

Computação Natural: **PSO**

Prof. Dr. Rafael Stubs Parpinelli

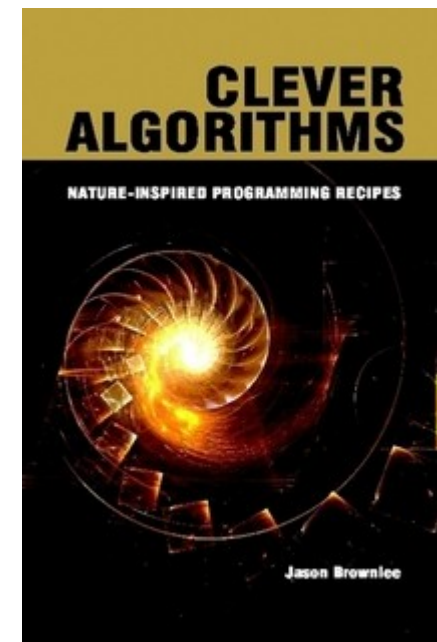
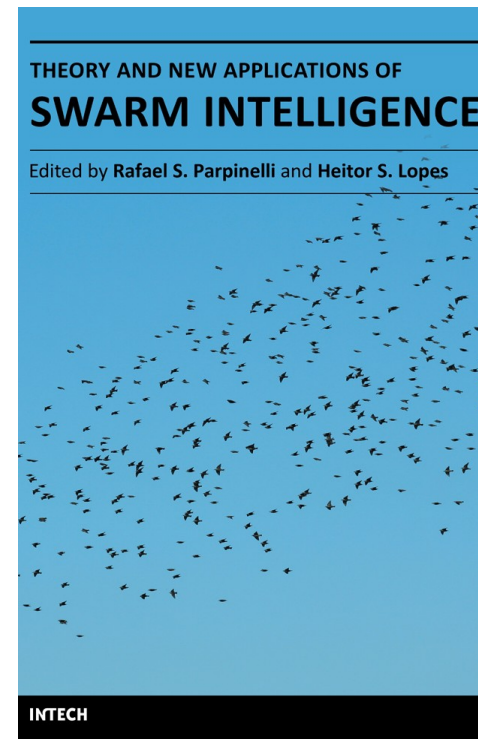
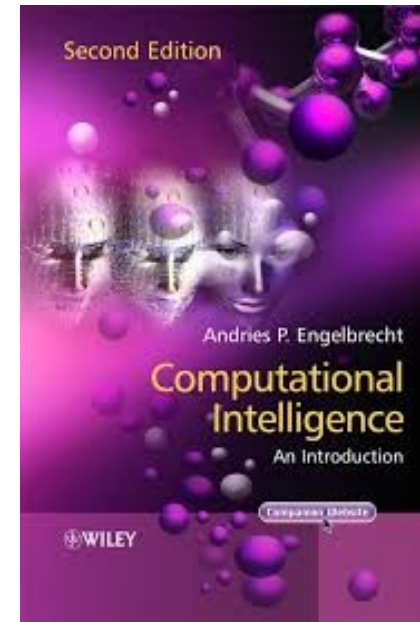
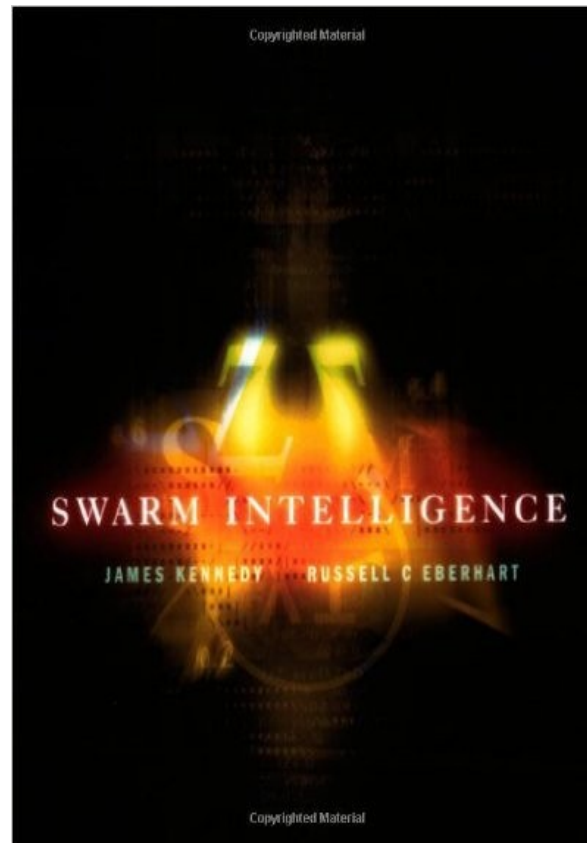
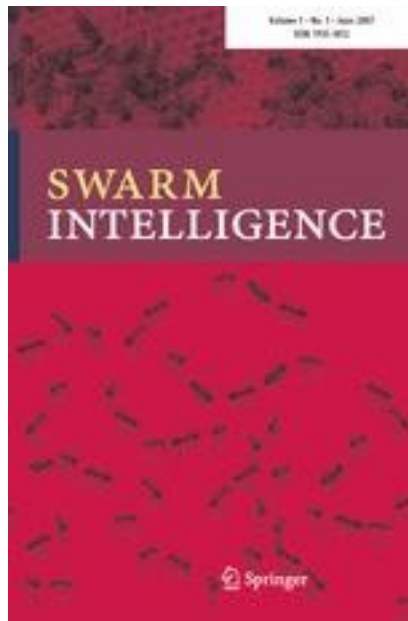
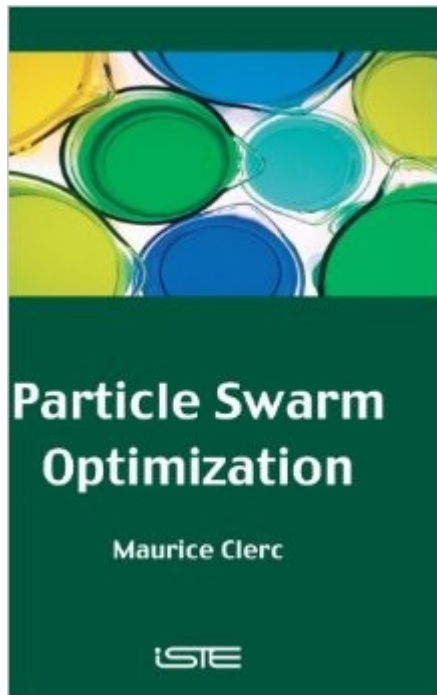
E-mail: rafael.parpinelli@udesc.br

