

III.1 Introduction

L'optimisation de l'écoulement de puissance consiste à répartir les puissances actives et réactives demandées entre les différentes centrales interconnectées dans un réseau électrique avec un coût minimal. Cette distribution doit évidemment respecter les limites de production des centrales et les capacités de transport des lignes électriques et les transformateurs. La variable à optimiser est donc le coût de production [1].

III.2 Définition de l'optimisation

Un problème d'optimisation se définit comme la recherche de l'optimum d'une fonction donnée [17]. On peut aussi trouver des problèmes d'optimisation pour lesquelles les variables de la fonction à optimiser sont des contraintes évoluant dans une certaine partie de l'espace de recherche. Dans ce cas, on a une forme particulière de ce que l'on appelle un problème d'optimisation sous contraintes. Donc optimiser, revient à minimiser ou maximiser une fonction en respectant certaines conditions préalables. Cette fonction dite « Objectif » peut être un cout (à minimiser), un profit (à maximiser), une production (à maximiser). Les fonctions objectifs sont diverses ainsi que les contraintes selon le problème à optimiser. [18]

Dans l'analyse de réseaux électrique plusieurs fonctions peuvent être optimisées (coût de production, l'émission des gaz toxiques, les pertes de transmission...etc.) avec la considération des contraintes d'égalité et d'inégalité pour obtenir des solutions optimales acceptables comme l'indique le tableau suivant:

Tableau .III. 1: Le tableau des fonctions objectives et leurs contraintes.

Contraintes	Variables de contrôles	Fonction objective
$P_g = P_D + \sum_{i=1}^{NG} P_i(1)$ $S_{transitée} \leq s_{max}$ (1):contrainte d'égalité (2):contrainte d'inégalité $P_{Gmin} \leq P_G \leq P_{Gmax}$ (2) $Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}$	P_G : puissance générée de chaque unité de production $P_{Gi} = P_i + P_D$	Minimiser le cout de production $F = \sum_{i=1}^{ng} (a_i + b_i P_{Gi} + c_i P_{Gi}^2)$
Les mêmes conditions	P_{Gi}	Pertes de puissance
Les mêmes conditions	P_{Gi}	Dégagement de gaz toxique

III.2.1 La fonction objective

La fonction objective est la formule de cout de production qui dépend de la puissance générée par les alternateurs et des constantes du cout unitaire du combustible par KWh caractérisant chaque centrale électrique. On suppose que le coût individuel de chaque centre de production dépende uniquement de la génération de la puissance active, alors la fonction objective égale la somme de tous les couts comme suit :

$$F = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \quad (III.1)$$

a_i, b_i, c_i représentent les coefficients de la fonction coût.

III.2.1.1 Les contraintes d'égalité

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.

$$\sum_{i=1}^{NG} P_{Gi} = P_D + P_L \quad (III.2)$$

Où P_D est la puissance totale demandée (charge du système) et P_L représente les pertes totale de transmission. La valeur exacte des pertes de transmission ne peut être obtenue qu'à partir d'une étude de l'écoulement de puissance .Néanmoins, dans les études du dispatching économique on exprime souvent les pertes de transmission en fonction des puissances actives générées. Cette technique est communément appelée la méthode des coefficients B.

Dans cette approche, les pertes sont approximées par la formule le plus précise de Kron :

$$P_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} B_{ij} P_{Gi} P_{Gj} + \sum_{j=1}^{NG} B_{0j} P_{Gi} + B_{00} \quad (III.3)$$

Où les termes B_{ij}, B_{0j} et B_{00} sont appelés coefficients des pertes ou les coefficients **B**.

Les coefficients des pertes ne sont pas constants, mais varient suivant les conditions de fonctionnement du système. Cependant, des résultats acceptables peuvent être obtenus si les conditions de fonctionnement actuelles sont relativement proches de celles pour lesquelles les coefficients

B ont été calculés. Il existe plusieurs méthodes basées sur l'étude de l'écoulement de puissance qui permettent de calculer les coefficients de pertes .

III.2.1.2 Les contraintes d'inégalité

Chaque alternateur possède une plage de fonctionnement limité par la puissance générée maximale et par la puissance générée minimale, les puissances maximale et minimale c'est une condition qu'exige le constructeur et ne peut pas dépasser cette limite pour le fonctionnement de l'alternateur.

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (\text{III.4})$$

- Le problème de dispatching économique optimal est donné sous une forme standard d'optimisation avec contraintes d'égalités et d'inégalités comme suit :

$$\left\{ \begin{array}{l} \mathbf{Min}F(P_{Gi}) = \sum_{i=1}^{ng} f_i = \sum_{i=1}^{ng} (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \\ \text{Sous:} \\ \mathbf{g}(x): \sum_{i=1}^{ng} P_{gi} - (P_d + P_l) = 0 \\ \text{et} \\ \mathbf{h}(x): P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max} \end{array} \right. \quad (\text{III.5})$$

On désigne les contraintes d'égalité par la fonction \mathbf{g} et les contrainte d'inégalité par la fonction \mathbf{h} [1].

Nous allons illustrer à la Fig. III.1 les conditions exigées pour pouvoir formuler un problème d'optimisation et la possibilité d'obtenir une solution optimale.

