

## *Chapitre 3*

### *L'optimisation par essaim particulaire*

## INTRODUCTION

L'optimisation est un paradigme actuel dans chaque aspect de la vie. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation. Bien que les problèmes d'optimisation soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes complexe, stochastique, ou fortement non-linéaire et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Dans ce chapitre, l'accent sera mis sur une méthode (algorithme) d'optimisation intelligente classée comme métaheuristique inspirée par analogie de l'éthologie. Il s'agit de l'algorithme d'optimisation par essaim particulaire (OEP). Nous nous intéressons, par la suite, notamment à la présentation des différents types de l'algorithme d'optimisation par essaim particulaire, et l'efficacité de cet algorithme dans la résolution des problèmes non linéaires.

### 1. Optimisation par essaim particulaire

La bio-inspiration est un changement de paradigme qui amène les ingénieurs à s'inspirer de la nature pour développer de nouveaux systèmes artificiels. Depuis son apparition, l'optimisation par essaim particulaire a gagné la popularité croissante parmi des chercheurs et des praticiens comme une technique robuste et efficace pour résoudre des problèmes d'optimisation difficiles [55].

L'optimisation par essaim particulaire (OEP) est classée dans les techniques d'optimisation stochastiques à population introduite en 1995 par *James Kennedy* (psychologue social et statisticien) et *Russel Eberhart* (ingénieur) sous le nom Particle Swarm Optimization (PSO) [51],[54],[55],[58]. PSO est inspiré par le comportement social des animaux, comme les essaims d'abeilles, les vols d'oiseaux et les bancs de poissons [53],[54],[55]. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture, ou émigrent, Si un membre voit un chemin souhaitable pour l'exploration, le reste de cet essaim suivra rapidement. Raison pour laquelle, cette technique est fondée sur la notion de coopération et l'échange d'information entre les particules. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

## 2. Les différents types du PSO

### 2.1. PSO de l'optimum global

Dans cette version le voisinage pour chaque particule est l'essaim entier. Le réseau social utilisé par ce type d'algorithme est la topologie d'étoile, figure (3.1). Pour la topologie de voisinage d'étoile, le composant social de la mise à jour de vitesse de particules reflète l'information obtenue à partir de toutes les particules dans l'essaim. Dans ce cas-ci, l'information sociale est la meilleure position trouvée par l'essaim, désigné sous le nom de  $g$ .

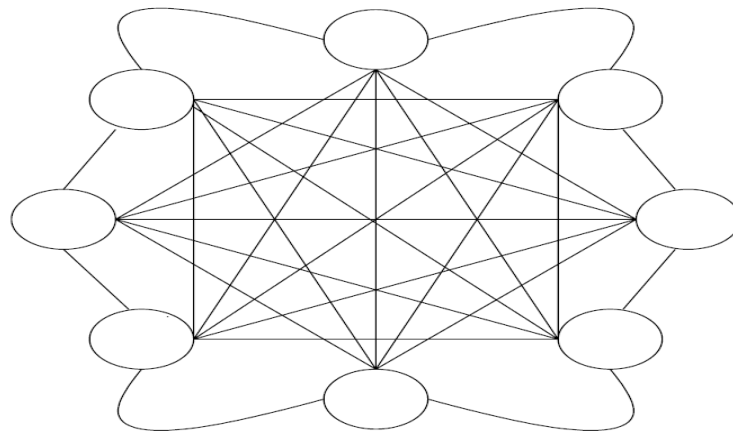


Figure 3.1 L'étoile topologie de voisinage.

#### 2.1.1. Principe de déplacement d'une particule dans PSO de l'optimum global

L'algorithme PSO dans sa version globale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche [53],[57]. Toutes les particules étant considérées voisines et chacune d'elle dispose d'une vitesse et une petite mémoire, lui permettant de se souvenir de sa meilleure performance, en position et en valeur. La performance de chaque particule est mesurée selon une fonction objective prédéfinie relative au problème à résoudre.

La théorie de la socio-psychologie suggère que les individus sont influencés par leur comportement passé et par le succès de leurs voisins dans le réseau social et non pas nécessairement dans l'espace [56]. Dans ce contexte, chaque particule interroge chaque congénère de proximité sur sa meilleure performance. Sur la base des informations dont elle dispose, chaque particule modifie sa vitesse d'une manière stochastique en fonction de sa propre vitesse actuelle, sa meilleure performance et la meilleure performance de ses voisins.