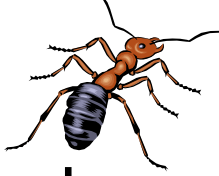


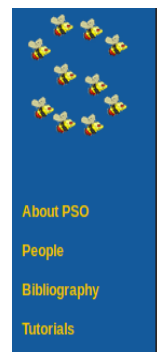
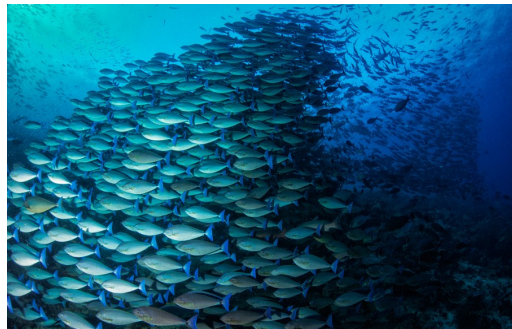
Programa de Pós-Graduação
em Computação Aplicada - UDESC



PSO :: *Algumas Referências*



- **Swarm Intelligence** → ACO e PSO
 - **Comportamentos emergentes** que surgem da coletividade de indivíduos simples que interagem entre si e com o ambiente
- ACO – *Ant Colony Optimization*: 
 - Inspirado no comportamento de busca por alimentos das formigas
 - Aplicações em problemas de otimização combinatorial
- PSO – *Particle Swarm Optimization*:
 - Inspirado na coreografia dos pássaros e cardumes