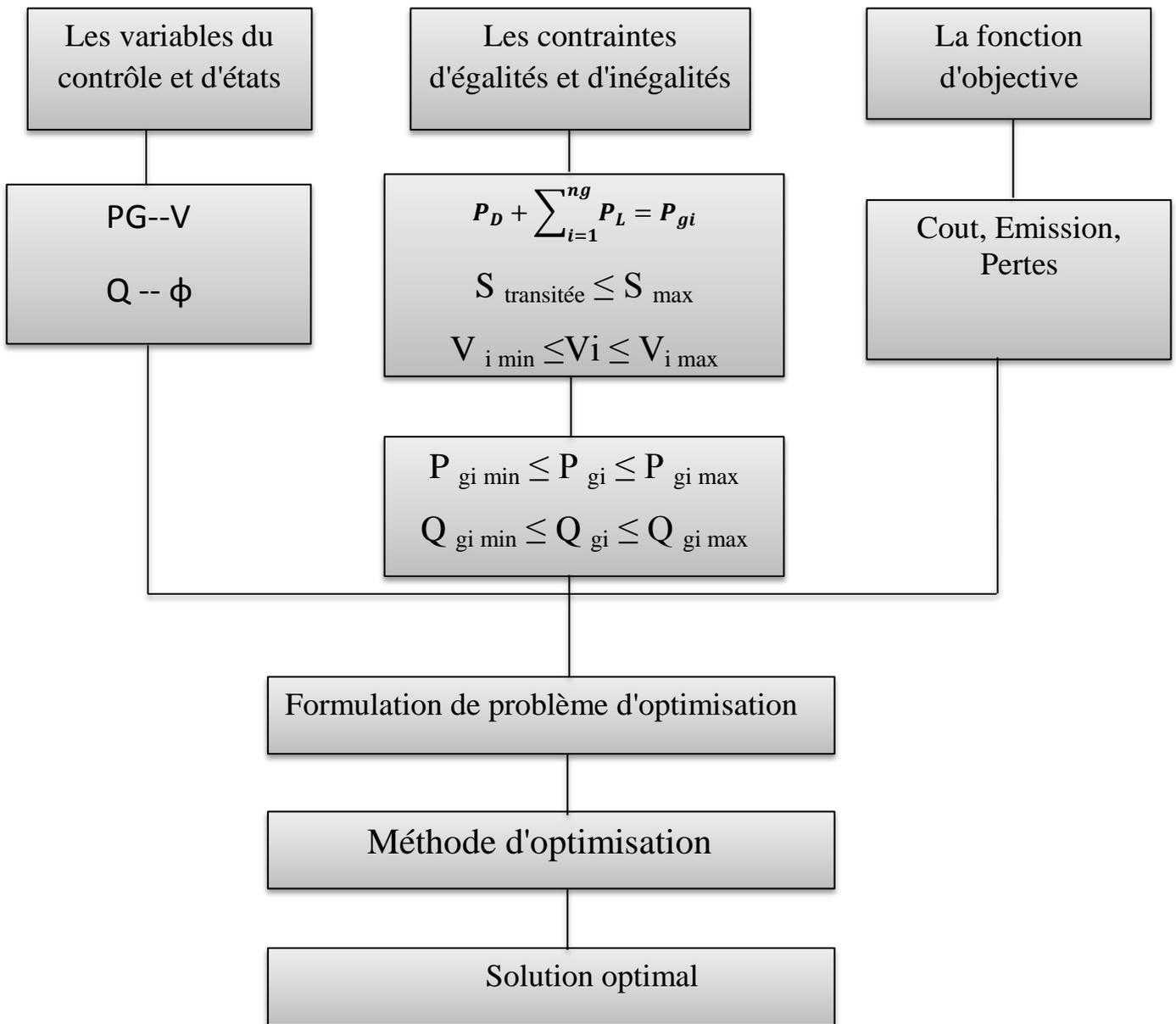


Fig. III. 1:Le schéma bloc représente les conditions exigées pour pouvoir formuler un problème d'optimisation

III.3.1 La méthode de Kuhn-Tucker

Cette méthode exige que la production en puissance par les centrales qui font part du système électrique doit satisfaire la demande, ceci peut être traduit par la formule suivante :

$$\sum_{i=1}^{Ng} PG = PD \tag{III.6}$$

Dans la pratique, les unités thermiques peuvent avoir des régions de fonctionnement indésirables pour des problèmes liés à la stabilité ou des limitations physiques de fonctionnement au niveau des composants des machines (groupes turbo-alternateurs) [18 ; 20 ; 21].

Ceci peut être traduit par les conditions suivantes :

$$P_{Gi} - P_{GiMax} \leq 0 \quad \text{Et} \quad P_{GiMini} - P_{Gi} \leq 0 \quad (\text{III.7})$$

$$Q_{Gi} - Q_{GiMax} \leq 0 \quad \text{Et} \quad Q_{GiMini} - Q_{Gi} \leq 0 \quad (\text{III.8})$$

La méthode de « Kuhn-Tucker » consiste à construire le Lagrangien qui tient compte des contraintes d'égalité en négligeant les contraintes d'inégalité dans un premier temps, sachant que ces contraintes peuvent être écrites sous les formes suivantes :

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad \text{Avec} \quad \mathbf{h}(\mathbf{x}) = \sum_{i=1}^{Ng} P_{Gi} - P_D = \mathbf{0} \quad (\text{III.9})$$

Donc le lagrangien peut s'écrire comme suit :

$$L(\mathbf{x},) = F(\mathbf{x}) + \lambda \mathbf{h}(\mathbf{x}) \quad (\text{III.10})$$

$$L(\mathbf{x},) = F(\mathbf{x}) \left(\sum_{i=1}^{Ng} P_{Gi} - P_D \right) \quad (\text{III.11})$$

$F(\mathbf{x})$: est la fonction objective du cout total à optimiser ; définie par :

$$F(\mathbf{x}) = \sum_{i=1}^{Ng} F_i(P_{Gi}) \quad (\text{III.12})$$

$$F_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 \quad (\text{III.13})$$

Dans le cas général pour étudier n'importe quelle fonction et trouver l'optimum, il suffit de la dériver.

Dans ce cas il faut dériver notre Lagrangien en fonction des P_{Gi} et du coefficient de Lagrange λ et annuler ces dérivées de sorte à obtenir les conditions sur l'optimum, ces conditions sont appelées: conditions nécessaires d'optimalité ou les conditions de Kuhn Tucker:

$$\frac{\partial L}{\partial P_{Gi}} = \frac{dF_i}{dP_{Gi}} - \lambda = 0 \quad (\text{III.14})$$

$$\frac{\partial L}{\partial \lambda} + P_D - \sum_{i=1}^{Ng} P_{Gi} = 0 \quad (\text{III.15})$$

La dérivée $\frac{dF_i}{dP_{Gi}}$ mathématiquement connue sous le nom : multiplicateur de Lagrange.

Physiquement c'est le « coût incrémental ou marginal », elle représente l'accroissement du coût correspondant à la production d'une unité de puissance supplémentaire, d'une autre manière le coût marginal de la production de l'électricité est en fonction de la demande future.

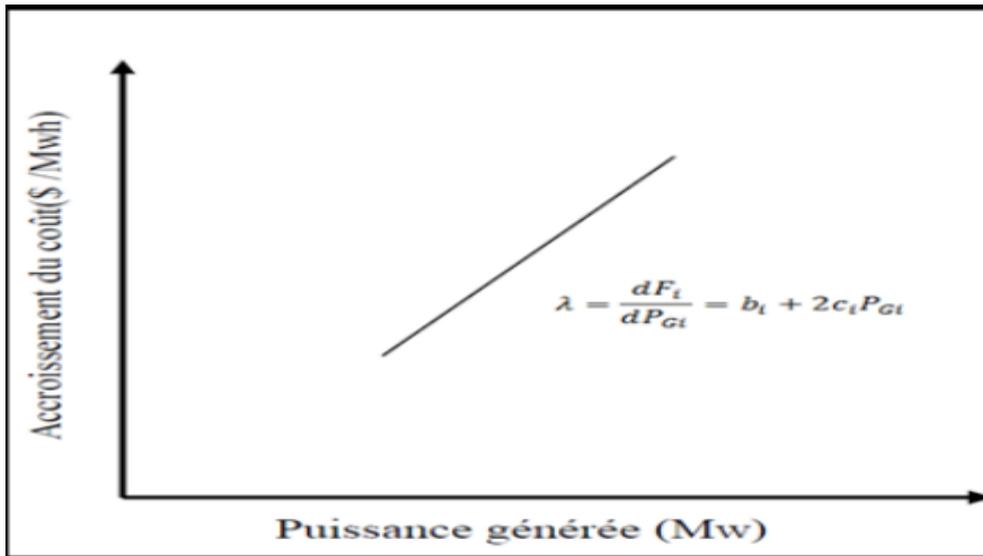


Fig. III. 2: La Courbe de l'accroissement du coût de combustible [19].

D'après la première condition, on peut calculer :

$$\lambda = \frac{dF_i}{dP_{Gi}} = b_i + 2c_i P_{Gi} \quad (\text{III.16})$$

Ce qui nous conduit à trouver l'expression de P_{Gi} :

$$P_{Gi} = \frac{(\lambda - b_i)}{2c_i} \quad (\text{III.17})$$

On sait par ailleurs que

$$P_D = \sum_{i=1}^{Ng} P_{Gi} \text{ et donc } P_D = \sum_{i=1}^{Ng} \frac{(\lambda - b_i)}{2c_i} \quad (\text{III.18})$$

Et d'après la seconde condition, on peut déduire :

$$P_{Gi} = \frac{1}{2c_i} \left(\left(\sum_{i=1}^{Ng} \frac{1}{2c_i} \right)^{-1} \left(P_D + \sum_{i=1}^{Ng} \frac{b_i}{2c_i} \right) - b_i \right) \quad (\text{III.19})$$

À l'aide de cette dernière expression (III.19) nous pouvons calculer les P_{Gi} pour lesquelles le coût total soit minimal, si cet optimum vérifie les contraintes d'inégalité, il s'agit de la solution recherchée, dans le cas contraire, on transforme certaines inégalités non vérifiées en égalités (pour imposer ces inégalités à leurs limites) et on recalcule un nouvel optimum en tenant compte de ces nouvelles égalités. L'optimum sera atteint dès que toutes les contraintes d'inégalités seront vérifiées [19]. La technique de Kuhn Tucker nous permet de trouver l'optimum local mais seulement dans le cas où il n'y a pas des limites sur la puissance générée et ce n'est plus le cas dans la réalité.

