

Chapitre II

**Logique floue & technique d'optimisation
d'essaim de particules (PSO)**

II.1 Introduction

En engineering plusieurs problèmes nécessitent l'utilisation des méthodes d'optimisation pour s'approcher ou avoir la solution optimale. Les méthodes déterministes semblent avoir rempli ce besoin, mais le problème des optimums locaux constitue un véritable piège pour accéder à l'optimum global. D'où, l'utilisation des méthodes stochastiques dites métaheuristiques qui, en général, évitent ce problème et ayant une grande capacité de trouver l'optimum global [14].

Ce chapitre, présentera brièvement le principe de fonctionnement de la logique floue ainsi que les PSO en mettant en évidence leurs similitudes et leurs différences.

II.2 Logique Floue

II.2.1 Définition et historique

La logique floue a vu le jour après la logique booléenne. Cette dernière est basée sur le principe que toute proposition ne peut être que vraie ou fausse (tout ou rien), et pourtant la pensée humaine est beaucoup plus complexe et plus nuancée qu'un ensemble rigide de lois car, pour un être humain, le mot "rien" n'est pas tout à fait blanc ou noir, on trouve dans notre langage habituel "assez grand, presque chaud, pas chaud...etc.". Alors, les propositions nécessitent un élargissement à la logique classique [15].

C'est donc en 1965, que la logique floue est apparue avec la publication par Lotfi Zadeh de l'université de Californie, Berkeley, USA, d'un article intitulé "les ensembles flous" [16].

Les quelques points de repères historiques suivants permettent de situer dans le temps le développement de la logique floue et ses applications au réglage : Le 1973 Lotfi propose d'appliquer la logique floue aux problèmes de réglage. En 1974, première application du réglage par la logique floue appliquée à une turbine à vapeur, en 1975 E.H.Mamdani expérimente un régulateur flou qu'il perfectionne au cours des années suivantes. Sugéno décrit des applications industrielles possibles en régulation floue, tel que premières produits industriels (JAPON) utilisant le principe de la logique floue appliqué à des problèmes de réglage et de commande. En 1995 J.S.R.Jang élargit la logique floue aux systèmes à réseaux de neurones et l'intelligence artificielle [17].

II.2.2 Principe fondamental de la logique floue

Une des caractéristiques du raisonnement humain est qu'il est généralement fondé sur des données imprécises ou même incomplètes. En effet les connaissances dont nous disposons sur un système quelconque sont généralement incertaines ou vagues, soit parce que nous avons un doute sur leur validité ou alors nous éprouvons une difficulté à les exprimer clairement. Par exemple : dans la logique classique, une vitesse peut être qualifiée par les termes « Elevée ». Mais dans la logique floue, des échelons d'appréciation intermédiaires de la variable vitesse sont possibles. La «Vitesse» devient une variable linguistique dont les valeurs sont par exemple : « Très faible », « Faible »,

«Moyenne », «Elevée », « Très élevée». La logique floue peut être considérée comme une extension de la logique classique [18].

La commande par logique floue peut opérer lorsque les procédés à commander sont mal connus ou difficiles à décrire précisément, ou lorsque les variables sont évaluées subjectivement et exprimées en langage naturel et non numériquement. Elle est simple à réaliser, flexible et donc facilement adaptable aux conditions de fonctionnement du processus. Les règles sont faciles à comprendre et à modifier puisqu'elles sont exprimées par des termes du langage naturel [19]. L'architecture interne d'un régulateur flou est donnée par la figure (II.1).

Elle est composée de trois blocs :

- Fuzzification.
- Inférence.
- Défuzzification.

