surtout pour la production éolienne. Des erreurs de prévision peuvent aussi provenir de pannes d'unités.

La flexibilité des moyens de production signifie la vitesse à laquelle ils peuvent changer de production. Les centrales hydrauliques et à combustion sont plus flexibles, tandis que les centrales nucléaires et thermiques sont moins flexibles. Cette flexibilité doit être rapide après un signal donné.

III.1.3 Réseau de transport

Le rôle principal du réseau de transport est de relier les grands centres de consommation aux moyens de production. C'est important parce qu'il n'est pas possible de stocker l'énergie électrique à grande échelle en ce moment. Un réseau de transport doit être géré selon des limites spécifiques pour éviter l'instabilité. Ces limites sont exprimées par des valeurs maximales ou minimales sur des variables comme la fréquence, le flux de puissance et le niveau de tension. Les contraintes de capacité de transport concernent les flux maximaux de puissance dans le réseau, ce qui est crucial car les flux d'électricité sont difficiles à contrôler.

III.1.4 La consommation électrique

Connaître la consommation d'électricité pour l'avenir est essentiel pour gérer le système électrique. Diverses variables comme la température, l'heure, le jour de la semaine et le prix sont utilisées pour déterminer la consommation. L'impact de ces variables dépend des conditions climatiques et des habitudes des consommateurs.

Plus la prévision est faite tôt, moins elle est fiable, car les valeurs des variables peuvent changer. Les prévisions faites quelques jours avant se basent surtout sur les consommations des jours précédents et les prévisions climatiques. Par exemple, une baisse de température en hiver peut entraîner une augmentation significative de la consommation. Même si les prévisions deviennent plus précises à l'approche de la date, des écarts demeurent entre les prévisions et la consommation réelle. Ces écarts peuvent résulter d'erreurs dans les variables ou des simplifications des modèles de prévision.

III.1.5 Fluctuations de la consommation

La consommation d'électricité change tout le temps, selon les saisons, la journée, l'activité quotidienne et la météo. Les différentes façons d'utiliser l'énergie électrique entraînent de grandes variations dans le temps. Sur une demi-heure, ces fluctuations sont cycliques, mais elles peuvent changer rapidement, même de plus de 10 % en une heure. Des variations aléatoires se produisent aussi minute par minute.

III.1.6 Equilibre du système électrique

L'équilibre du système électrique nécessite que la puissance produite soit égale à la puissance consommée plus les pertes sur le réseau. Cela impose une contrainte en temps réel entre la production et la consommation. L'équilibre est crucial car les systèmes à courant alternatif fonctionnent comme une grande machine synchronisée, sujette aux instabilités si cet équilibre est rompu. Maintenir cet équilibre est essentiel, notamment lors d'incidents. Des défaillances peuvent provoquer des phénomènes en cascade, et des écarts légers de fréquence peuvent endommager le système. La règle du N-1, appliquée par les gestionnaires de réseau, vise à garantir le fonctionnement même en cas de défaillance d'un élément.

III.2 Limites thermiques des lignes électriques en fonctionnement normal

Les lignes électriques et les transformateurs ont des limites de transport à cause de l'échauffement des lignes dû à l'effet Joule. Il est crucial de garder le courant en dessous de l'intensité maximale admissible pour éviter des déconnexions.

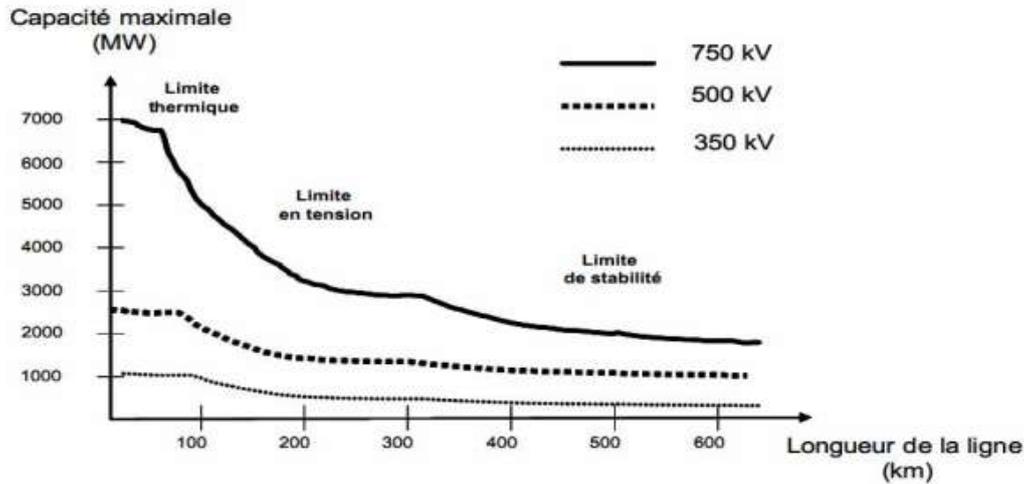


Figure 2: Limites thermiques, de tension et de stabilité de synchronisme des lignes de transport

III.2.1 La tenue de tension

Les limites de tension dans le réseau électrique ne doivent pas être dépassées pour plusieurs raisons. Les limites supérieures de tension sont déterminées par la capacité des équipements et les transformateurs. Dans les réseaux de distribution, une tension élevée peut abîmer les appareils. Les limites inférieures de tension garantissent le bon fonctionnement des appareils industriels et domestiques. Une tension trop basse peut entraîner des surcharges, des instabilités, et des pertes de production.

III.2.2 Modélisation du problème de l'écoulement de puissance optimal

Le problème de la répartition optimale des puissances vise à minimiser le coût total de production dans un réseau électrique. En ne considérant que la fonction objective, c'est une optimisation sans contraintes. Avec les équations de l'écoulement de puissance, c'est une optimisation avec contraintes d'égalités. En incluant les limites des puissances, c'est une optimisation avec contraintes d'égalités et d'inégalités [15][16].

min $F(x)$ (Fonction objective)

Selon

$$g_i(x) = 0 ; i = 1, 2, \dots, n \text{ (contraintes d'égalités)}$$

Et

$$h_j(x) = 0 ; j = 1, 2, \dots, m \text{ (contraintes d'égalités)}$$

Fonction objective :

Cette fonction reflète le besoin de minimiser le coût total de la production des puissances actives. On suppose que le coût individuel de chaque centre de production dépende uniquement de la génération de la puissance active [9][28][29].

$$F = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} C_i = \sum_{i=1}^{ng} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2$$

Contraintes d'égalités :

Ces contraintes sont l'image des lois physiques gouvernant le système électrique. Elles sont représentées par les équations non linéaires de l'écoulement de puissance. Il faut que la somme des puissances active et réactive injectées dans chaque jeu de barres soit égale à zéro [30][31].

$$g_i(x_1, x_2, \dots, x_n) = 0 \quad i = 1, \dots, n$$

$$\Delta P_i = 0 = V_i \sum_{j=1}^n V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) - P_{Gi} + P_{Di}$$

$$\Delta Q_i = 0 = V_i \sum_{j=1}^n V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) - Q_{Gi} + Q_{Di}$$

Contraintes d'inégalités :

En pratique, on ne doit pas dépasser les limites des éléments physiques du réseau électrique tels que les générateurs, les transformateurs à prises de charge, et les transformateurs de phase. En plus des contraintes sur les puissances actives à chaque générateur qui a une influence directe sur la fonction coût, on peut citer d'autres contraintes d'inégalités [11][31]:

- La puissance réactive générée Q_{Gi} qui est limitée par une borne inférieure $Q_{Gi \min}$ et une borne supérieure $Q_{Gi \max}$.

$$Q_{Gi \min} \leq Q_{Gi} \leq Q_{Gi \max} \quad i = 1, \dots, ng$$

- Les transformateurs à prises de charge ont des déviations max. et min. du niveau de tension par rapport à la tension nominale. De même les transformateurs à angles de phase ont des décalages max. et min. des phases des tensions. Les deux types de transformateurs forment les contraintes d'inégalités suivantes [31][32] :

$$\begin{cases} t_{ij \min} \leq t_{ij} \leq t_{ij \max} \\ \alpha_{ij \min} \leq \alpha_{ij} \leq \alpha_{ij \max} \end{cases}$$

- Pour maintenir la sécurité du système électrique, les lignes de transport et les transformateurs de puissances ont des limites sur le transit de puissance apparente. Ces limites sont dues aux pertes thermiques dans les conducteurs, et/ou la stabilité du système. Elles sont représentées par une contrainte d'inégalité, qui limitera le carré de puissance en MVA d'un transformateur ou d'une ligne de transport [28].

$$|S_{ij}|^2 - |S_{ij \max}|^2 \leq 0$$

- Pour garder la qualité de service électrique et la sécurité du système, les niveaux de tension des jeux de barres doivent toujours être entre leurs limites max. et min. Ces limites exigent encore l'addition des contraintes d'inégalités [30].

$$V_{i \min} \leq V_i \leq V_{i \max}$$

Donc il y a n contraintes d'égalités et m contraintes d'inégalités et le nombre des variables du problème est égal à la taille du vecteur des variables de contrôle (y compris puissances active et réactive générées, niveaux de tension des jeux de barres, prises des transformateurs, etc.).

