

# Um Novo Enfoque do Método do Enxame de Partículas

Evaristo Chalbaud Biscaia Jr., Marcio Schwaab, José Carlos Pinto

Programa de Engenharia Química / COPPE / Universidade Federal do Rio de Janeiro  
Cidade Universitária, CP: 68502 – CEP: 21945-970 – Rio de Janeiro – RJ – Brasil  
Telefone: (0-xx-21)2562-8364 – Fax: (0-xx-21)2562-8300 – E-mail: evaristo@peq.coppe.ufrj.br

A utilização de algoritmos heurísticos na otimização de processos químicos, em especial o método do enxame de partículas, tem recebido muita atenção pela sua simplicidade, robustez e eficiência, mesmo em problemas de grande dimensão. Neste trabalho o método do enxame de partículas é visto sob um novo enfoque, considerando a sua gênese na resposta dinâmica de sistemas lineares de segunda ordem, dando origem a um novo algoritmo de otimização. O algoritmo foi avaliado na minimização de funções teste e os resultados indicam que o novo método é capaz de encontrar o mínimo global com 100 % de eficiência na maioria das avaliações realizadas.

## Introdução

O método do enxame de partículas (Particle Swarm Optimization - PSO) é um algoritmo de otimização heurística baseado no comportamento gregário do movimento de animais (peixes, pássaros, etc). Proposto por Eberhart e Kennedy (1995), este método consiste na otimização de uma função objetivo através da troca de informações entre elementos (partículas) do grupo, resultando em um algoritmo de otimização não determinístico eficiente, robusto e de simples implementação computacional.

O movimento de cada partícula em cada iteração corresponde à soma de três termos distintos: o primeiro é um termo relativo à inércia da partícula e que traduz o modo com que a partícula vem se movendo; o segundo é um termo relativo à atração da partícula ao *melhor* ponto que já encontrou; e o terceiro termo é relativo à atração da partícula ao *melhor* ponto que todo grupo (ou uma parte do grupo) já encontrou.

Uma característica muito interessante do PSO é a realização de uma busca global nas iterações iniciais do procedimento de otimização, ocorrendo neste estágio a prospeção de várias regiões do espaço de busca, viabilizando a

localização de ótimo(s) global(is). Com o decorrer das iterações a busca passa a ser uma busca local em torno do valor mais promissor, chegando a valores próximos do ponto ótimo. Entretanto, o desempenho do algoritmo é bastante dependente da seleção dos valores de seus parâmetros. Trelea (2003) apresentam uma análise teórica do algoritmo, onde definem alguns critérios para uma boa escolha dos parâmetros de busca.

Apesar de ser recente, o PSO tem encontrado diversas aplicações, como nos trabalhos de Cockshott e Hartman (2001) e Costa Jr. et al. (2003) onde o PSO é usado para otimização de processos específicos, Ourique et al. (2002) que utilizaram o PSO na análise dinâmica não linear de processos químicos e Parsopulos e Vrahatis (2002) que aplicaram o PSO em problemas multi-objetivo.

Uma nova forma do algoritmo PSO, baseado na dinâmica de sistemas dinâmicos lineares de segunda ordem sub-amortecidos, é apresentada neste trabalho. As características do novo algoritmo são semelhantes às do PSO original, havendo a mesma capacidade de exploração global da região de busca e o posterior refinamento do ponto ótimo. Além disto, a concepção original do PSO se mantém, pois cada partícula é atraída para um ponto contido no interior da região delimitada pelo melhor

ponto encontrado pela partícula e pelo melhor ponto encontrado pelo grupo. A maior vantagem deste novo algoritmo é relativa a seus parâmetros, pois os mesmos podem ser relacionados ao amortecimento e às frequências de oscilação das respostas dinâmicas da cada partícula.

## Formulação do Algoritmo

O novo algoritmo é baseado na resposta dinâmica de sistemas lineares de segunda ordem, segundo a equação diferencial a seguir, que representa, por exemplo, o movimento de uma partícula ligada a uma mola:

$$\frac{d^2 x_k(t)}{dt^2} = -2 \cdot \xi \cdot \frac{dx_k(t)}{dt} - K \cdot [x_k(t) - X_k^{(i)}] \quad (1)$$

onde  $K$  é um ganho estático do sistema,  $\xi$  é a razão de amortecimento,  $x_k(t)$  é a posição da partícula  $k$  e  $X_k^{(i)}$  o ponto de equilíbrio do sistema. A analogia com a concepção original do PSO é traduzida no cálculo de  $X_k^{(i)}$  que é calculado como uma média ponderada entre o melhor valor atingido nas iterações anteriores pela partícula e o melhor valor global até então, de acordo com:

$$X_k^{(i)} = \lambda \cdot x_k^{(ind)} + (1 - \lambda) \cdot x^{(glo)} \quad (2)$$

onde  $\lambda$  é um número aleatório com distribuição uniforme  $\in [0, 1]$ ,  $i$  denota a iteração e  $k$  a partícula..