III.3.2 Méthode du gradient

Historiquement, les méthodes de gradient sont les plus anciennes. Elles permettent de résoudre des problèmes non linéaires et sont basées sur une hypothèse forte sur la connaissance de la dérivée de la fonction objective en chacun des points de l'espace [21].

Cette méthode peut être classée en deux catégories de premier ordre et de deuxième ordre, le premier ordre est basé sur une approximation linéaire en séries de Taylor avec initialisation du gradient, et le deuxième ordre est basé sur l'approximation quadratique en séries de Taylor avec initialisation du gradient en utilisant le Hessien.

III.3.2.1 Formulation mathématiques :

On choisit un point de départ x_0 . On calcule le gradient $\nabla f(x_0)$ en x_0 . Comme le gradient indique la direction de la plus grande pente de f , on se déplace d'une quantité λ_0 dans le sens opposé au gradient et on définit le point x_1 :

$$x_1 = x_0 - \lambda_0 \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|} \quad (\text{III.20})$$

Cette procédure est répétée et engendre les points $x_0, 1, \dots, k$. Ainsi, pas à pas, la distance entre le point d'indice k et l'optimum diminue.

$$x_{k+1} = x_k - \lambda_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \quad \text{Ou } \forall k, \lambda_k > 0 \quad (\text{III.21})$$

λ_k est le déplacement à chaque itération.

Si λ_k est fixé, on parle de méthode de gradient à pas prédéterminé. L'inconvénient de cette procédure est que la convergence est très dépendante du choix du pas de déplacement.

La convergence peut être très lente si le pas est mal choisi. L'intérêt principal de cette méthode est de pouvoir se généraliser aux cas des fonctions qui ne sont pas différentiables.

III.3.3 Méthode de Newton

La méthode de Newton est une méthode très puissante à cause de sa convergence rapide, en particulier si l'estimation initiale de la solution $x(0)$ est suffisamment proche de la solution optimale $x(m)$. L'idée de cette méthode est de minimiser, à chaque itération k , une approximation quadratique de la fonction objectif originale $f(x)$ au voisinage de l'estimation actuelle de la solution $x(k)$.

L'approximation quadratique de $f(x)$ est obtenue à partir du développement en série de Taylor à l'ordre 2 [22, 23].

$$f(x^{(k+1)}) \cong f(x^{(k)}) + [\nabla f(x^{(k)})]^T [\Delta x^{(k+1)}] + \frac{1}{2} [\Delta x^{(k+1)}]^T [\nabla^2 f(x^{(k)}) [\Delta x^{(k+1)}]]^T \quad (\text{III.22})$$

Ou: $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ est régulière (au moins différentiable). On cherche donc $x(m)$ tel que $F(x) = 0$ Pour toute $i = 1, \dots, n$.

III.3.4 La méthode d'itération de Lambda (Lambda itération méthode) :

La méthode d'itération de Lambda est une des méthodes utilisées pour trouver la valeur de lambda du système et trouver le dispatching économique optimal des générateurs.

Contrairement aux autres méthodes d'itération, comme : Gauss-Seidel et Newton-Raphson, lambda itération n'utilise pas la valeur précédente de l'inconnue pour trouver la valeur suivante c'est-à-dire il n'y a pas une équation qui calcule la valeur suivante en fonction de la valeur précédente. La valeur suivante est prédéfinie par intuition, elle est projetée avec interpolation de la bonne valeur possible jusqu'à ce que le décalage spécifié soit obtenu. On va maintenant discuter comment trouver le dispatching économique optimal utilisant la méthode d'itération de Lambda [9].

- la méthode exige qu'il y ait une correspondance entre une valeur lambda et la sortie (en MW) de chaque générateur
- la méthode commence avec des valeurs de lambda en-dessous et en-dessus de la valeur optimale (qui est inconnue), puis par itération limite la valeur optimale.

On choisit λ^L et λ^H tel que :

$$\sum_{i=1}^m P_{Gi}(\lambda^L) - P_D < 0 \quad \sum_{i=1}^m P_{Gi}(\lambda^H) - P_D > 0 \quad (\text{III.23})$$

On pose :

$$\lambda^M = \frac{(\lambda^H + \lambda^L)}{2} \quad (\text{III.24})$$

$$\text{Si : } \sum_{i=1}^m P_{Gi}(\lambda^M) - P_D > 0 \quad (\text{III.25})$$

On pose :

$$\lambda^H = \lambda^M \quad (\text{III.26})$$

Si :

$$\sum_{i=1}^m P_{Gi}(\lambda^M) - P_D > 0 \quad (\text{III.27})$$

On pose :

$$\lambda^L = \lambda^M \quad (\text{III.28})$$

On refait le calcul jusqu'à

$$|\lambda^L - \lambda^H| > \varepsilon \quad (\text{III.29})$$

D'où ε est : la tolérance de convergence

Dans la Fig. III.3 ci-dessous pour chaque valeur de lambda il y a une P_{Gi} unique pour chaque générateur. Cette relation est la fonction $P_{Gi}(\lambda)$

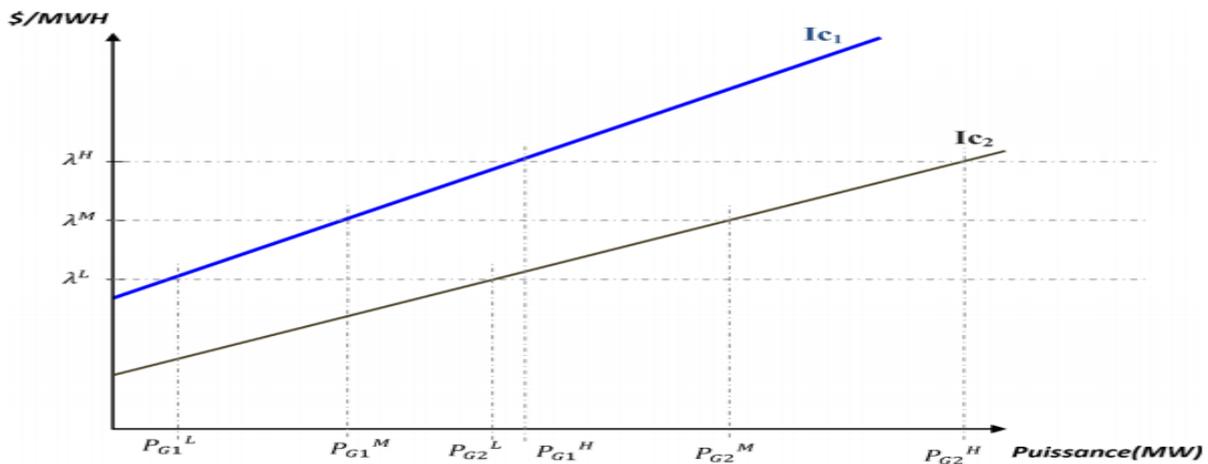


Fig. III. 3: changement de λ en fonction de la puissance de sortie.