La solution du problème d'OPF exige la formulation de la fonction Lagrangien appelée aussi la fonction de coût augmentée suivante [29][30]:

$$L = F + \sum_{i=1}^n \alpha_i g_i + \sum_{j=1}^m \mu_j h_j$$

Les conditions nécessaires pour trouver un minimum de L appelées conditions de KuhnTucker sont les suivantes [31]:

$$\begin{cases} \frac{\partial L}{\partial x_i} = 0 \text{ et } \frac{\partial L}{\partial \lambda_i} = g_i(x) = 0 & i = 1, \dots, n \\ \frac{\partial L}{\partial \mu_j} = h_j(x) \leq 0 \\ \mu_j h_j(x) = 0 \text{ et } \mu_j > 0 & j = 1, \dots, m \end{cases}$$

III.3 Dispatching économique

Les générateurs à combustibles distincts possèdent différents coûts pour fournir le même montant d'énergie électrique. C'est important de se rendre compte que le générateur le plus efficace du système ne peut pas produire de l'électricité au plus bas coût et qu'un générateur bon marché ne peut pas être le plus rentable. Puisqu'un générateur qui se trouve trop loin du centre de la charge donne des pertes de transmission énormes, et donc le rend peu économique de fonctionner [10].

Le problème est de réduire au minimum le coût de la puissance totale générée par l'ensemble des centrales interconnectées. Ce problème devient plus simple lorsque les limites des puissances de chaque générateur et les pertes dans le réseau sont négligées. Il est décrit comme suit :

$$F = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2$$

$$\text{Et } P_D = \sum_{i=1}^{ng} P_{Gi}$$

Une approche typique consiste à utiliser la méthode de Lagrange :

$$L = F + \lambda [P_D - \sum_{i=1}^{ng} P_{Gi}]$$

$$\frac{\partial L}{\partial P_{Gi}} = \frac{\partial F}{\partial P_{Gi}} + \lambda(0-1) = 0 \rightarrow \frac{\partial F}{\partial P_{Gi}} = \lambda$$

$$\text{Comme } F = \sum_{i=1}^{ng} f_i \text{ Alors } \frac{\partial F}{\partial P_{Gi}} = \frac{\partial f_i}{\partial P_{Gi}} = \lambda \text{ ou } i=1, \dots, ng$$

$$\lambda = \frac{\partial f_i}{\partial P_{Gi}} = \beta_i + 2 \gamma_i P_{Gi}$$

$$\frac{\partial L}{\partial \lambda} = [P_D - \sum_{i=1}^{ng} P_{Gi}] = 0 \text{ donne } \sum_{i=1}^{ng} P_{Gi} = P_D$$

Remplaçant et combinant les équations pour résoudre λ par les étapes suivantes :

De l'équation de λ on détermine la valeur de P_{Gi} comme suit :

$$P_{Gi} = \frac{1}{2 \gamma_i} (\lambda - \beta_i)$$

On a alors,

$$\sum_{i=1}^{ng} P_{Gi} = P_D$$

D'où,

$$\sum_{i=1}^{ng} \frac{(\lambda - \beta_i)}{2 \gamma_i} = P_D$$

On peut tirer alors la valeur de λ

$$\lambda = \left(\sum_{i=1}^{ng} \frac{1}{2 \gamma_i} \right)^{-1} \left(P_D + \sum_{i=1}^{ng} \frac{\beta_i}{2 \gamma_i} \right)$$

La valeur de puissance générée dans chaque jeu de barres est alors :

$$P_{Gi} = \frac{1}{2 \gamma_i} \left(\left(\sum_{i=1}^{ng} \frac{1}{2 \gamma_i} \right)^{-1} \left(P_D + \sum_{i=1}^{ng} \frac{\beta_i}{2 \gamma_i} \right) - \beta_i \right)$$

Cette dernière expression qui nous donne donc l'ensemble des puissances générées minimisant le coût total (contraintes d'inégalité négligées) et constituant notre premier optimum, est applicable s'il n'existe pas de limites sur les puissances générées [12].

III.3.1 Dispatching économique avec des limites sur les puissances générées

Dans le cas où les puissances des générateurs sont limitées par des bornes inférieures $P_{Gi \min}$ et des bornes supérieures $P_{Gi \max}$. Le problème d'optimisation est de la forme [12]:

$$\left\{ \begin{array}{l} \min F = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \\ \min F = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \\ \text{Selon } \sum_{i=1}^{ng} P_{Gi} = P_D \\ P_{Gi \min} \leq P_{Gi} \leq P_{Gi \max} \end{array} \right.$$

Les conditions de Kuhn-Tucker d'optimalité pour ce problème seront données par :

$$\left\{ \begin{array}{l} P_{Gi \min} \leq P_{Gi} \leq P_{Gi \max} \rightarrow \frac{\partial C_i}{\partial P_{Gi}} = \lambda \\ P_{Gi} = P_{Gi \max} \rightarrow \frac{\partial C_i}{\partial P_{Gi}} \leq \lambda \\ P_{Gi} = P_{Gi \min} \rightarrow \frac{\partial C_i}{\partial P_{Gi}} \geq \lambda \end{array} \right.$$

Et l'algorithme de résolution de ce problème est comme suit :

- 1- On calcule la puissance générée de chaque générateur par la formule :

$$P_{Gi} = \frac{1}{2 \gamma_i} \left(\left(\sum_{i=1}^{ng} \frac{1}{2 \gamma_i} \right)^{-1} \left(P_D + \sum_{i=1}^{ng} \frac{\beta_i}{2 \gamma_i} \right) - \beta_i \right)$$

- 2- On vérifie les dépassements des puissances générées :

$$\text{Si } P_{Gk} \geq P_{Gk \max}, P_{Gk} = P_{Gk \max}$$

$$\text{Si } P_{Gk} \leq P_{Gk \min}, P_{Gk} = P_{Gk \min}$$

- 3- On prend la puissance générée qui atteint sa limite min ou max comme une charge i.e :

$$P_{Dk}' = -P_{Gk} \text{ Pour toute puissance générée dépassée } k (k=1, \dots, nk)$$

- 4- On recalculer l'équation de l'équilibre de puissance comme suit :

$$\sum_{i=1}^N P_{Gi} = P_D + \sum_{k=1}^{nk} P_{Dk} \quad \text{Ou bien} \quad \sum_{i=1}^N P_{Gi} = P_D - \sum_{k=1}^{nk} P_{Gk}$$

5- Le processus itératif continue en retournant à l'étape 1 jusqu'à ce que toutes les contraintes soient satisfaites.

Cette méthode est applicable si les pertes dans le réseau sont vraiment négligeables. Sinon elle va nous donner de fausses informations de point de vue coût puisqu'elle va répartir la plupart de la demande sur les générateurs qui ont l'incrément du coût le plus petit malgré que ces générateurs soient les plus éloignés de la charge.

III.3.2 Dispatching économique avec des pertes constantes

Les pertes de puissances dans les lignes de transport varient en fonction de la répartition des puissances entre les centrales et la charge. Ainsi, contrairement à celui sans perte, le dispatching économique avec perte tient compte de la topographie du réseau. Pour pénaliser les centrales qui produisent de la puissance dont le transit provoque d'importantes pertes, nous multiplions leur coût incrémental par un facteur de pénalité. La justification physique de ce facteur de pénalité s'explique par le fait qu'à cause des pertes, il peut être plus intéressant de produire pour plus cher près du lieu de consommation que loin et pour moins cher. Le dispatching économique avec perte est un procédé itératif qui doit converger vers la solution optimale. Si on prend en considération les pertes de puissance constantes, on doit évaluer celles-ci et les inclure dans la demande [5].

$$F(x): \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2$$

$$g(x) : \sum_{i=1}^{ng} P_{Gi} = P_D + P_L$$

$$P_{Gi \min} \leq P_{Gi} \leq P_{Gi \max} \quad i=1, \dots, ng$$

L'équation résultante de l'optimisation :

$$\left\{ \begin{array}{l} L = F + \lambda \left(P_D + P_L - \sum_{i=1}^{ng} P_{Gi} \right) + \sum_{i=1}^{ng} \mu_{i \max} (P_{Gi \max} - P_{Gi}) + \sum_{i=1}^{ng} \mu_{i \min} (P_{Gi} - P_{Gi \min}) \\ P_{Gi} < P_{Gi \max} \rightarrow \mu_{i \max} = 0 \\ P_{Gi} > P_{Gi \min} \rightarrow \mu_{i \min} = 0 \end{array} \right.$$

Les conditions nécessaires pour trouver le minimum sont :

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial P_{Gi}} = 0 \\ \frac{\partial L}{\partial \lambda} = 0 \\ \frac{\partial L}{\partial \mu_{i(\max)}} = P_{Gi \max} - P_{Gi} = 0 \\ \frac{\partial L}{\partial \mu_{i(\min)}} = P_{Gi} - P_{Gi \min} = 0 \end{array} \right.$$

Quand les limites du générateur ne sont pas satisfaites :

$$\frac{\partial L}{\partial P_{Gi}} = 0 = \frac{\partial F}{\partial P_{Gi}} + \lambda \left(0 + \frac{\partial P_L}{\partial P_{Gi}} - 1 \right)$$

$$\frac{\partial F}{\partial P_{Gi}} = \frac{\partial}{\partial P_{Gi}} (f_1 + f_2 + \dots + f_{ng}) = \frac{\partial f_i}{\partial P_{Gi}}$$

$$\lambda = \frac{\partial f_i}{\partial P_{Gi}} + \lambda \frac{\partial P_L}{\partial P_{Gi}} = \left(\frac{1}{1 - \frac{\partial P_L}{\partial P_{Gi}}} \right) \frac{\partial f_i}{\partial P_{Gi}} = L_i \frac{\partial f_i}{\partial P_{Gi}} \text{ avec } L_i = \frac{1}{1 - \frac{\partial P_L}{\partial P_{Gi}}}$$

$$\frac{\partial L}{\partial \lambda} = 0 = P_D + P_L - \sum_{i=1}^{ng} P_{Gi}$$

$$\sum_{i=1}^{ng} P_{Gi} = P_D + P_L$$

Et l'algorithme de résolution de problème qui a été utilisé dans le problème sans pertes peut être utilisé dans ce problème, seulement on va modifier la puissance générée comme suit :

$$P_{Gi} = \frac{1}{2 \gamma_i} \left(\left(\sum_{i=1}^{ng} \frac{1}{2 \gamma_i} \right)^{-1} \left(P_D + P_L + \sum_{i=1}^{ng} \frac{\beta_i}{2 \gamma_i} \right) - \beta_i \right)$$