Particle Swarm Optimization

Introduction

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

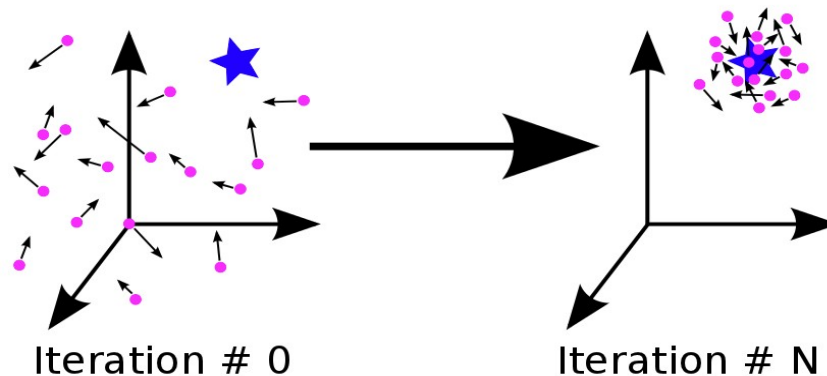
PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations.

<http://www.swarmintelligence.org/>
<http://particleswarm.info/>



PSO

- Enxame de Partículas:
 - Originalmente criado por Kennedy (psicólogo social) e Eberhart (engenheiro elétrico) (1995)
 - Inspirada na coreografia dos pássaros e cardumes
 - Simular o comportamento social de seres vivos
 - Ideia básica: A interação social é capaz de encontrar soluções ótimas para problemas difíceis



