4. MÉTODO DE ENXAME DE PARTÍCULAS (*PARTICLE SWARM)*

4. MÉTODO DE ENXAME DE PARTÍCULAS

4.1. Analogia Comportamental : *todos por um e um por todos*

Figura 1 –Uma imensa nuvem de pássaros [red-billed queleas] retornam a seu viveiro natural ao pôr do sol, Delta do Okavango, Botswana, África

O método de *enxame de partículas* pode ser explicado de uma forma simples através de uma analogia de comportamento social apresentada por M. Clerc [*The Swarm & The Queen. Towards a Deterministic and Adaptive Particle Swarm Optimization* de Maurice Clerc – Artigo disponibilizado na Internet na *Home Page* http://clerc.maurice.free.fr/PSO/]. O autor sugere a seguinte situação fictícia: "Suponha que você e alguns amigos estão à busca de um tesouro enterrado em uma ilha. Cada membro da expedição tem um rádio-comunicador-receptor, podendo se comunicar com todos os membros informando-os sobre a situação em que se encontra, bem como ouvir todos os comunicados trocados entre os membros. Assim, cada explorador sabe a localização de toda a equipe e está informado sobre a situação (quão próximo está do encontro do tesouro) de cada membro. O seu próximo deslocamento será feito fundamentado em sua própria experiência e nos relatos ouvidos em seu rádio. Desta forma, havendo real cooperação e troca de informações entre os membros da equipe, o tesouro será achado em um tempo bem menor do que se você fosse escavar sozinho vários buracos na ilha..." .Em suma, o aspecto cooperativo da busca do tesouro poderia ter como epíteto a famosa frase encontrada no final do Capítulo 9 do livro "Os Três Mosqueteiros" de Alexandre Dumas " *todos por um e um por todos, este é nosso lema, não é ?* ".

Este mesmo comportamento cooperativo e de troca de informações é encontrado entre várias espécies animais: em revoadas de pássaros, em cardumes de peixes, em enxames de abelhas, etc. Cientistas de diferentes áreas tentaram modelar estes comportamentos e simulá-los, em particular Reynolds [Reynolds, C.W. (1987) –*"Flocks, herds and schools : a distributed behavioral model"* em Computer Graphics, 21 (4): 25-34] discute a riqueza do movimento de revoadas de pássaros, de manadas de animais terrestres ou de cardumes de peixes procurando estabelecer regras básicas do comportamento individual que justifiquem o comportamento do grupo. O objetivo do trabalho foi o desenvolvimento da *animação*, por computação gráfica, do movimento de cada membro do grupo, estando portanto fora do escopo do presente curso. Entretanto, a conclusão mais pertinente do artigo se refere ao fato do movimento simulado da revoada de pássaros ser o resultado de comportamentos relativamente simples do movimento de cada um dos indivíduos (pássaros). Um outro trabalho pioneiro é o de Heppner & Grenander [Heppner, F & Grenander, U. (1990) – *"A stochastic nonlinear model for coordinated birds flocks"* em *"The Ubiquity of Chaos"* Editado por S. Krasner , AAAS Publications, Washington DC] trabalho conjunto entre um biólogo (primeiro autor!) e um matemático que busca estabelecer as regras que descrevam a perfeita sincronia do movimento de um grande número de pássaros tanto em suas súbitas e bem orquestradas mudanças de direção, como no seu espalhamento e posterior reagrupamento.

O primeiro trabalho sobre o algoritmo de otimização natural denominado *ENXAME DE PARTÍCULAS*(*Particle swarm* em Inglês) é o de Eberhart & Kennedy [Kennedy, J. & Eberhart, R. (1995), - *"Particle Swarm Optimization"* em *Proceedings IEEE International Conference on Neural Networks (Perth, Austrália)* páginas 1942-1948 , artigo disponibilizado na Internet na *Home Page*: http://www.engr.iupui.edu/~shi/Coference/psopap4.html] que, motivados pelo comportamento gregário do movimento de animais (pássaros, peixes, abelhas, gado, etc.) , propuseram um algoritmo de otimização não determinístico bastante eficiente, robusto e de simples implementação computacional.

A tradução matemática do algoritmo de Eberhart & Kennedy é apresentado no próximo item, entendendo-se que o termo partícula se refere a cada um dos indivíduos do grupo (termo equivalente a indivíduo no algoritmo genético) e o termo enxame se refere ao grupo de indivíduos. É importante enfatizar que neste algoritmo, de forma distinta do algoritmo genético, o indivíduo (a partícula) é mantido íntegro durante todo o processo, *sobrevivendo* sem envelhecer até o final do procedimento, a única modificação sofrida pelo indivíduo é sua localização no espaço.

voltar para ENXAME DE PARTÍCULAS

4.2. A Tradução Matemática: o algoritmo básico

No artigo de Eberhart & Kennedy (1995) não se encontra demonstração formal alguma do algoritmo proposto pelos autores, o método proposto já está apresentado em sua forma recursiva adequada para implementação computacional. Uma tentativa de apresentar este algoritmo é apresentada neste item, resultando desta nova formulação uma versão modificada do algoritmo

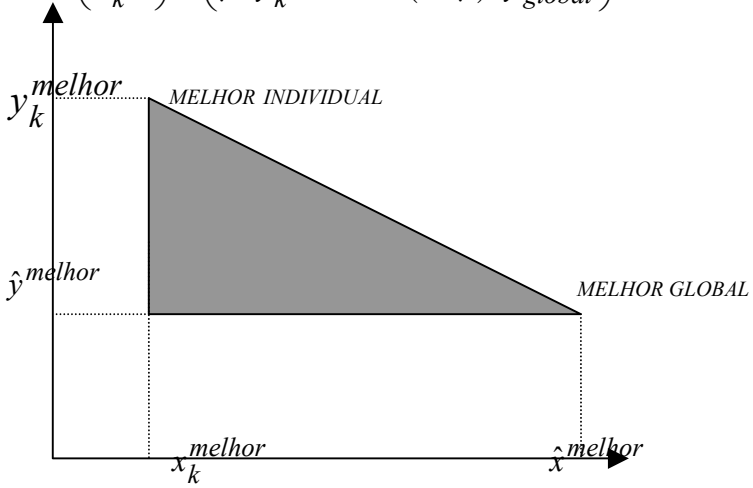
originalmente proposto.

