

Universidade Federal do Pará
Instituto de Tecnologia -ITEC
Disciplina: IA Bio-Inspirada e Otimização.



Particle Swarm Optimization - PSO

Profa: Jasmine Araújo.

Prof: Glauco.

Particle Swarm Optimization



- O PSO foi criado baseado no comportamento de enxames da natureza, tais como dos peixes e pássaros.
- Desde que o PSO foi proposto, tem crescido o interesse na área de pesquisa de inteligência de enxame.
- Este algoritmo busca no espaço de uma função objetivo através do ajuste da trajetória de agentes individuais, chamados de partículas, como a continuidade dos caminhos formados por vetores posicionais em uma maneira quasi-estocástica

Particle Swarm Optimization

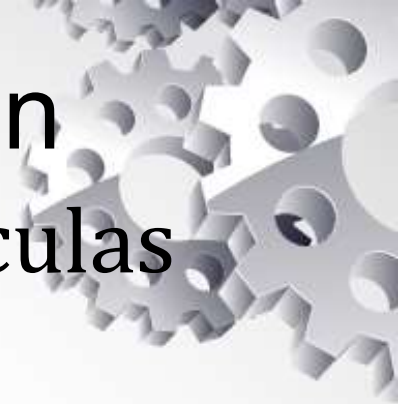
- O movimento de uma partícula do enxame consiste em 2 componentes principais:
 - Um componente estocástico
 - Um componente determinístico
- Cada partícula é atraída para a posição do melhor atual g^* best e para sua própria melhor localização x_i já ocorrida. Enquanto ao mesmo tempo tem uma tendência de se mover aleatoriamente.

Particle Swarm Optimization

- x_i e v_i são vetor posição e velocidade para a partícula i .
- O novo vetor velocidade é determinado pela fórmula:
- $$v_i^{t+1} = v_i^t + \underbrace{\alpha \varepsilon_1 [g^* - x_i^t]}_{\text{influência social}} + \underbrace{\beta \varepsilon_2 [x_i^* - x_i^t]}_{\text{influência pessoal}}$$

onde ε_1 e ε_2 são dois vetores aleatórios e seus valores variam de 0 a 1.

- Os parâmetros α e β são parâmetros de aprendizado ou constantes de aceleração.
Valores típicos $\alpha \approx \beta \approx 2$



Particle Swarm Optimization

- As localizações iniciais de todas as partículas deveriam se distribuir relativamente e uniformemente de maneira que pudessem ter uma amostra de todas as regiões, isso é especialmente importante para problemas multimodais.
- A velocidade inicial de uma partícula pode ser zero. $v_i^{t=0} = 0$
- A nova posição pode então ser atualizada:
- $x_i^{t+1} = x_i^t + v_i^{t+1}$
- Embora v_i possa ter qualquer valor mas tipicamente ele varia de 0 até v_{max}



Particle Swarm Optimization

- Existem muitas variantes do PSO, a melhoria mais conhecida é a inserção de um fator de inércia:
- $$v_i^{t+1} = \theta v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^* - x_i^t]$$
- *onde o θ pode ter valores entre 0 e 1.*
- No caso mais simples é escolhido um valor constante tipicamente entre 0.5 e 0.9 .
- Isso é equivalente a introduzir uma massa virtual para estabilizar o movimento das partículas e por esta razão é esperado que o algoritmo tenha uma convergência mais rápida.

Particle Swarm Optimization



- Exemplo de execução para a maximização da função:
- $f(x_1, x_2) = x_1^2 + x_2^2 - x_1x_2 + 2x_1 + 4x_2 + 3$
- $-5 \leq x_1, x_2 \leq 5$
- Parâmetros do PSO:
 - número de partículas=5,
 - α e $\beta = 1.5$,
 - *número de iterações* = 20,
 - *dimensão do problema* = 2.

