

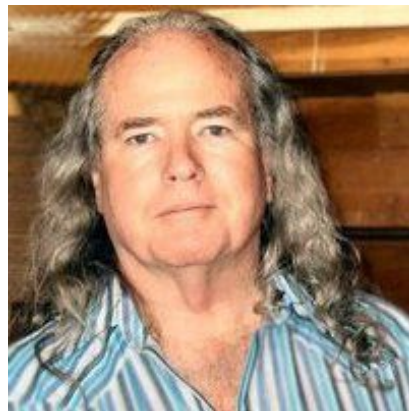
OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO)

Filipe Gomes Arante de Souza

Metaheurísticas 2023/2
Prof. Maria Claudia Silva Boeres

Introdução

- Criado por James Kennedy e Russel Eberhart em 1995;
- Surgiu de estudos oriundos de simulações de modelos sociais simplificados de um bando de pássaros;
- Aplicável em problemas de telecomunicações, controle, mineração de dados, design, otimização combinatória, sistemas de energia, processamento de sinais, ...;
- Um dos algoritmos populacionais mais conhecidos e amplamente utilizados atualmente;

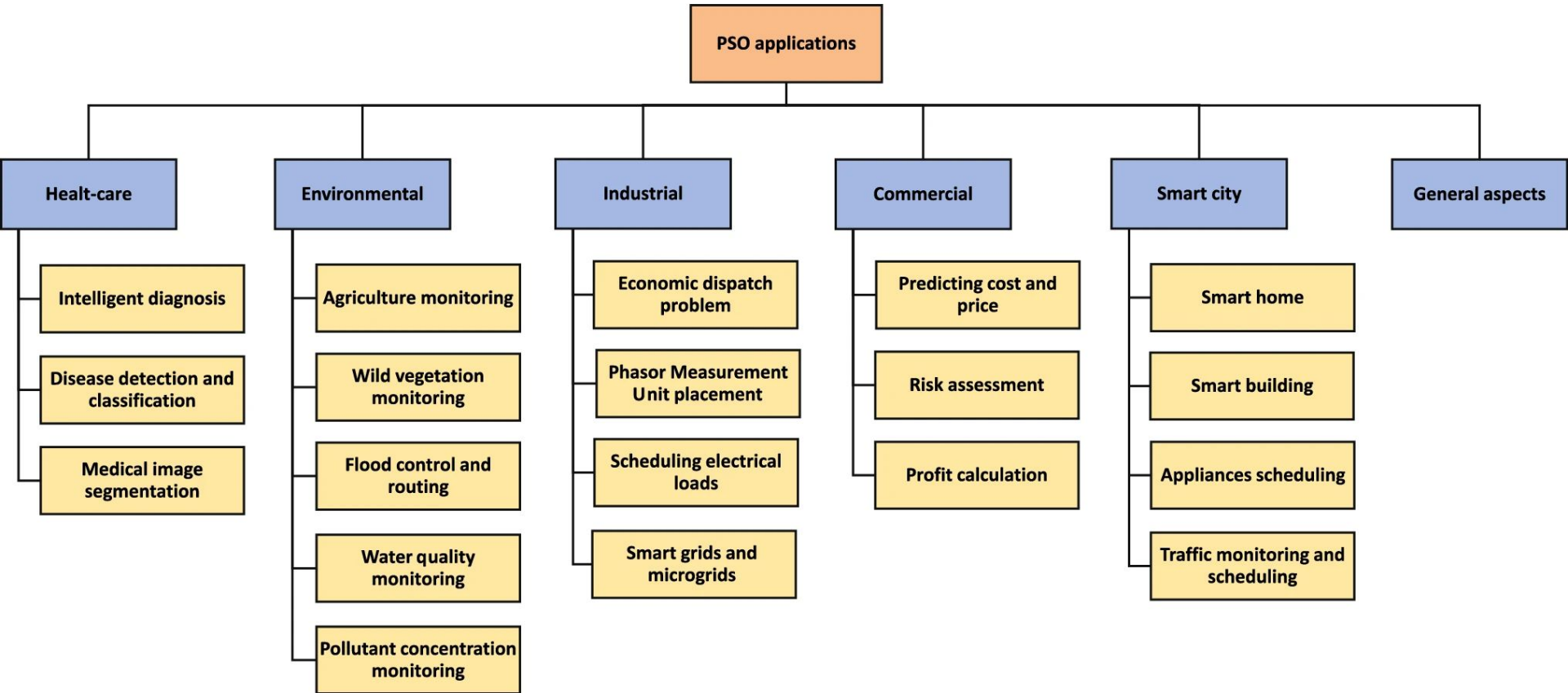


James K, Psicólogo Social



Russel E, Eng. Elétrico

Aplicações



Estado da Arte

- **Zhong Liu e Lei Huang em 2010:** resolvem o problema do Caixeiro Viajante com uma versão discreta do algoritmo, nomeada MDPSO, apresentando uma nova definição para a velocidade baseada no Algoritmo Genético;
- **Yannis Marinakis, Athanasios Migdalas e Angelo Sifaleras em 2017:** resolvem o problema do Caminho Mais Curto Restrito utilizando o PSO junto ao VNS para um cálculo otimizado da posição das partículas;
- **Jian Jiao, Seyed-mohsen Ghoreishi, Zohre Moradi e Khaled Oslub em 2022:** fazem uma análise do desempenho estático-dinâmico e ondas propagadas na nanoplaca magneto-eletro-elástica utilizando o PSO como um dos operadores do Algoritmo Genético;

Histórico

- Interesse em compreender o movimento em grupo de espécies de animais;
- Como formalizar matematicamente o movimento de uma partícula individual em função do bando?
- “O comportamento de cada um se baseava numa função dos esforços individuais para manter uma distância ideal entre eles mesmos e seus vizinhos”;
- Simulações computacionais foram feitas;