A equação 1 é resolvida no intervalo de tempo  $t_i$  e  $t_{i+1}$ , que mede, em sua forma discreta, as iterações do processo, já que é somente neste intervalo de tempo que as equações são válidas, pois o valor de  $X_k^{(i)}$  muda, conforme foi explicado acima. As condições iniciais para a solução da Equação 1 são:

$$x_k(t_i) = x_k^{(i)} \quad (3)$$

$$\left. \frac{dx_k(t)}{dt} \right|_{t_i} = v_k^{(i)} \quad (4)$$

ou seja, a posição e a velocidade da partícula no tempo  $t_i$  são conhecidos.

Para a solução da Equação 1 é feita a seguinte mudança de variável:

$$\bar{x}_k(t) = x_k(t) - X_k^{(i)} \quad (5)$$

e assim a Equação 1 e as condições de contorno assumem a seguinte forma:

$$\frac{d^2 \bar{x}_k(t)}{dt^2} + 2 \cdot \xi_k \cdot \frac{d\bar{x}_k(t)}{dt} + K \cdot \bar{x}_k(t) = 0 \quad (6a)$$

$$\bar{x}_k(t_i) = \bar{x}_k^{(i)} \quad (6b)$$

$$\left. \frac{d\bar{x}_k(t)}{dt} \right|_{t_i} = v_k^{(i)} \quad (6c)$$

A Equação 6a é uma equação diferencial homogênea com coeficientes constantes e apresenta a seguinte equação característica:

$$m^2 + 2 \cdot \xi \cdot m + K = 0 \quad (7)$$

cujas raízes são:

$$m = -\xi \pm \sqrt{\xi^2 - K} \quad (8)$$

Assim são possíveis três formas de solução da Equação 6a:

(i) se  $\xi^2 - K > 0$ , ou seja,  $K < \xi^2$ , têm-se duas raízes reais e distintas:

$$m_1 = -\xi - \sqrt{\xi^2 - K} \text{ e } m_2 = -\xi + \sqrt{\xi^2 - K}$$

e a solução da Equação 6a é:

$$\bar{x}_k(t) = c_1 \cdot \exp\left[\left(-\xi - \sqrt{\xi^2 - K}\right)t\right] + c_2 \cdot \exp\left[\left(-\xi + \sqrt{\xi^2 - K}\right)t\right] \quad (9)$$

ii) se  $\xi^2 - K = 0$ , ou seja,  $K = \xi^2$ , têm-se duas raízes reais e iguais:

$$m_1 = -\xi \text{ e } m_2 = -\xi$$

e a solução da Equação 6a é:

$$\bar{x}_k(t) = c_1 \cdot \exp(-\xi \cdot t) + c_2 \cdot t \cdot \exp(-\xi \cdot t) \quad (10)$$

iii) se  $\xi^2 - K < 0$  ou seja  $K > \xi^2$ , têm-se duas raízes complexas conjugadas:

$$m_1 = -\xi - \omega \cdot i \text{ e } m_2 = -\xi + \omega \cdot i$$

onde  $\omega = \sqrt{K - \xi^2}$ , e a solução da Equação 6a é:

$$\bar{x}_k(t) = \exp(-\xi \cdot t) \cdot [c_1 \cdot \cos(\omega \cdot t) + c_2 \cdot \sin(\omega \cdot t)] \quad (11)$$

Tanto a primeira forma (Equação 9) quanto a segunda (Equação 10) de solução não apresentam oscilação. Para potencializar a capacidade exploratória do método opta-se pela terceira forma de solução que é sub-amortecida, necessariamente oscilatória. Assim, escolhendo a terceira forma de solução através da imposição  $K > \xi^2$ , é garantido um movimento sub-amortecido das partículas, obrigando-as a oscilarem na região de busca em torno de um ponto intermediário entre o melhor valor individual de cada partícula e do melhor valor global.

