

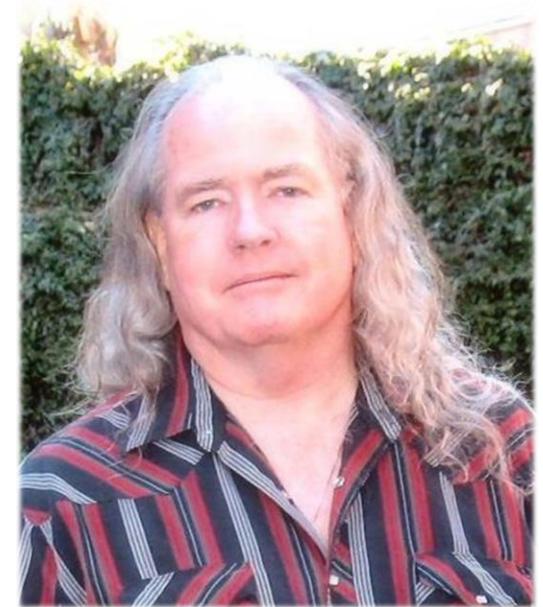
Particle Swarm Optimization (PSO)



Fabricio Breve – fabricio.breve@unesp.br
<https://www.fabriciobreve.com>

Particle Swarm Optimization

- Otimização por Enxame de Partículas
- Proposta em 1995 por:
 - James Kennedy: psicólogo social
 - Russ Eberhart: engenheiro eletricista
- Surgiu a partir de experimentos que modelavam o comportamento de enxames.
 - Visto em várias espécies de pássaros e de peixes.
 - Objetivo inicial era estudar o comportamento social e cognitivo de animais.



James Kennedy [*1950]



Russell C. Eberhart [*1950]

Kennedy, J., & Eberhart, R. (1995, November). **Particle swarm optimization**. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.

Modelagem do Comportamento de Enxames

- **Motivação:**
 - Criar uma simulação do comportamento social.
- **População de indivíduos capazes de interagir:**
 - Com o ambiente.
 - Uns com os outros.
 - Particularmente com um conjunto de vizinhos.



Bandos de Pássaros

- Vários modelos de movimento de organismos em bandos e cardumes foram investigados.
 - Craig Reynolds estudou coreografia dos bandos de pássaros.

Reynolds, C. W. (1987, August). **Flocks, herds and schools: A distributed behavioral model.** In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (pp. 25-34).

Craig W. Reynolds, é um especialista em vida artificial e computação gráfica, que criou a simulação de vida artificial Boids em 1986. Reynolds trabalhou no filme *Tron* como programador de cena e em *Batman Returns* como parte da equipe de imagens de vídeo.

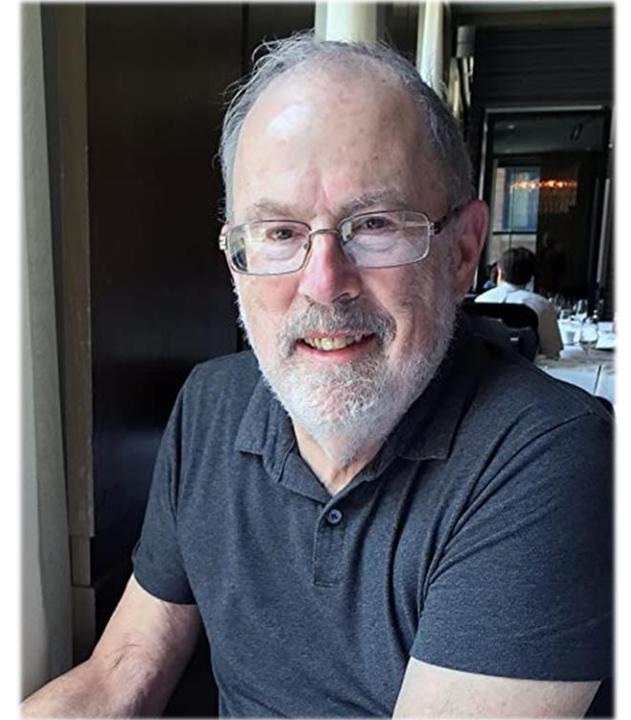


Craig W. Reynolds [*1953]



Bandos de Pássaros

- O biólogo Frank Heppner procurou as regras que possibilitam vôo sincronizado de grandes bandos de pássaros, mesmo com:
 - Pássaros mudando de direção subitamente.
 - Pássaros se espalhando e re-agrupando.