A idéia fundamental do algoritmo é o estabelecimento, em cada passo ou iteração, do movimento de cada uma das partículas do grupo composto por ***n*** partículas (sendo o valor de ***n*** escolhido pelo usuário!). Este movimento é norteado pela *lembrança* da *melhor* posição (melhor valor da função objetivo) no espaço que a partícula já encontrou em seu movimento e no *conhecimento* da *melhor* posição já encontrada por todo o grupo. A utilização da melhor posição individual da partícula pode ser classificada como uma espécie de autoconfiança (ou coloquialmente como o comportamento *sou-mais-eu*!) e a informação da melhor posição do grupo pode ser classificada como um comportamento gregário do indivíduo (ou coloquialmente como o comportamento *maria-vai-com-as-outras*!). Para assegurar a existência de uma certa *personalidade* em cada indivíduo se dota cada indivíduo do grupo de um comportamento distinto e aleatório em que a ponderação destas duas informações é *sorteada* ao longo do processo, assumindo assim um valor diferente para cada uma das partículas e variando este valor ao longo do processo iterativo (procurando traduzir uma certa mudança humor do indivíduo ao longo do tempo). Assim, considerando o movimento da partícula em um plano ou em um espaço bi-dimensional e sua localização neste espaço caracterizada por suas coordenadas:

Partícula ***k*** posição no plano *xy* no passo ou iteração ***i***: $\left(x_k^{(i)}, y_k^{(i)}\right)$ onde k = 1, 2,...,***n*** (número total de partículas no enxame) e i = 1, 2,...,***m*** (número total de passos); melhor posição da partícula ***k*** : $\left(x_k^{melhor}, y_k^{melhor}\right)$ e melhor posição já encontrada pelo enxame (todas as partículas) $\left(\hat{x}_{global}, \hat{y}_{global}\right)$.

A *Força Motriz* que movimentará cada uma das partículas será proporcional à distância entre a posição atual da partícula e o ponto resultante da média ponderada entre a melhor posição individual da partícula e a melhor posição do enxame, isto é :

$$\boldsymbol{X}_k^{(i)} = \begin{pmatrix} X_k^{(i)} \\ Y_k^{(i)} \end{pmatrix} = \begin{pmatrix} \lambda \cdot x_k^{(melhor)} + (1-\lambda) \cdot \hat{x}_{global} \\ \mu \cdot y_k^{(melhor)} + (1-\mu) \cdot \hat{y}_{global} \end{pmatrix}$$ onde λ e μ são dois números aleatórios ∈ [0,1].



O triângulo sombreado na figura acima representa a região no plano *xy* de todos os

pontos $\boldsymbol{X}_k^{(i)}$.

Considerando o movimento de cada partícula análogo ao do movimento do sistema mecânico carrinho+mola+amortecedor resulta no balanço de forças (em forma adimensional!):

Na direção *x*: $\dfrac{d^2 x_k(t)}{dt^2} = -2 \cdot \xi_k \cdot \dfrac{dx_k(t)}{dt} - K \cdot \left[x_k(t) - X_k^{(i)} \right]$

e na direção *y*: $\dfrac{d^2 y_k(t)}{dt^2} = -2 \cdot \varsigma_k \cdot \dfrac{dy_k(t)}{dt} - K \cdot \left[y_k(t) - Y_k^{(i)} \right]$

estas duas equações diferenciais são resolvidas no intervalo de tempo (tempo artificial adimensional, que na realidade *mede*, em sua forma discreta, as iterações do processo) : $t_i = i \cdot \Delta t < t \le (i+1) \cdot \Delta t = t_{i+1}$. No *início* do intervalo se tem: $x_k(t_i) = x_k^{(i)}$; $y_k(t_i) = y_k^{(i)}$; $\left.\dfrac{dx_k(t)}{dt}\right|_{t_i} = \nu_k^{(i)}$ e $\left.\dfrac{dy_k(t)}{dt}\right|_{t_i} = \upsilon_k^{(i)}$ (que são os dois componentes do vetor velocidade da partícula no início do intervalo). Os parâmetros ξ_k e ζ_k são dois números aleatórios (adimensionais!) $\in [0,1]$ e a constante *K* é um valor qualquer (escolhido pelo usuário) ≥ 1.