PSO

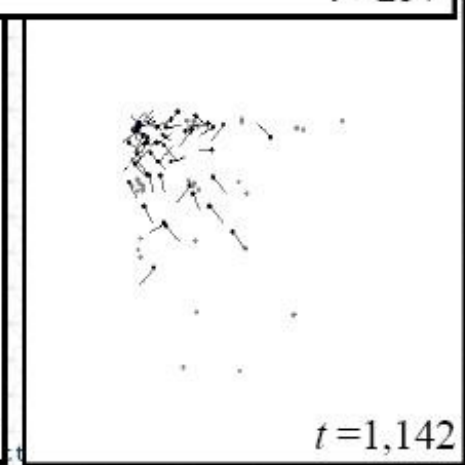
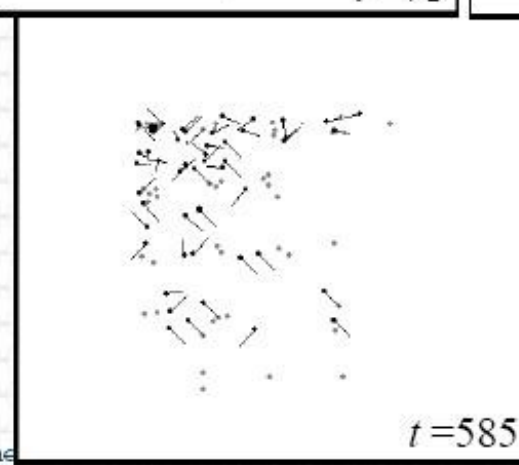
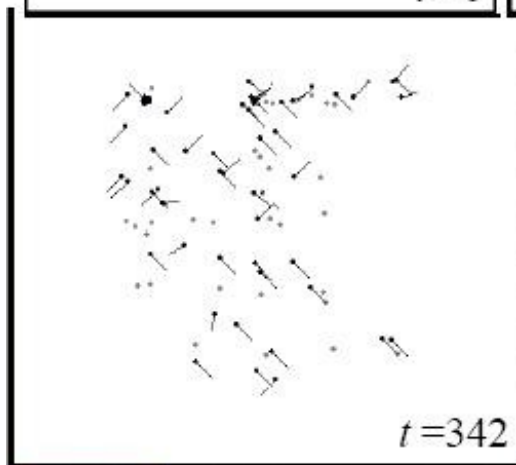
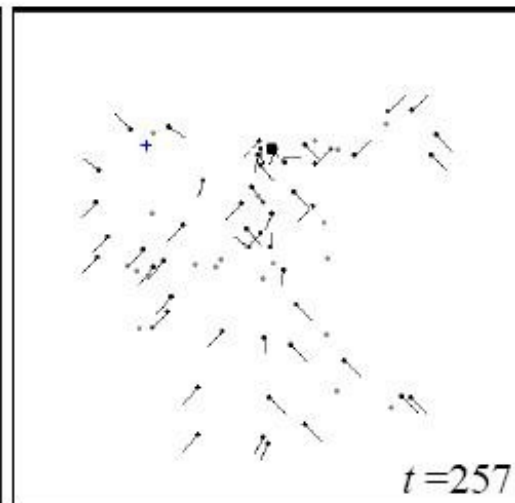
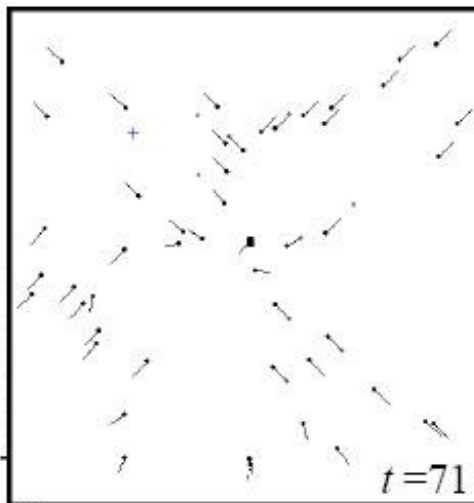
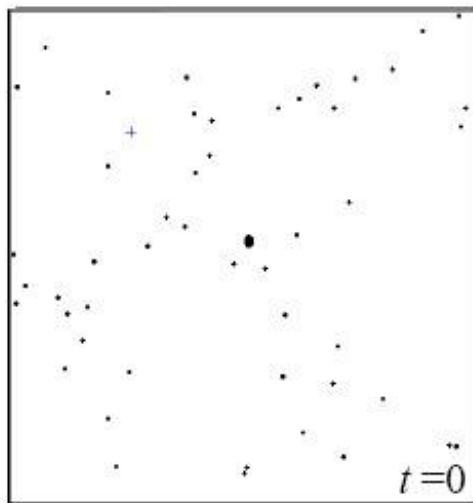
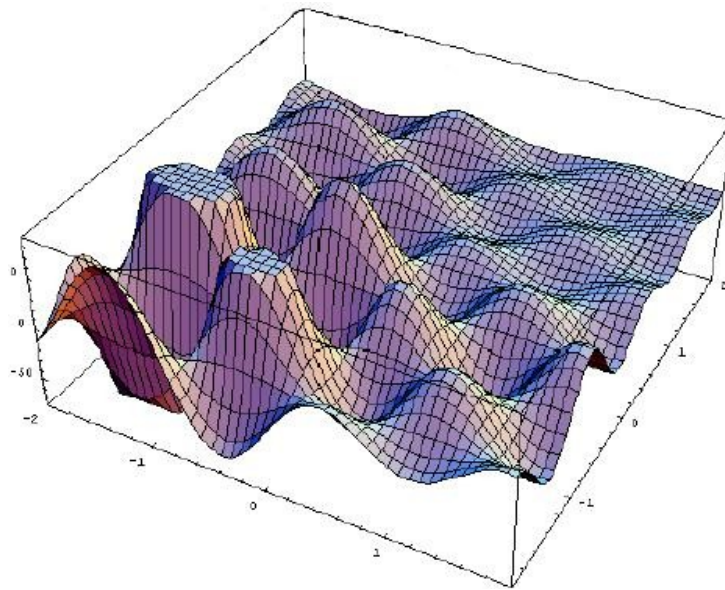
Comportamento Social

- Na natureza, quando um grupo de aves ou cardume de peixes procura alimento, pode -se considerar que cada indivíduo orienta sua busca por duas componentes:
 - a primeira é a sua própria experiência anterior em já conhecer locais onde costuma encontrar alimento (nostalgia)
 - a outra componente é a informação obtida através de outros indivíduos de seu grupo social.

