

---

# Optimisation par essaim particulaire

Guillaume CALAS

[guillaume.calas@gmail.com](mailto:guillaume.calas@gmail.com)

Spécialisation *Sciences Cognitives et Informatique Avancée*



14-16 rue Voltaire,  
94270 Le Kremlin-Bicêtre,  
France

---

**Mots clés:** optimisation, métaheuristiques, essaim de particules, essaim particulaire, OEP, *particle swarm optimisation*, PSO.

## Résumé

L'apparition des algorithmes évolutionnistes à fait l'effet d'une bombe dans les domaines de la résolution de problèmes complexes, et spécialement dans l'optimisation de fonction avec contraintes. L'optimisation par essaim de particules se présente comme une alternative aux algorithmes génétiques et aux colonies de fourmis pour l'optimisation de fonctions non-linéaires.

Ce document a pour objet de présenter succinctement l'origine de cette méthode avant d'aborder sa mise en œuvre et les réglages de ses paramètres.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Applications</b>	<b>1</b>
<b>3</b>	<b>Présentation de la méthode</b>	<b>1</b>
3.1	Principe . . . . .	1
3.2	Formalisation . . . . .	1
3.3	Configuration de la méthode . . . . .	2
3.3.1	Nombre de particules . . . . .	2
3.3.2	Topologie du voisinage . . . . .	2
3.3.3	Coefficients de confiance . . . . .	2
3.3.4	Vitesse maximale et coefficient de constriction . . . . .	2
3.3.5	Facteur d'inertie . . . . .	2
3.3.6	Initialisation de l'essaim . . . . .	3
3.3.7	Critères d'arrêt . . . . .	3
3.4	Algorithme de synthèse . . . . .	3

## 1 Introduction

L'optimisation par essaim de particules (*Particle Swarm Optimization* (PSO) dans la langue de SHAKESPEARE) est une méthode d'optimisation stochastique, pour les fonctions non-linéaires, basée sur la reproduction d'un comportement social et développée par le Dr. EBERHART et le Dr. KENNEDY [4, 2] en 1995.

L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons de REYNOLD [5], HEPPNER et GREANDER [3]. Ces simulations ont mis en valeur la capacité des individus d'un groupe en mouvement à conserver une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage.

D'autre part, ces simulations ont également révélé l'importance du mimétisme dans la compétition qui oppose les individus à la recherche de la nourriture. En effet, les individus sont à la recherche de sources de nourriture qui sont dispersés de façon aléatoire dans un espace de recherche, et dès lors qu'un individu localise une source de nourriture, les autres individus vont alors chercher à le reproduire.

Ce comportement social basé sur l'analyse de l'environnement et du voisinage constitue alors une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus.

## 2 Applications

Les essaims de particules sont essentiellement utilisés afin de trouver l'optimum de fonctions non-linéaires. Pour cette raison, cette méthode est utile pour optimiser l'entraînement des réseaux de neurones.

## 3 Présentation de la méthode

### 3.1 Principe

L'optimisation par essaim de particules repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors des particules, qui se déplacent dans l'hyper-espace de recherche et consistent, chacune, une solution potentielle.

Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage. À partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution

optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage.

À partir d'optimums locaux et empiriques, l'ensemble des particules va, normalement, converger vers la solution optimale globale du problème traité.

### 3.2 Formalisation

Un essaim de particule est caractérisé par :

- le nombre de particules** de l'essaim, noté  $nb$  (3.3.1) ;
- la vitesse maximale** d'une particule, notée  $\vec{v}_{max}$  (3.3.4) ;
- la topologie et la taille du voisinage** d'une particule qui définissent son réseau social (3.3.2).
- l'inertie** d'une particule, notée  $\Psi$  (3.3.5) ;
- les coefficients de confiance**, notés  $\rho_1$  et  $\rho_2$ , qui pondèrent le comportement conservateur (ie. la tendance à retourner vers la meilleure solution visitée) et le panurgisme (ie. la tendance à suivre le voisinage) (3.3.3).