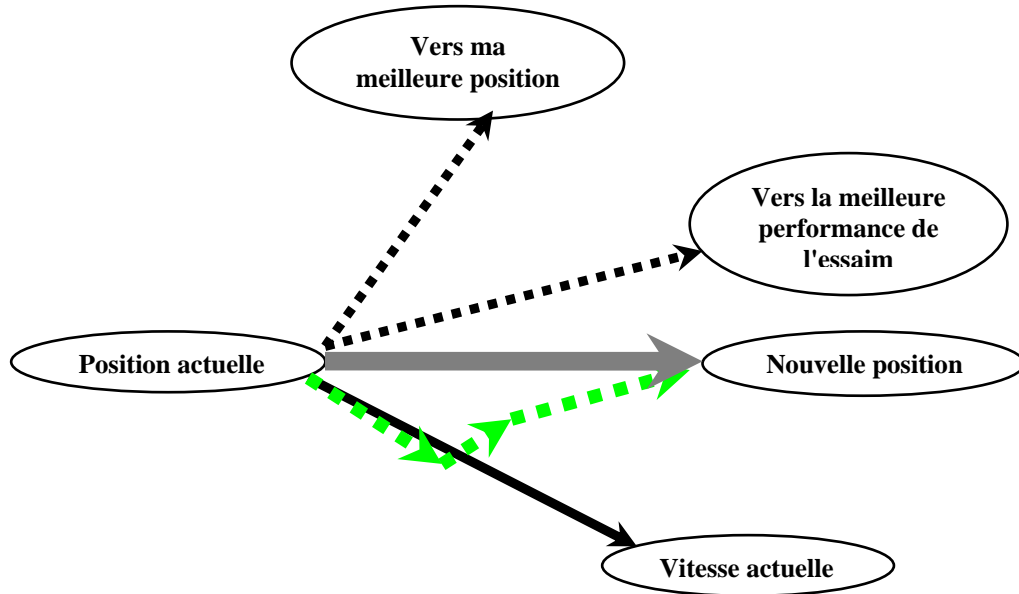


Figure 3.2 Principe de déplacement d'une particule dans PSO de l'optimum global.

La figure (3.2) montre que la décision de déplacement d'une particule est basée sur une combinaison linéaire de trois tendances à savoir de suivre sa propre voie, revenir vers sa meilleure performance et aller vers la meilleure performance de ses voisins.

Chaque membre de l'essaim conserve une connaissance partielle de sa meilleure position  $p^i$ , et la meilleure position de toutes les particules de l'essaim  $g$  et ajuste dynamiquement sa propre vitesse en se basant sur ces informations. L'ajustement de vitesse est basé sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons. La particule la mieux placée dans une région proche de l'endroit de la nourriture (optimum global) va motiver l'essaim d'explorer de proximité cette région et les régions adjacentes. De cette façon, les particules tendent à se rapprocher davantage de cet optimum d'une étape à l'autre durant le processus de recherche. Dans chaque étape on définit une nouvelle meilleure particule qui, à son tour, va attirer vers elle

l'ensemble de l'essaim.

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons .

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k+1) = wv^i(k) + c_1 rand_1 (p^i - x^i(k)) + c_2 rand_2 (g - x^i(k)) \quad (3.1)$$

Où

$k$  : est le nombre d'itérations,

$w$  : est le coefficient d'inertie,

$v^i(k)$  : est la vitesse courante de la particule  $i$  à l'itération  $k$ ,

$x^i(k)$  : est la position actuelle de la particule  $i$  à l'itération  $k$ ,

$c_1, c_2$  : sont les coefficients d'accélération,

$rand_1, rand_2$  : sont des nombres aléatoires de distribution uniforme sur  $[0,1]$ .

Chaque particule rejoint sa nouvelle position dans l'espace de recherche en utilisant l'équation suivante :

$$x^i(k+1) = x^i(k) + v^i(k+1) \quad (3.2)$$

Après un certain nombre d'itérations, les particules se regroupent par la suite dans la région où les solutions sont les plus convenables (probables). Si une particule tend à sortir de l'espace de recherche, on lui attribue la valeur du point frontière le plus proche avec ralentissement voir annulation de sa vitesse.