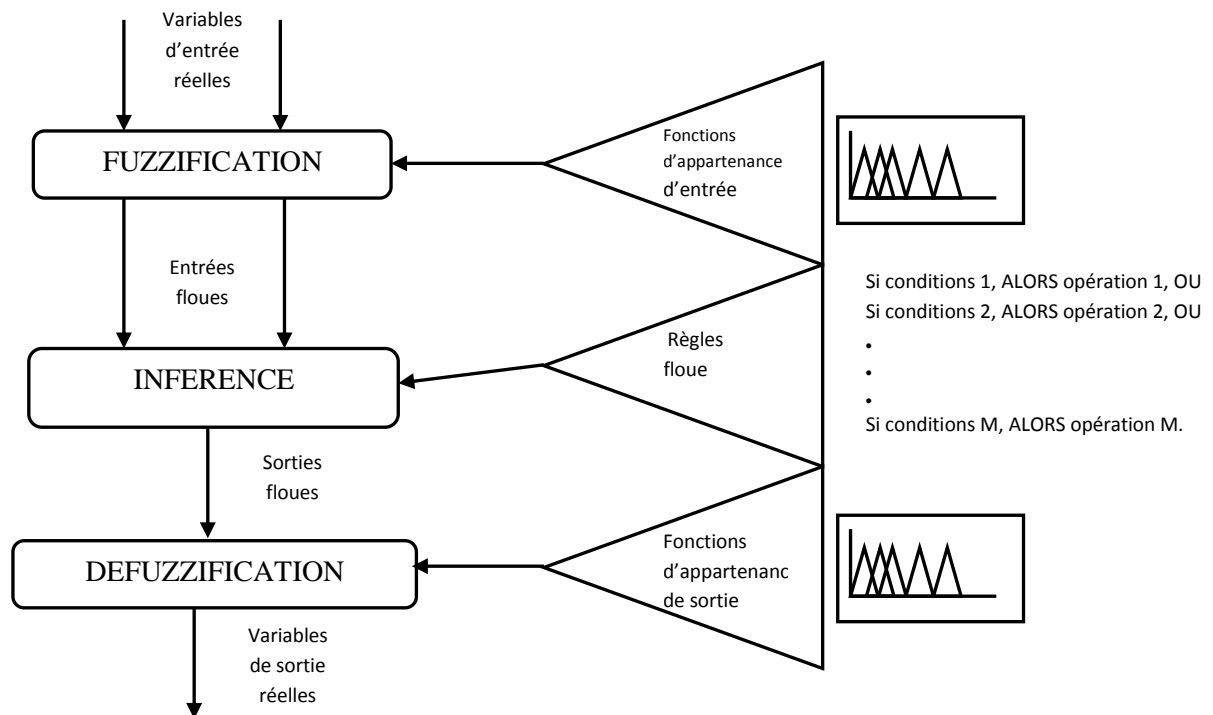


Figure II.1 Structure d'un régulateur flou [20].

II.2.2.1 Fuzzification

L'opération de fuzzification, permettant le passage des grandeurs réelles aux valeurs floues, consiste à déterminer les différents univers de discours correspondant aux domaines de variation des variables d'entrée et de sortie. Ensuite, définir pour chacune des variables ses sous-ensembles flous ainsi que leurs fonctions d'appartenance associée [19].

II.2.2.2 Inférence

L'opération d'inférence permet de relier les valeurs des variables linguistiques d'entrée à celles des variables linguistiques de sortie par une table de règles, qui doivent tenir compte du comportement statique et dynamique du système à commander. C'est un mécanisme qui, en manipulant les règles floues, établit une décision.

En réglage par logique floue, on utilise en général une des méthodes suivantes [21]:

- Méthode d'inférence Max-Min.
- Méthode d'inférence Max-Prod.
- Méthode d'inférence Somme-Prod.

Le nom de la méthode désigne les opérateurs utilisés respectivement pour l'agrégation et l'implication. Le tableau suivant indique la manière de leur utilisation :

Méthodes	Opérateurs sur Prémisses		Opérateur Implication	Opérateur Agrégation
	Ou	Et	Imp	
Max-min	Max	Min	Min	Max
Max-prod	Max	Min	Prod	Max
Som-prod	Som	Prod	Prod	Som

Tableau II.1 Description de la différent méthode d'inférence [22].

II.2.2.3 Défuzzification

Elle consiste à définir précisément quelle doit être l'action sur le processus. En effet, le procédé ne peut pas interpréter des ordres linguistiques fournis par les méthodes d'inférences. L'opération de défuzzification permet de calculer à partir de la fonction d'appartenance résultante, la valeur réelle de la variable de sortie à appliquer au processus. Il existe plusieurs méthodes de défuzzification :

- Méthode du maxima.
- Méthode du centre de gravité.
- Méthode de la moyenne des maximas.

Le choix d'une méthode de défuzzification est un point très délicat lors de l'élaboration d'une technique de contrôle en logique floue. Celui-ci conditionnera en effet grandement l'évolution dynamique de la commande. [23, 24].

Il est toutefois reconnu que la méthode du centre de gravité donne les meilleurs résultats. Notons que des blocs de normalisation et de dénormalisation sont ajoutés respectivement en entrée et en sortie du régulateur flou afin qu'il soit transportable et adaptable même avec des paramètres différents [19].