III.3.3 Dispatching économique avec les pertes en fonction des puissances générées

Dans les réseaux électriques réels les générateurs sont situés loin du centre de la charge électrique, alors les pertes de transport deviennent importantes. La forme la plus simple de ces pertes est :

$$P_L = \sum_{i=1}^{ng} \sum_{j=1}^{ng} P_{Gi} b_{ij} P_{Gj}$$

Une deuxième forme plus précise dite la formule de Kron est la suivante [12] [13]:

$$P_L = \sum_{i=1}^{ng} \sum_{j=1}^{ng} P_{Gi} b_{ij} P_{Gj} + \sum_{j=1}^{ng} b_{0j} P_{Gj} + b_{00}$$

Avec les b_{ij} sont les coefficients des Pertes, souvent supposés constants (en MW^{-1}). Le facteur de pénalité L_i est en fonction de l'accroissement de perte de transmission :

$$L_i = \left(1 - \frac{\partial P_L}{\partial P_{Gi}} \right)^{-1}$$

Le minimum du coût est obtenu quand l'accroissement du coût de chaque centrale multipliée par son facteur de pénalité est le même pour toutes les centrales de production en service.

$$P_L = \sum_{i=1}^{ng} \sum_{j=1}^{ng} P_{Gi} b_{ij} P_{Gj} + \sum_{j=1}^{ng} b_{0j} P_{Gj} + b_{00}$$

$$\frac{\partial P_L}{\partial P_{Gi}} = 2 \sum_{j=1}^{ng} b_{ij} P_{Gj} + b_{0i}$$

$$\frac{\partial f_i}{\partial P_{Gi}} = \beta_i + 2 \gamma_i P_{Gi}$$

$$\lambda = \frac{\partial f_i}{\partial P_{Gi}} + \lambda \frac{\partial P_L}{\partial P_{Gi}} = \beta_i + 2 \gamma_i P_{Gi} + 2 \sum_{j=1}^{ng} b_{ij} P_{Gj} + b_{0i}$$

on a:

$$\sum_{j=1}^{ng} b_{ij} P_{Gj} = b_{ii} P_{Gi} + \sum_{j=1, j \neq i}^{ng} b_{ij} P_{Gj}$$

On réarrange l'équation et on aura :

$$\left(\frac{\gamma_i}{\lambda} + b_{ii} \right) P_{Gi} + \sum_{j=1, j \neq i}^{ng} b_{ij} P_{Gj} = \frac{1}{2} \left(1 - b_{0i} - \frac{\beta_i}{\lambda} \right)$$

On peut écrire :

$$\begin{bmatrix} \frac{\gamma_1}{\lambda} + b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & \frac{\gamma_2}{\lambda} + b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & \frac{\gamma_n}{\lambda} + b_{nn} \end{bmatrix} \begin{bmatrix} P_{G1} \\ P_{G2} \\ \vdots \\ P_{Gn} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 - b_{01} - \frac{\beta_1}{\lambda} \\ 1 - b_{02} - \frac{\beta_2}{\lambda} \\ \vdots \\ 1 - b_{0n} - \frac{\beta_n}{\lambda} \end{bmatrix}$$

Si ces facteurs de pénalité sont calculés qu'une seule fois avant d'exécuter le dispatching en utilisant les informations prise de l'écoulement de puissance les solutions trouvées certes sont très proches de la solution optimale mais elles ne prennent pas vraiment avec exactitude l'effet des pertes avec la variation de la puissance générée d'où l'intérêt de la mise à jour du calcul de coefficient B et le recalcul du dispatching jusqu'à ce que la variation de la puissance du jeu de barres de référence sera négligeable.

III.4 Le réseau IEEE 30-bus

III.4.1 Description des Données Utilisées

L'étude a été menée sur le système standard IEEE à 30 nœuds, largement utilisé comme cas de test pour l'évaluation des algorithmes d'optimisation de l'écoulement de puissance. Le fichier de données comprend plusieurs matrices représentatives des différents composants du réseau électrique, nécessaires à la formulation du problème OPF[1].

Les principales structures de données extraites sont décrites ci-dessous :

- **Matpower Case (mpc)** : Il s'agit de la structure principale contenant toutes les informations du réseau, conformément au format de Matpower.
- **mpc.bus** : Cette matrice décrit les caractéristiques de chaque nœud (bus) du réseau. Chaque ligne correspond à un nœud et comprend notamment :
 - Le type de bus (1 = charge, 2 = PV, 3 = slack),
 - La demande active et réactive (Pd, Qd),
 - Les limites de tension (Vmin, Vmax),
 - L'angle de phase et la magnitude de la tension (θ , V).
- **mpc.gen** : Cette matrice contient les données relatives aux générateurs, y compris :
 - Le nœud d'injection,
 - La puissance active et réactive générée (Pg, Qg),
 - Les limites supérieures et inférieures (Pmax, Pmin, Qmax, Qmin),
 - La tension au terminal.
- **mpc.branch** : Cette matrice détaille les caractéristiques des lignes de transmission, notamment :
 - Les nœuds d'origine et de destination,
 - L'impédance (résistance R, réactance X) et la susceptance,
 - La capacité thermique maximale (rateA).
- **mpc.genconst** : Cette matrice contient les coefficients des fonctions de coût quadratiques associées à chaque générateur. Elle est utilisée dans la fonction objectif pour minimiser le coût total de production.

Ces données servent d'entrée pour la résolution du problème OPF, où l'objectif principal est de minimiser le coût de génération tout en respectant les contraintes physiques et opérationnelles du réseau électrique (équilibre puissance, limites de tension, capacité de ligne, etc.). Le fichier intègre également les résultats de simulation, notamment la distribution des puissances, les pertes dans les lignes et l'état final de chaque bus après optimisation.

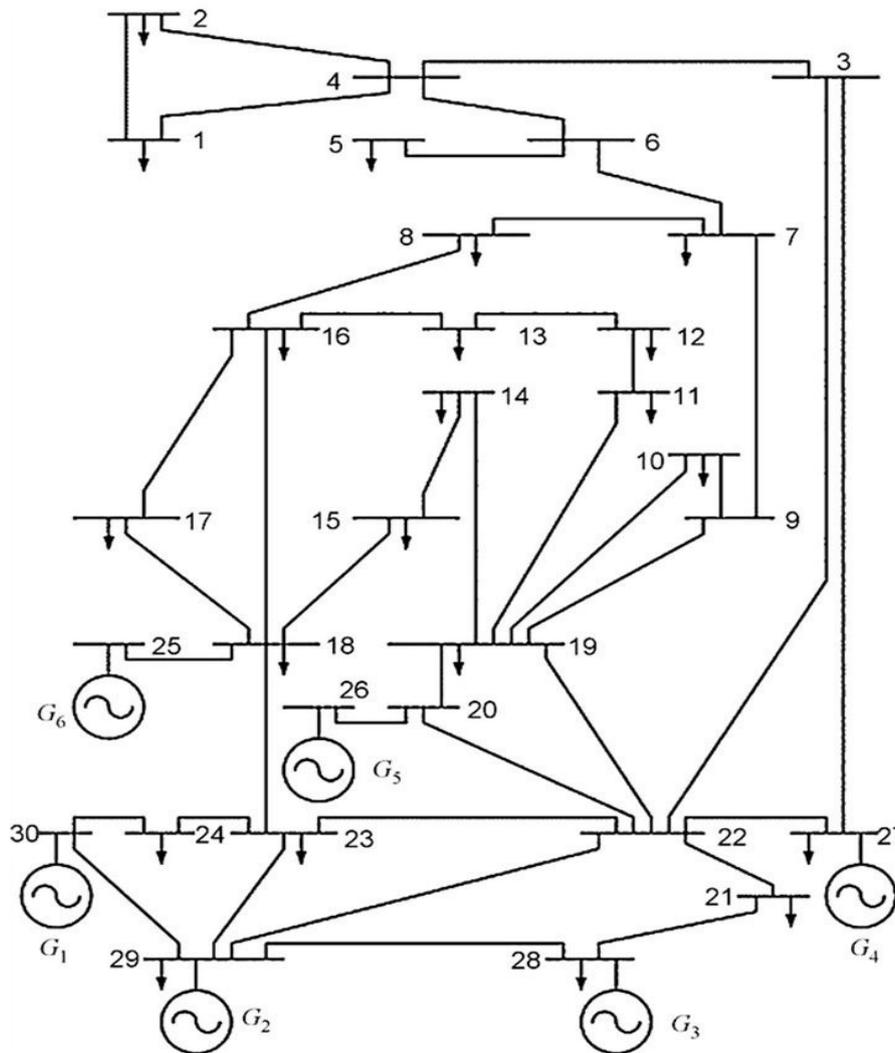


Figure 3: reseau IEEE 30 Bus

III.5 Optimisation de l'écoulement de puissance par le Matpower sur le réseau IEEE 30 jeux de barres

Le logiciel Matpower utilisé pour résoudre le problème de l'écoulement de puissance optimal est basé sur la méthode de point intérieur.