Frank H. Heppner [*1940]

Heppner, H., & Grenander, U. (1990). **A stochastic non-linear model for coordinated bird flocks.** In S. Krasner (Ed.), *The ubiquity of chaos* (pp. 233–238). Washington: AAAS

Bandos de Pássaros

- A Lição dos Gansos
 - Estudo dos motivos que levam os gansos a voarem em forma de “V” levaram a 5 descobertas.



Lessons from the Geese – by Robert McNeish

<https://web.archive.org/web/20220310115801/http://suewidemark.com/lessonsgeese.htm>

A Lição dos Gansos

- Descuberta I
 - À medida em que cada pássaro bate suas asas, ele cria sustentação para a ave seguinte.
 - Voando em formação “V”, o grupo consegue voar pelo menos 71% a mais do que se cada pássaro voasse isoladamente.



A Lição dos Gansos

- Descoberta 2
 - Sempre que um ganso sai fora da formação “V”, ele sente uma forte resistência ao tentar voar sozinho.
 - Volta rapidamente à formação anterior.
 - Por se beneficiar do poder de sustentação do(s) pássaro(s) imediatamente à sua frente.



A Lição dos Gansos

- Descoberta 3
 - Quando o ganso líder se cansa, ele vai para o fundo do “V”.
 - Enquanto um outro ganso assume a liderança.



A Lição dos Gansos

- Descoberta 4
 - Os gansos de trás grasnam para encorajar os da frente e manter o ritmo e a velocidade.



A Lição dos Gansos

- Descoberta 5
 - Quando um ganso adoece ou se fere e deixa o grupo, dois outros gansos saem da formação e acompanham o ganso doente até que a dificuldade seja superada.
 - Mais tarde, os três juntos reiniciam a jornada ou se unem a uma outra formação, até reencontrarem o antigo grupo.

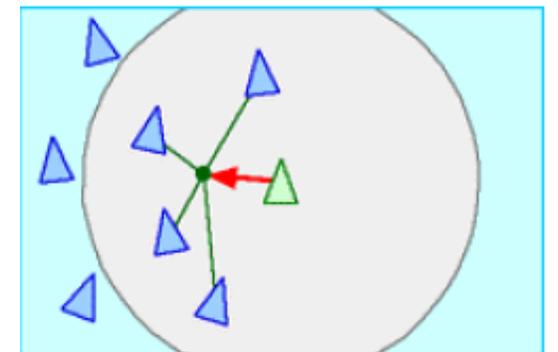
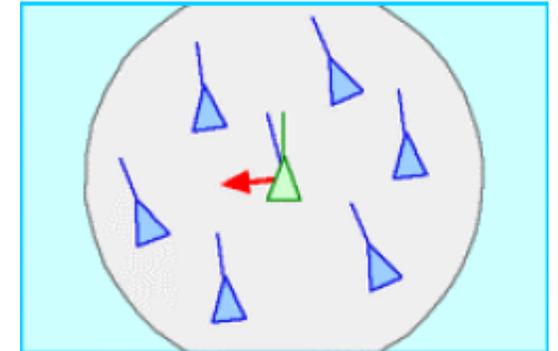
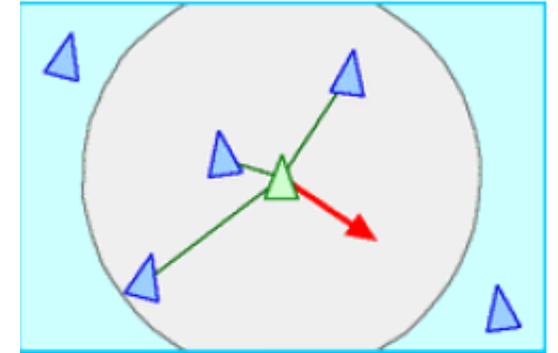


Boids



Reynolds, C. W. (1987, August). **Flocks, herds and schools: A distributed behavioral model.** In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (pp. 25-34).