Reescrevendo as duas equações diferenciais em termos dos dois componentes do vetor velocidade resulta no sistema de equações diferenciais de primeira ordem:

$$\begin{cases} \dfrac{d\nu_k(t)}{dt} = -2 \cdot \xi_k \cdot \nu_k(t) - K \cdot \left[x_k(t) - X_k^{(i)} \right] \\ \dfrac{dx_k(t)}{dt} = \nu_k(t) \\ \dfrac{d\upsilon_k(t)}{dt} = -2 \cdot \varsigma_k \cdot \upsilon_k(t) - K \cdot \left[y_k(t) - Y_k^{(i)} \right] \\ \dfrac{dy_k(t)}{dt} = \upsilon_k(t) \end{cases}$$

Resolvendo numericamente estas equações pelo método de Euler explícito, resulta no final do intervalo:

$$\begin{cases} \nu_k^{(i+1)} = \nu_k^{(i)} - 2 \cdot \xi_k \cdot \Delta t \cdot \nu_k^{(i)} - K \cdot \Delta t \cdot \left[x_k^{(i)} - X_k^{(i)} \right] \\ x_k^{(i+1)} = x_k^{(i)} + \Delta t \cdot \nu_k^{(i)} \\ \upsilon_k^{(i+1)} = \upsilon_k^{(i)} - 2 \cdot \varsigma_k \cdot \Delta t \cdot \upsilon_k^{(i)} - K \cdot \Delta t \cdot \left[y_k^{(i)} - Y_k^{(i)} \right] \\ y_k^{(i+1)} = y_k^{(i)} + \Delta t \cdot \upsilon_k^{(i)} \end{cases}$$

Definindo as variáveis : $V_x = \Delta t \cdot \nu$ e $V_y = \Delta t \cdot \upsilon$ resulta:

$$\begin{cases} V_{x,k}^{(i+1)} = V_{x,k}^{(i)} - 2\cdot\xi_k\cdot\Delta t\cdot V_{x,k}^{(i)} - K\cdot\Delta t^2\cdot\left[x_k^{(i)} - X_k^{(i)}\right] \\ x_k^{(i+1)} = x_k^{(i)} + V_{x,k}^{(i)} \\ V_{y,k}^{(i)} = V_{y,k}^{(i)} - 2\cdot\varsigma_k\cdot\Delta t\cdot V_{y,k}^{(i)} - K\cdot\Delta t^2\cdot\left[y_k^{(i)} - Y_k^{(i)}\right] \\ y_k^{(i+1)} = y_k^{(i)} + V_{y,k}^{(i)} \end{cases}$$

Agrupando os parâmetros das expressões acima chega-se à forma iterativa do método conforme sugerida por Eberhart & Kennedy (1995)[1]:

$$\begin{cases} V_{x,k}^{(i+1)} = \omega^{(i)}\cdot V_{x,k}^{(i)} + c_1\cdot\lambda\cdot\left[x_k^{melhor} - x_k^{(i)}\right] + c_2\cdot\mu\cdot\left[\hat{x}_{global} - x_k^{(i)}\right] \\ x_k^{(i+1)} = x_k^{(i)} + V_{x,k}^{(i+1)} \\ V_{y,k}^{(i+1)} = \omega^{(i)}\cdot V_{y,k}^{(i)} + c_1\cdot\eta\cdot\left[y_k^{melhor} - y_k^{(i)}\right] + c_2\cdot\varepsilon\cdot\left[\hat{y}_{global} - y_k^{(i)}\right] \\ y_k^{(i+1)} = y_k^{(i)} + V_{y,k}^{(i+1)} \end{cases}$$

para i = 0 , 1, ..., ***m***-1; sugere-se adotar a *ponderação* $\omega^{(i)}$ decrescente com i de forma linear segundo a expressão: $\omega^{(i)} = \omega_{inicial} + \left(\omega_{final} - \omega_{inicial}\right)\cdot\left(\frac{i}{\boldsymbol{m}-1}\right)$ com $\omega_{final} < \omega_{inicial}$; as constantes c_1 e c_2 são valores reais positivos escolhidos pelo usuário e os parâmetros λ, μ, η e ε são valores randômicos ou aleatórios *sorteados* em cada passo do processo iterativo. As posições e velocidades iniciais das partículas podem ser definidas pelo usuário ou então geradas aleatoriamente pelo computador.

Uma forma alternativa do procedimento recursivo pode também ser obtida (sendo referenciada a partir deste instante como algoritmo modificado de enxame de partículas, abreviadamente PSO-modificado, em contraposição ao de Eberhart & Kennedy denominado de PSO-clássico) através da solução analítica, em cada intervalo de tempo, do sistema original de EDO's:

$$\begin{cases} \frac{d^2x_k(t)}{dt^2} + 2\cdot\xi_k\cdot\frac{dx_k(t)}{dt} + K\cdot\left[x_k(t) - X_k^{(i)}\right] = 0 \\ \frac{d^2y_k(t)}{dt^2} + 2\cdot\varsigma_k\cdot\frac{dy_k(t)}{dt} + K\cdot\left[y_k(t) - Y_k^{(i)}\right] = 0 \end{cases}$$

congelando os valores de $X_k^{(i)}$ e $Y_k^{(i)}$ em seus valores no início do intervalo, *sorteando* os valores de ξ_k e ζ_k apenas uma vez em t_i e resolvendo o sistema com as condições iniciais:

---

[1] Note que na discretização das equações diferenciais o procedimento de integração da posição é o Euler implícito enquanto que o das velocidades é o Euler explícito!

$x_k(t_i) = x_k^{(i)}$; $y_k(t_i) = y_k^{(i)}$; $\left.\frac{dx_k(t)}{dt}\right|_{t_i} = \nu_k^{(i)}$ e $\left.\frac{dy_k(t)}{dt}\right|_{t_i} = \upsilon_k^{(i)}$, resulta:

$$\begin{cases} x_k^{(i+1)} = X_k^{(i)} + \exp(-\xi_k \cdot \Delta t) \cdot \left\{ \left(x_k^{(i)} - X_k^{(i)}\right) \cdot \cos(\omega \cdot \Delta t) + \left[\xi_k \cdot \left(x_k^{(i)} - X_k^{(i)}\right) + \nu_k^{(i)}\right] \cdot \frac{\operatorname{sen}(\omega \cdot \Delta t)}{\omega} \right\} \\ \nu_k^{(i+1)} = \exp(-\xi_k \cdot \Delta t) \cdot \left\{ \nu_k^{(i)} \cdot \cos(\omega \cdot \Delta t) - \left[\left(\omega^2 + \xi_k^2\right) \cdot \left(x_k^{(i)} - X_k^{(i)}\right) + \xi_k \nu_k^{(i)}\right] \cdot \frac{\operatorname{sen}(\omega \cdot \Delta t)}{\omega} \right\} \\ y_k^{(i+1)} = Y_k^{(i)} + \exp(-\varsigma_k \cdot \Delta t) \cdot \left\{ \left(y_k^{(i)} - Y_k^{(i)}\right) \cdot \cos(\sigma \cdot \Delta t) + \left[\varsigma_k \cdot \left(y_k^{(i)} - Y_k^{(i)}\right) + \upsilon_k^{(i)}\right] \cdot \frac{\operatorname{sen}(\sigma \cdot \Delta t)}{\sigma} \right\} \\ \upsilon_k^{(i+1)} = \exp(-\varsigma_k \cdot \Delta t) \cdot \left\{ \upsilon_k^{(i)} \cdot \cos(\sigma \cdot \Delta t) - \left[\left(\sigma^2 + \varsigma_k^2\right) \cdot \left(y_k^{(i)} - Y_k^{(i)}\right) + \varsigma_k \upsilon_k^{(i)}\right] \cdot \frac{\operatorname{sen}(\sigma \cdot \Delta t)}{\sigma} \right\} \end{cases}$$

onde: $\omega = \sqrt{K - \xi_k^2}$ e $\sigma = \sqrt{K - \varsigma_k^2}$.

Esta forma modificada do algoritmo só necessita (além da especificação do número de partículas , do número de iterações e dos valores iniciais das posições e velocidades das partículas) da especificação dos parâmetros $\Delta t > 0$ e K [$K \geq 1$ para assegurar o movimento oscilatório amortecido das partículas!]. Este é a grande vantagem deste método modificado (no algoritmo clássico se deve especificar os valores das constantes c_1, c_2 e de $\omega_{inicial}$ e ω_{final}) e a maior complexidade do algoritmo é apenas aparente já que o procedimento recursivo é análogo ao do clássico diferindo apenas no cálculo em cada passo dos valores de funções trigonométricas simples (seno e cosseno) e da função exponencial o que não aumenta em nada o custo computacional do código.

voltar para ENXAME DE PARTÍCULAS

4.3. A Programação do Algoritmo

As duas formas do algoritmo são, sob o ponto de vista de implementação, semelhantes diferindo apenas nas expressões matemáticas de cálculo da posição e da velocidade de cada partícula ao cabo de cada iteração e nas especificações das constantes pertinentes.

O algoritmo PSO-clássico para a busca do máximo de uma função *f(x,y)* pode assim ser sumarizado pelo pseudo-código abaixo:

Etapa Inicial *(Iteração zero)*: Entre com **n** (o número de partículas do enxame), **m** (o número de iterações), especifique os valores de ω_{final} , $\omega_{inicial}$, c_1, c_2 e os valores mínimos e máximos de *x* e de *y*. Gere as condições iniciais segundo o procedimento[2]:

---

[2] Nos procedimentos a função $.rnd(\alpha)$ gera números aleatórios com distribuição uniforme entre 0(zero) e α.

para k = 1, ..., **n** faça $\lambda \leftarrow rnd(1)$; $x_k \leftarrow x_{\min} + \lambda \cdot (x_{MAX} - x_{\min})$ e $x_k^{melhor} \leftarrow x_k$

$\mu \leftarrow rnd(1)$; $y_k \leftarrow y_{\min} + \mu \cdot (y_{MAX} - y_{\min})$ e $y_k^{melhor} \leftarrow y_k$

$V_{x,k} \leftarrow 0$ e $V_{y,k} \leftarrow 0$ (optou-se neste caso para partir com as partículas em repouso!).

Faça $\hat{x}^{melhor} \leftarrow x_1$; $\hat{y}^{melhor} \leftarrow y_1$; $f_1^{(melhor)} \leftarrow f(x_1, y_1)$ e $\hat{f}_{global} \leftarrow f_1^{(melhor)}$;

Etapa a) para i = 2,...., **n** faça $f_i^{(melhor)} \leftarrow f(x_i, y_i)$

se $f_i^{(melhor)} > \hat{f}_{global}$ então faça $\hat{x}^{melhor} \leftarrow x_i$; $\hat{y}^{melhor} \leftarrow y_i$ e $\hat{f}_{global} \leftarrow f_i^{(melhor)}$ volte para o início da Etapa a com o próximo i;

se $f_i^{(melhor)} \leq \hat{f}_{global}$ volte para o início da Etapa a com o próximo i;

Etapa 1 *(Iteração i)* Para i = 1, ..., **m** faça $\omega \leftarrow \omega_{inicial} + (\omega_{final} - \omega_{inicial}) \cdot \left(\frac{i-1}{\boldsymbol{m}-1} \right)$

Etapa 1-a) Para k = 1,...,**n** faça: $\lambda \leftarrow rnd(1)$; $\mu \leftarrow rnd(1)$; $\eta \leftarrow rnd(1)$ e $\varepsilon \leftarrow rnd(1)$

$$V_{x,k} \leftarrow \omega \cdot V_{x,k} + c_1 \cdot \lambda \cdot \left[x_k^{melhor} - x_k \right] + c_2 \cdot \mu \cdot \left[\hat{x}_{global} - x_k \right]$$

$$V_{y,k} \leftarrow \omega \cdot V_{y,k} + c_1 \cdot \eta \cdot \left[y_k^{melhor} - y_k \right] + c_2 \cdot \varepsilon \cdot \left[\hat{y}_{global} - y_k \right]$$

