Particle Swarm Optimization (PSO)

Particle Swarm Optimization

- Otimização por nuvem de partículas
- Desenvolvido pelo psicólogo social James Kennedy e o engenheiro eletricista Russel Eberhart, em 1995
- Originalmente desenvolvido para resolver problemas de otimização com variáveis contínuas

Princípio

- Simula o comportamento social de um bando de pássaros à procura de um alvo (alimento, local para pouso, proteção contra predadores etc.)
- Estudos apontam que o bando encontra seu alvo por meio de um esforço conjunto
- Isto sugere que eles compartilham informações

Princípio

- Algoritmo populacional
 - A população é chamada de "nuvem" ou "enxame"
 - Os indivíduos são chamados de "partículas"
- O enxame evolui por meio de cooperação e competição entre seus membros
- As partículas se beneficiam da sua própria experiência e da experiência de outros membros do enxame durante a busca pelo alvo

Notação

$$\mathbf{x}_{i} = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,n} \end{bmatrix}, \quad \text{posição da partícula } i \text{ (coordenadas)}$$

$$\mathbf{v}_{i} = \begin{bmatrix} v_{i,1} \\ v_{i,2} \\ \vdots \\ v_{i,n} \end{bmatrix}, \quad \text{velocidade da partícula } i$$

f(x_i), aptidão da partícula i
 m, tamanho da população de partículas

Notação

\mathbf{p}_i	pbest; (personal best)
	a melhor posição encontrada pela partícula i
g	gbest (global best)
	a melhor posição encontrada por
	todas as partículas
c_1, c_2	parâmetros cognitivo e social
	(também chamados de taxas de aprendizado)
W	ponderação de inércia
r_{1j}, r_{2j}	números aleatórios entre 0 e 1

Atualização de posição e velocidade

Atualização de velocidade na iteração k

$$v_{ij}^{k+1} = wv_{ij}^k + c_1r_{1j}(p_{ij}^k - x_{ij}^k) + c_2r_{2j}(g_j^k - x_{ij}^k)$$

para
$$i = 1, ..., m$$
 e $j = 1, ..., n$.

Atualização de posição na iteração k

$$\mathbf{X}_i^{k+1} = \mathbf{X}_i^k + \mathbf{V}_i^{k+1}$$

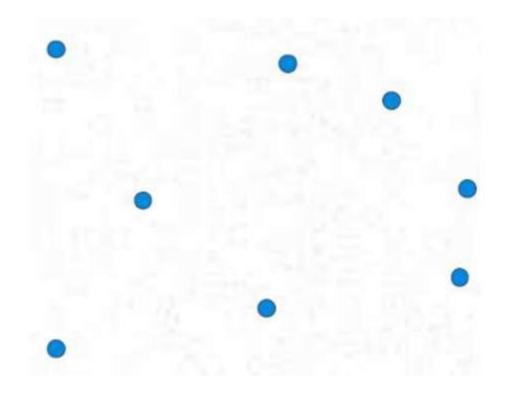
para
$$i = 1, \ldots, m$$

Componentes Cognitivo e Social

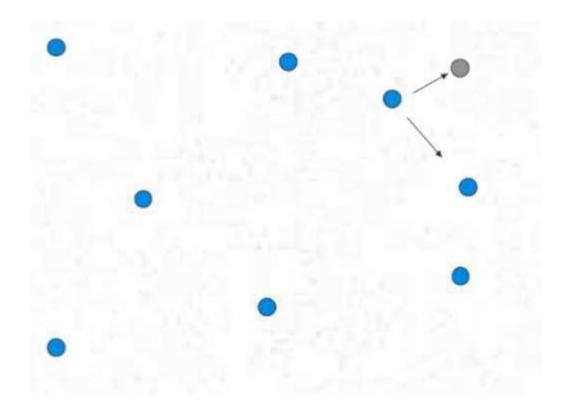
(p_i^k _i - x ^k) é o componente cepreisienta a experiência individual da partícula *i* até a *k*-ésima geração
(g^k -_ix ^k) é o componente sepreisenta a experiência da nuvem de partículas até a geração atual