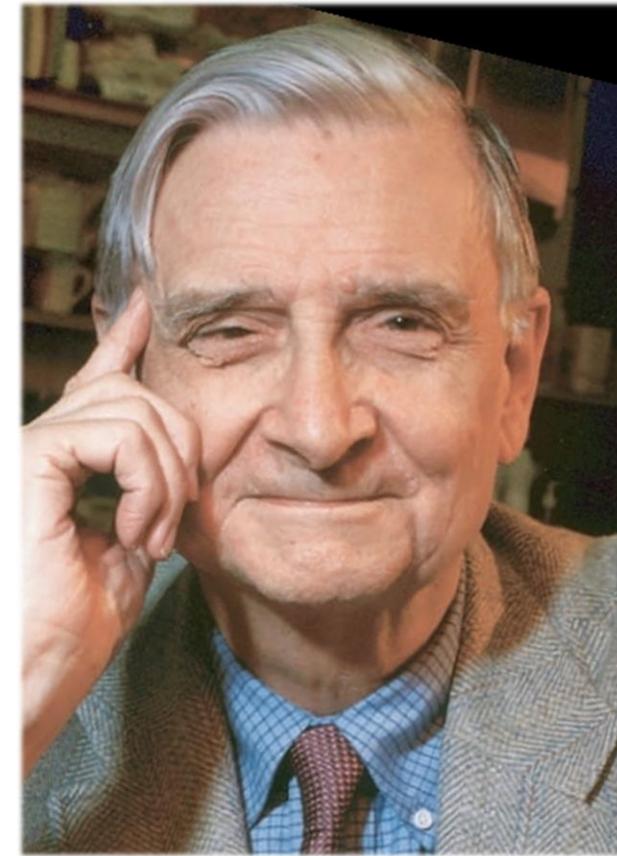
- Lembrando:
 - **Boids** é um programa de vida artificial, desenvolvido por Craig Reynolds em 1986.
 - Simula o comportamento de bandos de pássaros.
- Cada pássaro segue regras simples:
 - Separação:
 - Evitar colisão com outros pássaros do bando.
 - Alinhamento:
 - Voar no mesmo sentido dos demais pássaros do bando.
 - Coesão:
 - Voar em direção à posição média do bando.



Cardumes

Edward O. Wilson foi um renomado entomólogo e biólogo norte-americano conhecido por seu trabalho com ecologia, evolução e sociobiologia.

- De acordo com Edward Osborne Wilson:
 - “Pelo menos em teoria, membros individuais de um cardume podem se beneficiar das descobertas e experiências anteriores de outros membros do cardume na busca por alimentos.”
 - “Essa vantagem pode se tornar decisiva, compensando as desvantagens da competição por alimentos.”
- Compartilhamento de informação dentro de uma espécie gera uma vantagem evolutiva.
 - Hipótese fundamental para o desenvolvimento do PSO.



Edward Osborne Wilson [*1929 † 2021]



Bandos de Pássaros e Cardumes de Peixes

- Peixes e pássaros ajustam seu movimento físico para:
 - Evitar predadores.
 - Procurar por alimento e companheiro(a).
 - Otimizar parâmetros ambientais.
 - Temperatura.



Particle Swarm Optimization



- Principio Básico
 - Comportamento é governado por regras semelhantes em todas as sociedades.
 - Compartilhamento de informação entre os indivíduos pode oferecer alguma vantagem.
- PSO
 - Indivíduos aprendem com sua própria experiência e com a experiência dos outros.
 - Se avaliam e se comparam aos seus vizinhos e imitam apenas os vizinhos superiores a eles próprios.