L'algorithme PSO de l'optimum global a été décrite par plusieurs auteurs [51],[53],[52],[56]. L'organigramme de la figure (3.3) montre comment PSO de l'optimum global opère.

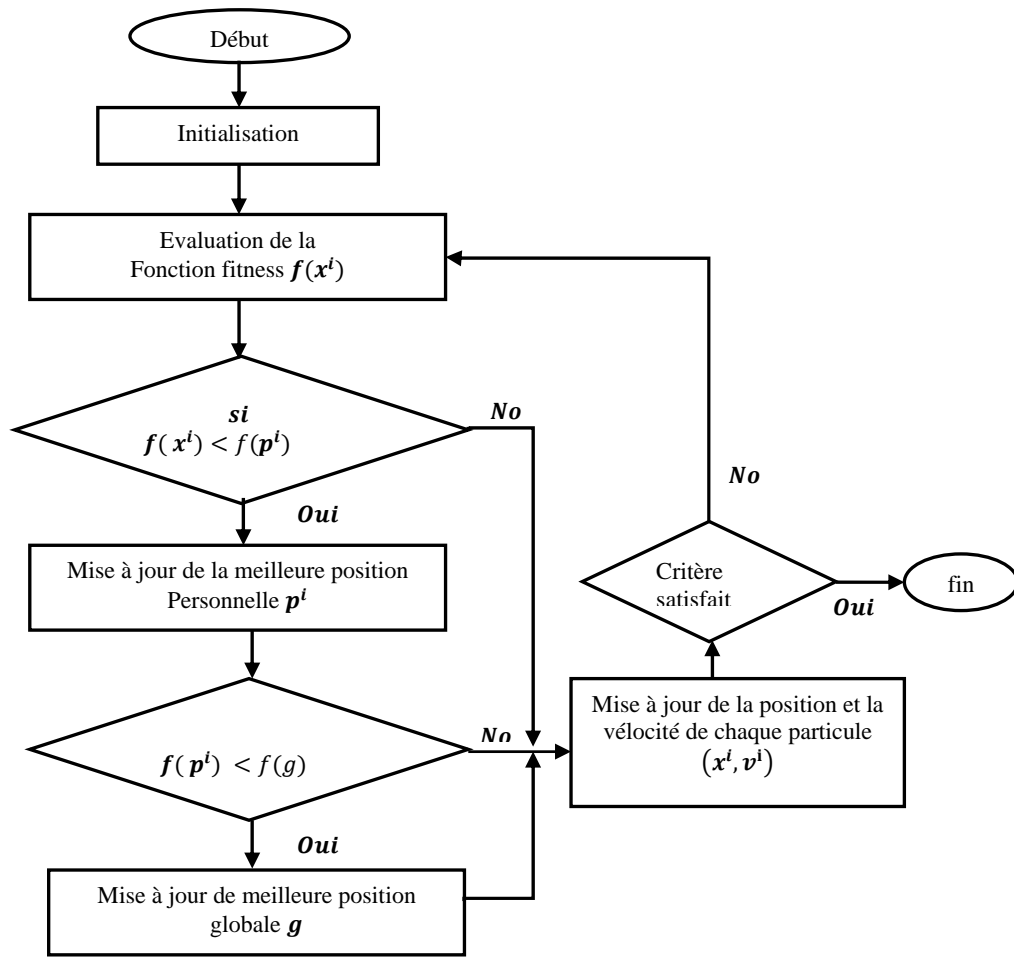


Figure 3.3 Organigramme de l'algorithme PSO de l'optimum global.

## 2.2. PSO de l'optimum local

Dans cette version de PSO, utilisations d'une topologie de réseau sociale d'anneau où des plus petits voisinages sont définis pour chaque particule, figure (3.4). Le composant social reflète l'information échangée dans le voisinage de la particule, reflétant la connaissance locale de l'environnement. La contribution sociale à la vitesse de particules est proportionnelle à la distance entre une particule et la meilleure position a trouvé par le voisinage des particules.