$x_k \leftarrow x_k + V_{x,k}$ e $y_k \leftarrow y_k + V_{y,k}$

Se $x_k > x_{MAX}$ faça $x_k \leftarrow x_{MAX}$ e $V_{x,k} \leftarrow 0$

Se $x_k < x_{\min}$ faça $x_k \leftarrow x_{\min}$ e $V_{x,k} \leftarrow 0$

Se $y_k > y_{MAX}$ faça $y_k \leftarrow y_{MAX}$ e $V_{y,k} \leftarrow 0$

Se $y_k < y_{\min}$ faça $y_k \leftarrow y_{\min}$ e $V_{y,k} \leftarrow 0$

Calcule $f_{atual} \leftarrow f(x_k, y_k)$

Se $f_{atual} > f_k^{(melhor)}$ faça $f_k^{(melhor)} \leftarrow f_{atual}$, $x_k^{(melhor)} \leftarrow x_k$ e $y_k^{(melhor)} \leftarrow y_k$; verifique a seguir se $f_{atual} > \hat{f}_{global}$ faça $\hat{f}_{global} \leftarrow f_{atual}$, $\hat{x}_{global} \leftarrow x_k$ e $\hat{y}_{global} \leftarrow y_k$; caso $f_{atual} \leq \hat{f}_{global}$ volte ao início Etapa 1-a) com o próximo k se

k < **n**, caso contrário [k≥ **n**] vá para o início da Etapa 1 com o próximo i se i < **m**, caso contrário [i ≥ **m**] vá para a Etapa 2

Se $f_{atual} \leq f_k^{(melhor)}$ volte ao início Etapa 1-a) com o próximo k se k < n, caso contrário [k≥ **n**] vá para o início da Etapa 1 com o próximo i se i < **m**, caso contrário [i ≥ **m**] vá para a Etapa 2

Etapa 2 O processo iterativo terminou os *melhores* valores de *x* e *y* são : $\hat{x}_{global}$ **e** $\hat{y}_{global}$ **e neste ponto o valor da função é :** $\hat{f}_{global}$

FIM

O algoritmo PSO-modificado para a busca do máximo de uma função *f(x,y)* pode assim ser sumarizado pelo pseudo-código abaixo:

Etapa Inicial *(Iteração zero)*: Entre com **n** (o número de partículas do enxame), **m** (o número de iterações), especifique os valores de $\Delta t > 0$ e K [$K \geq 1$] e os valores mínimos e máximos de *x* e de *y*. Gere as condições iniciais segundo o procedimento:

para k = 1, ..., **n** faça $\lambda \leftarrow rnd(1)$; $x_k \leftarrow x_{\min} + \lambda \cdot (x_{MAX} - x_{\min})$ e $x_k^{melhor} \leftarrow x_k$

$\mu \leftarrow rnd(1)$; $y_k \leftarrow y_{\min} + \mu \cdot (y_{MAX} - y_{\min})$ e $y_k^{melhor} \leftarrow y_k$

$\nu_k \leftarrow 0$ e $\upsilon_k \leftarrow 0$ (optou-se neste caso para partir com as partículas em repouso!).

Faça $\hat{x}^{melhor} \leftarrow x_1$; $\hat{y}^{melhor} \leftarrow y_1$; $f_1^{(melhor)} \leftarrow f(x_1, y_1)$ e $\hat{f}_{global} \leftarrow f_1^{(melhor)}$;

Etapa a) para i = 2,...., **n** faça $f_i^{(melhor)} \leftarrow f(x_i, y_i)$

se $f_i^{(melhor)} > \hat{f}_{global}$ então faça $\hat{x}^{melhor} \leftarrow x_i$; $\hat{y}^{melhor} \leftarrow y_i$ e

$\hat{f}_{global} \leftarrow f_i^{(melhor)}$ volte para o início da Etapa a com o próximo i;

se $f_i^{(melhor)} \leq \hat{f}_{global}$ volte para o início da Etapa a com o próximo i;

Etapa 3 *(Iteração i)* Para i = 1, ..., **m** execute Etapa 1-a)

Etapa 1-a) Para k = 1,...,**n** faça: $\lambda \leftarrow rnd(1)$; $\eta \leftarrow rnd(1)$; $\xi \leftarrow rnd(1)$ e $\varsigma \leftarrow rnd(1)$

Calcule : $\omega \leftarrow \sqrt{K - \xi^2}$; $X \leftarrow \lambda \cdot x_k^{(melhor)} + (1-\lambda) \cdot \hat{x}_{global}$; $A \leftarrow x_k - X$;

$$B \leftarrow \frac{\xi \cdot A + \nu_k}{\omega};\ f_1 \leftarrow \exp(-\xi \cdot \Delta t) \cdot \cos(\omega \cdot \Delta t) \text{ e } f_2 \leftarrow \exp(-\xi \cdot \Delta t) \cdot \operatorname{sen}(\omega \cdot \Delta t)$$

$$x_k \leftarrow X + A \cdot f_1 + B \cdot f_2$$

$$\nu_k \leftarrow \nu_k \cdot f_1 - (\omega \cdot A + \xi \cdot B) \cdot f_2$$

Se $x_k > x_{MAX}$ faça $x_k \leftarrow x_{MAX}$ e $\nu_k \leftarrow 0$

Se $x_k < x_{\min}$ faça $x_k \leftarrow x_{\min}$ e $\nu_k \leftarrow 0$

a seguir calcule : $\sigma \leftarrow \sqrt{K - \varsigma^2}$; $Y \leftarrow \eta \cdot y_k^{(melhor)} + (1-\eta) \cdot \hat{y}_{global}$ $C \leftarrow y_k - Y$;

$$D \leftarrow \frac{\varsigma \cdot C + \upsilon_k}{\sigma};\ g_1 \leftarrow \exp(-\varsigma \cdot \Delta t) \cdot \cos(\sigma \cdot \Delta t) \text{ e } g_2 \leftarrow \exp(-\varsigma \cdot \Delta t) \cdot \operatorname{sen}(\sigma \cdot \Delta t)$$

$$y_k \leftarrow Y + C \cdot g_1 + D \cdot g_2$$

$$\upsilon_k \leftarrow \upsilon_k \cdot g_1 - (\sigma \cdot C + \varsigma \cdot D) \cdot g_2$$