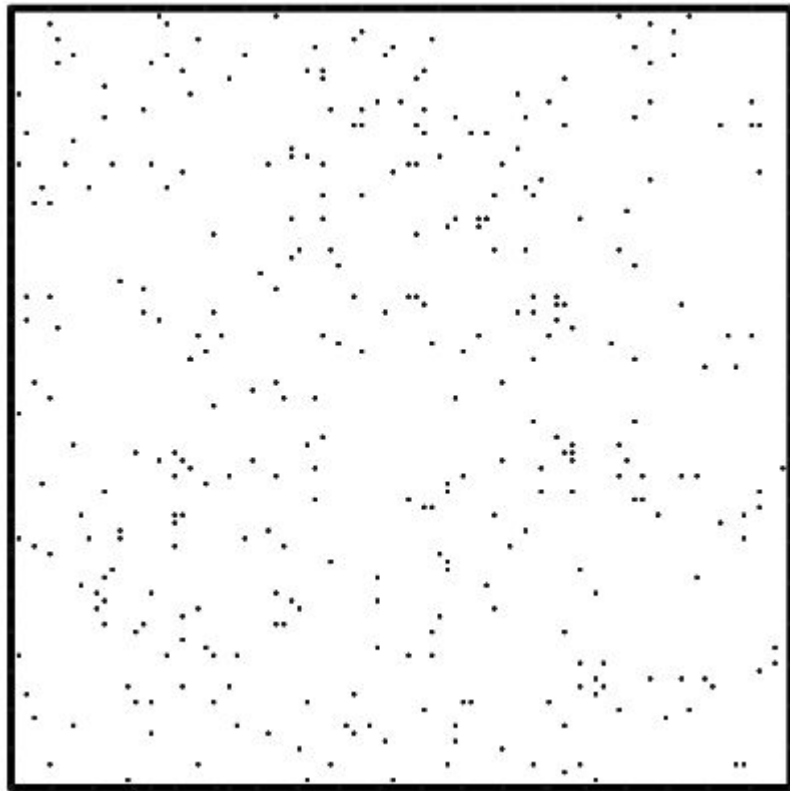
Bando de Pássaros



Cardume de Peixes

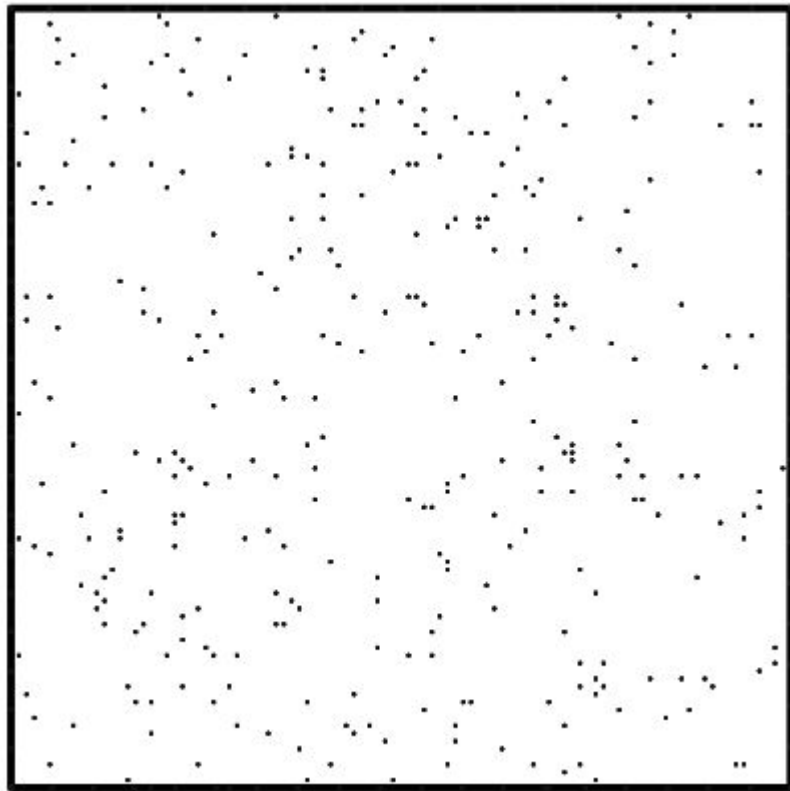
Histórico

- Primeira simulação: Correspondência de Velocidade do Vizinho Mais Próximo;
- Inicializar os pontos aleatoriamente numa grade de pixels, cada um com velocidade (v_x, v_y) ;
- A cada iteração os pontos adquirem a velocidade do vizinho mais próximo;
- Resultado: rapidamente o bando inteiro ficava numa direção uniforme;