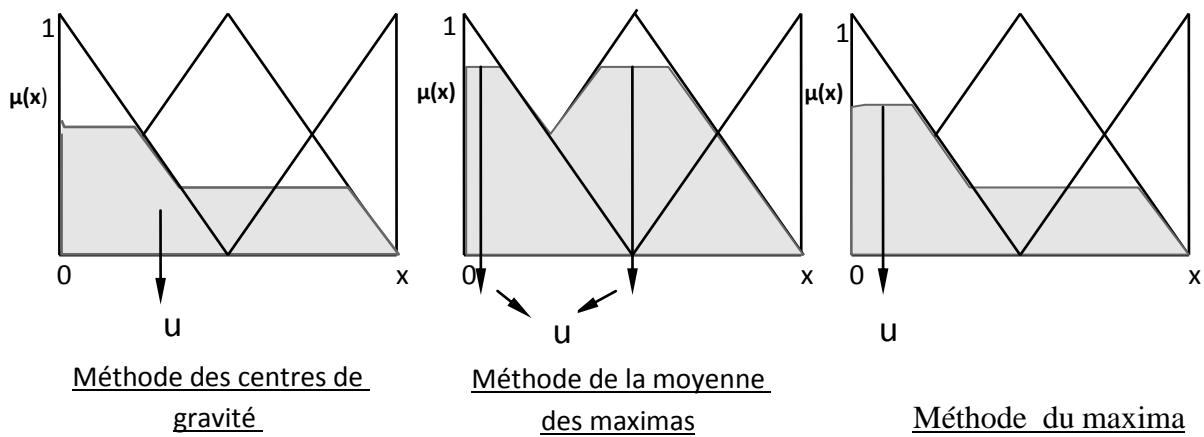


Figure II.2 Principe des différentes méthodes de défuzzification [22].

II.2.3 Avantage et désavantage du régulateur par la logique floue

Le réglage par logique floue réunit un certain nombre d'avantages et désavantages essentiels.

II.2.3.1 Les avantages [25]

- Le non nécessité d'une modélisation du système à régler.
- La possibilité d'implémenter des connaissances (linguistique) de l'opérateur de processus.
- La maîtrise du système à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).
- L'emploi possible aussi pour les processus rapides (grâce à des processeurs dédiés).
- La disponibilité de systèmes de développement efficaces soit pour microprocesseurs ou PC (solution logicielle), soit pour circuit intégrés (processeur dédiés, solution matérielle).

II.2.3.2 Les inconvénients [25]

- Le manque de directive précise pour la conception d'un réglage (choix des grandeurs à mesurer, détermination de la fuzzification, des inférences et de la défuzzification).
- L'approche artisanale et non systématique (implémentation des connaissances de l'opérateur souvent difficile).
- L'impossibilité de la démonstration de la stabilité du circuit de réglage en toute généralité (en l'absence d'un modèle valable).
- La possibilité d'apparition de cycles limites à cause du fonctionnement non linéaire.
- La précision de réglage souvent peu élevée.
- La cohérence des inférences non garantie a priori (apparition de règles d'inférence contradictoires possible).

II.3 Optimisation d'essaim de particules

II.3.1 Définition et historique

L'optimisation d'essaim de particules soit en anglais (Particle Swarm Optimization) (PSO) est une technique stochastique basée sur la population d'optimisation développée par Dr. Eberhart et Dr. Kennedy en 1995, inspiré par le comportement social des oiseaux s'assemblant ou le banc de poissons (Figure II.3) [26,27].



Figure II.3 Groupe de:(a) oiseaux, (b) poissons.

Depuis son apparition, PSO a gagné la popularité croissante parmi des chercheurs et des praticiens comme une technique robuste et efficace pour résoudre des problèmes d'optimisation difficiles. Dans le PSO, les particules individuelles d'un essaim représentent des solutions potentielles, qui se déplacent par le problème, fouillent dans l'espace cherchant un optimal, ou assez bon, la solution. La vitesse de chaque particule est ajustée en additionnant avec pondération sa vitesse à l'instant précédente; la différence entre la position précédente et sa meilleure position puis la différence entre la position précédente et la meilleure position globale. Quant à la position elle est mise à jours simplement par l'addition de la vitesse actuelle et la position à l'instant précédente [26,28].

PSO obtient de meilleurs résultats d'une manière plus rapide, comparé à d'autres méthodes. Une autre raison pour laquelle PSO est attrayant c'est qu'il y a peu de paramètres à ajuster [26,29].