III.5.1 Méthode de type Point Intérieur

A l'origine, les méthodes de type « Point Intérieur » ont été conçues pour résoudre les problèmes de programmation non linéaire. Des recherches plus approfondies sur ces méthodes ont montré qu'elles donnaient de très bonnes performances en termes de vitesse de convergence pour les problèmes de grande échelle. L'algorithme présenté dans cette section, connu sous le nom d'« algorithme primal-dual » est l'un des plus utilisés. Le principe de cette méthode est de rajouter à la fonction objective une fonction logarithmique « barrière » incluant des contraintes et qui décroît progressivement au fil de l'optimisation pour tendre vers 0. Typiquement, considérons un problème de la forme :

$$\text{Min } f(x) \text{ avec } h(x) \geq 0$$

On peut théoriquement transformer ce problème contraint, en incorporant les contraintes d'inégalités dans la fonction objective, en un problème non contraint :

$$\text{Min } f_u(x, \mu^k) \text{ avec } f_u(x, \mu^k) = f(x) - \mu^k \sum_i \ln h_i(x)$$

Où $\mu^k > 0$ est un paramètre de pénalisation qui tend vers 0 au fil des itérations par remise à jour appropriée. Le choix de la valeur initiale de μ^0 ainsi que sa procédure de remise à jour doivent être choisis de manière judicieuse pour éviter les problèmes de divergence.

III.6 METHODE AIS :

En ingénierie, de nouveaux défis exigent des approches innovantes, car les méthodes traditionnelles ne suffisent pas toujours. Les solutions novatrices s'inspirent souvent de techniques mathématiques ou de systèmes naturels. L'imitation des mécanismes vivants aide à résoudre les problèmes d'optimisation [21][22].

III.6.1 Méthodes Métaheuristiques

L'optimisation de l'écoulement de puissance dans les réseaux électriques est un défi complexe, surtout avec la nécessité de réduire les émissions de gaz à effet de serre et d'intégrer les énergies renouvelables. Les méthodes classiques d'optimisation sont souvent inadaptées à ces nouvelles exigences. Cela a conduit au développement des métaheuristiques, qui sont des approches d'optimisation offrant flexibilité et adaptées à des problèmes divers. Introduites dans les années 1980, ces techniques permettent de trouver des solutions satisfaisantes rapidement, même si elles ne garantissent pas toujours l'optimalité. Elles se divisent en algorithmes spécifiques pour des domaines particuliers et en algorithmes génériques [18][21][22].

Les métaheuristiques s'inspirent souvent de systèmes naturels, comme le recuit simulé ou les algorithmes génétiques. Elles montrent leur efficacité dans l'optimisation mono-objectif, mais sont aussi utilisées pour des problèmes d'optimisation multi-objectif, tenant compte de plusieurs contraintes dans les réseaux électriques.

Les systèmes immunitaires artificiels (AIS) sont un concept bio-inspiré basé sur le système immunitaire humain. Ils ont été utilisés dans divers domaines tels que la reconnaissance de motifs et la cybersécurité. Ce chapitre présente d'abord les bases biologiques du système immunitaire, qui protège l'organisme, et ses mécanismes de défense. L'immunité se compose de deux types : naturelle, avec des barrières physiques et moléculaires, et acquise, qui est spécifique et diversifiée. Le système immunitaire comprend plusieurs organes, principalement la moelle osseuse et le thymus, où les lymphocytes T et B se développent et se différencient. Les lymphocytes T sortant du thymus sont de différents types, ayant des fonctions précises dans la réponse immunitaire[4] [5].

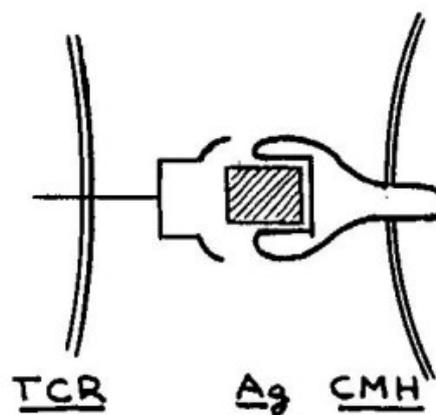


Figure 4: Structure de base d'une Lymphocyte T

Les organes lymphoïdes secondaires, comme les ganglions et la rate, sont les endroits où les antigènes interagissent avec les lymphocytes. Les cellules du système immunitaire comprennent les lymphocytes (B, T, NK), les macrophages, les cellules mononucléées, les cellules dendritiques et les granulocytes. Les lymphocytes T et B sont responsables de la réponse immunitaire adaptative, tandis que les autres cellules jouent des rôles dans l'immunité innée, comme stimuler les lymphocytes et faciliter l'élimination des antigènes.

Les cellules immunitaires utilisent des molécules qu'elles produisent. Certaines molécules agissent comme des protéines de liaison entre les cellules, d'autres agissent sur le site de la réaction immunitaire, et certaines sont des messagers qui diffusent à distance. Les lymphocytes sont essentiels pour l'immunité adaptative et chaque lymphocyte naïf a un récepteur d'antigène unique, déterminé par un mécanisme de réorganisation des gènes durant leur développement.

Les récepteurs des lymphocytes B, appelés immunoglobulines, sont présents sur leur membrane et servent à reconnaître les antigènes. Lorsqu'elles sont produites par les plasmocytes, elles se déplacent dans les liquides biologiques pour aider à la réponse immunitaire humorale. Les immunoglobulines sont composées de chaînes légères et lourdes, avec des domaines variables et constants. La variété des immunoglobulines provient de la diversité des segments géniques et des processus de réarrangement des gènes, qui permettent de créer des anticorps spécifiques.

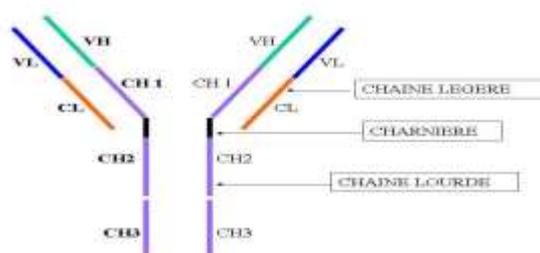


Figure 5: Structure de base d'une immunoglobuline

Le récepteur des lymphocytes T (TCR) est différent du récepteur des lymphocytes B (BCR) sur plusieurs points importants. Le TCR est uniquement présent sur la surface cellulaire, alors que le BCR existe sous forme soluble (anticorps). Le TCR reconnaît un antigène uniquement lorsqu'il est lié à une molécule du complexe majeur d'histocompatibilité (CMH). L'organisation génomique et le réarrangement des segments géniques du TCR sont semblables à ceux des immunoglobulines, permettant une grande diversité de récepteurs.

La sélection positive est cruciale pour la maturation des lymphocytes T. Seuls ceux qui peuvent interagir faiblement avec le CMH continuent leur développement, tandis que les autres subissent apoptose. Ce processus ne retient qu'environ 1 à 2 % des lymphocytes T générés.

Pour les lymphocytes B, la sélection négative se produit lors de leur maturation dans la moelle osseuse, éliminant les cellules qui reconnaissent fortement les antigènes du soi. Pour les lymphocytes T, un processus similaire dans le thymus élimine ceux ayant une affinité trop élevée pour les complexes peptide du soi-CMH, réduisant le risque de maladies auto-immunes.

Selon la théorie de la sélection clonale, chaque lymphocyte a un récepteur unique. Lorsqu'un pathogène entre dans l'organisme, seuls les lymphocytes avec des récepteurs correspondants sont activés et se multiplient. L'activation des lymphocytes B dépend de la nature des antigènes. Après activation, les lymphocytes B produisent des plasmocytes et des cellules mémoire, essentielles pour une réponse immunitaire rapide en cas de réexposition au même antigène.

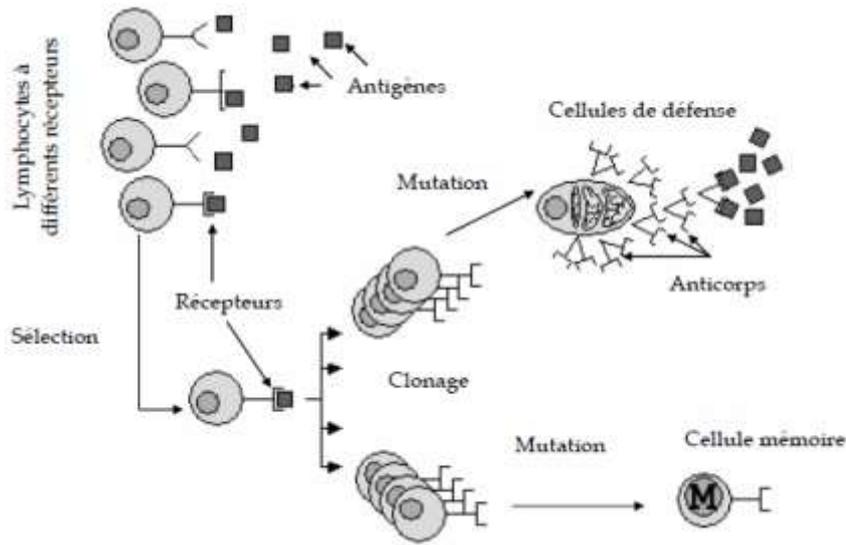


Figure 6: Une simple idée du processus de la sélection clonale.

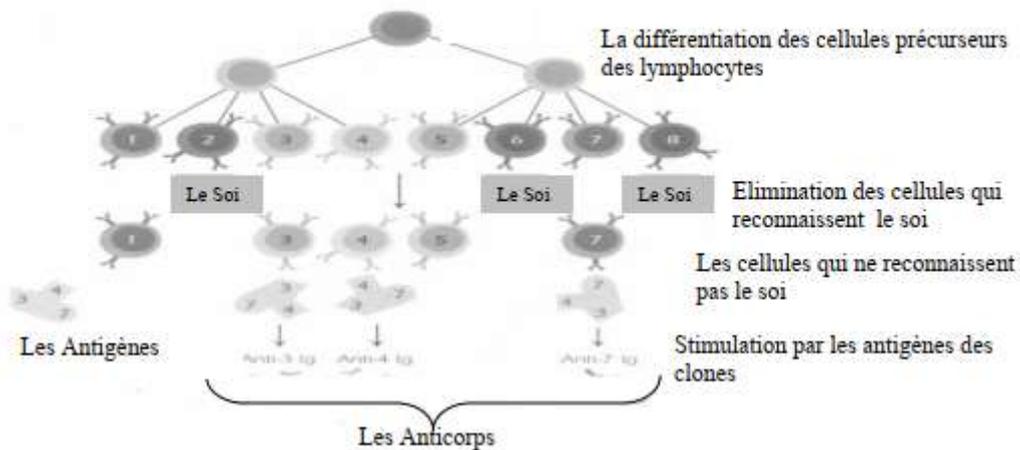


Figure 7: Une idée générale sur le processus de la sélection clonale et la sélection négative.