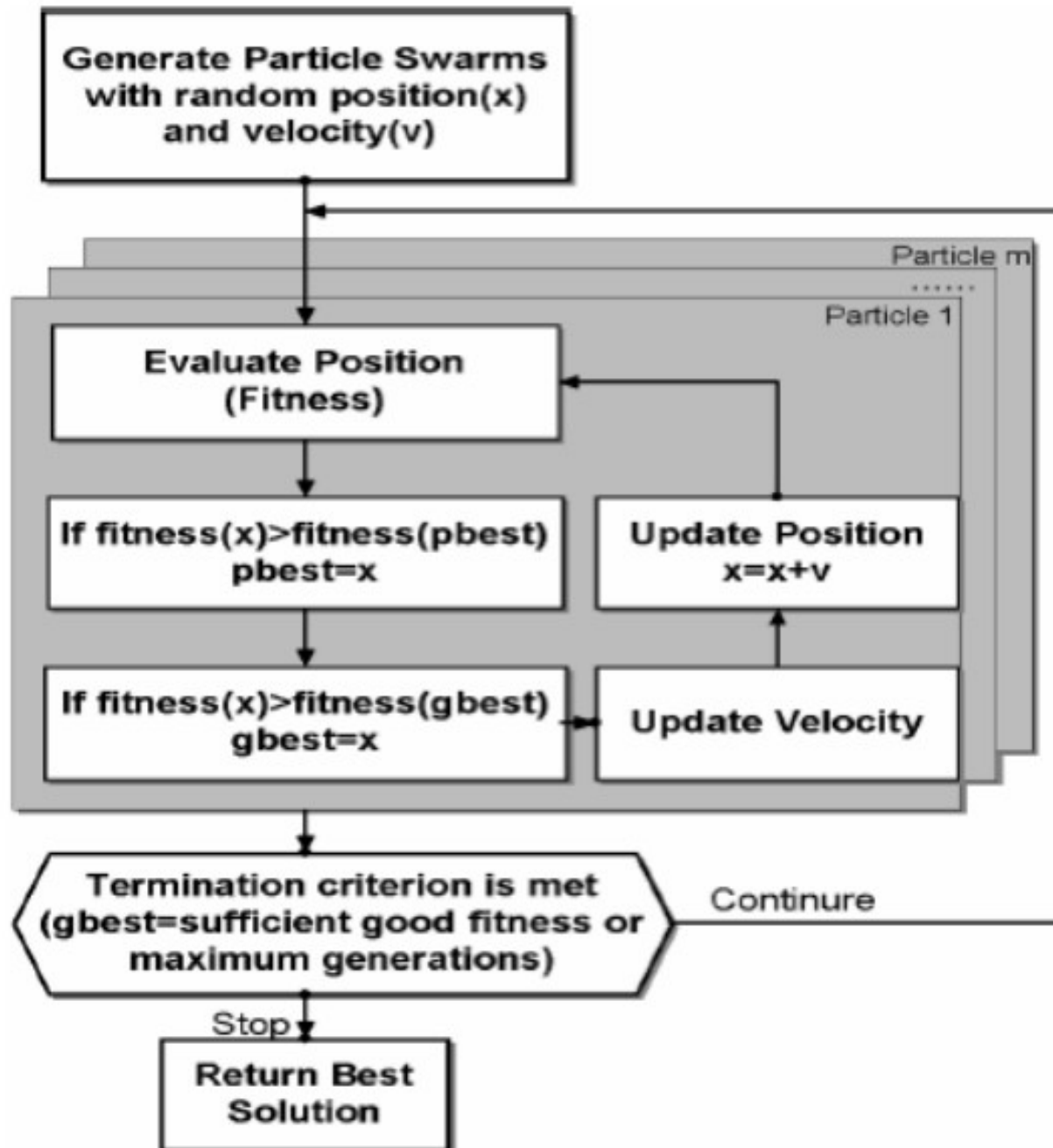
PSO

- Características:
 - Baseado em população
 - As potenciais soluções – partículas – “sobrevivem” um espaço de busca n-dimensional
 - A busca se dá através da aceleração e desaceleração das partículas: informação local (componente cognitivo) e informação global (componente social)