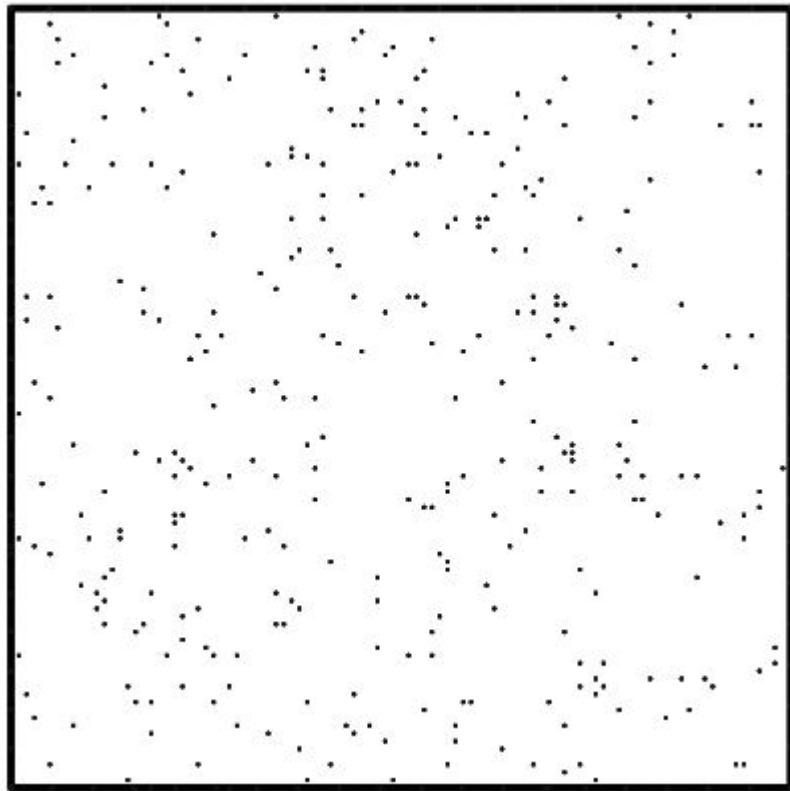
Histórico

- Segunda simulação: Correspondência de Velocidade e **Loucura** do Vizinho Mais Próximo;
- Inicializar os pontos aleatoriamente numa grade de pixels, cada um com velocidade (v_x, v_y) ;
- A cada iteração os pontos adquirem a velocidade do vizinho mais próximo;
- **É feita alguma alteração estocástica na velocidade de cada partícula;**
- Resultados melhores;