Se $y_k > y_{MAX}$ faça $y_k \leftarrow y_{MAX}$ e $\upsilon_k \leftarrow 0$

Se $y_k < y_{\min}$ faça $y_k \leftarrow y_{\min}$ e $\upsilon_k \leftarrow 0$

Calcule $f_{atual} \leftarrow f(x_k, y_k)$

Se $f_{atual} > f_k^{(melhor)}$ faça $f_k^{(melhor)} \leftarrow f_{atual}$, $x_k^{(melhor)} \leftarrow x_k$ e $y_k^{(melhor)} \leftarrow y_k$; verifique a seguir se $f_{atual} > \hat{f}_{global}$ faça $\hat{f}_{global} \leftarrow f_{atual}$, $\hat{x}_{global} \leftarrow x_k$ e $\hat{y}_{global} \leftarrow y_k$; caso $f_{atual} \leq \hat{f}_{global}$ volte ao início Etapa 1-a) com o próximo k se

k < **n**, caso contrário [k≥ **n**] vá para o início da Etapa 1 com o próximo i se i < **m**, caso contrário [i ≥ **m**] vá para a Etapa 2

Se $f_{atual} \leq f_k^{(melhor)}$ volte ao início Etapa 1-a) com o próximo k se k < n, caso contrário [k≥ **n**] vá para o início da Etapa 1 com o próximo i se i < **m**, caso contrário [i ≥ **m**] vá para a Etapa 2

Etapa 4 O processo iterativo terminou os *melhores* valores de *x* e *y* são : $\hat{x}_{global}$ **e** $\hat{y}_{global}$ **e neste ponto o valor da função é :** $\hat{f}_{global}$.

FIM

A seleção dos valores dos parâmetros/constantes dos dois algoritmos é uma tarefa de ***tentativa-e-erro***, a recomendação que se dá nestes casos e para estes tipos de algoritmos é a execução de um grande número de experimentos numéricos. A prática adquirida através da implementação computacional dos algoritmos e a resolução de exercícios simples é fundamental para se adquirir alguma sensibilidade sobre os valores destes parâmetros/constantes. Mesmo assim se deve alertar que não há *valores mágicos* destes parâmetros/constantes e sempre desconfie de artigos científicos que sugiram de forma categórica valores particulares destes parâmetros como os *melhores*.

voltar para ENXAME DE PARTÍCULAS

4.4. Exemplos Ilustrativos

Para entender o significado dos parâmetros do método, considere a mesma função considerada nos algoritmos genéticos, que é a minimização de F(x) apresentada abaixo.

$$F(x) = \frac{1}{10000}(x+10)(x+6)(x+5)(x+1)(x-7)(x-10)$$

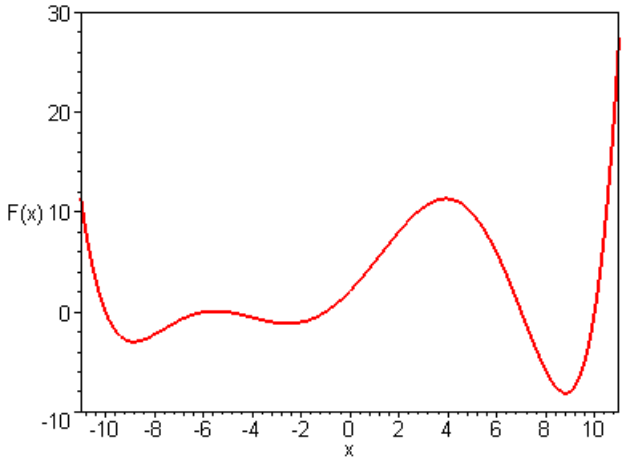


Figura 2 – Espaço de busca da função teste

- Observe que existem mínimos locais em x ≈-8,834, x ≈-2,546 e x ≈ 8,817, sendo este último o mínimo global. A função apresenta máximos locais em x ≈ -5,518 e x ≈ 3,914.

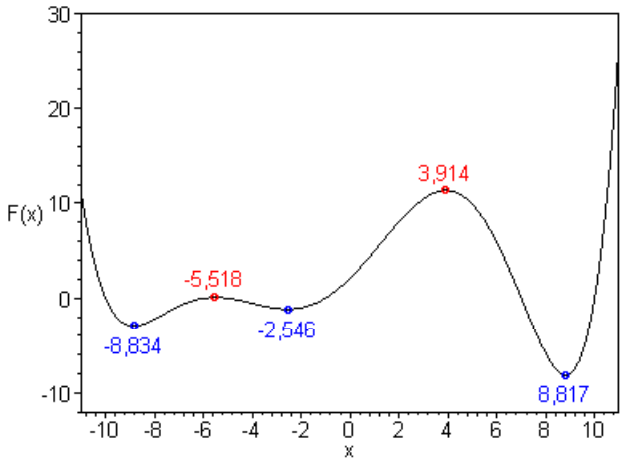


Figura 3 – Extremos da função teste

As duas versões do PSO foram implementadas computacionalmente. Abaixo é mostrada a evolução dos valores ótimos da função e da variável *x* ao longo do procedimento evolutivo com a forma clássica do algoritmo tendo sido utilizado os seguintes conjunto de parâmetros: $\omega_{inicial}$=0,9 ; ω_{final}=0 ; c_1=c_2 =1; x_{MAX} = +12 e x_{min} = -12.Adotando-se uma população de 10(dez) partículas, um máximo de 20 (vinte) iterações e partindo de localizações iniciais aleatórias (entre –12 e +12) e do repouso.