Exemplo de Visualização



Fluxo PSO



PSO

- Cada partícula é composta pelos seguintes vetores:

$$- X_i = (x_{i1}, x_{i2}, \dots, x_{in})$$

Posição atual

$$- V_i = (v_{i1}, v_{i2}, \dots, v_{in})$$

Deslocamento

$$- y_i = (y_{i1}, y_{i2}, \dots, y_{in})$$

Componente Cognitivo

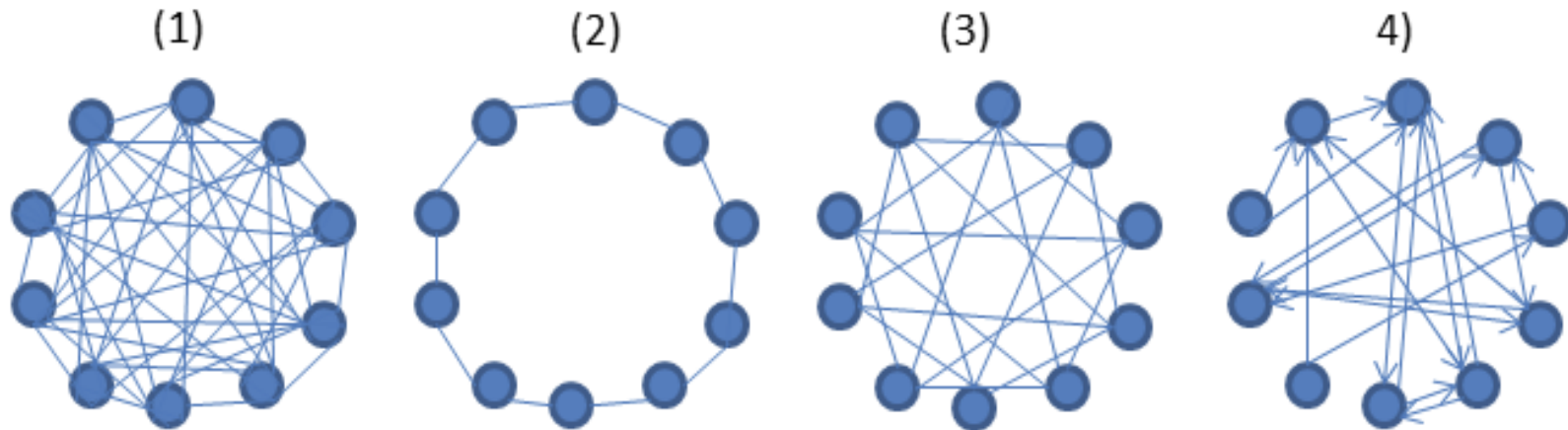
- Equação de atualização da melhor posição já visitada (problema de minimização):

$$y_i(t+1) = \begin{cases} y_i(t) & \text{se } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{se } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad [1]$$

PSO

- Existem duas versões do alg. PSO:
 - Gbest e Lbest
- A diferença baseia-se no grupo de partículas ao qual uma partícula irá interagir diretamente para identificar a informação da melhor partícula
 - Gbest: toda a população é considerada
 - Lbest: Somente alguns vizinhos especificados são considerados
- A melhor partícula será representada pelo símbolo: $\ddot{y}(t)$

PSO:: Algumas topologias



Graphical representation of (1) fully connected, (2) ring, (3) von Neumann and (4) random topology

PSO

- Possui duas sequências aleatórias independentes:
 - r_{1j} e $r_{2j} \in (0,1)$, onde $j \in 1..n$
- Possui duas constantes (coeficientes de aceleração):
 - $0 \leq c_1, c_2 \in \mathfrak{R} \leq 4$, usualmente $c_1 = c_2 = 2.05$
 - tamanho máximo de um passo que uma partícula pode dar em uma interação

PSO

- Equação de atualização das velocidades:

$$v_{ij}(t+1) = \begin{cases} v_{ij}(t) + c_1 \cdot r_{1j}(t) [y_{ij}(t) - x_{ij}(t)] + c_2 \cdot r_{2j}(t) [\ddot{y}_j(t) - x_{ij}(t)] \\ \quad \text{se } X_{\min} < x_{ij} < X_{\max} \quad \text{Onde } j \in 1..n \\ 0 \end{cases} \quad [2]$$

- Da equação acima:
 - c_1 regula o tamanho do passo na direção da melhor posição encontrada até o momento pela partícula (componente cognitivo)
 - c_2 regula o tamanho do passo na direção da melhor partícula encontrada globalmente/regionalmente (componente social)