Histórico

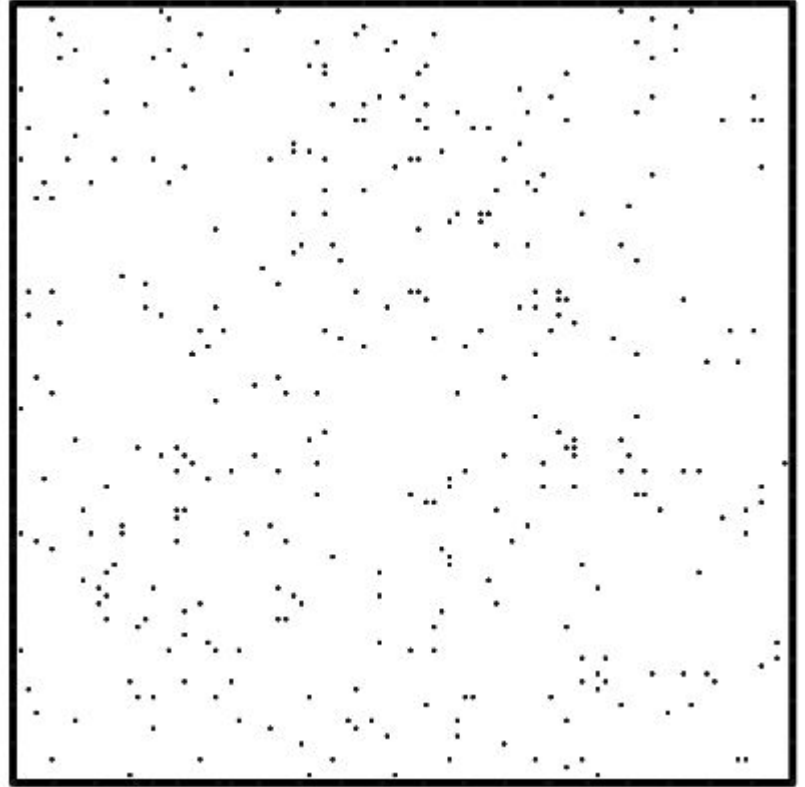
- Por que os pássaros andam em conjunto?
- R: Obter alimento, fugir de predadores, ...;
- Simulações anteriores apenas movimentam os pontos na grade de pixels;
- Colocar um objetivo para o bando;
- Como os pássaros procuram comida?
- Quando um encontra, como os outros são influenciados?



“Os membros individuais do bando podem aproveitar descobertas e experiências anteriores de todos os outros membros para alcançar os seus objetivos”

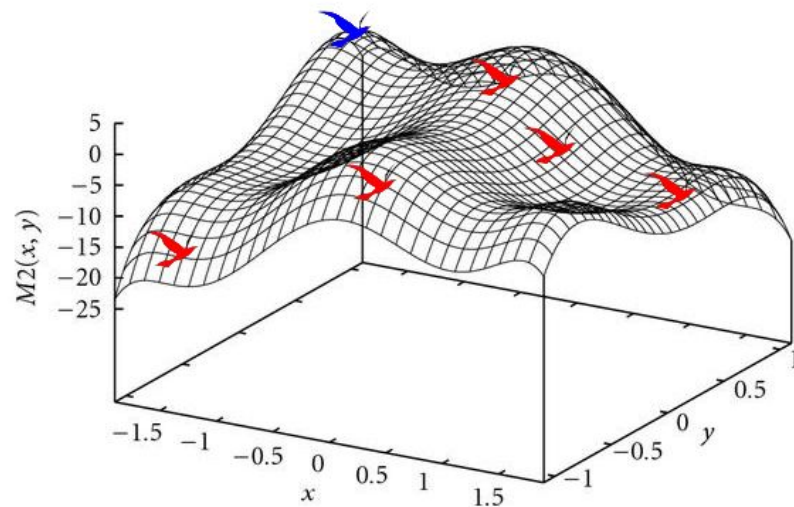
Histórico

- Estudo se torna um problema de otimização;
- Ideia do poleiro: O bando voa ao redor de um dado ponto (x, y) que possui alimentos;
- Cada um deles possui memória de sua melhor posição já percorrida e da melhor posição do bando já encontrada;
- Função objetivo depende do ponto (x, y) ;
- Problema: na prática não sabemos onde está o ponto ótimo;