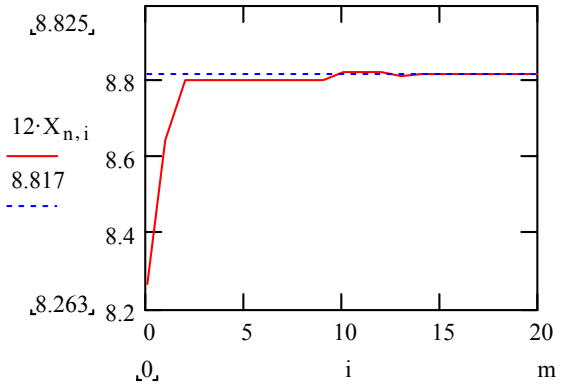


Figura 4 – Evolução dos Melhores Valores Globais da Função Teste

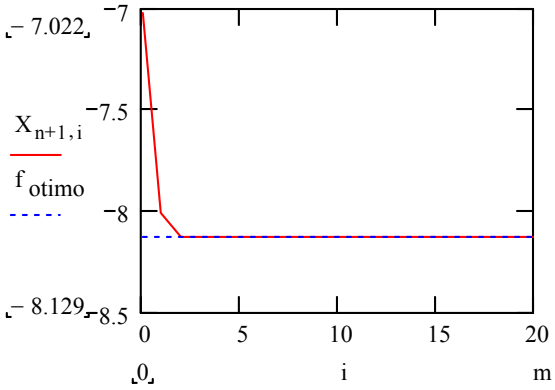


Figura 5 – Evolução dos Melhores Valores Globais da Variável

Uma animação desta solução pode ser reproduzida clicando-se no programa Univariável_PSO_clássico.

Abaixo é mostrada a evolução dos valores ótimos da função e da variável *x* ao longo do procedimento evolutivo com a forma modificada do algoritmo tendo sido utilizado os seguintes conjunto de parâmetros:Δt=1; $K = 2$; $x_{MAX} = +12$ e $x_{min} = -12$.Adotando-se uma população de 10(dez) partículas, um máximo de 20 (vinte) iterações e partindo de localizações iniciais aleatórias (entre –12 e +12) e do repouso.

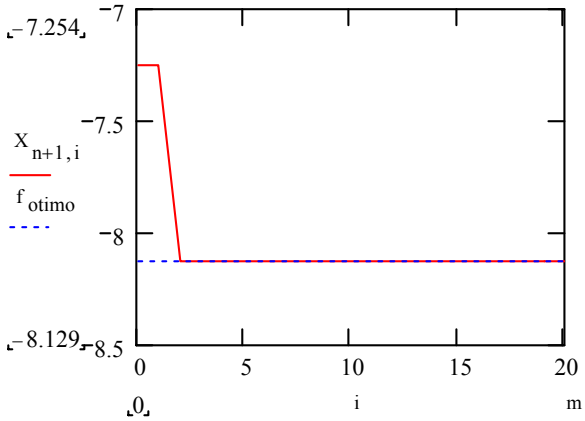


Figura 6 – Evolução dos Melhores Valores Globais da Função Teste

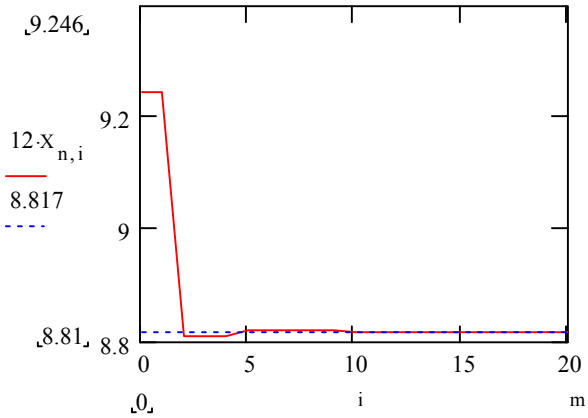


Figura 7 – Evolução dos Melhores Valores Globais da Variável

Uma animação desta solução pode ser reproduzida clicando-se no programa Univariável_PSO_modificado.

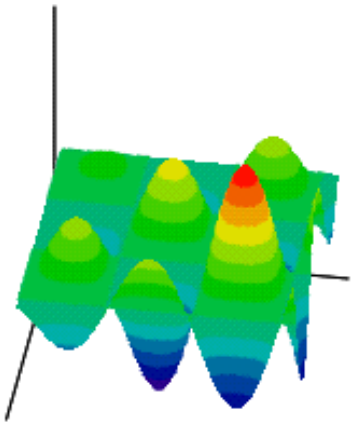
Notem que o desempenho das duas formas do algoritmo é idêntico, não sendo possível afirmar qual das duas formas é mais eficiente.