PSO

- Equação de atualização de posições

$$x_{ij}(t+1) = \begin{cases} x_{ij}(t) + v_{ij}(t+1) & \text{se } X_{\min} \leq (x_{ij}(t) + v_{ij}(t+1)) \leq X_{\max} \\ X_{\max} & \text{se } (x_{ij}(t) + v_{ij}(t+1)) > X_{\max} \\ X_{\min} & \text{se } (x_{ij}(t) + v_{ij}(t+1)) < X_{\min} \end{cases} \quad [3]$$

- Restrições

- Limites do espaço de busca para cada dimensão:
[X_{\min} , X_{\max}] ou [$-X_{\max}$, X_{\max}]
- V_{\max} : utilizada para limitar as velocidades, evitando que as partícula extrapolem o espaço de busca.
[V_{\min} , V_{\max}] ou [$-V_{\max}$, V_{\max}]
- Normalmente usa-se $V_{\min} = X_{\min}$ e $V_{\max} = X_{\max}$

PSO

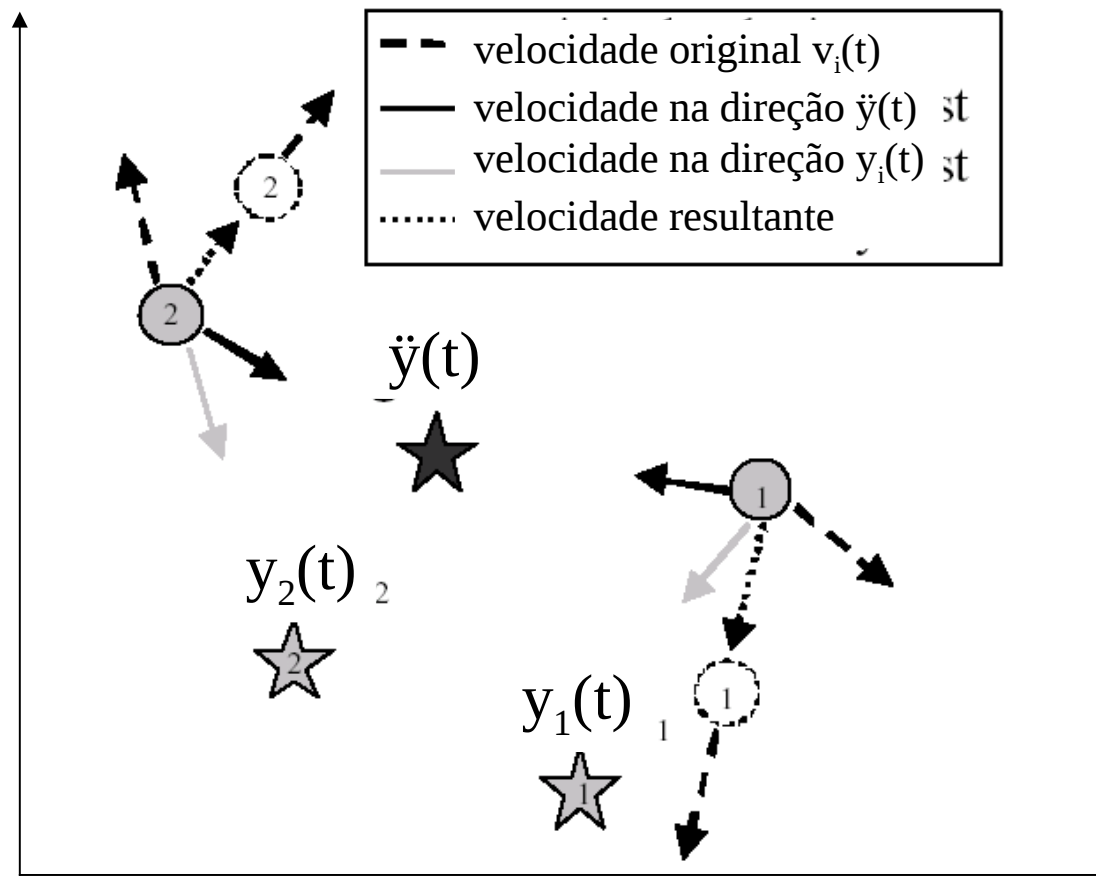
- Pseudo código:

```
Crie e inicialize um PSO n-dimensional: S
Repita
  Para cada partícula i
    Se  $f(x_i) < f(y_i)$  // identifica melhor componente cognitivo
      Então  $y_i = x_i$ 
    Se  $f(y_i) < f(\bar{y})$  // identifica melhor componente social
      Então  $\bar{y} = y_i$ 
  Fim_para
  Realize as atualizações no enxame S utilizando as equações (2) e (3)
Até condição de parada ser verdadeira
```