II.3.2 Principe de base d'un PSO

Dans le PSO, chaque individu de la population est dit (particule), tandis que la population est connue sous le nom (swarm). Il faut noter, que la particule peut bénéficier des mouvements des autres particules dans la même population pour ajuster sa position et sa vitesse durant le processus d'optimisation. Chaque individu utilise l'information locale à laquelle il peut accéder sur le déplacement de ses plus proches voisins pour décider de son propre déplacement. Des règles très simples comme "rester proche des autres individus", "aller dans la même direction", "aller à la même vitesse" suffisent pour maintenir la cohésion du groupe tout entier.

L'espace de recherche est défini comme « voisinage », différentes topologies sont considérées :

- Les voisinages géographiques : les voisins d'une particule sont ses voisines les plus proches. Ce type de voisinage impose l'utilisation d'une distance pour recalculer à chaque itération (ou toutes les k itérations) les voisins de chaque particule. Ci-dessous, la figure (II.4) est un exemple où les voisins d'une particule sont les deux particules qui lui sont le plus proche [30].

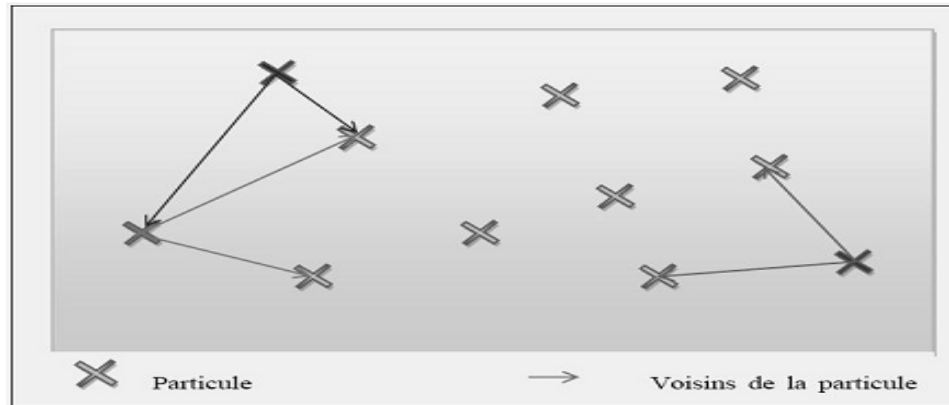


Figure II.4 Exemple de voisinage géographique [30].

- Les voisinages sociaux : les voisinages sont établis à l'initialisation et ne sont pas modifiés ensuite. Il existe différentes structures de voisinages sociaux, nous allons vous en présenter quelques-uns (Figure II.5) [30].

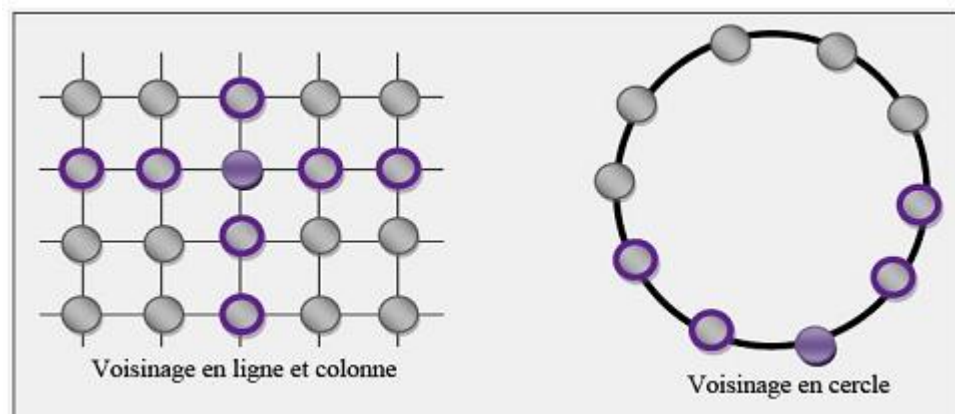


Figure II.5 Deux cas de voisinage social [30].

Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse 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, c'est-à-dire 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 congénères de son voisinage et d'obtenir de chacune d'entre elles sa propre meilleure performance.
- A chaque pas de temps, chaque particule choisit la meilleure des meilleures performances dont elle à connaissance, modifie sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence.

A partir des quelques informations dont elle dispose, une particule doit décider de son prochain mouvement, c'est-à-dire décider de sa nouvelle vitesse.