III.4 Les méthodes d'optimisation avancées

Les méthodes d'optimisation avancées comprennent les méthodes d'optimisation globales ou méta-heuristiques et sont inspirées parfois de la théorie d'évolution chez les sociétés d'animaux et d'insectes dans laquelle on trouve les algorithmes génétiques AG, parfois sont inspirées de la théorie d'éthologie de ces sociétés dans laquelle on cite les algorithmes d'optimisation par essaims particuliers PSO, les colonies des Fourmies ACO, les colonies des abeilles ABC...etc.

Ces algorithmes sont basés sur l'exploration aléatoire probabiliste d'une ou plusieurs régions de l'espace de recherche, cette exploration aléatoire guidée parfois par des fonctions probabilistes permet d'éviter les optimums locaux lors de l'exploration contrairement aux méthodes déterministes qui se bloquent en général dans un optima local ou bien si la fonction objective présente une certaine complexité mathématique.

Les premières méta-heuristiques datent des années 1980, si elles sont d'origine discrète, on peut les adapter à des problèmes continus. Elles sont utilisées généralement quand les méthodes classiques (mathématiques) ont échoué de trouver la solution souhaitée, leur efficacité n'est pas toujours garantie, elle dépend, de la nature du problème envisagé et les paramètres de l'algorithme. Ces méthodes sont largement appliquées aux différents domaines notamment dans le domaine de l'optimisation de l'énergie électrique [1].

III.4.1 Les algorithmes génétiques :

Les algorithmes génétiques AG sont inspirés du concept de sélection naturelle et de la théorie de l'évolution et de la génétique élaborée par Charles Darwin. [24]

Plusieurs programmes génétiques ont été introduits par Holland [24,25] en 1975 et largement développés par Koza en 1992. Les AG travaillent sur des chaînes de caractères de taille fixe. On définit au départ une population d'individus de façon aléatoire. Chaque individu est marqué par une chaîne de caractères particulière évaluant son chromosome. Les individus vont être évalués sur la base d'une fonction objective, sélectionnés, se reproduire et subir des mutations. C'est un processus itératif qui prend généralement fin lorsque la population n'évolue plus entre deux périodes.

III.4.1.1 Principe des algorithmes génétiques :

La structure générale d'un algorithme génétique est la suivante:

- Générer aléatoirement une population d'individus de taille donnée.
- Répéter jusqu'à ce qu'un individu s'impose (seuil à définir) .
- Sélectionner et appairer les individus.
- Combiner deux individus appariés pour créer deux nouveaux individus.
- Faire muter certains individus.
- Créer une nouvelle génération en remplaçant tous les anciens individus par les nouveaux.

La fonction à optimiser : Elle est appelée aussi fonction fitness ou fonction d'évaluation de l'individu.

Les opérateurs permettent de diversifier la population au cours des générations et d'explorer l'espace de recherche.

Les paramètres de dimensionnement : la taille de la population, le nombre total de générations ou le critère d'arrêt, les probabilités d'application des opérateurs génétiques [24].

Chaque individu représente un point de l'ensemble de l'espace de recherche auquel on associe la valeur du critère à optimiser. On génère une population d'individus aléatoirement pour

laquelle l'algorithme génétique s'attache à sélectionner les meilleurs individus tout en assurant une exploration efficace de l'espace de recherche. Après la sélection aléatoire des deux parents P_1 et P_2 , on applique aléatoirement l'opérateur de croisement avec une probabilité P_c qui génèrent deux enfants C_1 et C_2 (Fig. III.4). On modifie ensuite certains gènes de C_1 et C_2 en appliquant l'opérateur de mutation P_m , ce qui produit deux nouveaux individus C_1' et C_2' (Fig. III.5) pour les quels on évalue le nouveau d'adaptation avant de les insérer dans la nouvelle population. Contrairement à la reproduction et au croisement qui favorise l'intensification, l'opérateur de mutation favorise la diversification des individus. On réitère les opérateurs de sélection, de croisement et mutation afin de compléter une nouvelle population, ceci termine le processus d'élaboration d'une génération. On réitère les opérations précédentes jusqu'à ce qu'un critère d'arrêt soit satisfait [26].

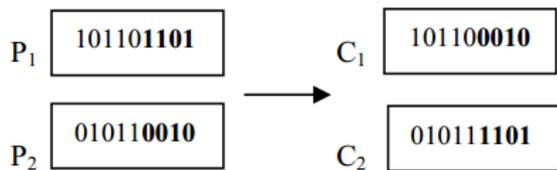


Fig. III. 4:L'opération de croisement

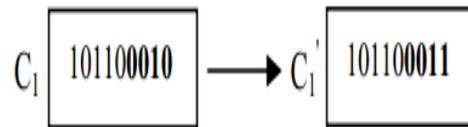


Fig. III. 5: L'opération de la mutation

Il existe trois principaux types de codage : binaire, gray ou réel. Nous pouvons facilement passer d'un codage à un autre.

Les chaînes binaires permettent de coder des structures complexes. C'est pourquoi, les algorithmes génétiques utilisent généralement cette représentation pour le codage des individus.

Il existe deux difficultés dans le choix du type de codage. D'une part celui-ci doit pouvoir adapter au problème de façon à limiter au mieux l'exploration de l'espace de recherche, et aussi de façon que les nouveaux génotypes engendrés par les opérateurs de recherche soient significatifs le plus souvent possible. D'autre part, la recherche doit être telle que les opérateurs de recherche standard soient suffisamment efficaces pour produire les solutions recherchées avec une bonne probabilité.

Le choix d'un codage approprié est un élément critique dont dépend grandement l'efficacité d'un algorithme génétique. Dans la représentation binaire, les individus se présentent sous forme d'une suite de 0 et 1, par contre dans la représentation réelle les individus se présentent sous forme de nombres entiers.

Dans notre travail nous avons utilisé un codage binaire. Nous codons les variables

P_{Gi} par les chaînes binaires de 12 bits comme suit :

La chaîne 000000000000 code $P_{Gi} = P_{Gi}^{min}$

La chaîne 111111111111 code $P_{Gi} = P_{Gi}^{max}$

La transformation d'une chaîne binaire « a » en un nombre réel P_G peut alors s'exécuter en deux étapes :

Conversion (base 2 \rightarrow base 10)

$$P'_G = \sum_{i=1}^{12} a_i 2^{1-i} \quad (\text{III.30})$$

Recherche du nombre réel correspondant :

$$P_G = P_G^{min} + P'_G \frac{P_G^{max} - P_G^{min}}{2^{12} - 1} \quad (\text{III.31})$$