Particle Swarm Optimization

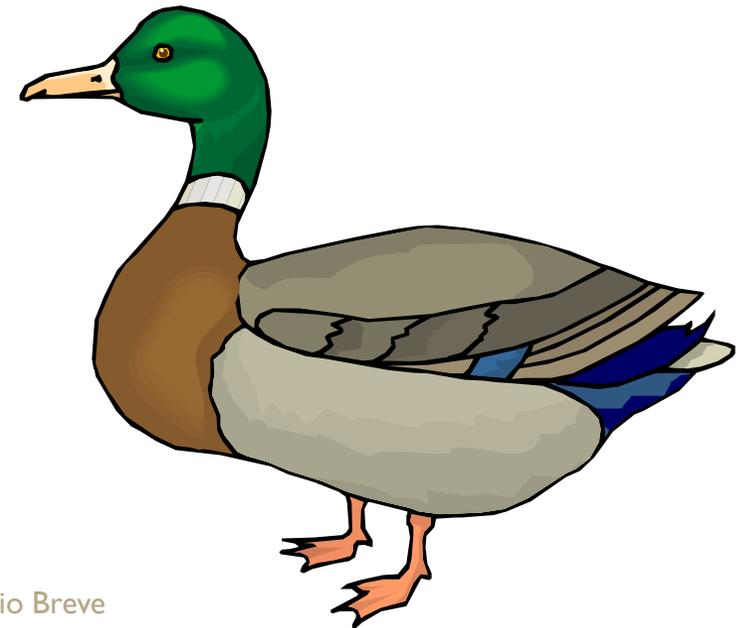
- Tem sido aplicado com sucesso a vários problemas de busca e otimização.
 - Engenharia
 - Computação
- Procedimento de otimização numérica.
 - Busca num espaço l -dimensional de valores reais.
 - Reveja slides de computação evolutiva sobre busca no espaço de possíveis soluções.



Princípios de Adaptação Cultural

- Cada indivíduo:
 - Tem sua própria experiência e sabe quão boa ela é.
 - Tem algum conhecimento de como os indivíduos de sua vizinhança tem se desempenhado.
- Corresponde a:
 - Aprendizado individual.
 - Transmissão cultural ou social.

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers. San Francisco.



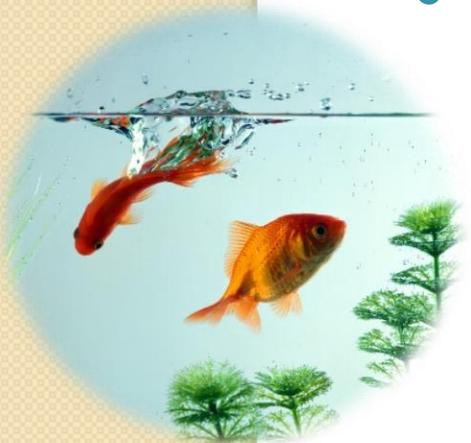
Princípios de Adaptação Cultural

- Probabilidade de um indivíduo tomar uma decisão depende de:
 - Quanto sucesso ele teve com essa decisão no passado.
 - Influências sociais.
- Algoritmo PSO
 - Indivíduos são influenciados pelo sucesso de sua vizinhança social.



Princípios de Adaptação Cultural

- **Avaliação:**
 - Capacidade de um indivíduo “sentir” o ambiente e quantificar um grau de quão bom ele é em relação a algum parâmetro ou tarefa.
- **Comparação:**
 - Pessoas usam outros como padrões para se avaliar, o que serve como motivação para aprender e mudar.
- **Imitação:**
 - Compreende tomar a perspectiva de outra pessoa, não apenas imitando um comportamento, mas também entendendo seus propósitos.



Particle Swarm Optimization

- Baseado em modelos desenvolvidos por Heppner
 - Exibe os mesmos comportamentos para bandos de pássaros encontrados em outros modelos.
- Apresenta uma característica adicional:
 - Os pássaros são atraídos para uma área de abrigo (poleiro).



Particle Swarm Optimization

- Supor um grupo de pássaros procurando por um abrigo.
 - Procura ocorre de forma aleatória.
 - Existe apenas um abrigo.
 - Nenhum pássaro sabe onde está o abrigo.
 - Mas todos sabem, a cada instante de tempo, quão longe estão do abrigo.
 - Qual a melhor estratégia para encontrar o abrigo?