Atualização de posição e velocidade

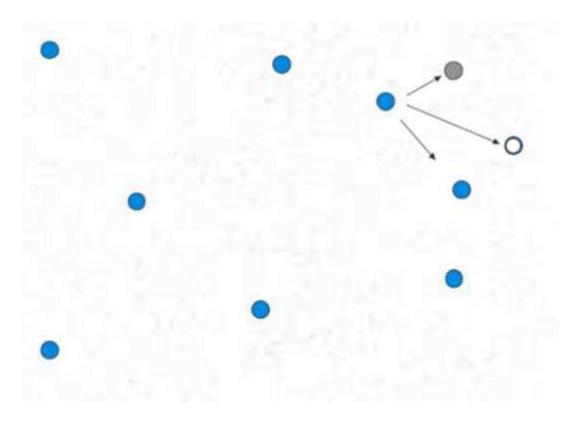
- A nova velocidade da partícula é influenciada por três fatores:
 - velocidade anterior
 - distância entre sua posição atual e melhor posição alcançada até então
 - distância entre sua posição atual e a melhor posição do grupo
- A partir do cálculo dessa nova velocidade, a partícula "voa" para sua nova posição



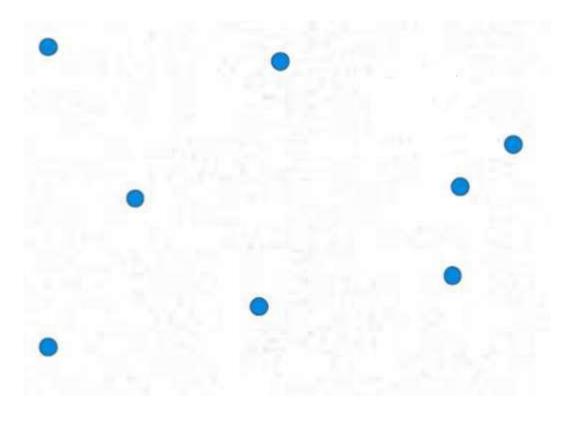
Inicialmente, diversas partículas são espalhadas aleatoriamente no espaço de busca



Cada partícula utiliza sua melhor posição no passado (em cinza) e sua melhor posição na vizinhança

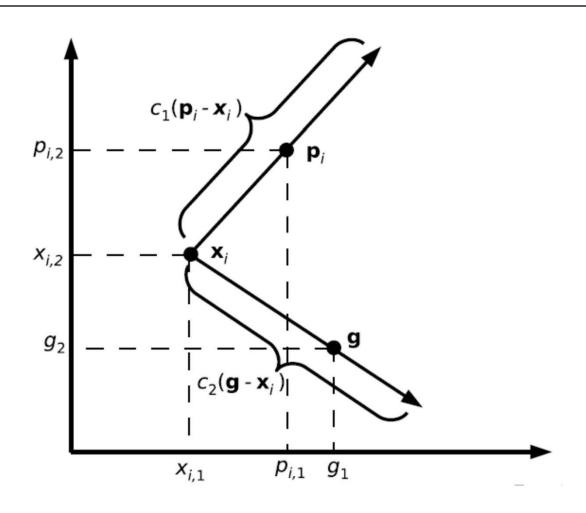


A partícula se move para a nova posição, pela combinação linear desses dois vetores, com pesos diferentes.

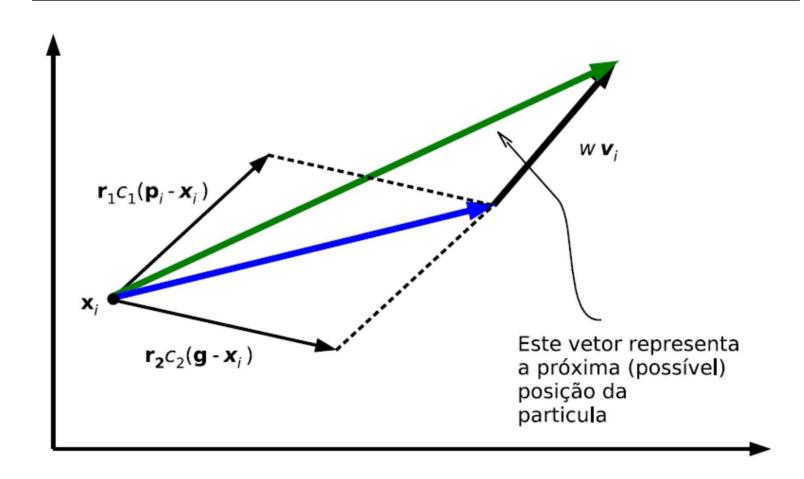


Nova posição para a partícula

Interpretação Geométrica



Interpretação Geométrica



Diversificação versus intensificação

 O algoritmo PSO fornece um mecanismo bem balanceado entre diversificação e intensificação:

$$V_{ij}^{k+1} = \underbrace{WV_{ij}^k}_{\text{diversificação}} + \underbrace{c_1r_{1j}(p_{ij}^k - X_{ij}^k) + c_2r_{2j}(g_j^k - X_{ij}^k)}_{\text{intensificação}}$$