III.4.1.2 Organigramme des algorithmes génétiques

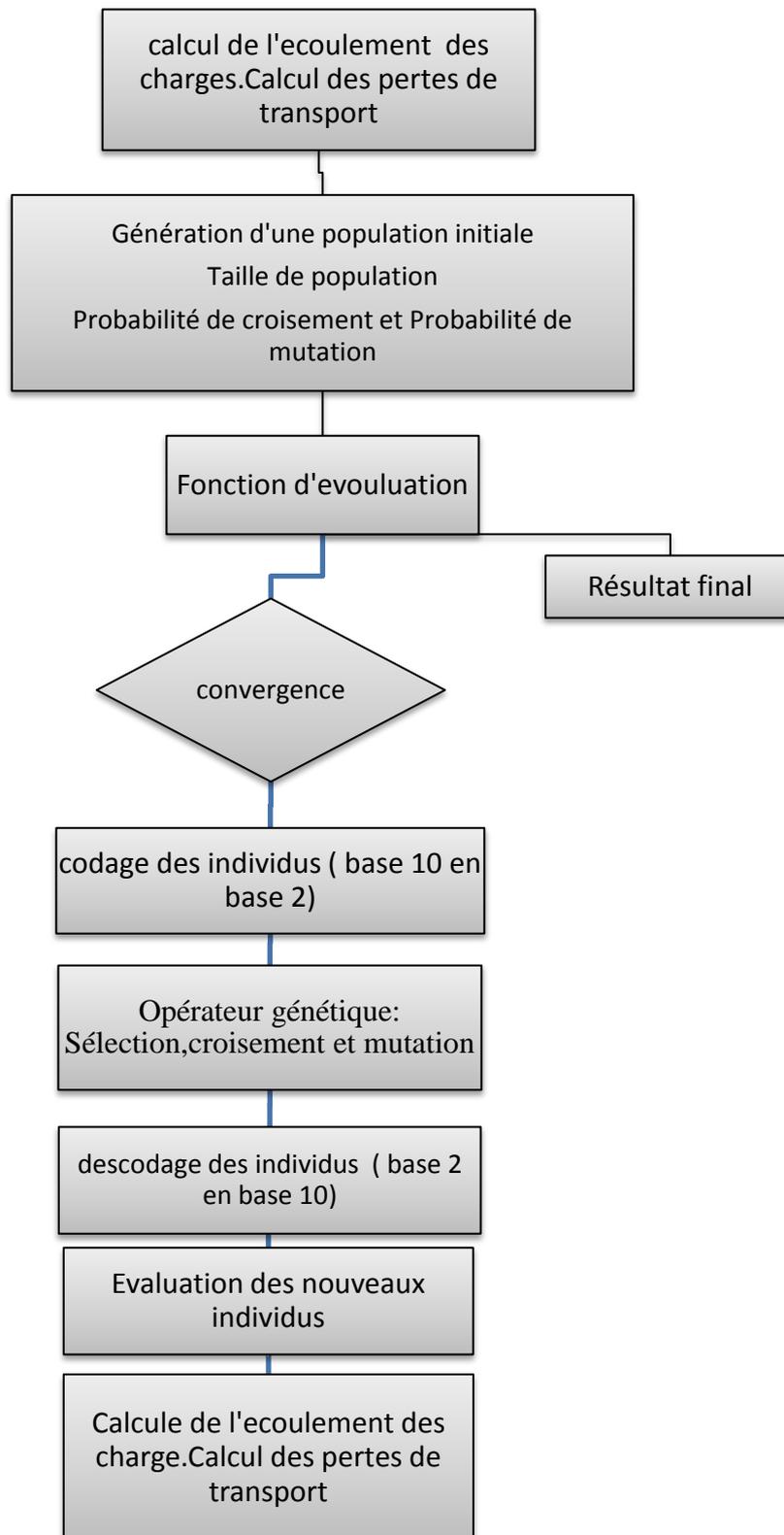


Fig. III. 6: L'organigramme de l'AG.

III.4.2 Optimisation par essaims de particules :

L'optimisation par Essaim Particulaire PSO a été proposée par Kennedy et Eberhart en 1995 [27]. Cette méthode est inspirée du comportement social des animaux évoluant en essaim. L'exemple le plus souvent utilisé est le comportement des bancs de poissons. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale de sa situation dans l'essaim. Un individu de l'essaim n'a pour connaissance que la position et la vitesse de ses plus proches voisins. Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement.

Kennedy et Eberhart se sont inspirés de ces comportements socio-psychologiques pour créer l'OEP. Un essaim de particules représentant les solutions potentielles au problème d'optimisation, "survole" l'espace de recherche, en quête de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes [27].

Pour des systèmes simples les principales caractéristiques sont :

1. L'information locale : Chaque individu ne possède qu'une connaissance partielle de l'environnement et n'a pas conscience de la totalité des éléments qui influencent le groupe,
2. L'ensemble de règles : Chaque individu obéit à un ensemble restreint de règles simples par rapport au comportement du système global,
3. Les interactions multiples : Chaque individu est en relation avec un ou plusieurs autres individus du groupe,
4. La collectivité : les individus trouvent un bénéfice à collaborer (parfois instinctivement) et leur performance est meilleure que s'ils avaient été seuls.

L'intelligence collective est observée notamment chez les insectes sociaux (fourmis, termites et abeilles) et les animaux en mouvement (oiseaux migrateurs, bancs de poissons). En conséquence, plusieurs algorithmes basés sur le principe d'intelligence collective ont été introduits : on peut citer les colonies de fourmis et les essaims particulaires.

III.4.2.1 Principe de fonctionnement :

Ainsi, grâce à des règles de déplacement très simples dans l'espace de solutions, les particules peuvent converger progressivement vers un optimum. Cette méta-heuristique semble cependant mieux fonctionner pour des espaces en variables continues. Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche de dimension D , chaque particule p est aléatoirement placée dans la position $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ de l'espace de recherche, chaque particule possède également une vitesse $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ aléatoire.

Ensuite, à chaque pas de temps :

- chaque particule est capable d'évaluer la qualité de sa position et de garder en mémoire sa meilleure performance $\vec{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$: la meilleure position qu'elle a atteinte jusqu'ici (qui peut en fait être parfois la position courante) et sa qualité (la valeur en cette position de la fonction à optimiser),
- chaque particule est capable d'interroger un certain nombre de ses informatrices et d'obtenir de chacune d'entre elles sa propre meilleure performance $\vec{g}_i = (g_{i1}, g_{i2}, \dots, g_{iD})$ (et la qualité afférente),
- à chaque pas de temps, chaque particule choisit la meilleure des meilleures performances dont elle a connaissance, modifie sa vitesse V en fonction de cette information et de ses propres données et se déplace en conséquence. La modification de la vitesse est une simple combinaison linéaire de trois tendances, à savoir sa propre vitesse courante, sa propre expérience et la tendance vers la meilleure performance de son essaim. La mise à jour des deux vecteurs la vitesse et la position, de chaque particule p dans l'essaim, est donnée par les équations (III.32) :

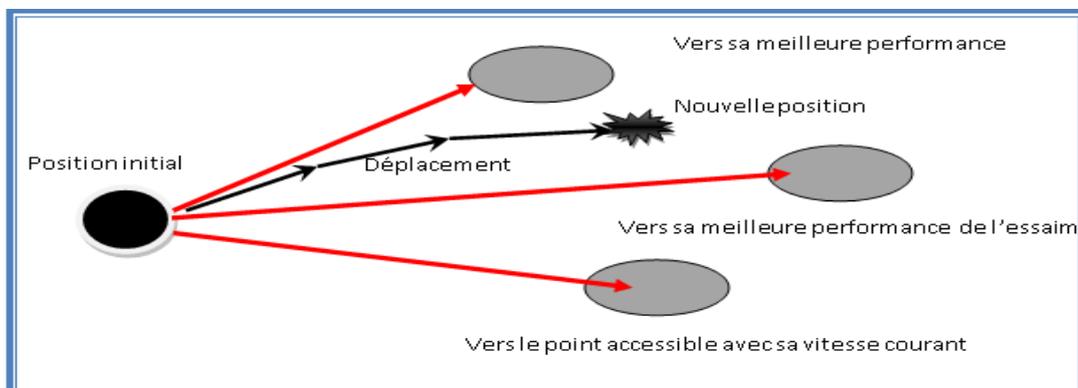


Fig. III. 7: La représentation schématique du déplacement d'une particule.

$$V(t + 1) = w * V(t) + C_1 * rand_1 * (Pbest(t) - X(t)) + C_2 * rand_2 * (Gbest(t) - X(t)) \quad (III.32)$$

Avec :

$X(t)$: Position de la particule $P(t)$.

$V(t)$: Vitesse de la particule $p(t)$.