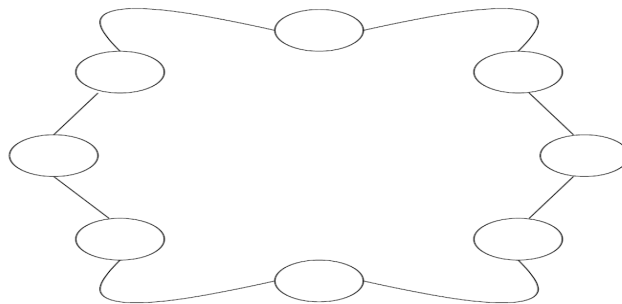


Figure 3.4 L'anneau topologie de voisinage.

### 2.2.1. Principe de déplacement d'une particule dans PSO de l'optimum local

L'algorithme PSO dans sa version locale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche[53]. comme illustré sur la figure (3.5), montre comment des particules sont influencées par leurs voisins immédiats. Pour maintenir le graphique lisible, seulement certains des mouvements sont illustrés, et seulement la direction globale de vitesse est indiquée. Le *voisinage 1*, les particules *a* et *b* se déplacent vers la particule *c*, qui est la meilleure solution dans ce voisinage. Considérant le *voisinage 2*, les mouvements des particules *d* vers *f*, fait ainsi *e*. Pour la prochaine itération, *e* sera la meilleure solution pour le *voisinage 2*. Mouvement maintenant de *d* et de *f* vers *e* comme illustré sur la figure (3.5) (b) (seulement une partie de l'espace de solution est illustrée). Les blocs représentent les positions précédentes. Noter que *e* reste la meilleure solution pour le *voisinage 2*. Noter également le mouvement général vers le minimum.

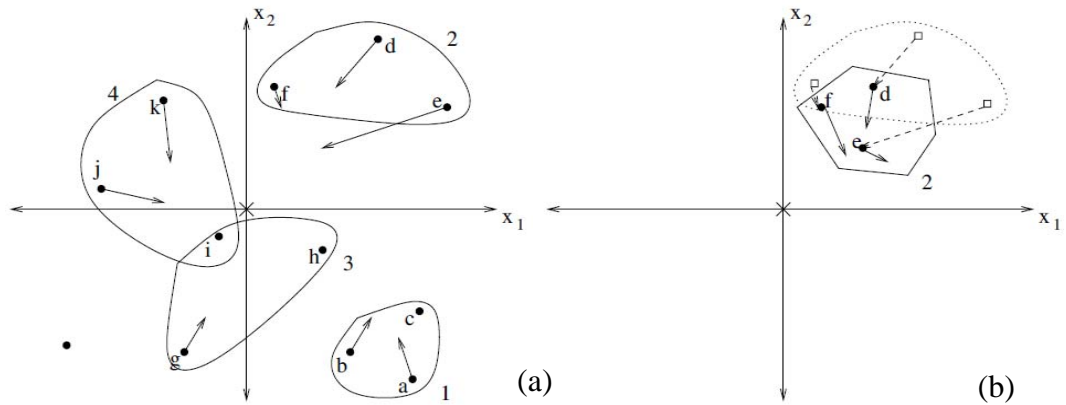


Figure 3.5 Principe de déplacement d'une particule dans PSO de l'optimum local.

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs voisins immédiats.

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k+1) = wv^i(k) + c_1 rand_1(p^i - x^i(k)) + c_2 rand_2(g_n - x^i(k)) \quad (3.3)$$

Où

$g_n$  : est la meilleure position découverte par les voisins immédiats .

la meilleure position découverte par les voisins immédiats  $N_i$  est défini comme suit :

$$g_n(k+1) \in \{N_i / f(g_n(k+1)) = \min\{f(x(k))\}, \forall x \in N_i\} \quad (3.4)$$

Il est important de noter que pour la PSO de base, les particules dans un voisinage  $N_i$  n'ont aucun rapport entre eux. Le choix des voisinages est fait à basé sur des index de particules. Cependant, des stratégies ont été développées où des voisinages sont formés ont basé sur la similitude spatiale.

L'algorithme PSO de l'optimum local a été décrite par plusieurs auteurs [51],[53],[52] [56]. L'organigramme de la figure (3.6) montre comment PSO de l'optimum local opère.