Neste caso, a velocidade da partícula  $k$  é dada pela seguinte equação:

$$v_k(t) = \exp(-\xi \cdot t) \cdot \left\{ (c_2 \cdot \omega - \xi \cdot c_1) \cdot \cos(\omega \cdot t) - (\xi \cdot c_2 + c_1 \cdot \omega) \cdot \sin(\omega \cdot t) \right\} \quad (12)$$

Para determinarmos os valores das constantes de integração  $c_1$  e  $c_2$  são usadas as condições iniciais definidas pelas Equações 6b e 6c. A forma recursiva do algoritmo é obtida quando usamos a Equação 11, já com as constantes definidas, para calcular os valores de  $x_k$  e  $v_k$  em  $t_{i+1}$ . As equações recursivas são:

$$x_k^{(i+1)} = X_k^{(i)} + \exp(-\xi \cdot \Delta t) \cdot \left\{ (x_k^{(i)} - X_k^{(i)}) \cdot \cos(\omega \cdot \Delta t) + \left[ \xi \cdot (x_k^{(i)} - X_k^{(i)}) + v_k^{(i)} \right] \cdot \frac{\sin(\omega \cdot \Delta t)}{\omega} \right\} \quad (13a)$$

$$v_k^{(i+1)} = \exp(-\xi \cdot \Delta t) \cdot \left\{ v_k^{(i)} \cdot \cos(\omega \cdot \Delta t) - \left[ K \cdot (x_k^{(i)} - X_k^{(i)}) + \xi \cdot v_k^{(i)} \right] \cdot \frac{\sin(\omega \cdot \Delta t)}{\omega} \right\} \quad (13b)$$

onde  $\Delta t = t_{i+1} - t_i > 0$ . O algoritmo tem três parâmetros de busca:  $\Delta t$ ,  $K$  e  $\xi$ . Porém, os valores de  $K$  e  $\xi$  estão sujeitos à restrição  $K > \xi^2$ , que garante o movimento oscilatório amortecido.

## Análise do Algoritmo

Uma vez proposto o novo algoritmo de otimização é necessário verificar as suas

características com relação à convergência e à exploração da região de busca, ou seja, a forma com que as partículas se movimentam pela região de busca até o equilíbrio. Para esta análise, uma versão simplificada do algoritmo de otimização é desenvolvida, resolvendo o sistema de equações recursivas (Equações 13a e 13b) através da transformada Z. O ponto de equilíbrio foi considerado igual a 0 e as equações obtidas são:

$$x_k(i) = \exp(-\xi \cdot \Delta t \cdot i) \cdot \left\{ x_k(0) \cdot \cos(\omega \cdot \Delta t \cdot i) + \left[ \xi \cdot x_k(0) + v_k(0) \right] \cdot \frac{\sin(\omega \cdot \Delta t \cdot i)}{\omega} \right\} \quad (14a)$$

$$v_k(i) = \exp(-\xi \cdot \Delta t \cdot i) \cdot \left\{ v_k(0) \cdot \cos(\omega \cdot \Delta t \cdot i) - \left[ K \cdot x_k(0) + \xi \cdot v_k(0) \right] \cdot \frac{\sin(\omega \cdot \Delta t \cdot i)}{\omega} \right\} \quad (14b)$$

onde se pode determinar a posição e a velocidade de cada partícula em cada iteração a partir de sua posição e velocidade iniciais. As equações do processo iterativo apresentadas nesta forma são úteis à análise da influência de cada parâmetro no movimento das partículas.. Assim, só é necessário analisar a evolução das partículas em uma única direção, já que a única conexão entre as direções de busca é a função objetivo, que nesta forma do algoritmo não está presente.

A análise do algoritmo foi realizada alterando os valores dos parâmetros de busca para identificar os efeitos de cada parâmetro no movimento das partículas. Foi avaliado o movimento de cinco partículas, cujas posições e velocidades iniciais foram as mesmas em cada simulação, conforme Tabela 1.

**Tabela 1:** Posição e velocidade inicial para as cinco partículas

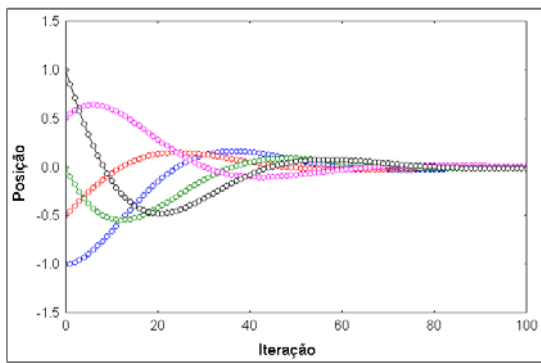
Partícula	Posição inicial	Velocidade inicial
1	-1,0	0,0
2	-0,5	0,5
3	0,0	-1,0
4	0,5	0,5
5	1,0	-1,0

Na Figura 1 podem ser vistos os resultados de 4 simulações, onde a posição de cada uma das cinco partículas é acompanhada durante 100 iterações do algoritmo, sendo o valor de  $\xi$  mantido igual a 0,5.