$Pbest(t)$: Meilleure fitness obtenue pour la particule $P(t)$.

$Gbest(t)$: Position de la particule $P(t)$ pour la meilleure fitness.

où w est en général une constante appelée, coefficient d'inertie, C_1 et C_2 sont deux constantes, appelées coefficients d'accélération, $rand_1$ $rand_2$, sont deux nombres aléatoires tirés uniformément de l'intervalle $[0,1]$ à chaque itération et pour chaque dimension.

- $w * V(t)$ correspond à la composante physique du déplacement. Le paramètre w contrôle l'influence de la direction de déplacement sur le déplacement futur. Il est à noter que, dans certaines applications, le paramètre w peut être variable.
- $C_1 * rand_1 * (Pbest(t) - X(t))$ correspond à la composante cognitive du déplacement.
- C_1 contrôle le comportement cognitif de la particule.
- $C_2 * rand_2 * (Pbest(t) - X(t))$ correspond à la composante sociale du déplacement. C_2 Contrôle l'aptitude sociale de la particule.

La combinaison des paramètres w , C_1 et C_2 permet de régler la balance entre les phases diversification et intensification du processus de recherche [27]. La position au temps t de la particule i est alors définie par (III.33).

$$X(t + 1) = X(t) + V(t + 1) \quad (III.33)$$

L'OEP (PSO) est un algorithme à population. Il commence par une initialisation aléatoire de l'essaim dans l'espace de recherche. A chaque itération de l'algorithme, chaque particule est déplacée suivant (III.32) et (III.33). Une fois le déplacement des particules effectué, les nouvelles positions sont évaluées. Les $Pbest(t)$ ainsi que $Gbest(t)$ sont alors mis à jour. Cette procédure est résumée par l'Algorithme (Fig.III.8). N est le nombre de particules de l'essaim. Le critère d'arrêt peut être différent suivant le problème posé. Si l'optimum global est connu a priori, on peut définir une "erreur acceptable" ϵ comme critère d'arrêt. Sinon, il est commun de fixer un nombre maximum d'évaluations de la fonction objectif ou un nombre maximum d'itérations.

III.4.2.2 Les étapes de la méthode d'optimisation par essaim de particelles

L'algorithme de cette méthode peut être décrit comme suit :

- **1ère étape** : Initialisation des coefficients $cr1$ et $cr2$, le coefficient d'inertie w .
- **2ème étape** : La création de la population initiale aléatoirement et le calcul de la fitness.
- **3ème étape** : Le calcul de la nouvelle vitesse et nouvelle position de chaque particule par l'utilisation des formules.
- **4ème étape** : Le calcul de la meilleure fitness de la population initiale et comparer par la précédente pour trouver la meilleure de toute les populations
- **5ème étape** : incrémentation du nombre d'itération $t = t+1$.
- **6ème étape** : Si un critère d'arrêt est satisfait alors passer à la 7ème étape. Autrement, aller à la 3ème étape.
- **7ème étape** : La position enregistrée dans la solution optimale

Organigramme de PSO :

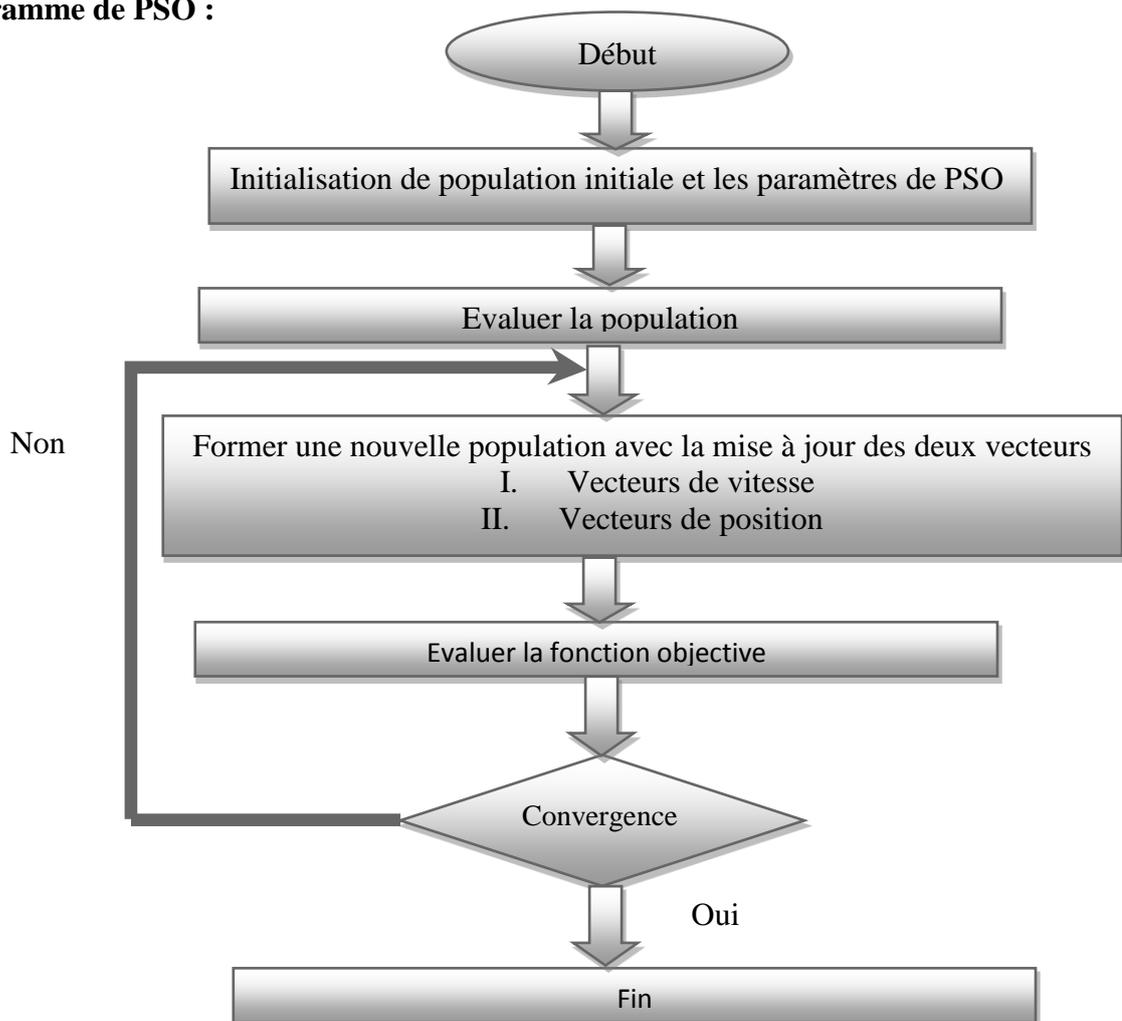


Fig. III. 8: L'organigramme de la méthode PSO.

III.4.3 Optimisation par l'algorithme de colonies de fourmis ACO

III.4.3.1 Les fourmis réelles

L'étude des fourmis à long temps été négligée par les entomologistes. Jusqu'à ce que, Hölldobler et Wilson ont corrigé cette lacune en 1990 en publiant un ouvrage concentrant tout ce que l'on connaissait alors des fourmis. Les fourmis constituent à l'heure actuelle un support majeur pour les théories développées en écologie comportementale et en sociobiologie[28]. On peut citer plusieurs raisons à cette inspiration:

L'influence des fourmis sur leur environnement naturel est extrêmement importante. Il a par exemple été montré (qu'elles déplacent plus de terre en forêt tropicale que les vers de terre, ou encore que le poids total des fourmis sur terre est du même ordre de grandeur que le poids des humains. De plus, la domination des fourmis est une preuve de leur adaptation à des environnements très variés)

- a) l'étude des fourmis se fait assez facilement en laboratoire car elles s'adaptent sans trop de difficultés à des environnements différents de leur habitat d'origine
- b) les fourmis possèdent une gamme de comportements très variés, collectifs ou Individuels.