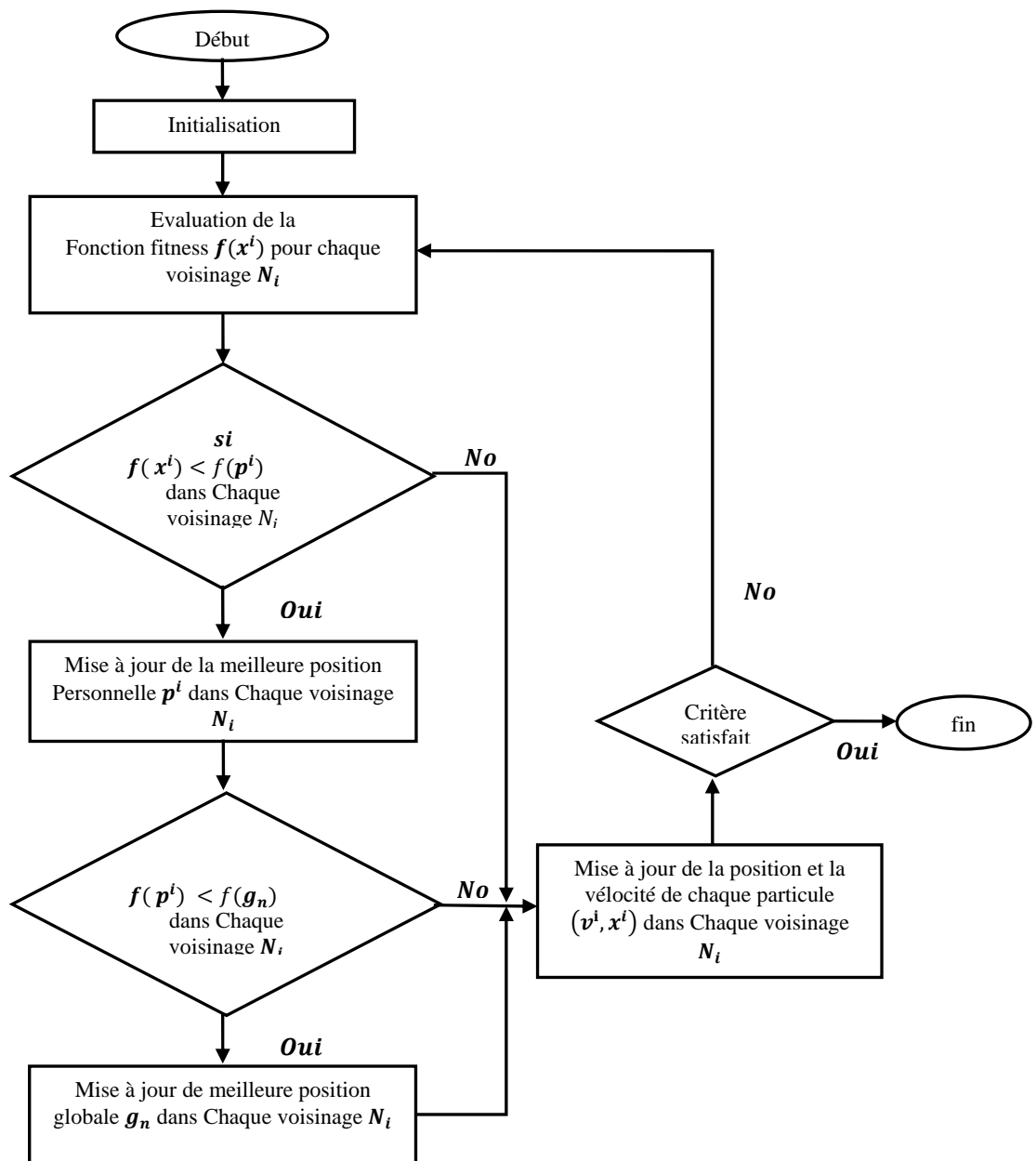


Figure 3.6 Organigramme de l'algorithme PSO de l'optimum local.