La théorie des réseaux idiotypiques, imaginée par Jerne en 1974, explique comment chaque anticorps interagit avec d'autres éléments du système immunitaire à travers des sites spécifiques et des déterminants idiotypiques. Lorsqu'un antigène est présent, il stimule la production d'un anticorps, modifiant l'équilibre du système immunitaire et augmentant la mémoire immunologique.

La théorie du danger de Matzinger suggère que le système immunitaire réagit principalement au danger, plutôt qu'au non-soi. Les cellules signalent leur détresse en cas de dommages, ce qui active les lymphocytes et les cellules B dans la zone de risque, sans toucher aux autres cellules. La nature du signal de danger peut être positive ou négative, ce qui remet en question certaines idées classiques sur le soi et le non-soi.

III.6.2 Le système immunitaire artificiel

III.6.2.1 Introduction

Les AIS (Artificial Immune Systems) forment un secteur de recherche assez récent comparé à d'autres modèles de calcul informatique s'inspirant de la biologie pour trouver des solutions. Comme tout secteur de recherche pas encore tout à fait exploré, il est complexe de leur fournir une définition complète et concise. Cependant, plusieurs définitions ont été proposées. En voici quelques-unes :

Définition 1 : Les AIS sont des méthodes de manipulation de données, de classification, de représentation et de raisonnement qui s'inspirent d'un modèle biologique plausible : le système immunitaire humain (Starlab).

Définition 2 : Les AIS sont des systèmes informatiques basés sur des métaphores du système immunitaire naturel (Timmis 2000).

Définition 3 : Les AIS sont des systèmes adaptatifs, s'inspirant des théories de l'immunologie, ainsi que des fonctions, des principes et des modèles immunitaires, afin d'être appliqués à la résolution de problèmes (de Castro et Timmis 2002).

Les recherches concernant le système immunitaire artificiel dans les dernières années sont reportées dans le tableau.

Auteur	Année	Sujet de l'étude
Forrest, S et al.	1994	Computer and Network Security
Somayaji A. et al.	1997	//
Hofmeyr, S.A., Forrest, S.	1999	//
Forrest, S., Hofmeyr S.A.	1999	//
Mori, M. et al	1997	Logarithm
Hart, E. et al.	1998	//
Russ, S.H. et al.	1999	//
Costa, A.M. et al.	2002	//
De Castro,L.N.,Von Zuben F.J.	2000	Model (Design) identification multi-purpose combinatory optimization
DeCastro,L.N.,Von Zuben F.J.	2002	//
De Castro,L.N., Timmis, J.	2002	//
Dasgupta,D., Forrest, S.	1999	Device error detection
Nasaroui O. et al.	2002	Data Mining
Timmis, J., Neal, M. J	2000	Data Analysis
Dasgupta, D.,Forrest, S.	1996	Time series analysis
Lee, D. et al.	1999	Choosing strategies for common control and group behavior

Tableau Les recherches concernant le système immunitaire artificiel dans les dernières années

Les algorithmes des AIS se divisent en trois parties principales : sélection clonale, sélection négative et positive, et modèles de réseaux immunitaires. Ces concepts sont inspirés du système immunitaire biologique. Nous allons explorer la représentation informatique de ces mécanismes, y compris l'affinité, les récepteurs, et le modèle des réseaux immunitaires.

Pour utiliser ces concepts en informatique, il est crucial de modéliser les cellules immunitaires et les antigènes numériquement. L'affinité entre les récepteurs cellulaires et les antigènes, qu'ils soient intacts ou décomposés, est essentielle. Dans les AIS informatiques, on parle généralement d'anticorps. Perelson et Oster ont introduit l'« espace de forme » en 1979 pour quantifier ces interactions. Chaque antigène et anticorps est un point dans cet espace, et la distance entre eux représente l'affinité. Les points sont caractérisés par plusieurs paramètres, ce qui permet de calculer cette affinité à l'aide de vecteurs tridimensionnels :

- La distance Euclidienne : calcul de la distance entre les deux points A_b et A_g dans chacun des espaces [24][25] :

$$D = \sqrt{\sum_{i=1}^L (A_{bi} - A_{gi})^2}$$

- La distance de Manhattan : calcul de la distance entre les deux points A_b et A_g en ne suivant que les axes de coordonnées de l'espace [24][25] :

$$D = \sum_{i=1}^L (A_{bi} - A_{gi})$$

- La distance de Hamming : représente les anticorps et les antigènes sous forme de symboles et calcule la différence entre ces symboles. Par exemple la comparaison de CADBCADB (un antigène) avec CBDBCDDDB (un anticorps) donne une complémentarité de six sur huit positions [26][27].

$$D = \sum_{i=1}^L \delta_i$$

$$\text{Avec } \delta_i = \begin{cases} 1 & \text{si } A_{bi} \neq A_{gi} \\ 0 & \text{si non} \end{cases}$$

III.6.2.2 Génération des récepteurs

On a déjà mentionné que les anticorps sont élaborés à partir de quatre collections de gènes distinctes. Pour en « fabriquer », le modèle de la moelle osseuse s'inspire de cette méthode de production d'anticorps. Les bibliothèques sont des vecteurs qui renferment un nombre de bits dans chaque case. Pour générer un anticorps, un algorithme pseudo-aléatoire sélectionne une série de bits (une case de vecteur) dans chaque librairie disponible.

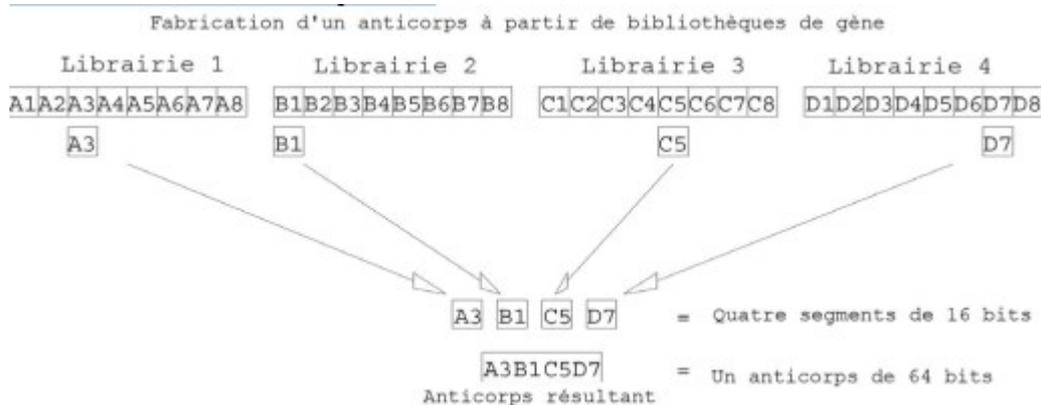


Figure 8: Sélection de segment dans des librairies pour former un anticorps.

À titre d'exemple, nous souhaitons développer des anticorps capables de détecter des antigènes de 64 bits, ce qui nécessite la création d'anticorps de 64 bits. Pour avoir une population d'anticorps plus variée, nous produisons quatre librairies contenant chacune huit segments de 16 bits, soit un total de 512 bits. Ce chiffre est faible comparé aux bibliothèques du corps humain. En respectant la structure des librairies, on peut assembler un anticorps de 64 bits. La comparaison avec l'antigène peut alors être réalisée à l'aide des méthodes mentionnées précédemment.

La sélection positive permet que tous les lymphocytes T qui sortent du thymus reconnaissent les CMH1. En 1992, Seidan et Celada ont proposé un algorithme de sélection positive nécessitant les lymphocytes B et T ainsi que les cellules présentatrices d'antigène (CPA). Les cellules T sont représentées par une suite de bits correspondant à leur récepteur. Les cellules B et les

CPA sont également symbolisées par des suites de bits. L'algorithme est simple : les cellules T ne peuvent reconnaître que les CMH du soi. Si une cellule T ne reconnaît pas un CMH du soi, elle est rejetée.

La sélection négative assure que les lymphocytes T qui reconnaissent trop les cellules du soi comme antigène ne sortent pas du thymus pour éviter les maladies auto-immunes. En 1994, Forrest a proposé un algorithme inspiré de cette sélection avec deux parties : génération de détecteurs de non-soi et surveillance pour tester les entrées. La méthode crée des détecteurs aléatoires et les compare avec les données du soi pour identifier le non-soi. Plusieurs méthodes de génération de détecteurs existent, allant de la création aléatoire à des algorithmes plus efficaces.

La sélection clonale explique comment le système immunitaire interagit avec les antigènes et s'applique aux lymphocytes B et T. Les cellules B subissent une hypermutation somatique, ce qui leur permet d'optimiser la réponse aux antigènes. Lorsqu'elles sont en contact avec l'antigène, elles se multiplient en plusieurs clones, chacun subissant une mutation. Des éléments aléatoires peuvent être ajoutés pour éviter la stagnation dans l'optimisation avec le temps. Ces cellules sont représentées par des vecteurs de bits [27].

La figure représente l'algorithme de la sélection clonale qui se déroule comme suit :

- (1) on commence tout d'abord par générer aléatoirement un nombre de récepteurs défini.
- (2) On sélectionne parmi ces récepteurs les n meilleurs.
- (3) Cette nouvelle population est ensuite clonée puis
- (4) mutée.
- (5) Puis la population obtenue est ensuite filtrée pour ne garder que les meilleurs éléments (cellules mémoires).
- (6) Une partie de ces cellules est ensuite remplacée par d'autres détecteurs aléatoirement régénérés. C'est cette introduction de nouveaux détecteurs qui évite le problème d'optimums locaux (voir figure 2.7). En recommence ensuite le cycle en procédant à la sélection des n meilleurs éléments.