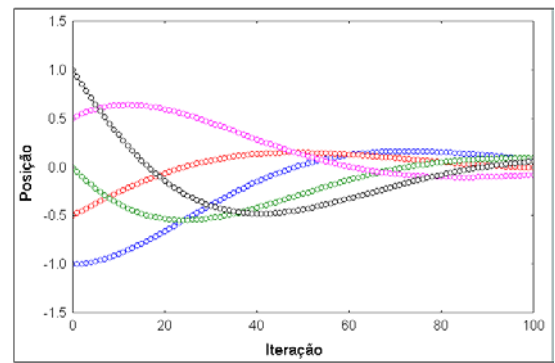
Observando a Figura 1, fica claro que o efeito do parâmetro  $\Delta T$  está relacionado com o número de iterações necessárias para convergência. Como o parâmetro  $\Delta T$  aparece nas exponenciais das equações do algoritmo, um aumento em  $\Delta T$  faz com que as partículas tenham uma

convergência com um número menor de iterações. Nas Figuras 1a-b, onde  $\Delta T$  é igual a 0,1, a convergência é alcançada com aproximadamente 70 iterações. Quando o valor de  $\Delta T$  é diminuído para 0,05 (Figuras 1c-d) o número de iterações necessárias para convergência é superior a 100.

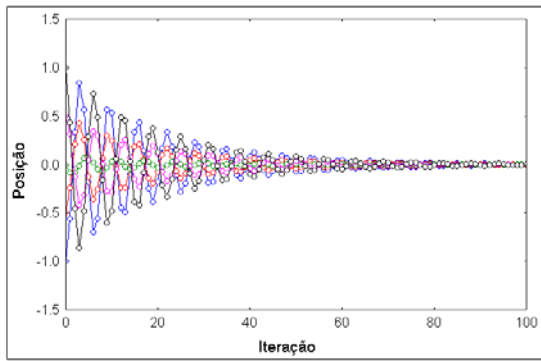
Já o efeito do parâmetro  $K$  está relacionado com a frequência das oscilações, isto é, aumentando o valor de  $K$  o número de oscilações aumenta para um mesmo número de iterações.



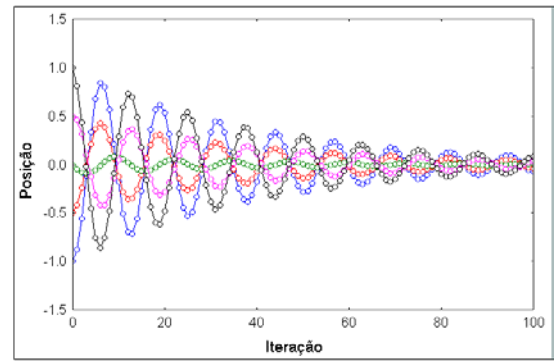
(a)  $\Delta T = 0,10$ ;  $K = 1$



(c)  $\Delta T = 0,05$ ;  $K = 1$



(b)  $\Delta T = 0,10$ ;  $K = 100$

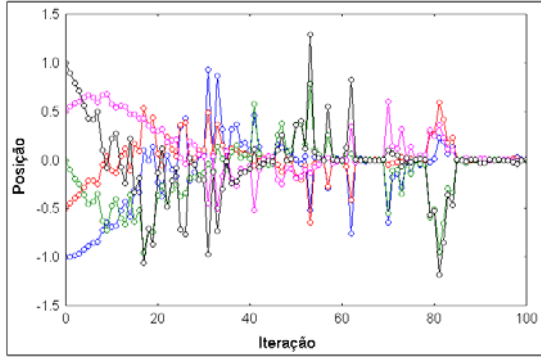


(d)  $\Delta T = 0,05$ ;  $K = 100$

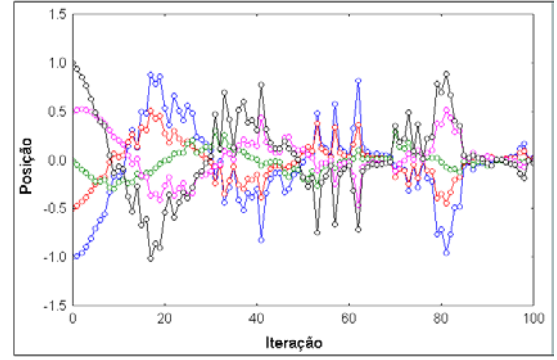
**Figura 1:** Evolução da posição das partículas ao longo das iterações, mantendo o valor de  $\xi$  igual a 0,5.