### 3. Les Composants de vitesse

Le calcul de vitesse du  $i^{eme}$  particule comme donné dans les équations (3.1) et (3.2) compose de trois termes :

- **La vitesse précédente**,  $v^i(k)$ , qui sert de mémoire de la direction précédente de vol, c.-à-d. mouvement dans le passé immédiat. Cette terme de mémoire peut être vue comme élan, qui empêche la particule de la direction rigoureusement changeante, et à la polarisation vers la direction courante. Ce composant désigné également sous le nom du composant d'inertie.
- **Le composant cognitif**,  $c_1 rand_1 (p^i - x^i(k))$ , qui mesure la performance de la particule  $i$  relativement à sa performance passée. Dans une certaine mesure, le composant cognitif ressemble la mémoire individuelle du position qui était la meilleure pour la particule. L'effet de cette terme est que ces particules sont attirées à leurs propres meilleures positions, ressemblant la tendance des individus de retourner aux situations ou aux endroits qui les ont satisfaits plus dans le passé. *Kennedy et Eberhart* se sont également référés au composant cognitif comme « nostalgie » de la particule.
- **Le composant social**,  $c_2 rand_2 (g - x^i(k))$ , dans la version globale de PSO ou ,  $c_2 rand_2 (g_n - x^i(k))$ , dans la version local ,qui mesure la performance de la particule  $i$  relativement à un groupe de particules, ou des voisins. Conceptuellement, le composant social ressemble à une norme ou à une norme de groupe que les individus cherchent à atteindre. L'effet du composant social est que chaque particule est attirée vers la meilleure position trouvée par son voisinage.

La figure (3.7) présente une illustration géométrique des mises à jour de vitesse et de position:

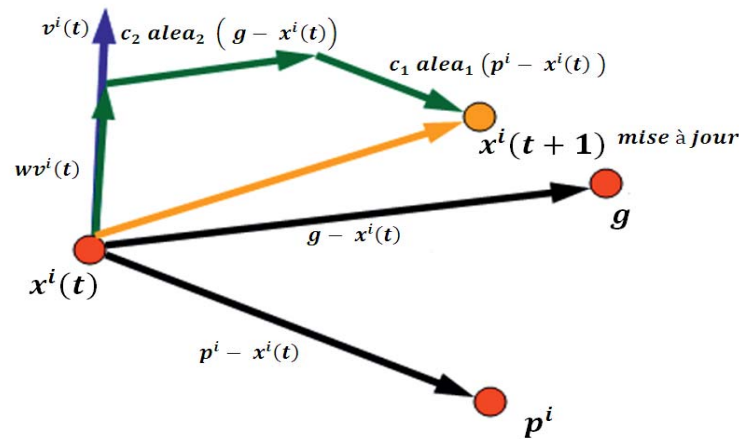


Figure 3.7 Schéma vectorielle du déplacement d'une particule.

#### 4. Les paramètres du PSO

Le PSO de base est influencé par un certain nombre de paramètres de contrôle, à savoir la dimension du problème, nombre de particules, coefficients d'accélération, poids d'inertie, voisinage taille, nombre d'itérations, et les valeurs aléatoires qui influencent la contribution des composants cognitifs et sociaux [53],[59].

- **Taille de l'essaim** : On lui a montré dans un certain nombre d'études empiriques que le PSO a la capacité de trouver les solutions optimales avec de petites tailles d'essaim de 10 à 30 particules.
- **Taille de voisinage** : La taille de voisinage définit l'ampleur de l'interaction sociale dans l'essaim. Plus les voisinages sont petits, moins l'interaction se produit. Tandis que de plus petits voisinages sont plus lents dans la convergence, ils ont une convergence plus fiable aux solutions optimales. Des plus petites tailles de voisinage sont moins susceptibles des minimum locaux. Pour profiter des avantages de petites et grandes tailles de voisinage, commencer la recherche avec de petits voisinages et augmenter la taille de voisinage proportionnellement à l'augmentation en nombre des itérations. Cette approche assure une première diversité élevée avec une convergence plus rapide pendant que les particules se déplacent vers un secteur prometteur de recherche.
- **Nombre d'itérations** : Le nombre d'itérations pour atteindre une bonne solution est également dépendante à la problème. Trop peu d'itérations peuvent terminer la recherche prématurément. Un nombre trop grand d'itérations a la conséquence de la complexité du calcul supplémentaire inutile (à condition que le nombre d'itérations est le seul état d'arrêt).