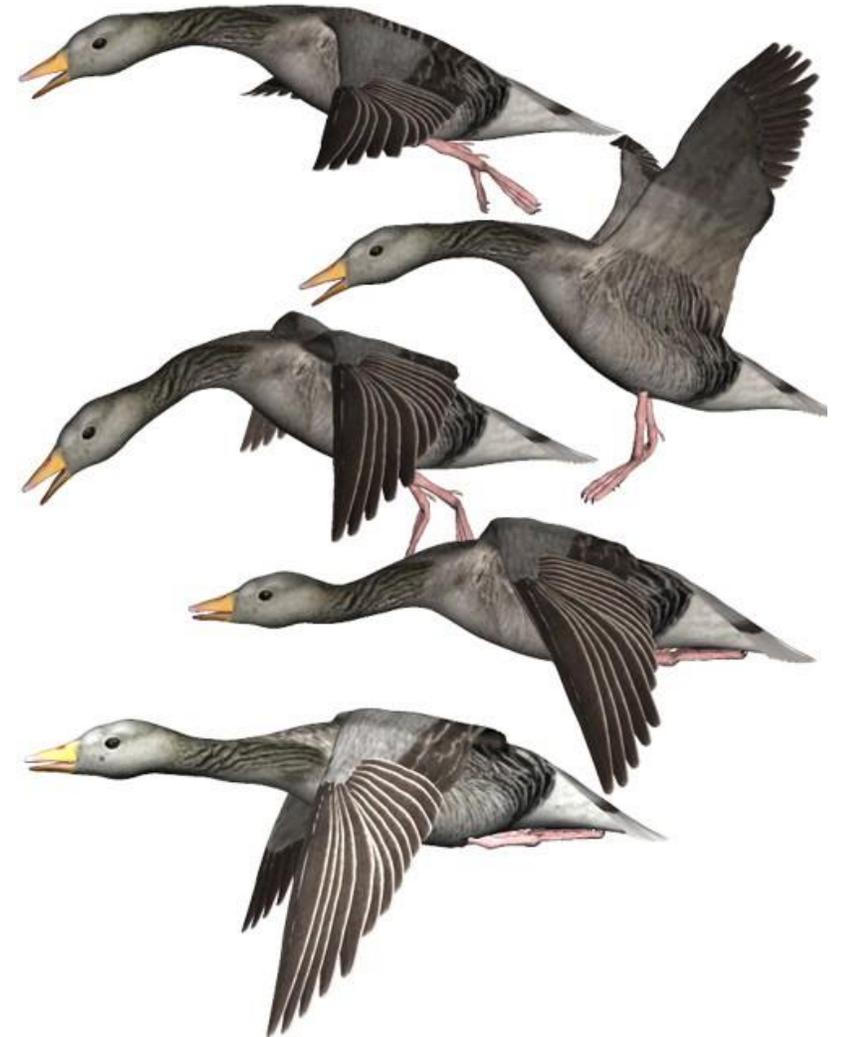


Particle Swarm Optimization

- Possível estratégia:
 - Seguir o pássaro que está mais próximo do abrigo.
 - Estratégia usada pelo PSO.



Bando de Pássaros em Brisbane, Queensland, Austrália – por Fabricio Breve



Particle Swarm Optimization

- Nas simulações realizadas:
 - Cada solução = um pássaro voando no espaço de busca.
 - Os pássaros inicialmente voavam ao acaso.
 - Sem um destino definido.
 - Espontaneamente, os pássaros formavam bandos.
 - Até um dos pássaros sobrevoar um abrigo (ou uma fonte de alimentos).
 - Se o desejo de pousar tivesse um valor maior que o desejo de permanecer voando (valores programados), o pássaro deixava o bando e pousava.

Particle Swarm Optimization

- Nas simulações realizadas (cont.):
 - Pássaros usavam regras simples para definir sua direção e velocidade.
 - Cada pássaro tentava:
 - Ficar no meio dos pássaros mais próximos.
 - Evitar colidir com esses pássaros.
 - Como resultado:
 - Quando um pássaro saía do bando para ir ao abrigo, pássaros próximos acompanhavam o movimento.
 - Cada vez mais pássaros desciam no abrigo.
 - Até que o bando inteiro tivesse descido.

Particle Swarm Optimization

- Encontrar um abrigo é análogo a encontrar uma solução no universo de possíveis soluções.
- Maneira como o pássaro que encontrou o abrigo lidera seus vizinhos em direção a ele aumenta a chance desses também encontrarem o abrigo.
 - Encontra eco na definição sócio-cognitiva de mente.
 - Que a mente, e portanto a inteligência, é social.



Particle Swarm Optimization

- Partículas podem voar sobre o espaço de soluções e pousar na melhor solução.
 - Como evitar que as partículas pousem em qualquer solução?
 - E não apenas na melhor?
 - Indivíduos aprendem a partir do sucesso de seus vizinhos
 - Princípio importante da visão sócio-cognitiva.
 - Como implementar este princípio em um algoritmo?