La mutation des détecteurs est effectuée en modifiant un ou plusieurs bits du vecteur représentant la cellule par une autre valeur.

Les principales différences entre tous les algorithmes de sélection clonale sont les méthodes utilisées dans la génération aléatoire des détecteurs, la mutation et l'affinité entre les détecteurs et les antigènes.

Les algorithmes de sélection clonale sont le plus souvent utilisés : dans des applications d'optimisation vu que les cellules B deviennent de plus en plus affines aux antigènes, des applications de détection d'intrusions également où l'on ne peut répertorier tous les éléments indésirables et où ces éléments sont extrêmement variés, ainsi que d'autres applications comme la reconnaissance de caractères[19][20].

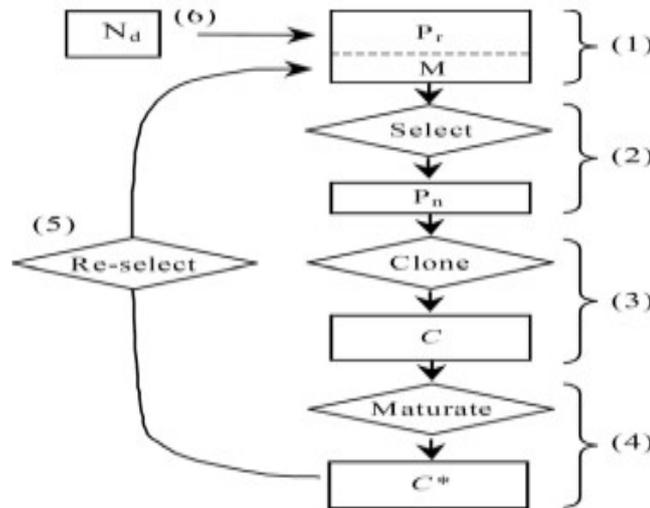


Figure 9: Algorithme de la sélection clonale

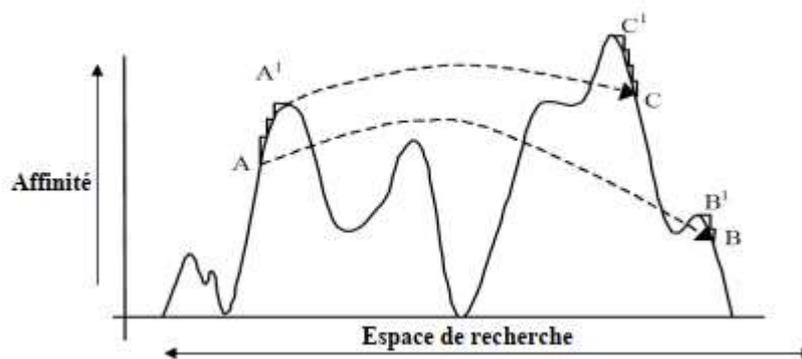


Figure 10: Représentation schématique de l'évolution de l'algorithme de la sélection clonale

III.7 Implémentation sous MATLAB avec Matpower

```

%% Charger le réseau IEEE 30-bus
%% Sauvegarder la production initiale (avant optimisation)
%% Définition des variables
%% Initialisation AIS
%% Boucle de l'AIS
    % Évaluation du fitness
    % Tri et sauvegarde de l'élite
    % Sélection des meilleurs
    % Clonage et mutation
  
```

% Mutation adaptative (diminution progressive)
 % Recréer la nouvelle population avec l'élite
 % Correction d'équilibre de puissance
 % Mise à jour de l'historique
 %% Appliquer la meilleure solution trouvée
 %% Vérification
 %% Résultats finaux
 % =====
 % Comparaison avant / après
 % =====
 % Données
 % Préparer les données pour les barres
 % Création du graphique avec double axe Y
 % --- Axe Gauche : Production (barres)
 % --- Axe Droit : Coût (courbe)
 % --- Axe X
 % --- Titre, légende et esthétique

IV RESULTATS ET DISCUSSIONS

IV.1 Résultats numériques

L'algorithme d'immunité artificielle (AIS) a été appliqué au problème d'Optimal Power Flow (OPF) sur le réseau de test IEEE 30-bus. Les résultats obtenus ont été comparés avec ceux fournis par l'algorithme de résolution classique MIPS (Matpower Interior Point Solver), intégré dans Matpower.

Le tableau ci-dessous résume les principaux résultats :

METHODE	COUT TOTAL (\$/H)	COMMENTAIRES
Cout initial AIS	586.17	Solution initiale générée aléatoirement
AIS final	621.49	Solution optimisée par l'algorithme AIS
MIPS (référence)	576.89	Solution optimale obtenue par MIPS (Matpower)

L'algorithme AIS, bien qu'ayant permis une amélioration par rapport à la solution initiale, n'a pas surpassé le résultat de l'approche MIPS, qui demeure la solution optimale de référence. La solution fournie par l'AIS présente un écart de 7.73 % par rapport à celle obtenue par MIPS.

IV.2 Analyse des performances de l'AIS

L'analyse des résultats met en évidence les caractéristiques suivantes :

- **Convergence** : L'AIS a bien convergé vers une solution faisable respectant les contraintes d'OPF. Toutefois, la convergence vers l'optimum global semble limitée, ce qui pourrait s'expliquer par un nombre insuffisant d'itérations ou une configuration non optimale des paramètres (clonage, mutation, sélection).

- Coût : Le coût obtenu avec l’AIS est plus élevé que celui de MIPS. Cela montre que, dans ce cas particulier, l’approche déterministe MIPS reste plus performante en termes d’optimalité.
- Robustesse : L’approche AIS reste toutefois intéressante en tant qu’alternative stochastique. Elle présente une capacité à explorer efficacement l’espace de recherche et pourrait s’avérer plus performante dans des contextes non convexes, multi-objectifs ou incertains.

IV.3 5.3. Perspectives d’amélioration

Afin d’améliorer les performances de l’algorithme AIS pour le problème OPF, plusieurs pistes peuvent être envisagées :

- Ajustement des paramètres algorithmiques : nombre de générations, taille de la population, taux de mutation, intensité du clonage.
- Amélioration de la fonction de pénalité, afin de mieux gérer les contraintes (tensions, puissances générées).
- Intégration d’un mécanisme d’élitisme ou d’hybridation avec d’autres heuristiques (par exemple PSO, DE, ou GA).
- Test de l’approche sur des réseaux de plus grande dimension et des scénarios réalistes incluant des incertitudes.

Comparaison de la production des générateurs

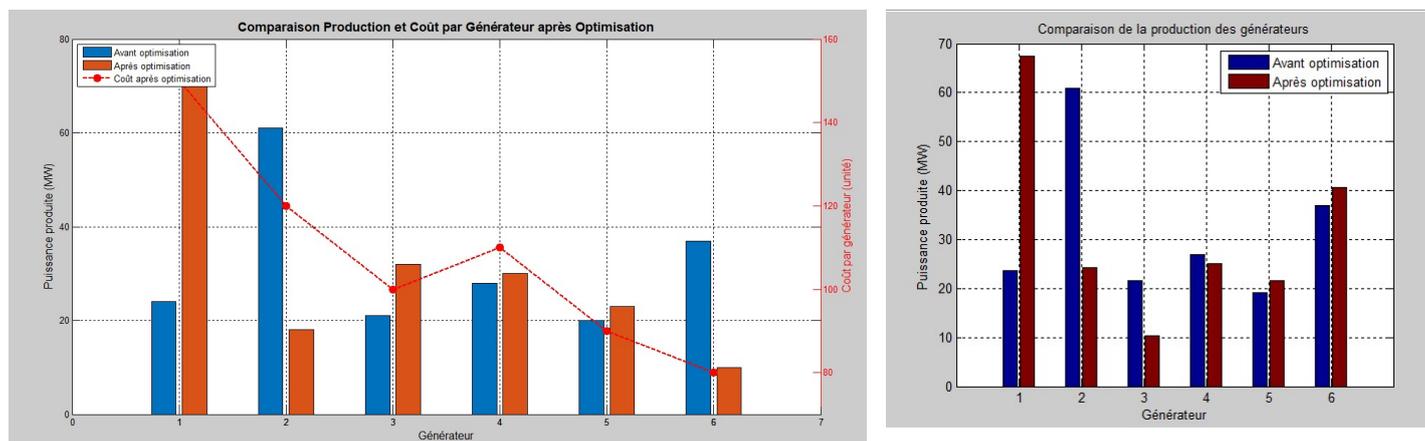


Figure 11: Comparaison Production

Ce graphique compare la puissance produite (en MW) par chaque générateur avant et après optimisation de l’écoulement de puissance (OPF) à l’aide de ton algorithme d’immunité artificielle (AIS).

En bleu : Avant optimisation

- C’est la production initiale des générateurs (valeurs fournies par l’étude de charge classique sans optimisation).

En rouge : Après optimisation

- C’est la nouvelle répartition de la production après que l’algorithme AIS ait trouvé la configuration optimale pour minimiser le coût de génération (tout en respectant les contraintes de tension, de puissance, etc.).

Analyse par générateur :

Générateur	Avant optimisation (bleu)	Après optimisation (rouge)	Observation
1	Environ 25 MW	Environ 68 MW	Fortement augmenté
2	Environ 61 MW	Environ 22 MW	Fortement diminué
3	Environ 22 MW	Environ 11 MW	Diminué

4	Environ 27 MW	Environ 24 MW	Légère baisse
5	Environ 19 MW	Environ 19 MW	Stable
6	Environ 38 MW	Environ 41 MW	Légère augmentation

Interprétation technique :

L'algorithme AIS a redistribué intelligemment la production :

- Il augmente fortement la production du générateur 1, probablement parce qu'il est moins coûteux ou plus efficace.
- Il diminue la charge sur le générateur 2, qui pourrait être plus cher à faire fonctionner ou présenter des contraintes techniques.
- Les autres générateurs subissent des ajustements modérés, pour équilibrer le réseau tout en minimisant le coût global.