- **Les coefficients d'accélération:** influencent la valeur maximale du déplacement qu'une particule peut prendre durant une seule itération.  $c_1$ ,  $c_2$  sont désignés également sous le nom de paramètres cognitif et social, respectivement. Les nombres aléatoires  $rand_1$ ,  $rand_2$  sont employés pour affecter la nature stochastique de l'algorithme.
- **Le coefficient (poids) d'inertie :** a été développée par *Eberhart et al*, régit la quantité de vitesse actuelle qui devrait être maintenue pour influencer le calcul de la nouvelle vitesse. De cette façon, le paramètre  $w$  règle les capacités d'exploration globale et locale de l'essaim et influence le comportement de convergence de l'algorithme. Un grand poids d'inertie facilite l'exploration globale (exploration de nouveaux secteurs), alors qu'une petite valeur tend à faciliter l'exploration locale (exploration fine du secteur courant de recherche). Une valeur appropriée du poids de l'inertie fournit habituellement l'équilibre entre les capacités globales et locales d'exploration et par conséquent une réduction du nombre d'itérations nécessaires pour localiser la solution optimale. En règle générale, il est souhaitable de commencer le poids d'inertie à une grande valeur, afin de faire une meilleure exploration globale de l'espace de recherche, et le diminue graduellement pour affiner l'exploration locale.

La figure (3.8) illustre le diminue progressif en poids de l'inertie:

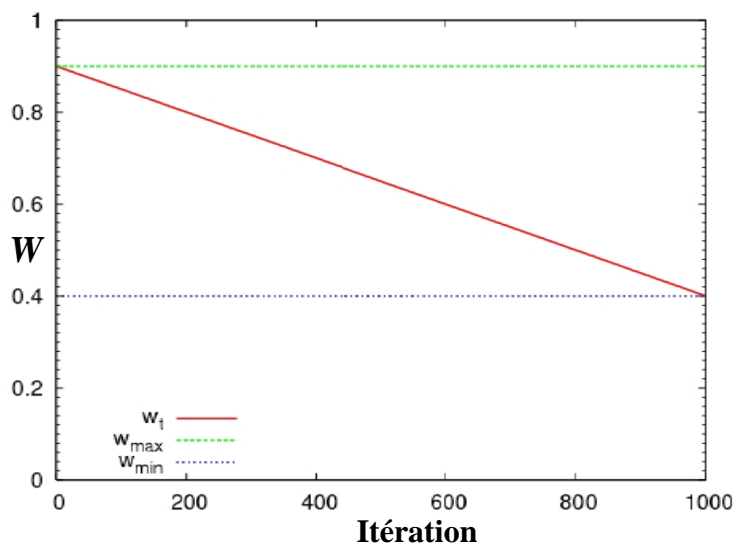


Figure 3.8 Le diminue progressif en poids de l'inertie.

## 5. La comparaison entre les deux versions du PSO

Il y a deux différences principales entre les deux approches en ce qui concerne leurs caractéristiques de convergence :

- En raison de la très grande interconnectivité des particules du PSO de l'optimum global, il converge plus rapidement que le PSO de l'optimum local. Cependant, cette convergence plus rapide vient au coût de moins de diversité que le PSO de l'optimum local.
- Comme conséquence de la très grande diversité du PSO de l'optimum local (qui a comme conséquence de plus grandes parties de l'espace de recherche étant couvert), le PSO de l'optimum local est moins susceptible de l'emprisonnement dans des minimums locaux. En général (selon le problème), les structures de voisinage telles que la topologie d'anneau utilisée dans le PSO de l'optimum local améliore les performances.

## 6. Test de l'efficacité du PSO

Dans le but de test les capacités du PSO, nous avons choisi de chercher l'optimum global de la fonction Rosenbrock définie par :

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (3.5)$$

Cette fonction, comme le montre la figure (3.9), a plusieurs optimums locaux et n'admet qu'un seul optimum global défini au point (1,1) tel que  $f(1,1) = 0$ .