Pour ce faire, elle combine trois informations (voir figure II.6):

- ✚ Sa vitesse actuelle.
- ✚ Sa meilleure position actuelle.
- ✚ La meilleure performance (vitesse et position) de ses voisines.

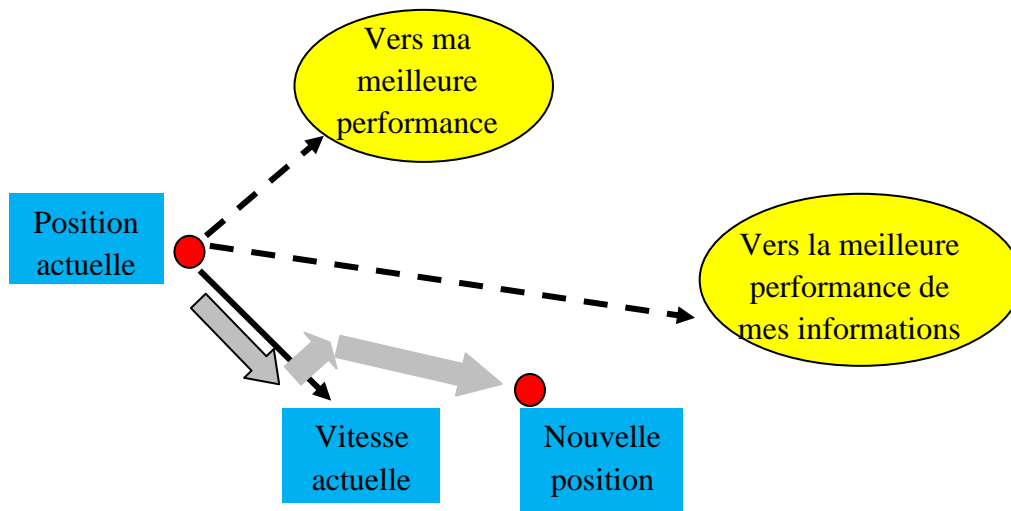


Figure II.6 Schéma de principe du déplacement d'une particule.

Le hasard joue un rôle, grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l'exploration de l'espace de recherche.

Naturellement, pour pouvoir être programmé, tout ceci est formalisé dans des équations de mouvement. Un point intéressant est que, contrairement à bien d'autres heuristiques qui restent purement expérimentales, il existe une analyse mathématique précisant les conditions de convergence et le choix des paramètres [27,30].

II.3.3 Formulation mathématique du PSO

Considérons une population (swarm) composée de K particules.

Les équations formalisant le mouvement des particules sont données par [31] :

$$V_i(t+1) = w \cdot v_i(t) + \varphi_1 \cdot r_1(t) \cdot (p_{bi}(t) - x_i(t)) + \varphi_2 \cdot r_2(t) \cdot (p_g(t) - x_i(t)) \quad (\text{II.1})$$

$$x_i(t+1) = x_i(t) + v_i(t) \quad (\text{II.2})$$

Chaque particule P_i ($i = 1, 2, \dots, K$) dans le Swarm est caractérisé par :

- 1) Sa position $x_i(t) \in \mathbb{R}^d$ courante, qui réfère à une solution candidate pour le problème d'optimisation considéré à l'itération t ;
- 2) Sa vitesse $v_i(t) \in \mathbb{R}^d$;
- 3) La meilleure position $p_{bi}(t) \in \mathbb{R}^d$ identifiée durant sa trajectoire antécédente. Soit $p_g(t) \in \mathbb{R}^d$ la meilleure position globale identifiée dans le processus de recherche pour toutes les particules dans le swarm. La position optimal est mesurée avec une fonction dit fitness définit suivant le problème d'optimisation. Durant l'optimisation, les particules se déplacent suivant les équations (II.1) et (II.2).
- 4) $r_1(\cdot)$ et $r_2(\cdot)$ sont des variables aléatoires générés d'une distribution uniforme dans l'intervalle $[0,1]$ afin de fournir un poids stochastique aux différentes composantes participant dans la définition de la vitesse de la particule.
- 5) φ_1 et φ_2 sont deux constantes d'accélération régulant les vitesses relatives par rapport aux meilleurs positions locales et globales. Ces paramètres sont considérés comme des facteurs d'échelle utilisés pour déterminer les mouvements relatifs de la meilleure position de la particule ainsi que de la meilleure position globale. Ce sont des facteurs qui déterminent le degré d'influence des positions passées de la particule elle mêmes et celles des autres particules dans le swarm.
- 6) L'inertie W est utilisée comme un compromis entre l'exploration locale et globale du swarm. Des valeurs élevées de ce paramètre permettent une bonne exploration globale, tandis que pour des valeurs minimales une recherche fine est réalisée.