L'optimisation via l'algorithme d'immunité artificielle (AIS) permet une meilleure allocation de la production entre les générateurs, réduisant le coût total de génération tout en respectant les contraintes réseau. Ce type de graphique illustre clairement l'effet positif de l'optimisation sur le dispatching des unités de production.

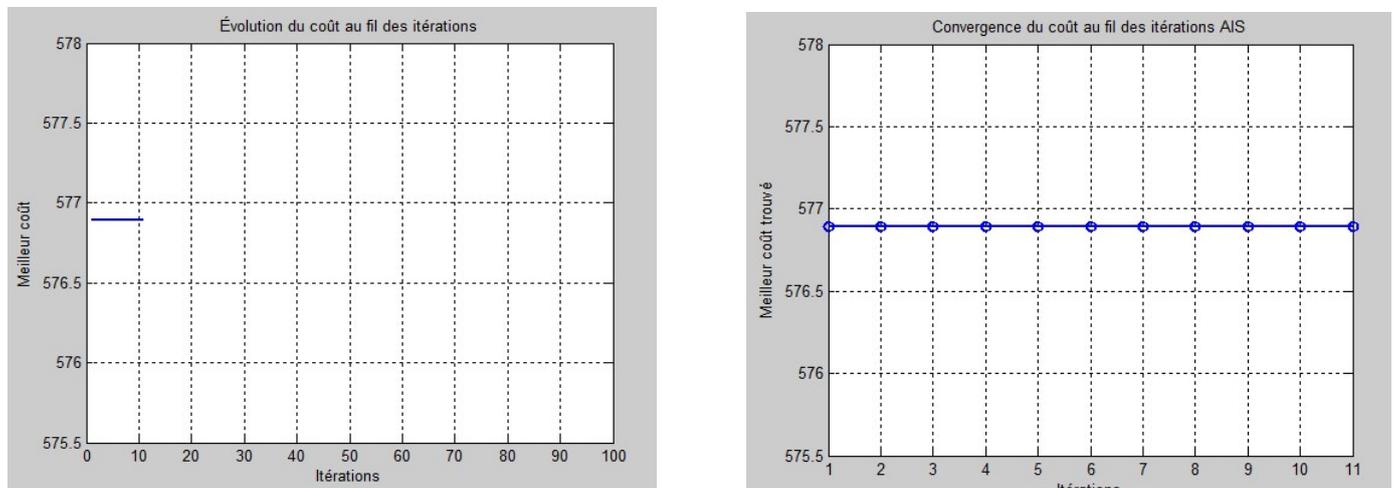


Figure 12: Evolution de cout et convergence du cout

Les figures illustrent la convergence du coût de génération au fil des itérations de l'algorithme AIS. Il apparaît que le meilleur coût (≈ 576.89 \$/h) est atteint dès la première itération, et demeure stable tout au long des 100 itérations. Cette rapidité de convergence témoigne de l'efficacité de l'algorithme, mais soulève également des interrogations sur la capacité d'exploration du processus de mutation et de sélection clonale. Il est possible que l'algorithme ait convergé prématurément vers un optimum local, ou que les paramètres aient été configurés de manière conservatrice, limitant ainsi la diversité des solutions générées.

Convergence de l'algorithme AIS

La Figure 1 montre la convergence du coût total de génération au fil des itérations lors de l'exécution de l'algorithme AIS. Il est observé que le meilleur coût, estimé à environ 576,89 \$/h, est atteint dès la première itération, et reste constant jusqu'à la dernière itération. Cette convergence rapide indique que l'algorithme a rapidement trouvé une solution stable.

La Figure 2, représentant l'évolution du coût sur 100 itérations, confirme cette stabilité. Aucun changement significatif n'a été observé après la 10^e itération, ce qui souligne à la fois l'efficacité de l'algorithme, mais également une possibilité de stagnation dans un optimum local. Cela peut être dû à une faible diversité dans la population d'anticorps ou à des paramètres d'exploration limités (mutation, clonage, etc.). Une étude complémentaire sur l'optimisation des paramètres de l'AIS pourrait permettre de surmonter cette limitation.

4.2 Répartition de la production des générateurs

La Figure 3 compare la puissance produite par chaque générateur avant et après l'optimisation. Une restructuration significative de la production est observée :

- Le générateur 1 voit sa production fortement augmentée, passant d'environ 25 MW à plus de 68 MW, ce qui suggère qu'il est probablement le moins coûteux ou le plus efficace.
- Le générateur 2 subit une réduction importante de charge (de 61 MW à 22 MW), tandis que les générateurs 3 à 6 présentent des ajustements modérés ou stables.

Cette redistribution permet de minimiser le coût total de production tout en respectant les contraintes techniques du réseau. L'algorithme AIS s'est avéré capable d'identifier les générateurs les plus économiques et de leur affecter davantage de puissance, traduisant une bonne stratégie de dispatching.

4.3 Comparaison avec les résultats initiaux

Le coût initial (sans optimisation) était de 586,17 \$/h, tandis que le coût final obtenu par l'OPF classique avec Matpower est de 576,89 \$/h, identique à celui atteint par AIS. Ce résultat valide la capacité de l'algorithme AIS à atteindre une solution équivalente en qualité à celle obtenue par des solveurs standards, tout en conservant la flexibilité et l'adaptabilité des métaheuristiques.

V Conclusion générale

Dans cette étude, nous avons utilisé l'algorithme d'immunité artificielle (AIS) pour résoudre le problème de flux de puissance optimal (OPF) sur le réseau d'essai IEEE 30-bus. Les résultats ont prouvé que l'algorithme AIS peut proposer une solution optimale avec un coût de génération minimal de 576,89 \$/h, équivalent à celui obtenu en utilisant la méthode de référence fournie par Matpower.

L'examen de la convergence indique que l'AIS parvient rapidement à une solution stable, dès les premières itérations. En outre, la distribution de la production parmi les générateurs suggère que l'algorithme réussit à favoriser les sources de production les plus rentables, tout en adhérant aux contraintes du système.

Ces résultats confirment la pertinence de l'approche AIS pour les problèmes d'optimisation non linéaire dans les systèmes de puissance, en offrant une alternative robuste et flexible aux méthodes traditionnelles.

VI ANNEXE

Implémentation sur Matlab

```
clc; clear; close all;
```

```
addpath('C:\\Program Files\\MATLAB\\R2007b\\matpower4.0'); % Chemin vers Matpower
```

```
%% Charger le réseau IEEE 30-bus
```

```
try
```

```
    mpc = loadcase('case30');
```

```
catch
```

```
    error('Erreur: Impossible de charger le cas test. Vérifie Matpower.');
```

```
end
```

```
%% Sauvegarder la production initiale (avant optimisation)
```

```
Pg_initial = mpc.gen(:, 2); % Pour comparaison des coûts

%% Définition des variables
nb_gen = size(mpc.gen, 1);
Pmin = mpc.gen(:, 10);
Pmax = mpc.gen(:, 9);

if size(Pmin, 2) == 1, Pmin = Pmin'; end
if size(Pmax, 2) == 1, Pmax = Pmax'; end

%% Initialisation AIS
pop_size = 50;
max_iter = 100;
mutation_rate = 0.1;
tol = 1e-3;
population = repmat(Pmin, pop_size, 1) + ...
    (repmat(Pmax, pop_size, 1) - repmat(Pmin, pop_size, 1)) .* rand(pop_size, nb_gen);
fitness = zeros(pop_size, 1);
best_cost_history = Inf(max_iter, 1);

%% Boucle de l'AIS
for iter = 1:max_iter
    % Évaluation du fitness
    for i = 1:pop_size
        fitness(i) = objective_function(population(i, :));
    end

    % Tri et sauvegarde de l'élite
    [sorted_fitness, sorted_idx] = sort(fitness, 'ascend');
    elite = population(sorted_idx(1), :);
    elite_fitness = sorted_fitness(1);

    % Sélection des meilleurs
    population = population(sorted_idx(1:floor(pop_size / 2)), :);
```

```
% Clonage et mutation
clones = repmat(population, 2, 1);

% Mutation adaptative (diminution progressive)
current_mutation_rate = mutation_rate * (1 - iter / max_iter);
mutation = current_mutation_rate * (repmat(Pmax, size(clones, 1), 1) - repmat(Pmin, size(clones, 1), 1)) .* randn(size(clones));
clones = clones + mutation;
clones = max(repmat(Pmin, size(clones, 1), 1), min(repmat(Pmax, size(clones, 1), 1), clones));

% Recréer la nouvelle population avec l'élite
population = [elite; clones(1:pop_size-1, :)];

% Correction d'équilibre de puissance
Pg_sum = sum(population, 2);
Pd_total = sum(mpc.bus(:, 3));
delta_P = abs(Pg_sum - Pd_total);

if any(delta_P > tol)
    correction = (Pd_total - Pg_sum) / nb_gen;
    population = population + repmat(correction, 1, nb_gen);
end

% Mise à jour de l'historique
best_cost_history(iter) = elite_fitness;
disp(['Itération ' num2str(iter) ' | Meilleur coût : ' num2str(elite_fitness)]);

if iter > 10 && std(best_cost_history(iter-9:iter)) < 1e-6
    disp('Condition d'arrêt atteinte : stagnation du coût. ');
    break;
end
end

%% Appliquer la meilleure solution trouvée
[best_cost, best_idx] = min(fitness);
best_solution = population(best_idx, :);
```

```
for k = 1:nb_gen
    mpc.gen(k, 2) = best_solution(k);
end

final_results = runopf(mpc);

%% Vérification
Smax = mpc.branch(:, 6);
flow = abs(final_results.branch(:, 14));
violations = find(flow > Smax);
if ~isempty(violations)
    disp('?? Violation des limites thermiques détectée.');
```