Apesar das partículas oscilarem bem ao longo das iterações, o caminho de cada partícula é muito uniforme, tornando o algoritmo determinístico. Com o objetivo aumentar o caráter aleatório da busca, o valor do parâmetro  $\xi$  deve assumir valores uniformemente distribuídos entre 0 e 1, pois como o parâmetro aparece tanto na função exponencial como nas funções

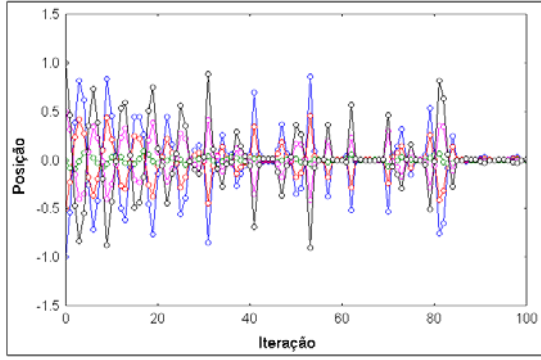
trigonométricas das equações do algoritmo, esta mudança deve provocar efeitos tanto na forma de convergência das partículas quanto nas oscilações das mesmas. Na Figura 2 as simulações apresentadas acima foram refeitas considerando o novo valor do parâmetro  $\xi$ , um número uniformemente distribuído entre 0 e 1, sorteado em cada iteração para cada partícula.



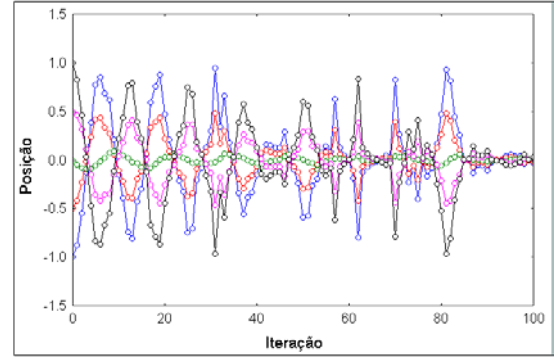
(a)  $\Delta T = 0,10$ ;  $K = 1$



(c)  $\Delta T = 0,05$ ;  $K = 10$



(b)  $\Delta T = 0,10$ ;  $K = 100$



(d)  $\Delta T = 0,05$ ;  $K = 100$

**Figura 2:** Evolução da posição das partículas ao longo das iterações, com o valor de  $\xi$  distribuindo-se uniformemente entre 0 e 1.

O resultado obtido com  $\xi$  distribuindo-se uniformemente entre 0 e 1 foi uma maior variedade nos caminhos de cada partícula, com cada partícula tendo uma oscilação distinta e, mesmo após a convergência, as partículas foram capazes de *escapar* do domínio de atração do ponto de equilíbrio. Esta característica potencializa a capacidade do algoritmo na exploração de ótimos globais.

### Avaliação do Algoritmo

A forma do algoritmo representada pelas Equações 15 foi utilizada na otimização de funções teste para a avaliação da eficiência do algoritmo em encontrar o mínimo global. As funções utilizadas foram:

a) Função de Levy No. 5:

$$f(x_1, x_2) = \sum_{i=1}^5 \left[ i \cdot \cos((i-1) \cdot x_1 + i) \right] \cdot \sum_{j=1}^5 \left[ j \cdot \cos((j+1) \cdot x_2 + j) \right] + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2 \quad (15)$$

onde  $-10 \leq x_i \leq 10$ , e  $i=1, 2$ . Esta função tem por volta de 760 mínimos locais e um mínimo global com o valor da função igual a  $f^* = -176.1375$ , no ponto  $(x_1, x_2)^* = (-1.3068, -1.4248)$  (Parsopoulos e Vrahatis, 2002). Quando a função assumir um valor inferior a -175, o problema será considerado resolvido.

b) Função de Rosenbrock:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[ 100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (16)$$

onde  $n = 30$ ,  $-30 \leq x_i \leq 30$ , e  $i = 1 \dots n$ , tem um ponto de mínimo em  $f^* = 0$ , em  $x_i=1$  para  $i=1,\dots,n$ , e o problema é considerado resolvido se a função assumir um valor inferior a 100 (Trelea, 2003).