Une particule est caractérisée, à l'instant  $t$ , par :

- $\vec{x}_i(t)$  : sa position dans l'espace de recherche ;
- $\vec{v}_i(t)$  : sa vitesse ;
- $\vec{x}_{pbest_i}$  : la position de la meilleure solution par laquelle elle est passée ;
- $\vec{x}_{vbest_i}$  : la position de la meilleure solution connue de son voisinage ;
- $pbest_i$  : la valeur de *fitness* de sa meilleure solution ;
- $vbest_i$  : la valeur de *fitness* de la meilleure solution connu du voisinage ;

---

#### Algorithme 1 version simpliste (sans voisinage)

---

ENTRÉES:  $0 < \rho < 1$

répéter

pour  $i = 1$  jusqu'à  $nb$  faire

si  $F(\vec{x}_i) > pbest_i$  alors

$$pbest_i = F(\vec{x}_i)$$

$$\vec{x}_{pbest_i} = \vec{x}_i$$

fin si

$$\vec{v}_i = \vec{v}_i + \rho(\vec{x}_{pbest_i} - \vec{x}_i)$$

$$\vec{x}_i = \vec{x}_i + \vec{v}_i$$

fin pour

jusqu'à (un des critères de convergence est atteint)

---

### 3.3 Configuration de la méthode

#### 3.3.1 Nombre de particules

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

#### 3.3.2 Topologie du voisinage

La topologie du voisinage défini avec qui chacune des particules va pouvoir communiquer. Il existe de nombreuses combinaisons dont les suivantes sont les plus utilisées :

- topologie en étoile** : chaque particule est reliée à toutes les autres, ie. l'optimum du voisinage est l'optimum global ;
- topologie en anneau** : chaque particule est reliée à  $n$  particules (en général,  $n = 3$ ), c'est la topologie la plus utilisée ;
- topologie en rayon** : les particules ne communiquent qu'avec une seule particule centrale ;

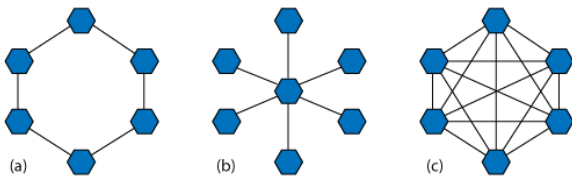


FIGURE 1 – (a) anneau (avec  $n = 2$ ), (b) rayon, (c) étoile.

Sous licence Creative Common Attribution ShareAlike 2.5.

Le voisinage géographique auquel nous sommes amenés à penser en premier lieu n'est pas nécessairement pertinent car, d'une part, il s'agirait d'un voisinage trop local, et d'autre part car la sociabilisation des particules tend à rendre tout voisinage social en voisinage géographique. Enfin, c'est un voisinage très lourd en terme de calculs car nécessitant de recalculer le voisinage de chaque particule à chaque itération.

#### 3.3.3 Coefficients de confiance

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme. Les variables aléatoires  $\rho_1$  et  $\rho_2$  peuvent être définies de la façon suivante :

$$\begin{cases} \rho_1 = r_1 \cdot c_1 \\ \rho_2 = r_2 \cdot c_2 \end{cases}$$

où  $r_1$  et  $r_2$  suivent une loi uniforme sur  $[0..1]$  et  $c_1$  et  $c_2$  sont des constantes positives déterminées de façon empirique et suivant la relation  $c_1 + c_2 \leq 4$ .

#### 3.3.4 Vitesse maximale et coefficient de constriction

Afin d'éviter que les particules ne se déplacent trop rapidement dans l'espace de recherche, passant éventuellement à côté de l'optimum, il peut être nécessaire de fixer une vitesse maximale (notée  $\vec{v}_{max}$ ) pour améliorer la convergence de l'algorithme.

Cependant, on peut s'en passer si on utilise un coefficient de constriction  $\kappa$  — introduit par Maurice CLERC [1] — et qui permet de resserrer l'hyper-espace de recherche.

L'équation de la vitesse devient alors :

$$\kappa = 1 - \frac{1}{\rho} + \frac{\sqrt{|\rho^2 - 4\rho|}}{2}$$

avec  $\rho = \rho_1 + \rho_2 > 4$ .

$$\begin{aligned} \vec{v}_i(t) &= \kappa \cdot (\vec{v}_i(t-1)) \\ &+ \rho_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i(t)) \\ &+ \rho_2 \cdot (\vec{x}_{vbest_i} - \vec{x}_i(t)) \end{aligned}$$

Les études de SHI et EBERHART indiquent que l'utilisation d'un coefficient de constriction donne généralement un meilleur taux de convergence sans avoir à fixer de vitesse maximale. Cependant, dans certains cas, le coefficient de constriction seul ne permet pas la convergence vers la solution optimale pour un nombre d'itérations donné. Pour résoudre ce problème, il peut être intéressant de fixer  $\vec{v}_{max} = \vec{x}_{max}$  en plus du coefficient de constriction, ce qui, selon les études de SHI et EBERHART, permet d'améliorer les performances globales de l'algorithme.

