

Particle Swarm Optimization (PSO)

Carolina Ribeiro Xavier

Junho de 2024

1 Inspiração biológica

O particle swarm optimization (PSO ou otimização por nuvem de partículas) foi desenvolvido pelo psicólogo social James Kennedy e o engenheiro eletricitista Russel Eberhart, em 1995. Originalmente, desenvolvido para resolver problemas de otimização com variáveis contínuas.

O PSO é inspirado pelo comportamento social e cooperativo exibido por várias espécies de forma a realizar as

suas necessidades no espaço de busca. Em termos gerais, o algoritmo guia-se por experiência pessoal (P_{best}) e por experiência geral (G_{best}).

O PSO resolve o problema criando uma população de soluções candidatas, também conhecidas como partículas, e movendo estas partículas em torno do espaço de busca, de acordo com fórmulas matemáticas simples sobre a posição e a velocidade da partícula. O movimento de cada partícula é influenciado pela sua melhor posição conhecida e a melhor posição conhecida por sua vizinhança.

O PSO pode simular, por exemplo, 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, o que sugere que eles compartilham informações.

O PSO é um algoritmo populacional, onde a população é chamada de "nuvem" ou "enxame" e os indivíduos são chamados de "partículas". O enxame evolui por meio de cooperação e competição entre seus membros, e as partículas se beneficiam da sua própria experiência e da experiência de outros membros do enxame durante a busca



Figura 1: Bando de pássaros

pelo alvo.

1.1 Atualização de posição e velocidade

A atualização da velocidade é dada pela Equação 1:

$$v_{ij}^{k+1} = wv_{ij}^k + c_1r_{1j}(p_{best_{ij}}^k - x_{ij}^k) + c_2r_{2j}(g_{best_j}^k - x_{ij}^k) \quad (1)$$

para $i = [1, \dots, m]$ e $j = [1m\dots, n]$.

A parte em vermelho é responsável pela diversificação da solução, enquanto a parte em azul é responsável pela intensificação.

A posição de cada partícula depende da velocidade e é dada pela Equação 2

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Como podemos observar, a nova velocidade da partícula é influenciada por três fatores:

- velocidade anterior;
- distância entre sua posição atual e melhor posição alcançada até então (cognitivo);
- distância entre sua posição atual e a melhor posição do grupo conhecido por ela (social);

A partir do cálculo dessa nova velocidade, a partícula “voa” para sua nova posição.

As Figuras em 2 exemplificam a mudança de posição de uma partícula:

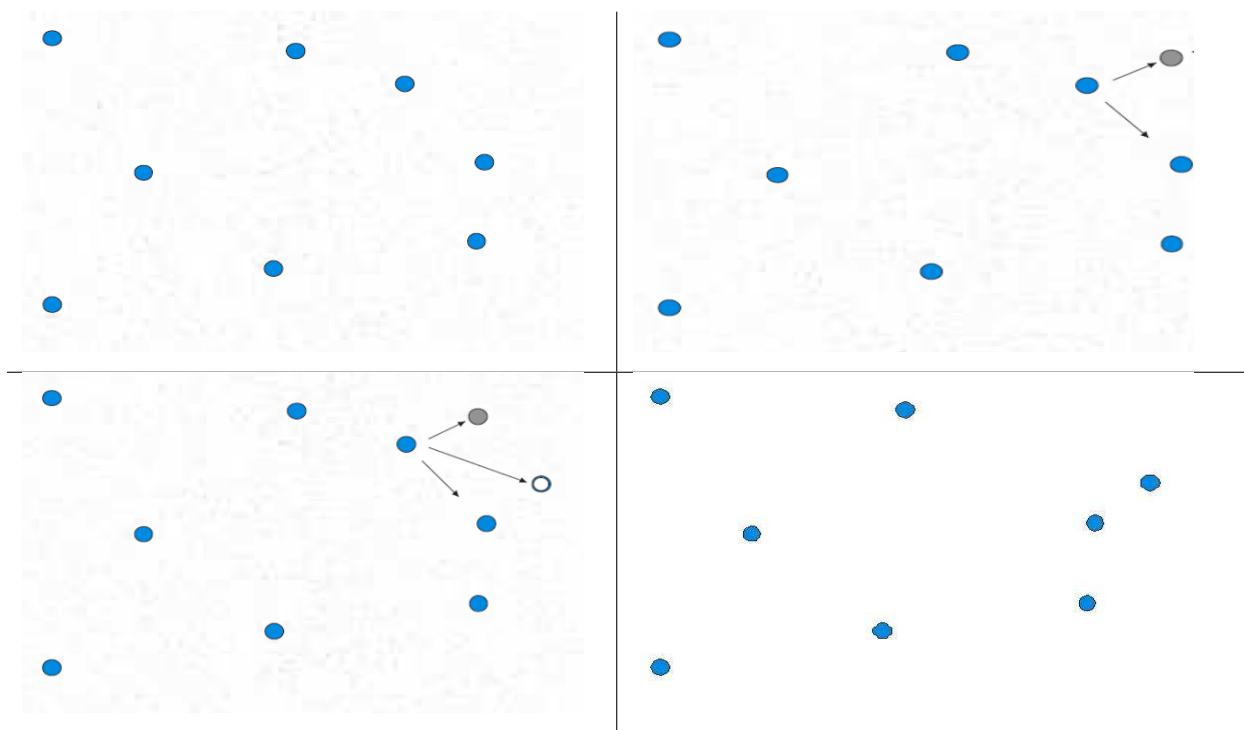


Figura 2: Exemplo de mudança de posição de uma partícula

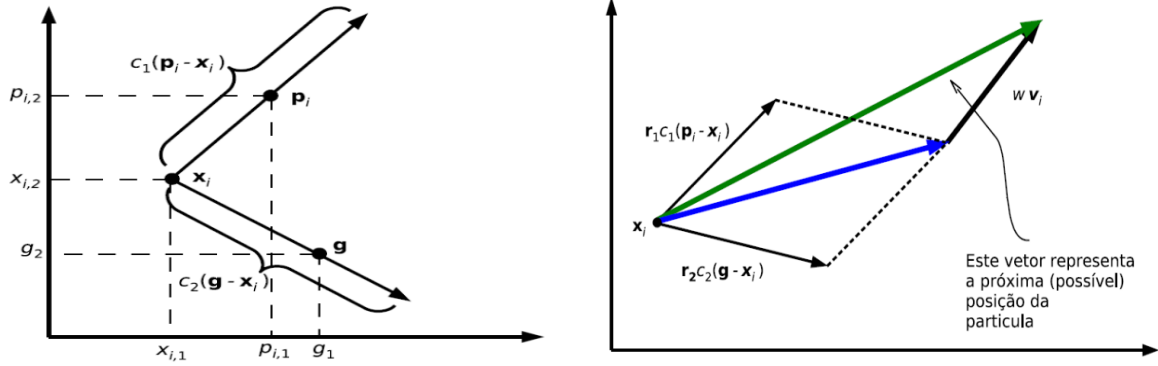


Figura 3: Interpretação geométrica da mudança de posição de uma partícula

A interpretação geométrica dessa mudança de posição é um conjunto de operações simples sobre vetores, como ilustra a Figura 3.

1.2 Pseudocódigo

Dada a dimensão da função que se deseja otimizar, teremos os vetores para representar a soluções do problema:

```
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
         $\mathbf{g} = \mathbf{x}_i$ 
      fim se
    fim se
    para  $j = 1$  até  $n$ 
       $r_1 = \text{rand}()$  ,  $r_2 = \text{rand}()$ 
       $v_{ij} = wv_{ij} + c_1r_1(p_i - x_{ij}) + c_2r_2(g_j - x_{ij})$ 
    fim para
     $\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i$ 
  fim para
até satisfazer o critério de parada
```

Figura 4: Pseudocódigo

2 Funções Benchmark

Faremos nosso tutorial para aplicação do PSO para otimizar funções benchmark de variáveis reais e contínuas.

2.1 Questões de projeto

As principais questões para o PSO estão as listadas a seguir:

- Configuração inicial;
- Função objetivo;
- Representação;
- Parâmetros;
- Atualização de posição.
- Topologia;

2.2 Configuração inicial

Cada partícula deve ter uma localização, esta pode ser aleatória ou distribuída de forma dirigida pelo espaço de busca.

2.3 Função objetivo

A função objetivo mapeia a qualidade do indivíduo diante da solução. Nosso objetivo é minimizar uma função de parâmetros reais.

2.4 Representação

Cada partícula i terá três vetores de n dimensões, onde n é igual ao número de dimensões:

$$X_i = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad V_i = \begin{bmatrix} v_0 \\ v_1 \\ \dots \\ v_n \end{bmatrix} \quad P_{best_i} = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_n \end{bmatrix}$$

sendo X o vetor de coordenadas da posição atual da partícula i , V o vetor da velocidade da partícula i e o

vetor P_{best_i} guarda as coordenadas da melhor posição conhecida pela partícula. Cada partícula possui ainda o valor de sua aptidão $f(x_i)$ e o índice da melhor partícula de sua vizinhança (topologia).