O funcionamento do algoritmo de enxame de partículas (PSO para os íntimos) é agora ilustrado com a maximização de uma função de duas variáveis, a chamada Função Alpina que é representada pela equação :

$$f(x,y) = \sqrt{x \cdot y} \cdot \text{sen}(x) \cdot \text{sen}(y)$$

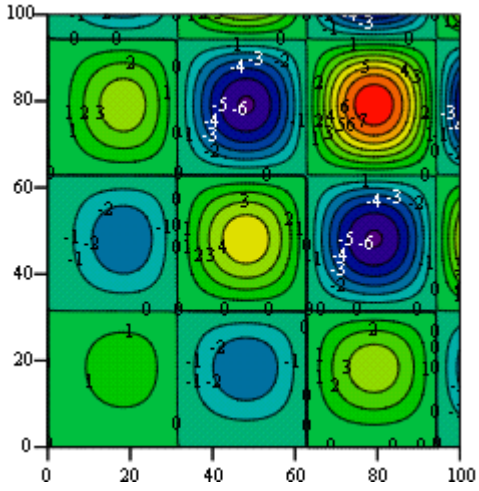
no domínio : $0 < x < 10$ e $0 < y < 10$.

Esta função é mostrada nas duas figuras a seguir:



a

Figura 8 – A Função Alpina



a

Figura 9 – A Função Alpina em Curvas de Nível

O máximo global desta função, na região considerada, é em $x = y \approx 7{,}917$ onde a função assume o valor $f_{MAXIMO} \approx 7,8856$.

Abaixo é mostrada a evolução dos valores ótimos da função ao longo do procedimento evolutivo com a forma clássica do algoritmo tendo sido utilizado os seguintes conjunto de parâmetros: $\omega_{inicial}=0{,}9$; $\omega_{final}=0$; $c_1=c_2=1$; $x_{MAX}= y_{MAX} = +10$ e $x_{min} = y_{min} = 0$ .Adotando-se uma população de 20(vinte) partículas, um máximo de 20 (vinte) iterações e partindo de localizações iniciais aleatórias [entre (0,0) e (10,10)] e do repouso.

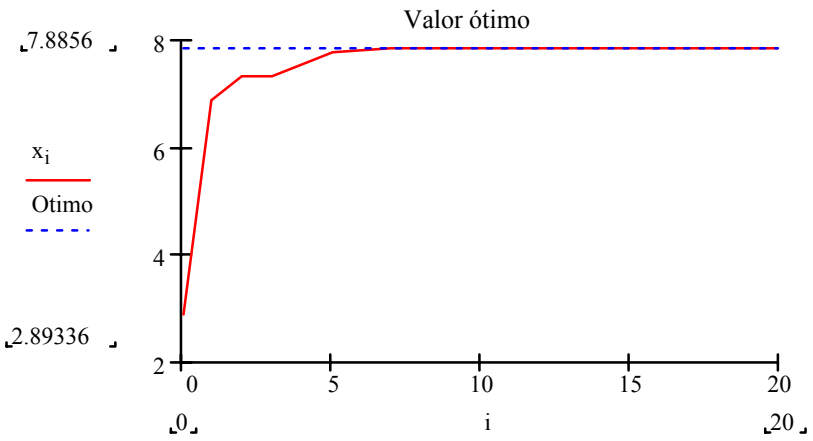


Figura 10 – Evolução dos Melhores Valores Globais da Função Alpina

Uma animação desta solução pode ser reproduzida clicando-se no programa Alpina_PSO_Clássico.

Abaixo é mostrada a evolução dos valores ótimos da função ao longo do procedimento evolutivo com a forma modificada do algoritmo tendo sido utilizado os seguintes conjunto de parâmetros: :Δt=1; K = 2; x_{MAX}= y_{MAX} = +10 e x_{min} = y_{min} = 0 .Adotando-se uma população de 20(vinte) partículas, um máximo de 20 (vinte) iterações e partindo de localizações iniciais aleatórias [entre (0,0) e (10,10)] e do repouso.

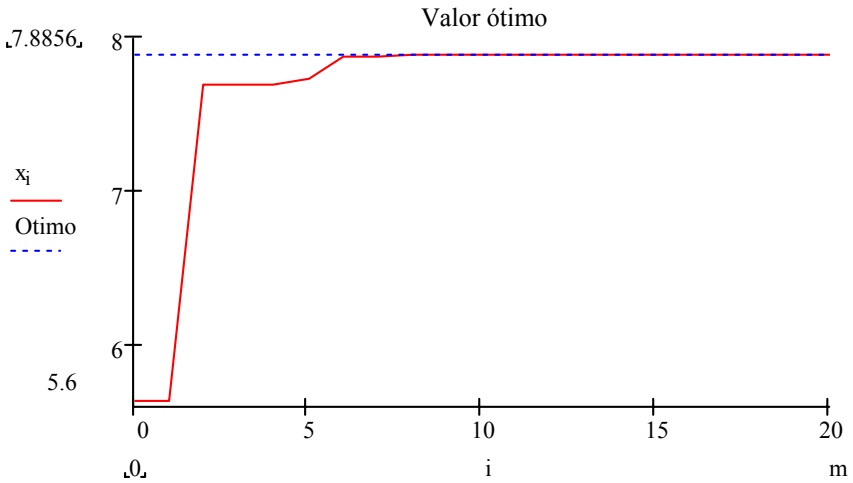


Figura 11 – Evolução dos Melhores Valores Globais da Função Alpina

Uma animação desta solução pode ser reproduzida clicando-se no programa Alpina_PSO_Modificado.

É importante novamente enfatizar que os *melhores* valores dos parâmetros/constantes das duas formas do algoritmo são muito dependentes do problema particular que se está resolvendo e que uma análise da sensibilidade da resolução do problema a valores particulares dos

parâmetros/constantes só pode ser feita após a execução de um grande número de simulações. Desta maneira, recomendamos fortemente que sejam feitos **todos** os exercícios da lista sobre o assunto e que dúvidas sejam tiradas por E-mail ou durante as sessões do *chat-room.*

voltar para ENXAME DE PARTÍCULAS