L'équation (II.1) permet le calcul de la vitesse à l'itération $t + 1$ pour chaque particule en combinant linéairement la position et la vitesse (à l'itération t) et les distances qui séparent la position courante de la particule de son antécédente meilleure position et la meilleure position globale, respectivement. La mise à jour de la position de la particule est réalisée à travers l'équation (II.2). Les équations (II.1) et (II.2) sont itérées jusqu'à ce que la convergence soit atteinte [26,27].

Un organigramme pour être dresser comme suit (figure II.7):

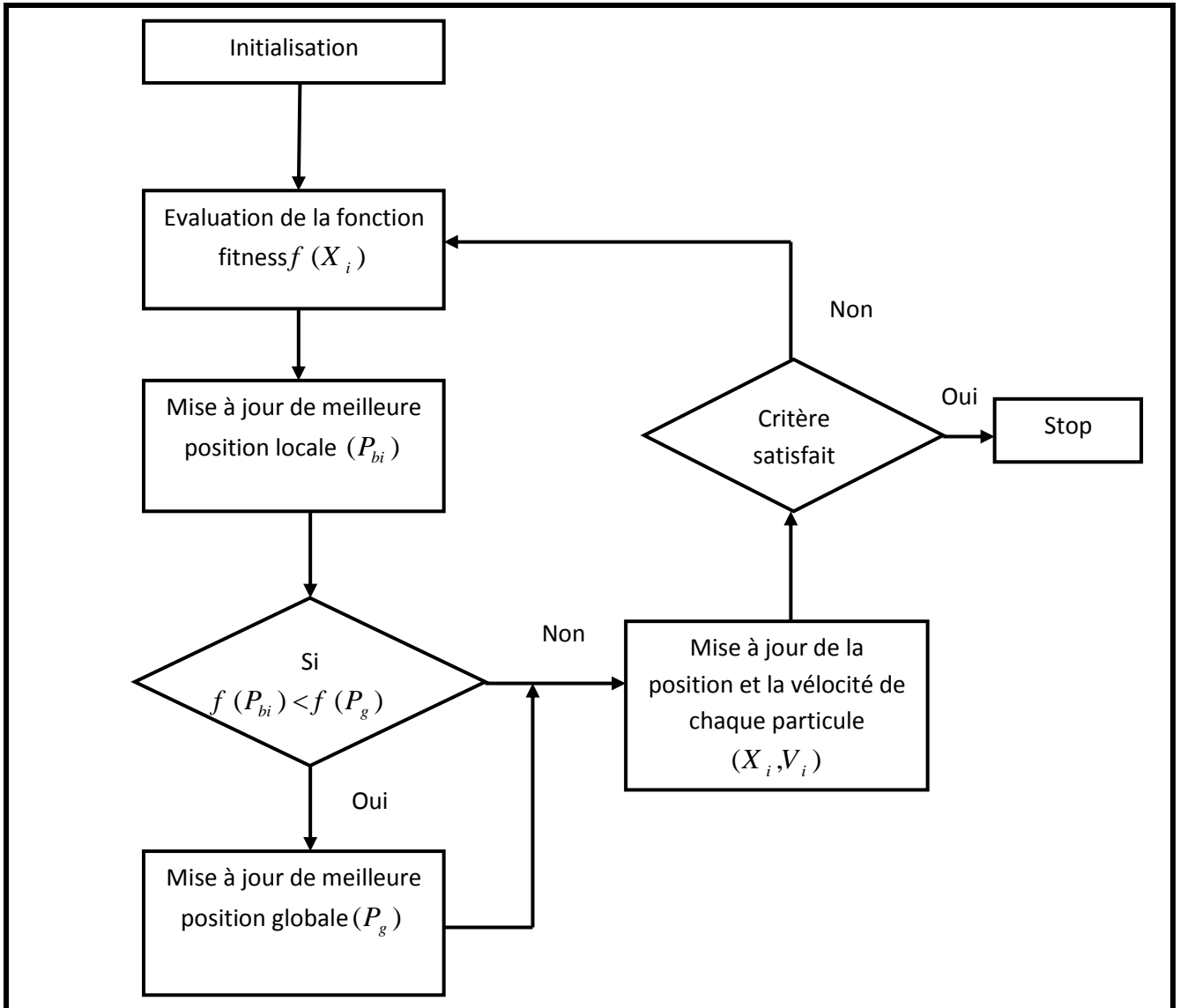


Figure II.7 Organigramme de l'algorithme de PSO [26].