Globalmente tem-se o vetor $G_{best_i} = \begin{bmatrix} g_0 \\ g_1 \\ \dots \\ g_n \end{bmatrix}$ que guarda

as coordenadas da melhor solução visitada pelo enxame ou pela vizinhança da partícula, vai depender da topologia. Podemos ter somente o índice da partícula que acessou o melhor valor e acessar seu atributo P_{best} .

2.5 Parâmetros

- m é o tamanho do enxame, o número de partículas;
- c_1 e c_2 são parâmetros dados pelo usuário que significam respectivamente uma taxa de aprendizado cognitiva e uma taxa de aprendizado social.
- w é um parâmetro também dado pelo usuário, sendo esse uma ponderação de inércia.

2.6 Topologia

Diferentes topologias podem dar diferentes soluções para o problema que se deseja otimizar. A Figura 5 ilustra alguns exemplos. A letra (a) o conceito de melhor global é aplicado e todas as partículas levaram em conta o mesmo g_{best} , na letra (b), cada partícula só "enxerga" as melhores posições de seus vizinhos imediatos, na letra (c) cada partícula possui três vizinhos, entanto na letra (d), as partículas internas possuem 4 vizinhos, e as das bordas possuem dois os três vizinhos.

Quanto mais vizinhos em comum as partículas tiverem, mais rápida será a convergência

2.7 Critério de parada

- Número máximo de iterações
- Estagnação (mínimo local)
- Convergência (mínimo global)

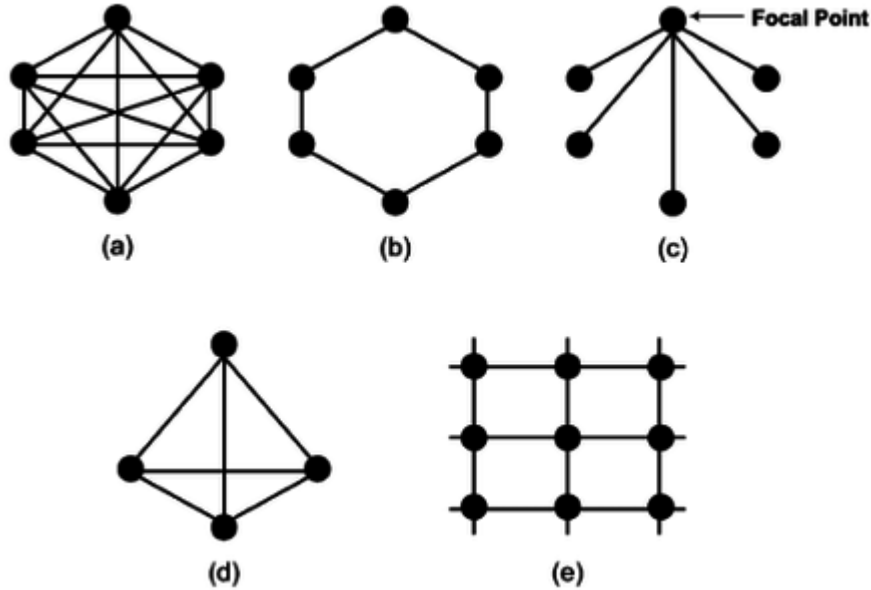


Figura 5: Topologias

3 Implementação

Alguns detalhes para implementação do PSO:

De acordo com a função objetivo deve-se estabelecer limites para as posições de uma partícula $x_{ij} \in [x_{min}, x_{max}]$

Caso x_{ij} saia deste intervalo, recuar para o limite extrapolado:

no caso de obtenção de valores inferiores a x_{min} , $x_{ij} = x_{min}$, em caso de obtenção de valores superior ao x_{max} ,

$$x_{ij} = x_{max}.$$

Pode-se estabelecer os limites de velocidade:

$$v_{min} \leq v \leq v_{max}$$

Agora vamos implementar um algoritmo baseado em enxame de partículas.

Escolha uma função do artigo para otimizar, no artigo tem as soluções para auxiliar na avaliação dos resultados. Não use funções com somente 1 ou 2 dimensões para que fique "mais difícil" de encontrar o valor ótimo e você possa explorar mais os parâmetros dos algoritmos.

Você deve definir a estrutura de dados que você irá armazenar suas partículas, suas soluções, as soluções de sua vizinhanças e seus respectivos valores de *fitness*.

As operações com vetores são simples e existem diversas bibliotecas que podem te ajudar nisso.

Defina 3 dos valores listados a seguir com um experimento fatorial:

- w (fator de diversificação);
- c_1 (intensificação, fator de cognitivo);
- c_2 (intensificação, fator de social);

- m (número de partículas);
- topologias;
- k número máximo de iterações.