

Inteligência de Enxame

 Inteligência de enxames é a denominação aplicada a tentativa de desenvolvimento de algoritmos para a solução distribuída de problemas inspirando-se no comportamento coletivo de colônias de insetos sociais e outras sociedades de animais.

 Existem vários algoritmos ou meta-heurísticas de inteligência de enxame:

- ★ Otimização por Enxame de Partículas – PSO
- ★ Otimização por Conjunto de tribos – TRIBES
- ★ Otimização por Colônia de Formigas – ACO
- ★ Otimização por Colônia de Abelhas – ABC
- ★ Sistemas Artificiais Imunológicos – AIS

Inteligência de Enxame: *PSO*



A otimização por enxame de partículas:

- ★ É baseada em uma estratégia inspirada no voo dos pássaros e movimento de cardumes de peixes;
- ★ Permite a otimização global de um função objetivo
 - ★ A função objetivo pode ser qualquer função 1 a d dimensões
- ★ Utiliza um enxame de partículas
- ★ Realiza a movimentação das partículas dentro do espaço de busca;
 - ★ O espaço de busca é um hyper-paralelepípedo:
 - ★ Sendo multidimensional de 1 a d dimensões,
 - ★ Para cada dimensão d , é definido um intervalo:
 $[min_d, max_d]$

Inteligência de Enxame: *PSO*

🐜 Função objetivo: parábola em 3D

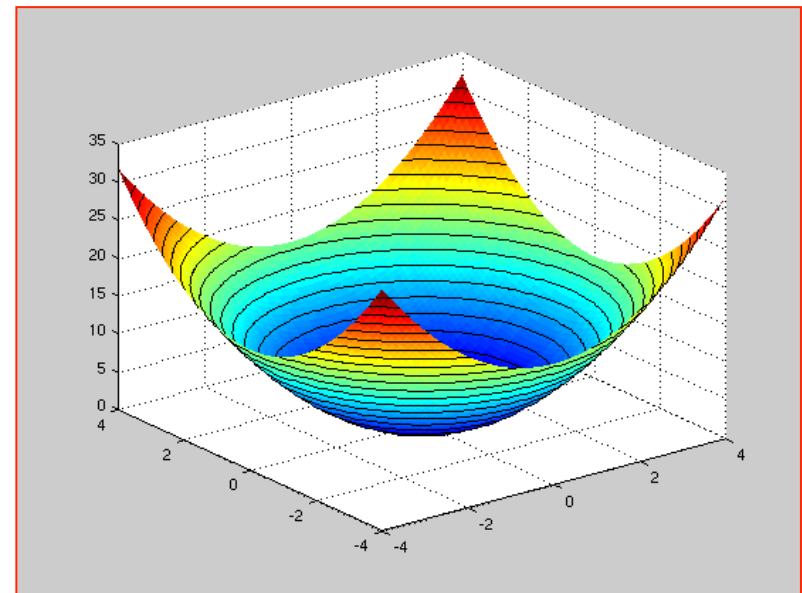
$$f_d(\mathbf{X}) = \sum_{i=1}^d x_i^2, \mathbf{X} = [x_1, x_2, \dots, x_d]$$

$$f_2(x, y) = x^2 + y^2$$

🐜 Espaço de busca de 2 dimensões:

★ x em $[-4, +4]$

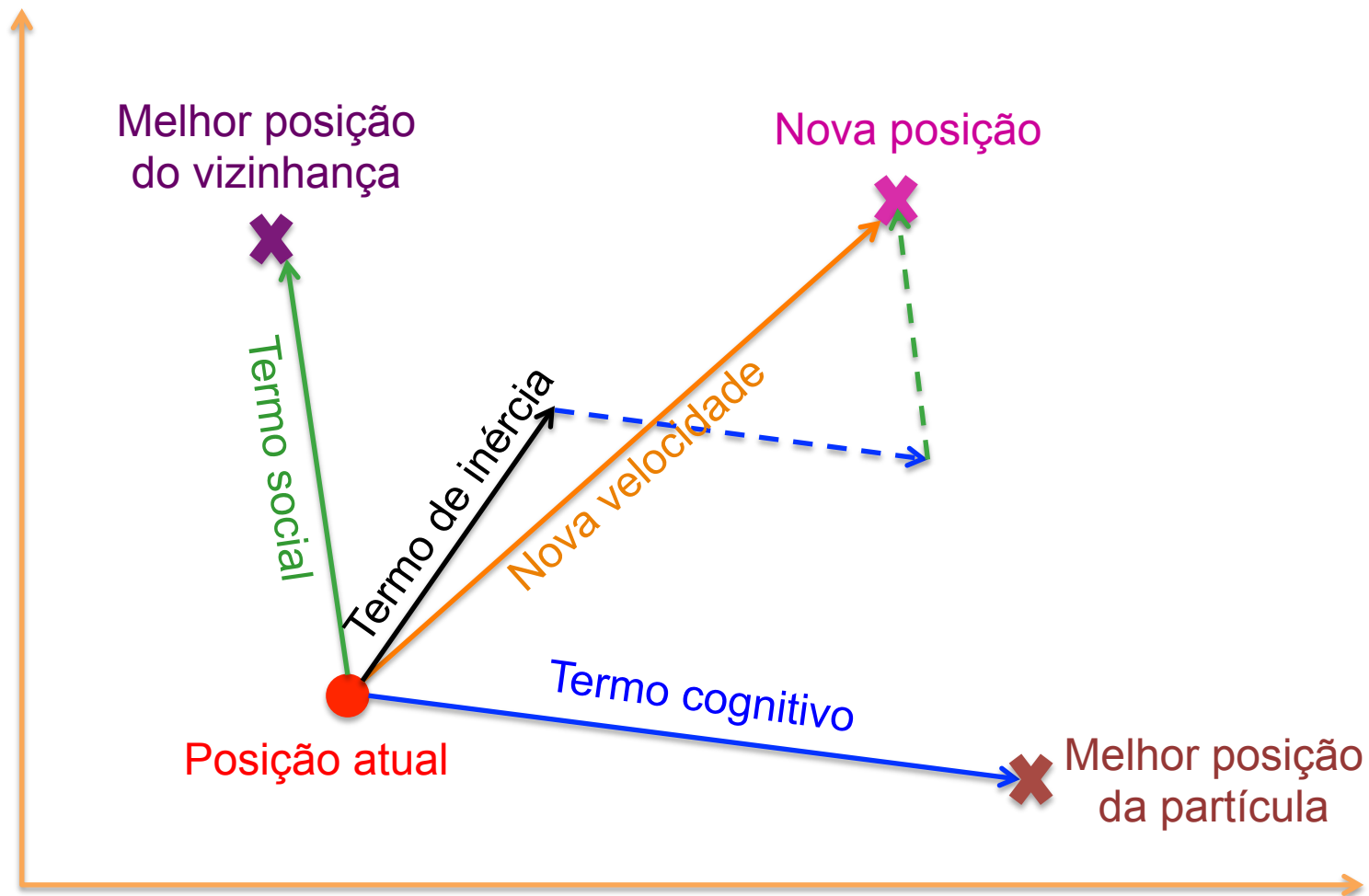
★ y em $[-4, +4]$



Inteligência de Enxame: *PSO*

-  A estratégia usa um conjunto de N partículas que se movimentam dentro do espaço de busca.
-  Cada partícula:
 - ★ É um ponto mapeado no espaço de busca;
 - ★ Representa uma solução em potencial;
 - ★ É definida pelas d coordenadas no espaço de busca;
 - ★ É avaliada usando a função objetivo;
 - ★ É movimentada no espaço de busca usando uma velocidade, que tem 3 termos;
 - ★ Termo de inércia
 - ★ Termo cognitivo
 - ★ Termo social

Inteligência de Enxame: *PSO*



$$v_{ij}(t+1) = \omega v_{ij}(t) + \varphi_1 r_{1j}(t)(p_{ij}(t) - x_{ij}(t)) + \varphi_2 r_{2j}(t)(g_{ij}(t) - x_{ij}(t))$$

$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$, onde $\omega, \varphi_1, \varphi_2$ são os parâmetros a ser definidos.

Inteligência de Enxame: *PSO*

Algoritmo de inicialização do PSO

$vbest_i := \infty$

Para i de 1 a n Faça

Inicializar x_i e v_i aleatoriamente

$p_i := x_i$

$pbest_i = f(p_i)$

Se $pbest_i < vbest$ Então

$vbest_i := pbest_i$

$g_i := p_i$

Fim Se

Fim Para

Fim inicialização do PSO

Inteligência de Enxame: *PSO*

Algoritmo de inicialização de x_i e v_i

Para j de 1 a d Faça

$x_{ij} := \text{rand}(\min_j, \max_j)$

$v_{ij} := \text{rand}(0, v\max_j)$

Fim Para

Fim inicialização de x_i e v_i

Inteligência de Enxame: *PSO*

Algoritmo do PSO

Inicialização

Repita

Para i de 1 a n Faça

Para j de 1 a d Faça

$$v_{ij} := \omega * v_{ij} + \varphi_1 * rand() * (p_{ij} - x_{ij}) + \varphi_2 * rand() * (g_{ij} - x_{ij})$$