L'implémentation des PSO a été effectuée en se référant à l'organigramme suscit. Mais avant tout, on donne quelques explications aux symboles utilisés [26]:

P_{bi} : représente la i^{eme} meilleure solution (position) de la particule i .

P_g : représente la meilleure solution (position) globale du groupe.

X_i : représente la solution (position) de la i^{eme} particule.

V_i : représente la vitesse (vitesse) de la i^{eme} particule.

f : La fonction fitness à optimiser.

φ_1, φ_2 : sont des pondérations choisies généralement égales à 2.

r_1, r_2 : sont deux variables aléatoires uniformément distribuées.

W : représente l'inertie est fixée à 0.8.

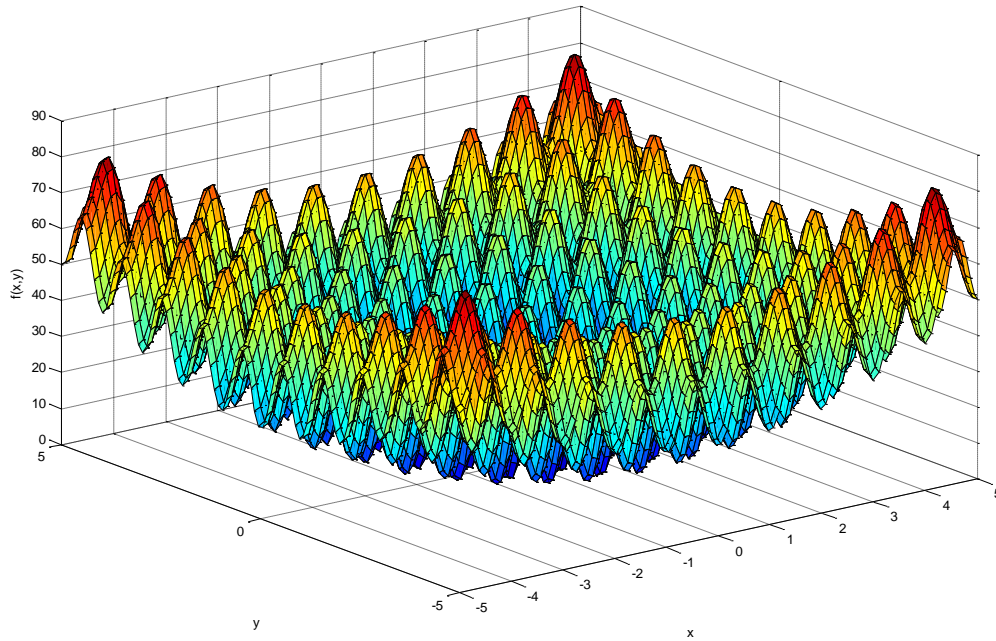
II.3.4 Exemple d'Application

Pour voir l'efficacité et la robustesse de la méthode PSO, cette partie présente un exemple qui montre comment trouver le minimum de la fonction de « *Rastrigin* » [32], qui est une fonction employée pour examiner la qualité de la convergence de cette méthode PSO.

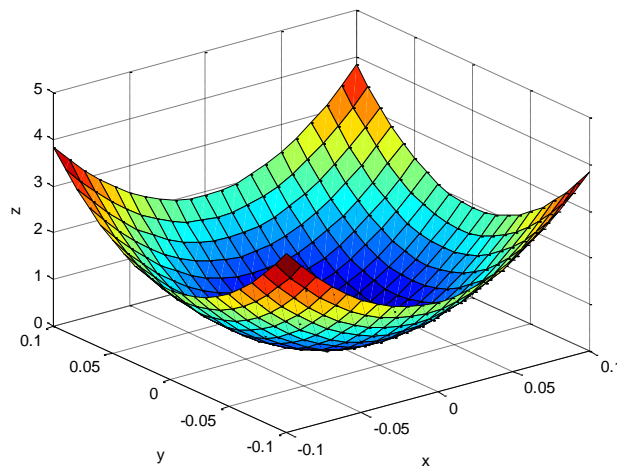
Cette fonction a deux variables x et y où elle est définie par [32]:

$$f(x, y) = 20 + x^2 + y^2 - 10 \cdot (\cos(2\pi \cdot x) + \cos(2\pi \cdot y)) \quad (\text{II.3})$$

où : $-5 < x < 5$ et $-5 < y < 5$



(a) avec plusieurs optimums locaux.



