Sistemas Inteligentes

– Particle Swarm Optimization (PSO) –
(Otimização por Enxame de Partículas)

Prof. Ivan Nunes da Silva

1. Inteligência Coletiva (Swarm)

Conceitos introdutórios

- É aquela inteligência que acaba estando distribuída quando se considera o todo, que é incessantemente valorizada, que é coordenada em tempo real, e que resulta em mobilização efetiva de competências bem simples.
- Os sistemas de Inteligência Coletiva são compostos tipicamente por:
  - Uma população de organismos que interagem localmente um com o outro e com seu ambiente.
  - Os agentes seguem regras bem simples, sendo que nenhuma estrutura centralizada de controle rege como os agentes (organismos) individuais devem se comportar.
  - Essas interações locais entre tais agentes conduzem à resolução de problemas com comportamento global complexo.
- Assim como os Algoritmos Genéticos, métodos baseados em Inteligência Coletiva (*Particle Swarm Optimization – PSO //* ***Enxame de Partículas***) são aplicados tipicamente em problemas de **otimização de sistemas**.

1. Inteligência Coletiva (Swarm)

Características de inteligência coletiva

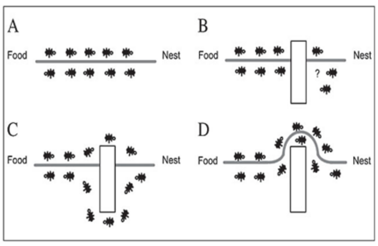
- São notáveis a interação social e a capacidade de adaptação de vários organismos.
- Segundo a ótica da inteligência artificial e da psicologia social:
  - Inteligência e sucesso, em muitos casos, provêm muito mais da sociabilidade do que da capacidade individual.
- A inteligência artificial distribuída é uma área de pesquisa que aborda fenômenos ligados a “inteligência social”:
  - Formigas;
  - Cupins;
  - Abelhas;
  - Peixes;
  - Pássaros.

A inteligência de Enxames vem sendo designada como classe de algoritmos de otimização.

1. Inteligência Coletiva (Swarm)

Estratégia de mapeamento de problemas

- As estratégias modelam o “comportamento social” observado em diversos tipos de colônias.
  - Inicialmente, os indivíduos se movimentam sem nenhuma orientação prévia;
  - Se aglomeram em colônias, até que um consegue encontrar o ninho ou a comida;
  - Este atrai os que estiverem mais próximos e a chance dos outros encontrarem aumenta consideravelmente.
- Os algoritmos possuem um vetor de velocidades e outro de posições para cada uma das partículas.



2. Contexto de Métodos Swarm

Mecanismos de funcionamento

- **Sempre baseados em duas variáveis básicas:**
  - **Posição**
  - **Velocidade**
- **Como exemplo, têm-se os pássaros:**
  - Pássaros estão dispostos aleatoriamente e estes estão à procura por alimento e um local para construir o seu ninho.
  - Eles, à priori, não sabem onde está esse lugar, mas este é único.
  - Qual o melhor comportamento que os pássaros terão que realizar para conseguir efetuar seu objetivo?
- **Estratégia de busca da melhor solução:**
  1. Inicialmente, os pássaros voam sem nenhuma orientação prévia;
  2. Se aglomeram em bandos, até que um consegue encontrar o ninho ou a comida;
  3. Este atrai os que estiverem mais próximos;
  4. A chance dos outros encontrarem aumenta consideravelmente.