III.4.3.2 Les colonies de fourmis

L'optimisation par colonies de fourmis, s'inspire du comportement des fourmis lorsque celles-ci sont à la recherche de nourriture. Une fourmi en se déplaçant pose de la phéromone, substance olfactive et volatile, sur son chemin. Les fourmis se dirigent de manière probabiliste en tenant compte de la quantité de phéromone qui est autour d'elles et qui a été précédemment déposée par les autres membres de la colonie. Plus la quantité de phéromone indiquant un chemin est grande, plus la fourmi a tendance à suivre ce chemin. Cependant, comme la phéromone s'évapore progressivement, le choix probabiliste que prend une fourmi pour choisir son chemin évolue continuellement.

Comme l'illustre la Fig.III. 9, les fourmis qui reviendront au nid le plus rapidement, après avoir visité la source de nourriture, sont celles qui auront emprunté les deux branches les plus courtes. Ainsi, la quantité de phéromone présentée sur le plus court chemin est relativement plus importante que celle présente sur le chemin le plus long. En sachant qu'une piste présentant une plus grande concentration en phéromone est plus attirante pour les fourmis, elle aura une plus grande probabilité d'être choisie. Le chemin le plus court sera renforcé par les passages successifs et deviendra à terme la piste empruntée par la majorité des insectes [8].

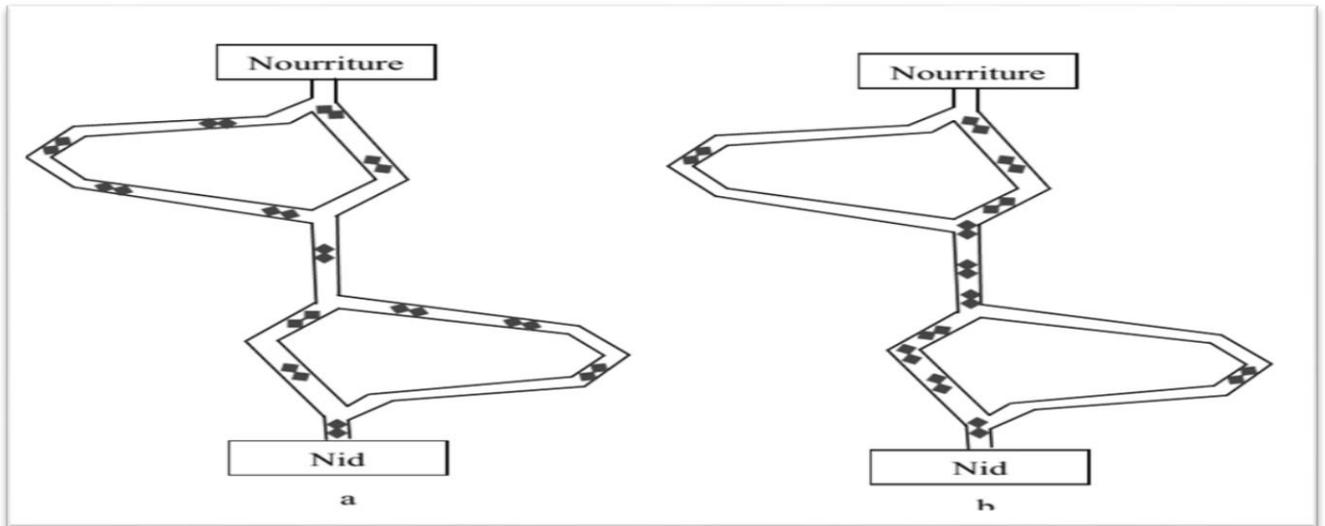


Fig. III. 9:Expérience de sélection des branches les plus courtes par une colonie de fourmis [8]: (a) au début de l'expérience, (b) à la fin de l'expérience



a) : Des fourmis réelles suivant un chemin entre le nid et une source de nourriture

b) :Un obstacle survient sur le chemin les fourmis choisissent de tourner à gauche ou à droite, avec des probabilités égales



c) : La quantité de phéromone est déposée plus rapidement sur le chemin le plus court

d) : Toutes les fourmis ont choisi le chemin le plus court

Fig. III. 10:Illustration de la capacité des fourmis à chercher de la nourriture en minimisant leur parcours.

Une autre expérience qui illustre l'intelligence collective des fourmis concerne la coupure d'une piste de phéromone, voir la Fig.III. 10. On suppose que les fourmis suivent une piste de phéromones sur laquelle est déposé un obstacle. Les fourmis choisissent alors de contourner l'obstacle par la gauche ou par la droite. Dans la mesure où aucun marquage n'existe autour de l'obstacle, les fourmis partiront à gauche ou à droite dans les mêmes proportions. Le chemin de droite étant plus court que celui de gauche, les fourmis qui l'empruntent, retrouveront plus vite la piste de phéromone déjà existante. A partir de ce moment, les fourmis qui arriveront devant l'obstacle préféreront suivre la piste de droite. Le nombre de fourmis passant par la droite augmentera proportionnellement avec la concentration de phéromones. De plus, l'évaporation des phéromones sera plus rapide sur la piste de gauche du fait de sa plus grande longueur. Elle sera moins imprégnée et les fourmis suivront la piste la plus courte.

III.4.3.3 Organigramme de la technique ACO appliquée à l'OPF

Les étapes principales de l'algorithme ACO pour résoudre le problème d'OPF sont illustrées dans la Fig.III.11 et discutées ci-dessous:

- a. Etape 1:** Initiation
- b. Etape 2 :** évaluation de la fonction objective
- c. Etape 3:** répartition des fourmis
- d. Etape 4:** Critère d'arrêt