Controle de v_i

$$x_{ij} := v_{ij} + x_{ij}$$

Confinamento de p_i

Fim Para

Atualização de $pbest_i$ e $vbest_i$ considerando x_i

Fim Para

Até (condição de parada) Fim PSO

Inteligência de Enxame: *PSO*

Algoritmo do controle da velocidade da partícula i

Para j de 1 a d Faça

Se $v_{ij} > v_{max_j}$ Então

$v_{ij} := v_{max_j}$

Fim Se

Fim Para

Fim controle de velocidade

Algoritmo do confinamento da partícula i

Para j de 1 a d Faça

Se $x_{ij} < min_j$ or $x_{ij} > max_j$ Então

$$x_{ij} := (min_j + max_j)/2$$

Fim Se

Fim Para

Fim confinamento

Inteligência de Enxame: *PSO*

Algoritmo do Atualização do $pbest_i$ e $vbest_i$

$fitness := f(x_i)$

Se $fitness < pbest_i$ Então

$pbest_i := fitness$

$p_i := x_i$

Se $fitness < vbest_i$ Então

$vbest_i := fitness$

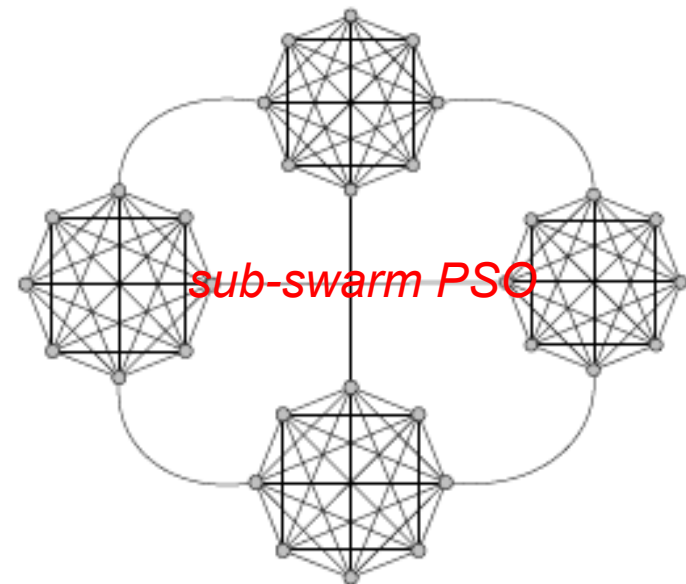
$g_i := x_i$

Fim Se

Fim Atualização

Sistemas Multi-agentes: *Organização*

🐜 Topologias de vizinhanças mais usadas:



🐜 *Local best PSO*:

- ★ Vizinhança da partícula i : $(i - 1) \bmod N$ e $(i + 1) \bmod N$
- ★ Cada partícula tem seu $lbest_i$

🐜 *Global best PSO*:

- ★ Vizinhança da partícula i : $1, \dots, i-1, i+1, \dots, N$
- ★ Todas as partículas tem um único $gbest$

Inteligência de Enxame: *PSO*



A condição de parada que pode ser:

- ★ O número máximo de iteração;
 - ✦ A processo pode parar um pouco antes de chegar ao ótimo global;
 - ✦ A processo pode iterar muito mais do que é preciso.
- ★ Um limiar do erro
 - ✦ Só pode ser usada quando a solução do problema é conhecida *a priori*;
- ★ O número máximo de avaliações da função objetivo;
 - ✦ Apresenta as mesmas desvantagens que o número máximo de iterações;
 - ✦ Avalia melhor o esforço computacional.

Inteligência de Enxame: *PSO*



Uma aplicação do PSO é definida por 3 elementos que definem o problema:

- ★ A função objetivo;
 - ★ Número de dimensões;
- ★ O espaço de busca;
 - ★ Intervalos válidos para cada dimensão.
- ★ O critério de parada.

Inteligência de Enxame: *PSO*



O algoritmo de PSO requer a configuração dos seguintes parâmetros:

- ★ O tamanho de enxame definido pelo número total de partículas N ;
- ★ O coeficiente ω de inércia;
 - ★ Pode ser adaptativo conforme a evolução do processo;
 - ★ O ajuste requer conhecimento da aptidão da solução ótima;
- ★ O peso φ_1 que valoriza a experiência pessoal da partícula (cognitivo);
- ★ O peso φ_2 que valoriza a experiência grupal do enxame (social).
- ★ A topologia da vizinhança da partícula.

Inteligência de Enxame: *PSO*

- 🐜 Para permitir uma boa convergência, os coeficientes de inércia ω , cognitivo φ_1 e social φ_2 devem ser configurados com cuidado.

$$\omega = \frac{1}{\phi - 1 + \sqrt{\phi^2 - 2\phi}} \text{ e } \varphi_1 = \varphi_2 = \phi\omega \text{ com } \phi > 2$$

$$\phi = 2 \Rightarrow \omega = 1 \text{ e } \varphi_1 = \varphi_2 = 2$$

$$\phi = 2,1 \Rightarrow \omega = 0,641742430504416 \text{ e } \varphi_1 = \varphi_2 = 1,347659104059274$$

- 🐜 Heurísticas para escolha dos valores de ω , φ_1 e φ_2 :

$$0 \leq \omega < 1$$

$$1 > \omega > \frac{1}{2}(\varphi_1 + \varphi_2) - 1 \geq 0$$

$$\varphi_1 + \varphi_2 \approx 4$$