(b) optimum global au point $f(0,0)=0$.

Figure II.8 Fonction de Rastrigin.

Comme le montre la figure (II.8a et b), nous constatons que la fonction précédente présente plusieurs optimums locaux et admis un seul optimum global au point (0,0) tel que $f(0,0)=0$.

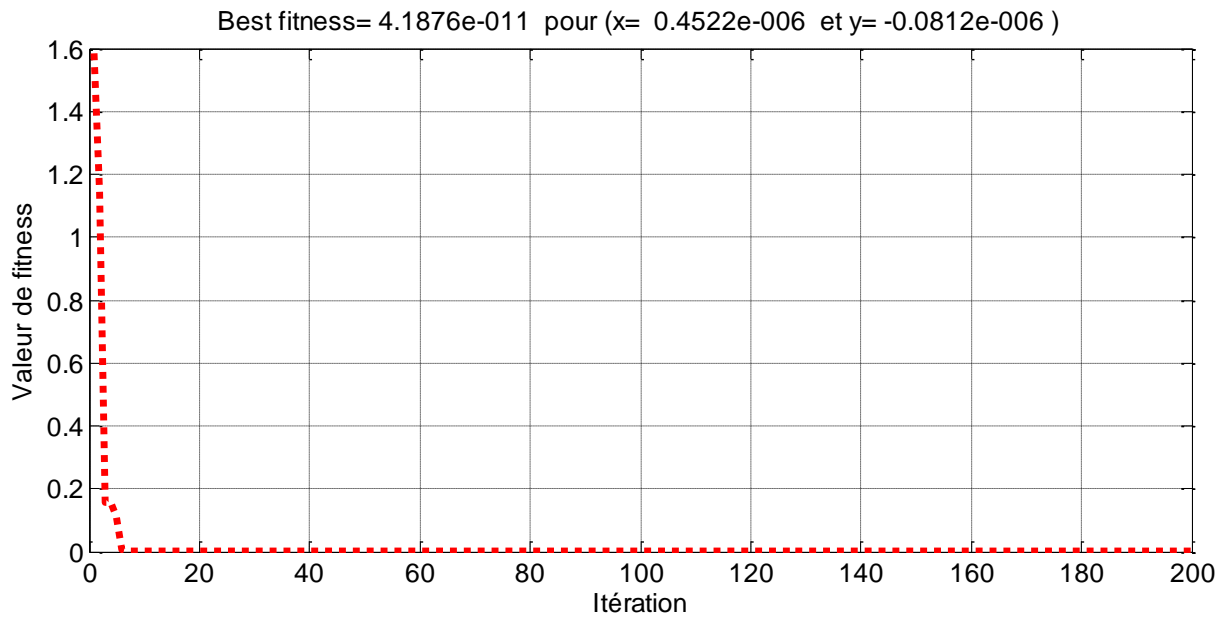


Figure II.9 Convergence de la fonction objectif vers l'optimum global (PSO).

II.4 conclusion

Dans ce chapitre, nous avons présenté deux techniques heuristiques. D'abord on a présenté la technique de la logique floue, ainsi que son principe de fonctionnement. Ensuite, on a évoqué aussi la deuxième technique qui est l'optimisation d'essaim de particules soit en anglais (Particle Swarm Optimization), en exposant son principe de base ainsi que sa formulation mathématique avec un exemple d'application pour montrer son efficacité.

En général, les méthodes déterministes convergent vers l'optimum le plus proche qu'il soit local ou global. A l'opposée, les méthodes stochastiques convergent, grâce à l'intelligence collective, vers l'optimum global. Parmi ces méthodes stochastiques, la technique PSO, comme une autre alternative très simple à mettre en œuvre, montrant, récemment, une grande aptitude à résoudre des problèmes d'optimisation dans un grand nombre d'applications de l'engineering.

Comme les méthodes déterministes ne permettent pas de tenir compte directement d'éventuelles variations paramétriques d'une machine électrique. Alors on a opté pour les méthodes stochastiques en particulier l'optimisation d'essaim de particules pour le calcul adaptatif des paramètres du régulateur PI afin d'améliorer les performances de la commande vectorielle adoptée.

Le dernier chapitre sera consacré à l'application de ces deux techniques pour l'ajustage des paramètres du régulateur PI associé à la MAS.