Durante as otimizações a posição inicial e velocidade de cada partícula foram geradas aleatoriamente dentro das faixas definidas acima. Porém durante a busca tanto a velocidade quanto a posição de cada partícula não foi restrita a esta faixa,

permitindo que partículas movimentassem fora região de busca.

Com a função de Levy No. 5 os resultados foram obtidos para 10, 20 e 50 partículas, sendo feitas 50 repetições para avaliar a eficiência do algoritmo. Na Tabela 2 são apresentados os resultados obtidos nesta avaliação.

Já com a função de Rosenbrock os resultados foram obtidos com 30 e 60 partículas, sendo feitas 20 repetições para a avaliação da eficiência do algoritmo.

**Tabela 2:** Eficiência na busca do mínimo global da função de Levy No. 5.

Partículas	$\Delta T$	K				
		500	750	1000	2500	5000
10	0.10	90	90	76	82	66
	0.05	88	94	88	98	98
	0.01	74	76	78	86	98
20	0.10	100	98	96	98	86
	0.05	100	100	100	100	100
	0.01	82	88	88	96	100
50	0.10	100	100	100	100	100
	0.05	100	100	100	100	100
	0.01	98	100	100	100	100

**Tabela 3:** Eficiência na busca do mínimo da função de Rosenbrock.

Partículas	$\Delta T$	K		
		1000	5000	10000
30	0.005	45	70	70
	0.010	45	65	95
	0.050	0	100	0
60	0.005	95	90	100
	0.010	85	100	100
	0.050	10	100	10

Os resultados acima mostram que esta nova versão do algoritmo é capaz de encontrar o mínimo global das funções testadas. Para a função de Levy No 5 a eficiência do algoritmo, com os valores de K e  $\Delta T$  da tabela 2, foi praticamente de 100 % em todos os casos. Isto mostra uma grande capacidade de otimização global, já que esta função apresenta muitos mínimos locais. Já com a função de Rosenbrock o

algoritmo foi capaz de atingir 100 % de eficiência para algumas combinações de K e  $\Delta T$ . Para esta função, o valor de  $\Delta T$  deve ser menor que para o caso anterior, já que como a dimensão deste segundo problema é muito maior, o número de iterações deve ser maior e o parâmetro K deve ser maior pois é preciso cobrir bem toda a região de busca para garantir que o mínimo global seja encontrado.

## Conclusões

O novo algoritmo é eficiente na otimização de funções matemáticas e, sua gênese garante um comportamento oscilatório do movimento das partículas e uma grande robustez. Apresentou uma eficiência excelente na minimização da função de Levy No. 5 com praticamente 100% de eficiência. Já na função de Rosenbrock a eficiência de 100% foi alcançada para algumas combinações dos parâmetros.

Assim, o desempenho do algoritmo depende de uma escolha criteriosa de seus parâmetros. Porém esta tarefa é pouca árdua, já que cada parâmetro tem seu efeito qualitativo preestabelecido de acordo com: (i)  $\Delta T$  no número total de iterações necessárias e (ii)  $K$  na frequência de oscilação do movimento de cada partícula.

## Referências

- COCKSHOT, A.R., HARTMAN, B.E. Improving the fermentation medium for Echinocandin B production part II: Particle swarm optimization. *Process Biochemistry* 36 (2001) 661-669.
- COSTA Jr., E.F., LAGE, P.L.C., BISCAIA Jr., E.C. On the numerical solution and optimization of styrene polymerization in tubular reactors. *Computers and Chemical Engineering* 27 (2003) 2591-1604.
- OURIQUE, C.O., BISCAIA Jr., E.C., PINTO, J.C. The use of particle swarm optimization for dynamical analysis in chemical processes. *Computers and Chemical Engineering* 26 (2002) 1783-1793.
- PARSOPOULOS, K.E., VRAHATIS, M.N. Particle swarm optimization method in multiobjective problems. *Proceedings ACM Symposium on Applied Computing (SAC 2002)* 603-607.
- PARSOPOULOS, K.E., VRAHATIS, M.N. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing* 1 (2002) 235-306.
- TRELEA, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* 85 (2003) 317-325.