Definição do PSO

- Algoritmo heurístico baseado no comportamento social de um bando de pássaros;
- Procura a solução ótima num determinado espaço de busca através da troca de informações entre indivíduos de uma população;
- Funciona para d dimensões;



Movimento das Partículas

$$v_{k+1} = w \cdot v_k + c_1 \cdot r_1 \cdot (p_{best_k} - x_k) + c_2 \cdot r_2 \cdot (g_{best} - x_k)$$

$$x_{k+1} = x_k + v_k$$

Pseudocódigo do PSO

Algorithm 1 Particle Swarm Optimization

```
1: function PSO()
2:    $S \leftarrow \text{INITIALIZE\_SWARM}()$ 
3:
4:   while (stop condition is not met) do
5:     for  $\forall$  particle  $p \in S$  do
6:       if  $f(p) < f(best_p)$  then
7:          $best_p \leftarrow p$ 
8:       end if
9:
10:      if  $f(best_p) < f(best_{swarm})$  then
11:         $best_{swarm} \leftarrow best_p$ 
12:      end if
13:    end for
14:
15:     $\text{UPDATE\_SWARM\_VELOCITIES}(S)$ 
16:     $\text{UPDATE\_SWARM\_POSITIONS}(S)$ 
17:  end while
18:
19:  return  $best_{swarm}$ 
20: end function
```

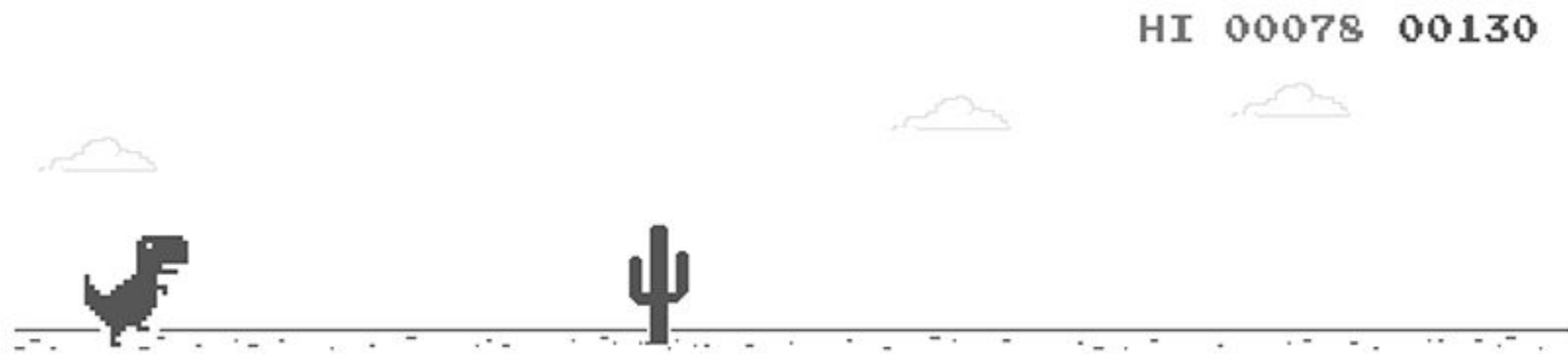
Observações

- A versão clássica do PSO é melhor para problemas de otimização contínua;
- Movimento das partículas frequentemente geram soluções inválidas;
- Deve passar por uma série de adaptações para ser capaz de resolver problemas discretos;



Exemplo de Aplicação: Chrome Dino

- Classificador de árvore de decisão controla as ações do dinossauro;
- PSO ajusta os parâmetros de entrada do classificador;



Exemplo de Aplicação: Chrome Dino

- Ideia do classificador: determinar o melhor momento para pular um obstáculo e o melhor momento para abaixar quando se está no ar;
- Ideia: parâmetros α e β ;
- Coeficientes de proporcionalidade entre as distâncias limite;