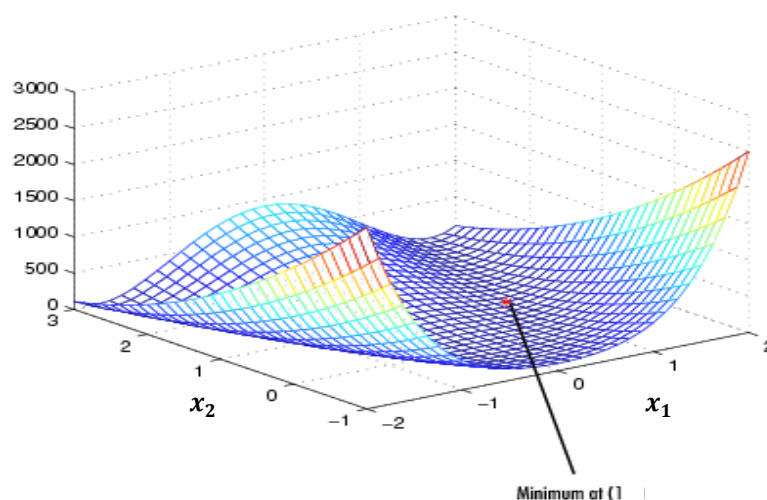


Figure 3.9 Fonction de Rosenbrock.

On prend :

- $S = 20$  ; taille de l'essaim.
- $k = 1000$  ; nombre d'itérations.
- $c_1 = c_2 = 1.8$
- $w$  ; d'après *Eberhart et al*
- Condition d'arrêt est ( $k = 1000$  Ou la valeur du fonction objective "erreur" =0)

Le résultat obtenu est :  $x_1 = 1$  ,  $x_2 = 1$  , *erreur*  $\cong 0$  *après 357 itirations*

La figure (3.10) montre la convergence de la fonction objective vers l'optimum global.

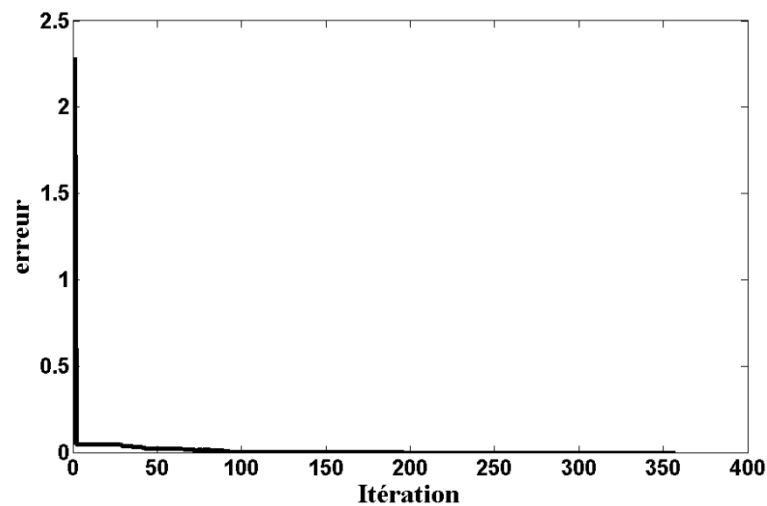


Figure 3.10 convergence de la fonction objective vers l'optimum global.

## 7. Les champs d'application du PSO

Cet algorithme se révèle efficace pour les problèmes d'optimisation non linéaires à variables continue ou discrète. Il a fait sa preuve dans des domaines variés entre autres comme l'optimisation de:

- l'apprentissage de réseaux de neurones [3],
- la commande par logique floue [3],
- les régulateurs PID [5],
- régulation et optimisation de l'écoulement de puissance [4],
- l'identification et le diagnostic des machines électriques [4]
- la commande des convertisseurs de puissance [4].
- ...etc.

## CONCLUSION

Parce qu'elle est simple à comprendre, à programmer et à utiliser, PSO se révèle efficace pour les problèmes d'optimisation non linéaires. La puissance de cet algorithme découle de la nature même de son principe fondé sur la notion de coopération entre les particules. PSO semble très performant en termes de la précision des solutions trouvées, du temps de calcul et du nombre de paramètres à ajuster. Pour notre système, nous avons opté pour l'algorithme PSO de l'optimum global car il semble bien adapté au problème posé en termes de temps de calcul.

L'analyse et l'étude des performances de ces méthodes choisies appliquées au domaine de contrôle et de surveillance des eaux propres. Constituent notre principal objectif. Une étude en simulation ayant pour but d'évolution des performances de ces techniques proposées, fera l'objet du chapitre suivant. L'évaluation des résultats, reflétant les performances obtenues, nous conduira au meilleur choix de la méthode hybride la mieux adaptée à l'application. L'architecture du système de contrôle et de surveillance proposé est aussi présentée.