5

2. Contexto de Métodos Swarm

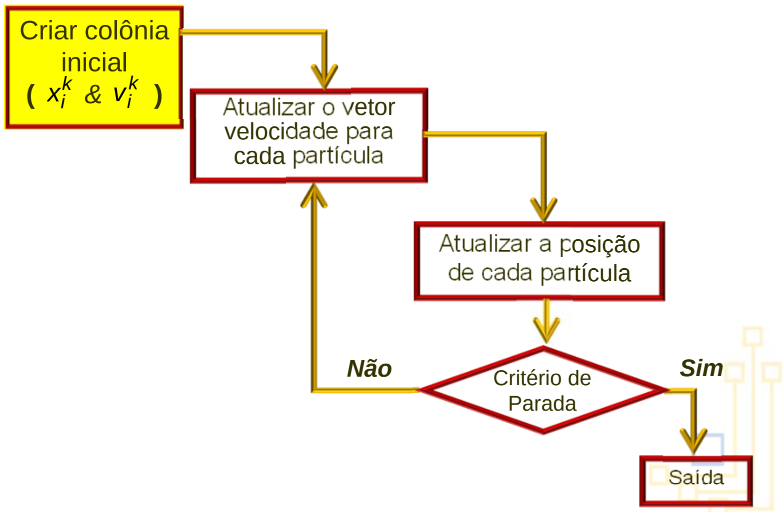
Terminologia dos componentes

- Para o caso dos pássaros, estes são denominados de as "**partículas**" do processo:
  - Durante a busca por alimento ou ninho, as "partículas" usam de suas experiências individuais e da experiência do próprio bando.
- Assim, o PSO é um algoritmo que possui um vetor de **velocidades** (v_i^k) e outro de **posição** (x_i^k);
  - v_i^k é a velocidade corrente da partícula *i* na iteração *k*.
  - x_i^k é a posição corrente da partícula *i* na iteração *k*.
- A **posição** de cada partícula é atualizada de acordo com a velocidade atual, levando-se em conta os seguintes aspectos:
  - Conhecimento adquirido pela "partícula" (*pbest*).
  - Conhecimentos adquirido pelo "bando" (*gbest*).

6

3. Algoritmo de Métodos Swarm

Diagrama de blocos principais



7

3. Algoritmo de Métodos Swarm

Criando colônia inicial (posição)

- A inicialização da população é normalmente obtida com as partículas dispostas aleatoriamente sobre o domínio de definição do problema;
- A equação seguinte mostra uma maneira de se obter a posição inicial:

$$x_i^{inicial} = x_{\min} + rand_1 \cdot (x_{\max} - x_{\min}) \qquad (1)$$

onde:

- $x_i^{inicial}$ é a posição inicial da partícula *i* ;
- $x_{\min}$ é o limite inferior do domínio de definição;
- $x_{\max}$ é o limite superior do domínio de definição;
- $rand_1$ é um número aleatório entre 0 e 1.

8

3. Algoritmo de Métodos Swarm

Criando colônia inicial (velocidade)

- Cada partícula possui também uma velocidade que foi inicializada aleatoriamente.
- A equação seguinte mostra uma maneira de se obter o vetor de velocidades iniciais:

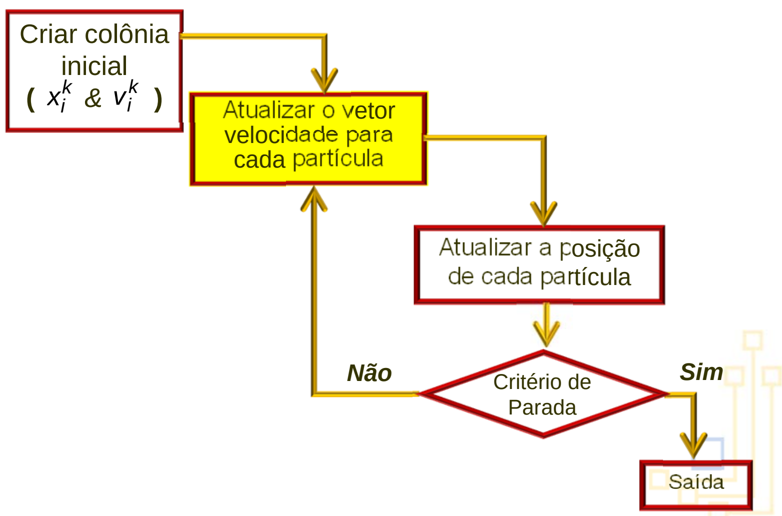
$$\boxed{v_i^{inicial} = \frac{x_{min} + rand_2 \cdot (x_{max} - x_{min})}{\Delta t}} \qquad (2)$$

onde:

- $v_i^{inicial}$ é a velocidade inicial da partícula *i* ;
- x_{min} é o limite inferior do domínio de definição;
- x_{max} é o limite superior do domínio de definição;
- $rand_2$ é um número aleatório entre 0 e 1;
- Δt é o intervalo de tempo considerado.

3. Algoritmo de Métodos Swarm

Atualizando vetor de velocidades



3. Algoritmo de Métodos Swarm

Atualizando vetor de velocidades (regra de ajuste)

- O vetor de velocidades pode ser atualizado conforme a seguinte equação:

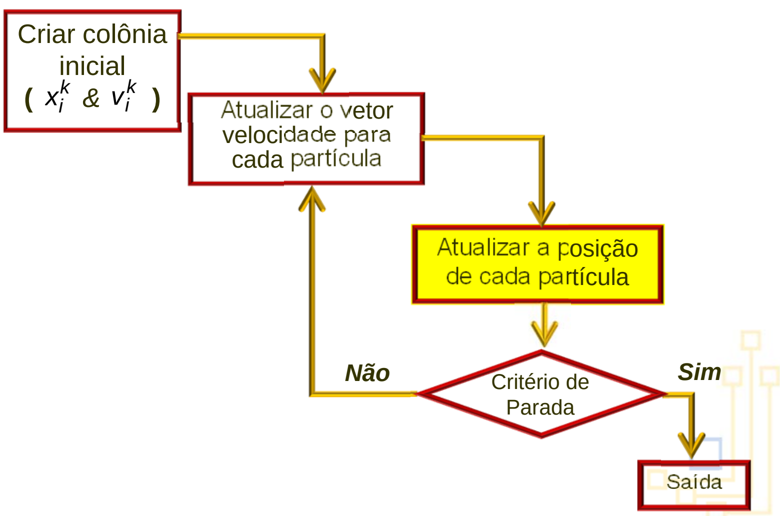
$$v_i^{k+1} = w_i \cdot v_i^k + c_1 \cdot rand \cdot (pbest_i - x_i^k) + c_2 \cdot rand \cdot (gbest^k - x_i^k) \quad (3)$$

onde:

- v_i^k é a velocidade corrente da partícula *i* na iteração *k* ;
- v_i^{k+1} é a velocidade atualizada da partícula *i* ;
- x_i^k posição corrente da partícula *i* na iteração *k* ;
- $pbest_i$ é o melhor valor de retorno (custo) da partícula *i* ;
- $gbest^k$ é o melhor valor de retorno do bando na iteração *k* ;
- *rand* é um número aleatório entre 0 e 1;
- c_1 e c_2 são coeficientes de ponderação; ***{parâmetros de confiança}***
- w_i é uma função de ponderação para a velocidade da partícula *i* . ***{representa a sua inércia}***

3. Algoritmo de Métodos Swarm

Atualizando vetor de posições



3. Algoritmo de Métodos Swarm

Atualizando vetor de posições (regra de ajuste)

- A próxima posição das partículas é calculada usando-se a seguinte expressão:

$$\boxed{x_i^{k+1} = x_i^k + v_i^{k+1} \cdot \Delta t} \qquad (4)$$

onde:

- x_i^{k+1} é a posição de cada partícula *i* na iteração *k*+1;
- x_i^k é posição corrente da partícula *i* na iteração *k* ;
- v_i^{k+1} é o vetor de velocidade da partícula *i* na iteração *k*+1 ;
- Δt é o intervalo de tempo considerado.

13

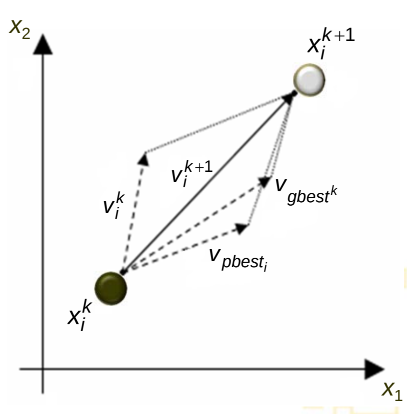
3. Algoritmo de Métodos Swarm

Atualizando vetor de posições (interpretação geométrica)

$$\boxed{v_i^{k+1} = w_i \cdot v_i^k + c_1 \cdot rand \cdot (pbest_i - x_i^k) + c_2 \cdot rand \cdot (gbest^k - x_i^k)}$$

$$\boxed{x_i^{k+1} = x_i^k + v_i^{k+1} \cdot \Delta t}$$

- x_i^k é a posição corrente;
- x_i^k é a posição de busca atualizada;
- v_i^k é a velocidade corrente;
- v_i^{k+1} é a velocidade atualizada;
- v_{pbest_i} é a velocidade em função de $pbest_i$;
- v_{gbest^k} é a velocidade em função de $gbest^k$.



14

3. Algoritmo de Métodos Swarm

Atualizando vetor de posições (velocidade máxima)

- Um outro parâmetro presente é o v_{max} , vinculado a um procedimento de limitação de velocidade para as partículas;
- Este procedimento impede que o algoritmo desenvolva valores de velocidades não pertinentes ao problema;
- Para quaisquer dimensões *d* do vetor v_i^k, tem-se:

$$-v_{\max} \le v_i^k \le v_{\max} \quad (5)$$

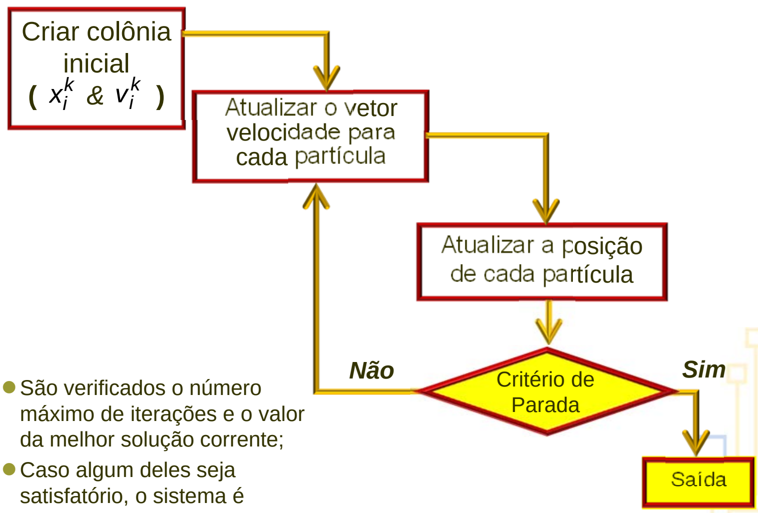
- Caso quaisquer das partículas do sistema desenvolva, em qualquer uma de suas dimensões, uma velocidade além de seus limites, este valor é automaticamente saturado, ou seja:

$$v_i^k = \begin{cases} -v_{\max}, \text{ se } v_i^k \le -v_{\max} \\ v_{\max}, \text{ se } v_i^k \ge v_{\max} \end{cases} \quad (6)$$

15

3. Algoritmo de Métodos Swarm

Critério de parada



- São verificados o número máximo de iterações e o valor da melhor solução corrente;
- Caso algum deles seja satisfatório, o sistema é finalizado.

16

3. Algoritmo de Métodos Swarm

Pseudocódigo computacional

//Inicialização – Código em Pseudocódigo da *Particle Swarm Optimization*//

1. Inicialize aleatoriamente as posições $\boldsymbol{x}_i^k$ das partículas.
2. Inicialize aleatoriamente as velocidades $\boldsymbol{v}_i^k$ das partículas.
3. Inicialize $\boldsymbol{pbest}_i$ e $\boldsymbol{gbest}^k$

//Construção – loop principal//

4. Para cada partícula ***i*** , faça as seguintes instruções:

    //Atualiza a melhor posição da partícula atual *pbest*$_i$ //

    Se $\boldsymbol{x}_i^k > \boldsymbol{p}_{best_i}$ então

        Para cada dimensão ***d*** do problema faça $\boldsymbol{p}_{best_i} = \boldsymbol{x}_i^k$

    //Atualiza a melhor partícula//

    Para cada partícula vizinha, faça

        Se $\boldsymbol{pbest}_i > \boldsymbol{gbest}^k$ então $\boldsymbol{gbest}^k = \boldsymbol{pbest}_i$

    //Atualização das posições//

    Para cada dimensão ***d*** do problema faça

        Calcula $\boldsymbol{v}_i^{k+1}$ de acordo com a expressão (3).

    Verifique a condição de $\boldsymbol{v}_{max}$ dada em (5). Caso necessário, aplique a expressão (6).

5. Atualiza $\boldsymbol{x}_i^{k+1}$ de acordo com a equação (4).

//Condição de término//

6. Caso a condição de término não seja satisfeita, volte ao passo 4.

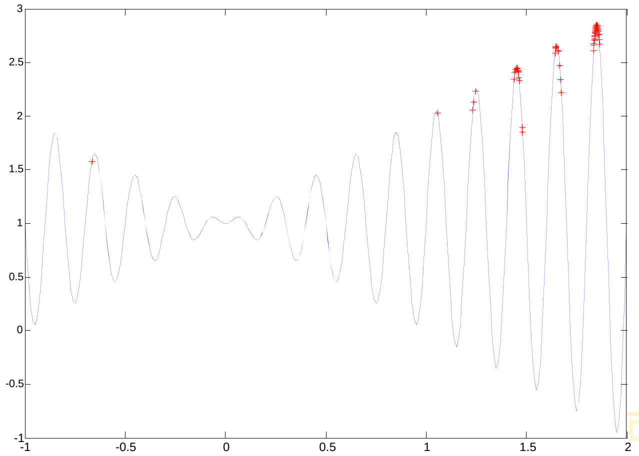
17

4. Exemplo Ilustrativo

Otimização não-linear de sistemas

$$f(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 1$$

$$gbest^{100} = 1.8506$$
$$f(gbest^{100}) = 2.8502$$



18

5. Considerações Adicionais

Requisitos de aplicabilidade & tópicos promissores

- **Requisitos de aplicabilidade:**
  - Conversão das possíveis soluções do problema a partir de sua representação por um vetor de posições e velocidades.
  - Existência de um método para medir a qualidade (Função Objetivo) de uma solução potencial.
- **Tópicos de investigação promissores:**
  - Sistemas híbridos.
  - Otimização multi-objetivo.
  - Aspectos de convergência, escabilidade e análise de complexidade.
  - Auto-adaptação dos seus parâmetros internos.

19

Fim da Apresentação

Núcleo de Pesquisa em
AUTOMAÇÃO INTELIGENTE
de Processos e Sistemas

20