#### 3.3.5 Facteur d'inertie

Le facteur d'inertie  $\Psi$  — introduit par SHI et EBERHART — permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Une grande valeur de  $\Psi$  ( $> 1$ ) est synonyme d'une grande amplitude de mouvement et donc, *in fine*, d'exploration globale. *A contrario*, une faible valeur de  $\Psi$  ( $< 1$ ) est synonyme de faible amplitude de mouvement et donc, d'exploration locale. Fixer ce facteur, revient donc à trouver un compromis entre l'exploration locale et l'exploration globale.

Le calcul de la vitesse est alors défini par :

$$\begin{aligned} \vec{v}_i(t) &= \Psi \cdot \vec{v}_i(t-1) \\ &+ \rho_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i(t)) \\ &+ \rho_2 \cdot (\vec{x}_{vbest_i} - \vec{x}_i(t)) \end{aligned}$$

La taille du facteur d'inertie influence directement la taille de l'hyper-espace exploré et aucune valeur de  $\Psi$  ne peut garantir la convergence vers la solution optimale.

Les études menées par SHI et EBERHART indiquent une meilleure convergence pour  $\Psi \in [0.8, 1.2]$ . Au delà de 1.2, l'algorithme tend à avoir certaines difficultés à converger.

Enfin, il est également possible de faire diminuer le facteur d'inertie au cours du temps, un peu à la manière de la température dans un algorithme de recuit simulé (*Simulated Annealing*). De bons résultats ont été trouvés pour une valeur décroissant linéairement de 0.9 à 0.4.

Pour de plus amples informations sur le réglage de ce paramètre, veuillez vous référer à la thèse de VAN DEN BERGH[6].

### 3.3.6 Initialisation de l'essaim

La position des particules ainsi que leur vitesse initiale doivent être initialisés aléatoirement selon une loi uniforme sur  $[0..1]$ . Cependant, en ce qui concerne la position des particules, il est préférable d'utiliser un générateur de séquence de SOBOLOV qui est plus pertinent dans la disposition homogène des particules dans un espace de dimension  $n$ .

### 3.3.7 Critères d'arrêt

Comme indiqué précédemment, la convergence vers la solution optimale globale n'est pas garantie dans tous les cas de figure même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une porte de sortie en définissant un nombre maximum d'itération (que nous noterons  $nbIter_{max}$ ).

L'algorithme doit alors s'exécuter tant que l'un des critères de convergence suivant n'a pas été atteint :

- $nbIter_{max}$  a été atteint ;
- la variation de la vitesse est proche de 0 ;
- le *fitness* de la solution est suffisant.

## 3.4 Algorithme de synthèse

### Algorithme 2 version simpliste (avec voisinage)

**répéter**

**pour**  $i = 1$  **jusqu'à**  $nb$  **faire**

**si**  $F(\vec{x}_i) > pbest_i$  **alors**

$pbest_i = F(\vec{x}_i)$

$\vec{x}_{pbest_i} = \vec{x}_i$

**fin si**

**si**  $F(\vec{x}_i) > vbest_i$  **alors**

$vbest_i = F(\vec{x}_i)$

$\vec{x}_{vbest_i} = \vec{x}_i$

**fin si**

**fin pour**

**pour**  $i = 1$  **to**  $nb$  **faire**

$\vec{v}_i = \kappa(\vec{v}_i + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i) + \rho_2(\vec{x}_{vbest_i} - \vec{x}_i))$

$\vec{x}_i = \vec{x}_i + \vec{v}_i$

**fin pour**

**jusqu'à** (*un des critères de convergence est atteint*)

## Références

- [1] Maurice Clerc and James Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, 6(1) :58–73, 2002.
- [2] Russell C. Eberhart, Yuhui Shi, and James Kennedy. *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, USA, 2001.
- [3] F. Heppner and U. Grenander. *A stochastic nonlinear odel for coordinated bird flocks*. AAAS Publication, Washington, DC, 1990.
- [4] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1(1) :33–57, 2007.
- [5] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. *Computer Graphics*, 21(4) :25–34, 1987.
- [6] F. van den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, Department of Computer Science, University of Pretoria, November 2002.