Algoritmo PSO

```
inicialize a nuvem de partículas
repita
    para i = 1 até m
       se f(\mathbf{x}_i) < f(\mathbf{p}_i) então
            \mathbf{p}_i = \mathbf{x}_i
            se f(\mathbf{x}_i) < f(\mathbf{g}) então
               \mathbf{q} = \mathbf{x}_i
            fim se
        fim se
        para j = 1 até n
            r_1 = \text{rand}(), r_2 = \text{rand}()
            v_{ij} = wv_{ij} + c_1r_1(p_i - x_{ij}) + c_2r_2(g_i - x_{ij})
        fim para
       \mathbf{X}_i = \mathbf{X}_i + \mathbf{V}_i
    fim para
até satisfazer o critério de parada
```

Algoritmo PSO

Passo 0 (Definição das variáveis):

P o tamanho da população do PSO.

PSO[i] a posição da i-ésima partícula da população do PSO, que representa uma solução candidata para o problema.

fitness[i] o custo da função da i-ésima partícula.

V[i] a velocidade da partícula.

Gbest um índice para a melhor posição global.

P_{best}[i] a posição de melhor localização da i-ésima partícula.

P_{best_fitness}[i] o melhor fitness local visitado pela i-ésima partícula.

Passo 1 (Inicialização): Para cada partícula i na população:

Passo 1.1: Inicialize o PSO[i] randomicamente.

Passo 1.2: Inicialize V[i] randomicamente.

Passo 1.3: Avalie o fitness[i].

Passo 1.4: Inicialize Ghest

Passo 1.5: Inicialize $P_{best}[i]$ com uma cópia de $PSO[i] \forall i \leq P$.

Passo 2: Repita até que um critério de parada seja satisfeito

Passo 2.1: Encontre o G_{best} tal que $fitness[G_{best}]$ fitness[i] $i \leq P$.

Passo 2.2: Para cada partícula i:

 $P_{best}[i] = PSO[i] \text{ se fitness}[i] > P_{best \text{ fitness}}[i] \forall i \leq P.$

Passo 2.3: Para cada particular i:

Atualize V[i] e PSO[i] de acordo com as equações 1 e 2.

Passo 2.4: Avalie o $fitness[i] \forall i \leq P$

Detalhes de implementação

 Limites para as posições de uma partícula:

$$x_{ij} \in [x_{min}, x_{max}]$$

Caso x_{ii} saia deste intervalo, fazer:

Velocidade máxima:

$$V_{min} \le V \le V_{max}$$

Não é necessário armazenar **g** no computador. Basta armazenar o índice i tal que **p**_i = **g**

Melhoramentos

- Redução linear da ponderação de inércia
 - Para reduzir gradativamente a influência da diversificação
- Fator de constrição
 - Para evitar a mudança brusca de velocidade

Redução linear da ponderação de inércia

A cada iteração k a ponderação é reduzida:

$$W^{k+1} = W_{\text{max}} - k \left(\frac{W_{\text{max}} - W_{\text{min}}}{k_{\text{max}}} \right)$$

onde k_{max} é o número máximo de iterações.

Shi e Eberhart (1998) relataram que

$$W_{\text{max}} = 0, 9$$

 $W_{\text{min}} = 0, 4$
 $C_1 = C_2 = 2$

deu bons resultados em uma variedade de problemas.

Fator de constrição

Atualização de velocidade:

$$v_{ij}^{k+1} = \chi \left[v_{ij}^{k} + c_{1} r_{1j} (p_{ij}^{k} - x_{ij}^{k}) + c_{2} r_{2j} (g_{j}^{k} - x_{ij}^{k}) \right]$$

$$\chi = \frac{2\kappa}{|2 - \varphi - \sqrt{\varphi^{2} - 4\varepsilon}|}$$

onde χ é o fator de constrição, $\varphi = c_1 + c_2$, $\varphi > 4$.

Valores usuais, $\kappa = 1, \varphi = 4, 1 \Rightarrow \chi = 0,73$. $c_1 = c_2 = 2,05$.