Particle Swarm Optimization



- Inicialmente como o número de partículas foi determinado como 5, esse valor é preenchido aleatoriamente entre -5:5.
- Exemplo:
 - -0.9355 3.1836
 - -1.7027 -0.3482
 - 4.5879 -3.5555
 - -4.9214 1.3059
 - 3.5489 1.0510
- A função fitness de cada uma dessas posições é:
 - 27.8516 0.6293 47.9569 30.7330 24.2714
- o maior valor é 47.9569 que corresponde a posição 4.5879 -3.5555 (gbest)
- todas as posição de pbest.

Particle Swarm Optimization



- a velocidades de cada partícula são:

- 0.0443 0.0443
- 0.0443 0.0443
- 0.0443 0.0443
- 0.0443 0.0443
- 0.0443 0.0443

- atualizando as velocidades:

- $$v_i^{t+1} = \theta v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^* - x_i^t]$$
- para $v11 = v11 + \alpha \varepsilon_1 [g_{best} - x_i^t] + \beta \varepsilon_2 [p_{best} - x_i^t]$
- $v11 = 0.0443 + 1.5 * rand * (4.5879 - (-0.9355)) + 1.5 * rand * (-0.9355 - (-0.9355))$
- $v11 = 2.9719$

- *atualizando a posição:*

- $$x_i^{t+1} = x_i^t + v_i^{t+1}$$
- $x11 = x11 + v11 = -0.9355 + 2.9719 = 3.9074.$

*verifica-se se o novo valor de x11 está dentro dos limites -5,5 se sim então o valor alterado.
calcula-se o valor de x12 e assim sucessivamente para cada partícula.*



Particle Swarm Optimization

Iteração 1:

- Aleatoriamente definir velocidades entre 0 e 1
- Aleatoriamente definir os valores de x entre -5 e 5
- Calcular os valores da função objetivo $f(x)$
- Como não existe iteração anterior a primeira vez p_{best} é igual a x .
- Calcular o g_{best} de acordo com a finalidade da otimização minimizar ou maximizar

v (velocity)	x (position)	$f(x)$	p_i^g (pbest)
$\begin{bmatrix} \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \end{bmatrix}$	$\begin{bmatrix} \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \end{bmatrix}$	$\begin{bmatrix} \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \end{bmatrix}$	$\begin{bmatrix} \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \\ \text{.....} & \text{.....} \end{bmatrix}$
$p^g(g_{best}) = [\text{.....} \quad \text{.....}]$			

Particle Swarm Optimization

Iteração 2:

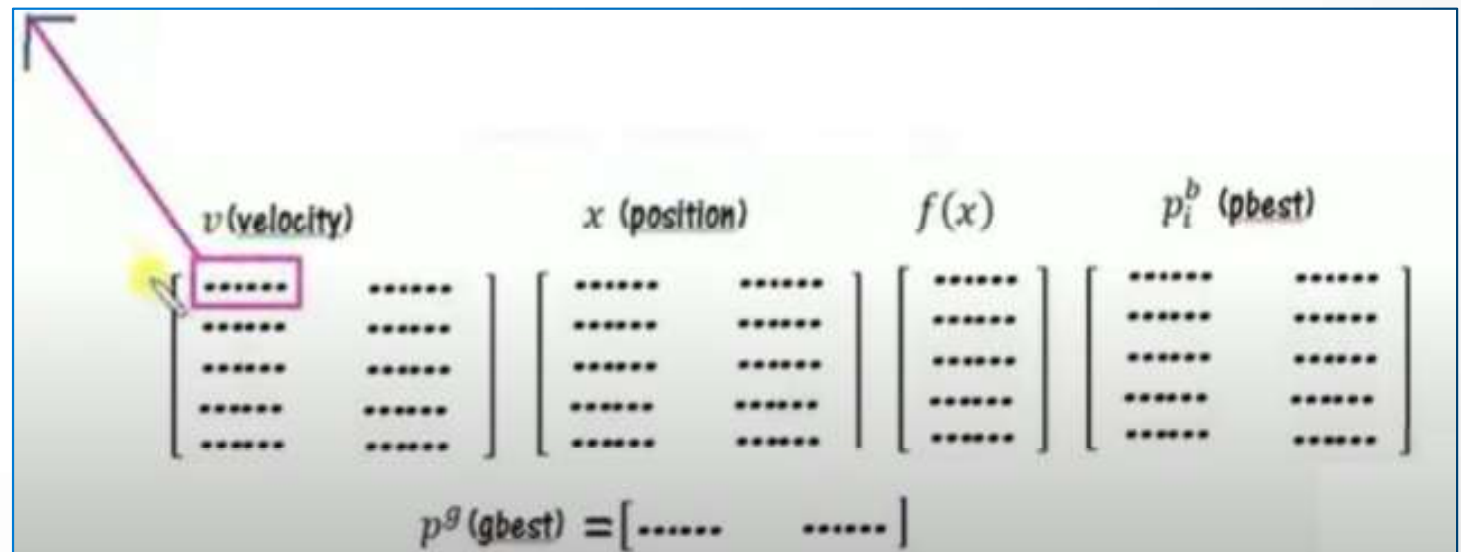
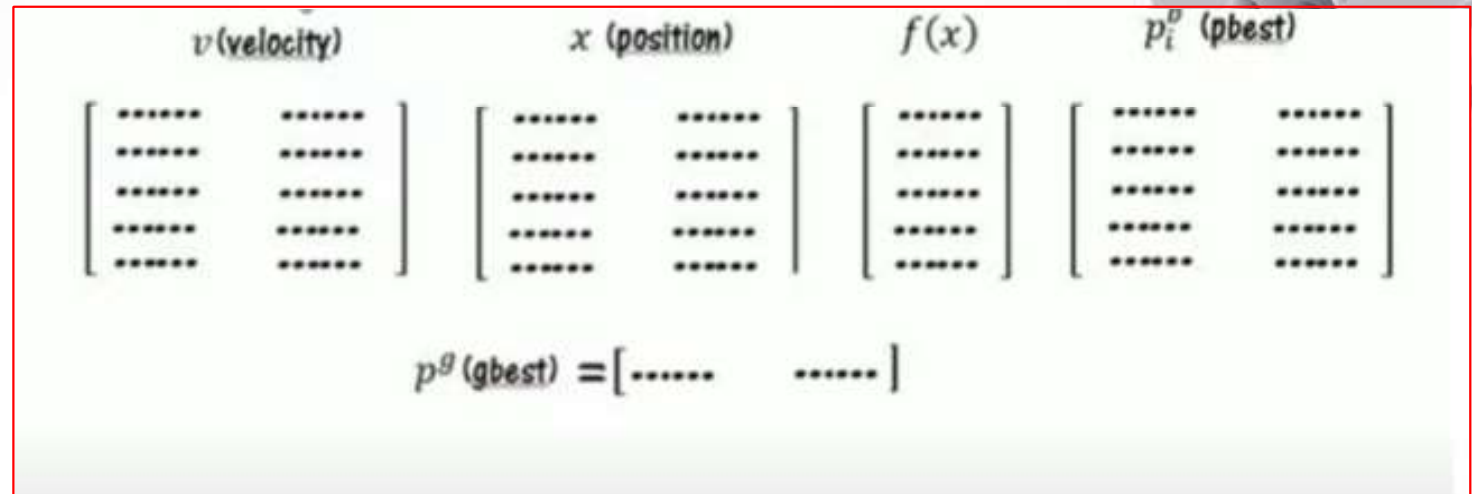
- Definir velocidades de acordo com a fórmula:

$$v_i^{t+1} = \theta v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^* - x_i^t]$$

- atualizando a posição:*

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

- Calcula para todas as partículas nova velocidade e nova posição*
- Ao final calcula então $f(x)$ e escolhe max ou min e separa gbest e o pbest*



Particle Swarm Optimization



- Iteração 3 em diante:
- Atualiza velocidade e posição
- Calcula a função fitness
- Atualiza o gbest
- Atualiza o pbest

Particle Swarm Optimization