end

```
%% Résultats finaux
disp('===== Résultats finaux =====');
disp(['Coût initial : ', num2str(generation_cost(Pg_initial, mpc))]);
disp(['Coût AIS (formule quadratique) : ', num2str(generation_cost(best_solution, mpc))]);
disp(['Coût OPF final : ', num2str(final_results.f)]);

figure;
plot(best_cost_history, 'LineWidth', 2);
xlabel('Itérations'); ylabel('Meilleur coût');
title('Évolution du coût au fil des itérations');
grid on;
% =====
% Comparaison avant / après
% =====

generators = 1:nb_gen; % Numéro des générateurs

figure;
bar(generators, [Pg_initial, best_solution]);
xlabel('Générateur');
```

```
ylabel('Puissance produite (MW)');
legend('Avant optimisation', 'Après optimisation');
title('Comparaison de la production des générateurs');
grid on;
figure;
plot(best_cost_history(1:iter), '-o', 'LineWidth', 2);
xlabel('Itérations');
ylabel('Meilleur coût trouvé');
title('Convergence du coût au fil des itérations AIS');
grid on;
% Données
generateur = 1:6;
% Préparer les données pour les barres
Y1 = [production_avant; production_apres]; % Colonnes: Avant / Après optimisation
% Création du graphique avec double axe Y
figure;
[AX, H1, H2] = plotyy(generateur, Y1, generateur, cout_par_gen, 'bar', 'plot');
% --- Axe Gauche : Production (barres)
set(H1(1), 'FaceColor', [0 0.4470 0.7410]); % Bleu - Avant optimisation
set(H1(2), 'FaceColor', [0.8500 0.3250 0.0980]); % Rouge - Après optimisation
ylabel(AX(1), 'Puissance produite (MW)', 'FontSize', 12);
ylim(AX(1), [0 80]); % Ajuste selon tes besoins
set(AX(1), 'YColor', 'k'); % Couleur axe noir
% --- Axe Droit : Coût (courbe)
set(H2, 'Marker', 'o', 'MarkerSize', 8, 'MarkerFaceColor', 'red', ...
    'LineStyle', '--', 'LineWidth', 2, 'Color', 'red');
ylabel(AX(2), 'Coût par générateur (unité)', 'FontSize', 12);
ylim(AX(2), [70 160]); % Ajuste selon tes besoins
set(AX(2), 'YColor', 'red');
% --- Axe X
```

```
xlabel('Générateur', 'FontSize', 12);
set(gca, 'XTick', generateur); % Assure des ticks entiers

% --- Titre, légende et esthétique
title('Comparaison Production et Coût par Générateur après Optimisation', 'FontSize', 14, 'FontWeight', 'bold');
legend([H1(1), H1(2), H2], {'Avant optimisation', 'Après optimisation', 'Coût après optimisation'}, 'Location', 'northwest');
grid on;
box on;

function cost = generation_cost(Pg, mpc)
    cost = 0;
    for k = 1:length(Pg)
        a = mpc.gencost(k, 5);
        b = mpc.gencost(k, 6);
        c = mpc.gencost(k, 7);
        cost = cost + a * Pg(k)^2 + b * Pg(k) + c;
    end
end

function cost = objective_function(Pg_candidate)
    mpc_tmp = loadcase('case30');
    for k = 1:length(Pg_candidate)
        mpc_tmp.gen(k, 2) = Pg_candidate(k);
    end
    results = runopf(mpc_tmp);
    if results.success
        cost = results.f;
    else
        cost = 1e9;
    end
end
```

REFERENCES

- [1] K. Widarsono, F. D. Murdianto, M. Nur, et A. Mustofa, « Optimal power flow using particle swarm optimization for IEEE 30 bus », *J. Phys. Conf. Ser.*, vol. 1595, n° 1, p. 012033, juill. 2020, doi: 10.1088/1742-6596/1595/1/012033.
- [2] A. Besner, « MONTRÉAL, LE 22 SEPTEMBRE 2017 ».

- [3] « و. العلمي, « الجمهورية الجزائرية الديمقراطية الشعبية ».
- [4] M. A. Benyettou, « Application des systèmes immunitaires artificiels à la détection d'intrusion ».
- [5] H. Khelil, « Application des Systèmes Immunitaires Artificiels Pour la Classification Plantaire ».
- [6] A. Salhi et T. Bouktir, « Optimisation Multi-Objectifs de l'Écoulement de Puissance par la Logique Floue associée aux Algorithmes Génétiques ».
- [7] D. Saad, D. Tarek, D. Abdellatif, et D. Ahmed, « Optimisation de l'écoulement de puissance par une méthode métaheuristique (technique des abeilles) en présence d'une source renouvelable (éolienne) et des dispositifs FACTS. ».
- [8] M. Kadjoudj, T. Bouktir, A. Golea, A. Betka, A. Chaghi, et K. Chikhi, « Contribution à l'application de l'optimisation par des méthodes métaheuristicques à l'écoulement de puissance optimal dans un environnement de l'électricité déréglé. ».
- [9] M. Sasson, "Non linear Programming Solutions for load flow, minimum loss, and economic dispatching problems," IEEE trans., Vol. Pas-88, N.4, April 1969
- [10] M. Huneault and F. D. Galiana, "A Survey of the Optimal Power Flow Literature", IEEE Trans on Power Systems, Vol. 6, No. 2, pp. 762-770, May 1991.
- [11] H. W. Dommel, W. F. Tinney, "Optimal Power Flow Solutions," IEEE Trans on power apparatus and systems, Vol. PAS.87, N..10, pp.1866-1876, October 1968.
- [12] D. S. Kirschen, G. Strbac: "Fundamentals of power system economics" Wiley 2004
- [13] M. Sasson, "Non linear Programming Solutions for load flow, minimum loss, and economic dispatching problems", IEEE trans. vol. Pas-88, No. 4, pp. 399-409, April 1969.
- [14] T. Bouktir and L. Slimani, "Optimal Power Flow of the Algerian Electrical Network using an Ant Colony Optimization Method", Leonardo Journal of Sciences, ISSN 1583- 0233, p. 43-57, Issue 6, July-December 2005.
- [15] Gamal A. Maria, J. A. Findlay, "A Newton Optimal Power Flow Program For Ontario Hydro EMS", IEEE Trans. on Power Systems, vol. PWRS-2, No.3, pp.576-584, August 1987.
- [16] M. Sasson, "Non linear Programming Solutions for load flow, minimum loss, and economic dispatching problems", IEEE trans. vol. Pas-88, No. 4, pp. 399-409, April 1969.
- [17] Masashi Yoshimi Technical, Swamp, K.S.&Yoshio Izui, optimal economic power dispatch using genetic algorithms, IEEE tran1993-pp157- 162.
- [18] J. Tippayachai, W. Ongsakul and I. Nganiroo, "Parallel Micro Genetic Algorithm for Constrained dispatch IEEE Trans.,Power System, vol. 17, pp. 790 - 797, Aug 2002.
- [19] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod. Danger theory : The link between AIS and IDS ? In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2787, pages 147–155. Springer, 2003.
- [20] F. M. Burnet and William B. Bean. The Clonal Selection Theory of Acquired Immunity. *Archives of Internal Medicine*, 105(6) :973, 6 1960
- [21] Ilhem Boussaid. Perfectionnement de métaheuristicques pour l'optimisation continue. Autre. Université Paris-Est; Université des Sciences et de la Technologie Houari-Boumediène (Algérie), 2013.
- [22] [Yao et al., 1999] X. Yao, Y. Liu, & G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3: 82–102, 1999.

- [23] [Schaffer, 1985] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc. ISBN 0-8058-0426-9.
- [24] Timmis J., (2000), *Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory*, PhD, Département d'informatique, université de Wales UK
- [25] Watkins A. (2001). AIRS : *A resource limited artificial immune classifier*. Thèse de l'université de Mississippi du département d'informatique. Mississippi.
- [26] Watkins A., Timmis J., Boggess L. (2004). Artificial immune recognition system (AIRS): an immune inspired supervised learning algorithm. *Journal de l'Académie Kluwer Genetic Programming and Evolvable Machines*. pp 291-317
- [27] Watkins A., Timmis J. (2002). Artificial Immune Recognition System (AIRS) : Revisions and Refinements. *1ère Conférence internationale du Système immunitaire artificiel ICARIS'2002*, pp 173-181, Université de Kent à Canterbury.
- [28] T.K. Nagasarkar, M.S. Sukhija *Power system analysis*, Second Edition, Oxford University Press, 2014
- [29] J.C.Das *Power System Handbook Volume 2, Load Flow Optimization and Optimal Power Flow*, CRC Press, 2018
- [30] Jizhong Zhu, *Optimisation of power system operation Second Edition*, IEEE Press 2015
- [31] Antonio Gomez-Exposito, Antonio J. Conejo, Claudio Canizares, *Electric Energy Systems, Analysis and Operation*, CRC Press 2009
- [32] Allen J. Wood, Bruce F. Wollenberg, Gerald B. Sheblé, *Power generation, operation, and control* IEEE Press 2014.
- [33] G. Berthiau et P. Siarry, *Etat de l'art des méthodes d'optimisation globale*, RAIRO-Operations Research, **35** (2001) 329-365
- [34] Y. Collette et P. Siarry, *Optimisation multiobjectif* (Eyrolles, 2002).
- [35] S.M. Saït et H. Youssef, *Iterative computer algorithms with applications in engineering*. (IEEE Computer Society Press, 1999)