- Passo de inicialização
 - Inicializar aleatoriamente cada coordenada x_{ij} dentro de seu domínio
 - Inicializar aleatoriamente cada v_{ij} dentro de seu domínio
 - Atribuir $y_i = x_i$ para toda partícula

PSO

- A cada passo de atualização, o comportamento de uma partícula é um balanço de 3 possíveis escolhas:
 - Seguir a trajetória original: $v_i(t)$
 - Ir na direção do componente social: $\ddot{y}(t)$
 - Ir na direção do componente cognitivo: $y_i(t)$



PSO

- Melhoria: peso de inércia (w)
 - Controla a influência da velocidade em um instante de tempo anterior
- Velocidade define a intensidade da busca: valores altos resultam em movimentos bruscos; valores baixos podem resultar na exploração insuficiente do espaço de busca

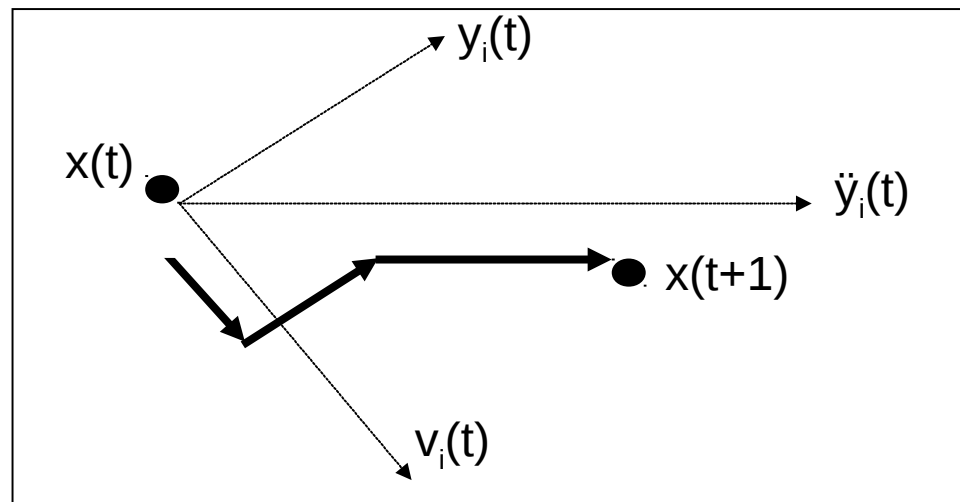
$$v_{ij}(t+1) = w.v_{ij}(t) + c_1.r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2.r_{2j}(t)[\ddot{y}_j(t) - x_{ij}(t)]$$

Onde $j \in 1..n$

- w regula o *trade-off* entre exploração local e global: $0 < w \in \Re < 1.5$

PSO

- Peso de inércia:
 - Os três coeficientes social/cognitivo quantificam respectivamente:
 - O quanto a partícula confia em si mesma (w)
 - O quanto ela confia na sua experiência (c_1)
 - O quanto ela confia nos seus vizinhos (c_2)



PSO

- Melhoria: fator de constrição (K)
 - Atua no componente resultante da composição dos movimentos

$$v_{ij}(t+1) = k[v_{ij}(t) + c_1 \cdot r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 \cdot r_{2j}(t)[\ddot{y}_j(t) - x_{ij}(t)]]$$

Onde $j \in 1..n$

- Onde

$$k = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}$$

$$\phi = c_1 + c_2, \phi > 4$$

Aplicações

- Originalmente projetado para otimização contínua:
- Otimização de funções
- Treinamento de redes neurais
- Evolução da arquitetura de redes neurais
- Controle de sistemas *fuzzy*
- Balanceamento de ingredientes para maximizar a produção
- Cálculo de carga em sistemas elétricos
- Segmentação de imagens
- Simulação de sinais de EEG
- Verificação de voz,