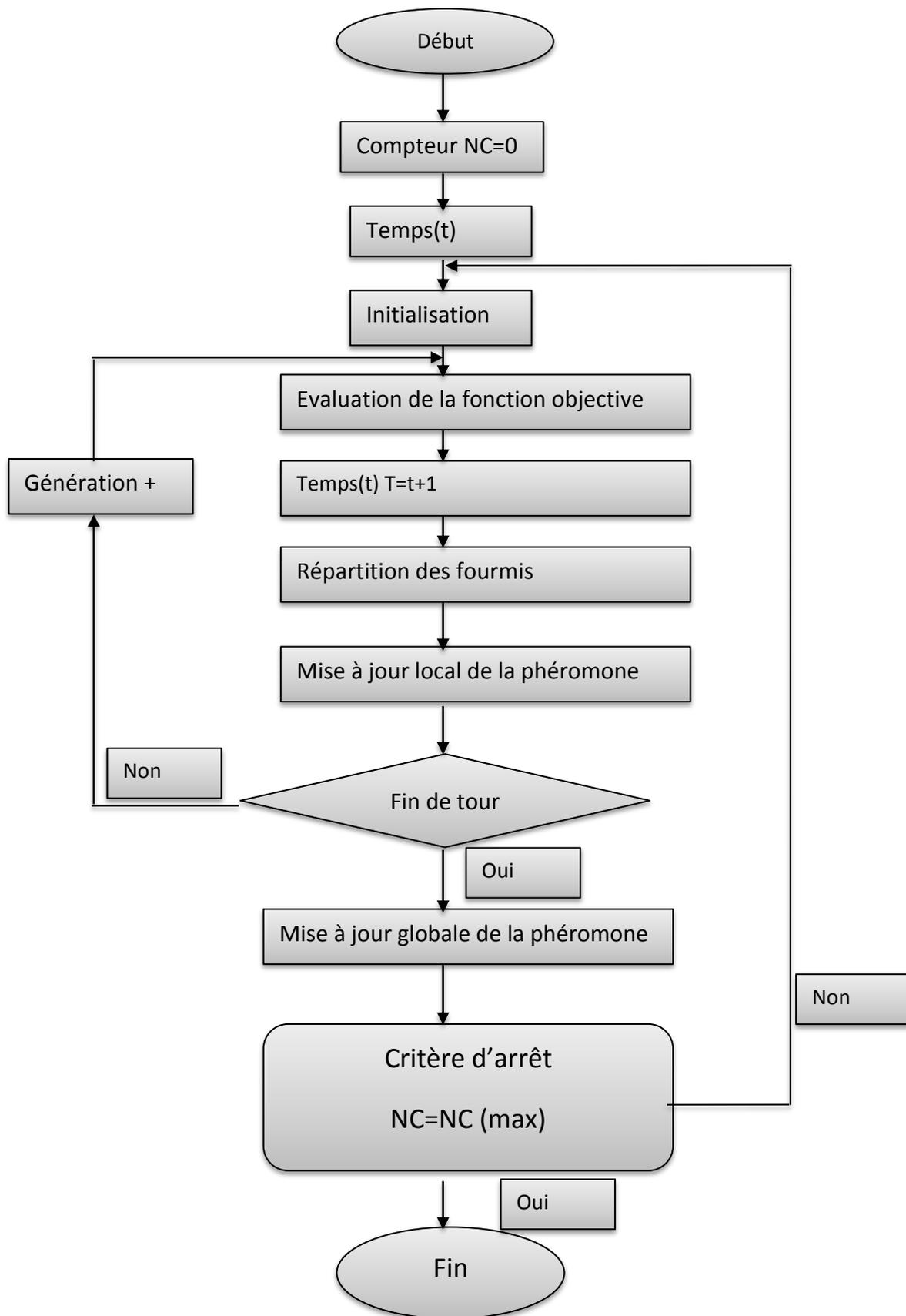


Fig. III. 11: L'organigramme de l'ACO

III.4.4 Algorithme d'optimisation de colonie d'abeilles artificielle ABC

L'algorithme ABC (Artificiel Bee Colony) est développé par Karaboga et Basturk en 2005, en inspectant les comportements des abeilles réelles pour trouver la source de nourriture, qui s'appelle le nectar, et partager l'information des sources de nourriture aux autres abeilles dans le nid. Dans cet algorithme, les abeilles artificielles sont définies et classifiées en trois groupes : abeilles employeuses (abeilles qui recherche la nourriture), spectatrices (abeilles d'observation) et scouts (éclaireuses) sont chargées de trouver de nouvelles nourritures, (le nectar de nouvelles source). [29]

Pour chaque source de nourriture, il y a seulement une abeille employeuse. C'est-à-dire, le nombre d'abeilles employeuses est égal au nombre de sources de nourriture. [30]

Si l'abeille employeuse d'un site ne réussit pas de trouver la source de nourriture, elle doit être forcément devenir un scout pour rechercher aléatoirement de nouvelles sources de nourriture.

Les abeilles employeuses partagent l'information avec les abeilles spectatrices dans une ruche de sorte que les abeilles spectatrices puissent choisir une source de nourriture pour l'explorer. Le processus de l'algorithme ABC est présenté comme suit:

Etape 1- Initialisation:

On commence par sélectionner le pourcentage de population de façon aléatoire dans l'espace de recherche en utilisant l'équation suivante:

$$U_j = U_j^{min} + n_j * (U_j^{max} - U_j^{min})n_j \in [0,1] \quad (III.34)$$

Sachant que chaque abeille porte un vecteur 'U' de 'n' solution. A titre d'exemple le cas qu'on va simuler est expliqué comme suit:

$$U_{controle} = (P_i, V_i, T_i) \Rightarrow \left\{ \begin{array}{l} abeille_1 = (P_1, V_1, T_1) \\ abeille_2 = (P_2, V_2, T_2) \\ abeille_3 = (P_3, V_3, T_3) \\ \cdot = \cdot \\ \cdot = \cdot \\ \cdot = \cdot \\ abeille_i = (P_i, V_i, T_i) \end{array} \right\} \quad (III.35)$$

P_i : puissance générée au jeu de barres i ;

V_i : tension générée au jeu de barres i ;

T_i : prises en charge des transformateurs;

Puis, on les évalue dans la fonction objective (équation de coût), ensuite on calcule leurs valeurs Fitness, appelées la quantité de nectar par l'équation suivante :

$$\mathbf{Fitness} = \frac{1}{F_{objective}} \quad (\text{III.36})$$

F représente le rapport des abeilles dans la population totale. Une fois que ces populations sont placées dans l'espace de recherche, elles prennent le nom : les abeilles employeuses.

Étape 2- Déplacement des abeilles employeuses: Calculer la probabilité de choisir une source de nourriture par l'équation (III.37),

$$P_i = \frac{0.9 * \text{Fitness}_i}{\max(\text{Fitness}_i)} + 0.1 \quad (\text{III.37})$$

Puis sélectionner une source de nourriture et ensuite déterminer ses quantités de nectar. L'équation de mouvement des abeilles observatrices est donnée ci-dessous:

$$m_{ij}(t + 1) = x_{kj} + y(x_{ij}(t) - x_{kj}(t)) \quad y \in [0,1] \quad (\text{III.38})$$

Tel que m_{ij} est la $i^{\text{ème}}$ position de l'abeille spectatrice, t est le nombre d'itération, x_{ij} est l'abeille utilisée choisie aléatoirement, ' j ' représente la dimension du vecteur de solution qui produit une série de variables aléatoires dans la gamme $[-1,1]$; où $k \in \{1,2,3 \dots \dots \dots, N\}$ et $j \in \{1,2, \dots \dots, D\}$ sont choisis aléatoirement; ' D ' est le nombre de paramètres à optimiser; [31].

' K ' est aussi choisi aléatoirement mais doit être différent de l'indice ' i '

Étape 3-Déplacer les scouts : Si les valeurs de Fitness des abeilles employeuses ne sont pas améliorées par un nombre d'itérations prédéterminé, appelé "max-cycle", ces sources de nourriture sont abandonnées et l'abeille trouvée dans cet emplacement passera aléatoirement pour explorer d'autres nouveaux emplacements. (Abeilles employeuses deviennent des Scouts). Cette explication est traduite mathématiquement par l'équation (III.39):

$$V_{ij} = V_{ij}^{min} + \varphi_{ij} * (V_{ij}^{max} - V_{ij}^{min}) \varphi_{ij} \in [0,1] \quad (\text{III.39})$$

Étape 4-Mettre à jour la meilleure source de nourriture trouvée jusqu'ici : Apprendre la meilleure valeur de Fitness et la position, qui sont trouvées par les abeilles, et les mémoriser.

Étape 5-Critère d'arrêt

Vérifier le processus de calcul jusqu'à ce que le nombre d'itérations atteigne la valeur maximale prédéfinie ou qu'une solution de la fonction objective acceptable soit trouvée.

III.5 Conclusion

Dans ce chapitre qui est consacré à l'étude des méthodes d'optimisation avancées, nous avons étudié d'une manière explicite le principe de fonctionnement des algorithmes génétiques GA, l'algorithme d'optimisation par essais de particules PSO et l'algorithme des colonies des fourmis ACO, l'optimisation par colonie des abeilles ABC. Où nous avons clarifié leur rôle à l'optimisation et leurs processus algorithmiques.