Árvore de Decisão



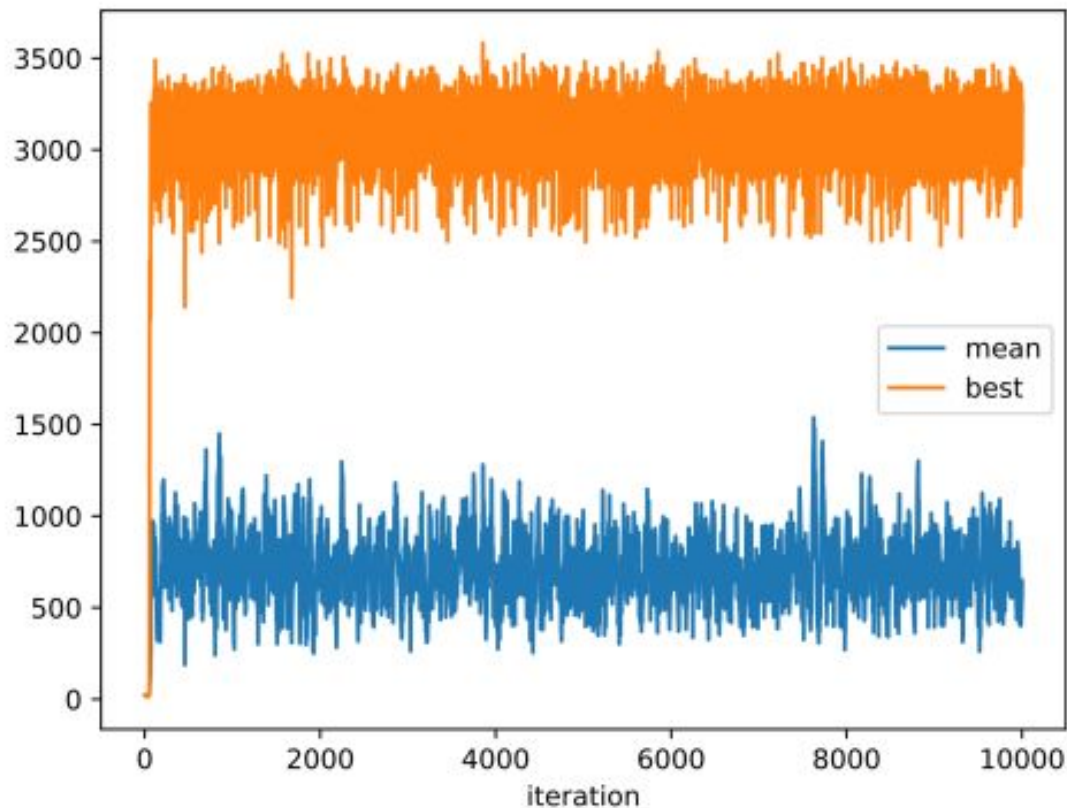
Exemplo de Aplicação: Chrome Dino

- Valores utilizados:

$$w = 0.08, \quad c_1 = 0.2, \quad c_2 = 0.05$$

- **Quantidade iterações:** 10.000;
- **Tamanho da população:** 50;
- **Função objetivo:** média da pontuação de 5 corridas;
- **Tempo de execução:** 8 horas;
- **Solução encontrada pelo PSO:** $\alpha = 23,891$ e $\beta = 6,298$;
- **Média da pontuação em 30 corridas:** 3.290,54;
- **Desvio padrão em 30 corridas:** 256,43

Iteração vs Melhor Pontuação / Média da população



Resultado Final

https://www.youtube.com/watch?v=YqOPyTOzwjo&ab_channel=FilipeGomes

Referências

- [1] J. Kennedy and R. Eberhart. Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, volume 4, pages 1942–1948 vol.4, 1995.
- [2] Zhong Liu and Lei Huang. A mixed discrete particle swarm optimization for tsp. In 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), volume 2, pages V2–208–V2–211. IEEE, 2010.
- [3] Yannis Marinakis, Athanasios Migdalas, and Angelo Sifaleras. A hybrid particle swarm optimization – variable neighborhood search algorithm for constrained shortest path problems. European journal of operational research, 261(3):819–834, 2017.

Referências

- [4] Jian Jiao, Seyed-mohsen Ghoreishi, Zohre Moradi, and Khaled Oslub. Coupled particle swarm optimization method with genetic algorithm for the static–dynamic performance of the magneto-electro-elastic nanosystem. *Engineering with computers*, 38(Suppl 3):2499–2513, 2022.
- [5] Diogo Freitas, Luiz Guerreiro Lopes, and Fernando Morgado-Dias. Particle swarm optimisation: A historical review up to the current developments. *Entropy*, 22(3), 2020.
- [6] M. Dorigo, M. A. Montes de Oca, and A. Engelbrecht. Particle swarm optimization. *Scholarpedia*, 3(11):1486, 2008. revision #91633.
- [7] Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, 29, 04 2022.

OBRIGADO!