Partículas

- Não têm inteligência.
- Voam através das coordenadas de um espaço n -dimensional.
- Quando se movem, mandam suas coordenadas para uma função de aptidão.
 - Mede aptidão da partícula.
 - Relacionada ao problema investigado.



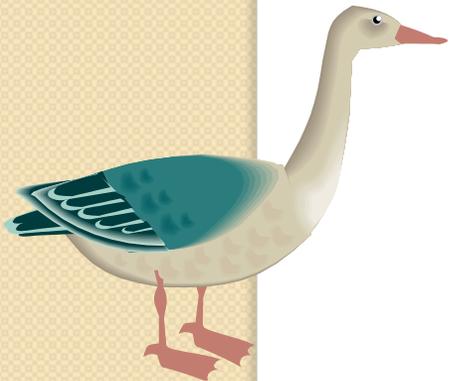
Partículas

- Uma partícula lembra de:
 - Suas coordenadas atuais.
 - Sua velocidade atual.
 - Quão rápido está se movendo ao longo das dimensões do espaço de soluções.
 - Seu melhor valor de aptidão até então recebido.
 - Coordenadas de onde foi calculado seu melhor valor de aptidão.



Partículas

- Além disso, cada partícula tem acesso às seguintes informações:
 - Melhor posição encontrada até o momento por uma partícula em sua vizinhança no espaço de soluções.
 - Aptidão desta solução.



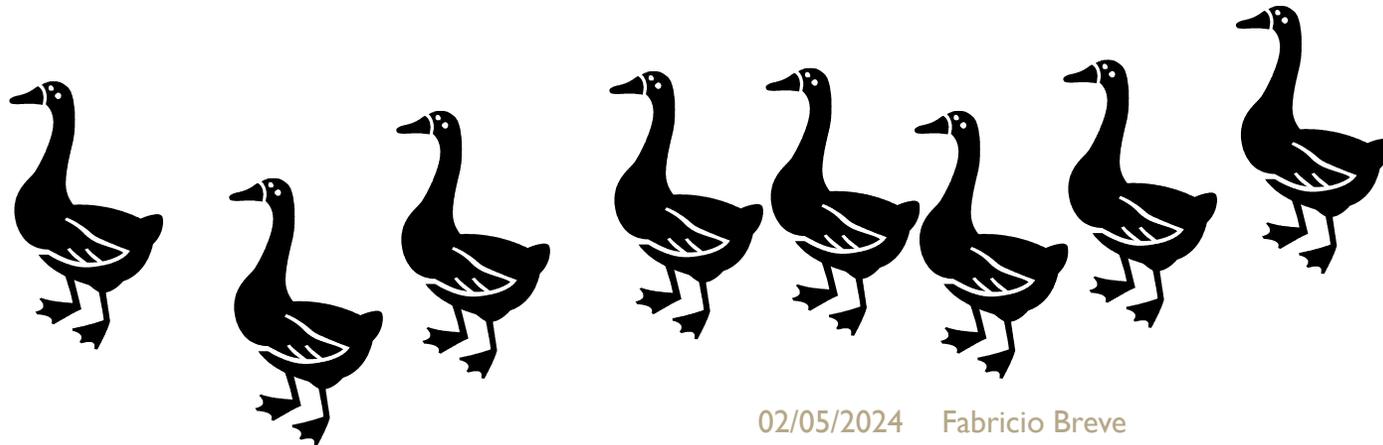
Partículas

- A cada iteração, cada partícula tem seus parâmetros atualizados usando:
 - Melhor solução obtida até o momento pela partícula.
 - Melhor solução obtida até o momento por qualquer partícula da sua vizinhança.



Vizinhança

- Não está relacionada à proximidade física dos indivíduos no espaço de soluções.
 - Vizinhanças são definidas antes das partículas iniciarem a caça.
 - Buscas dentro de uma vizinhança podem ocorrer em áreas distintas do espaço de soluções.



Vizinhança

- Quem são os vizinhos de uma partícula?
 - Sucesso dos vizinhos irá influenciar sucesso da partícula.
 - Veremos algumas alternativas...



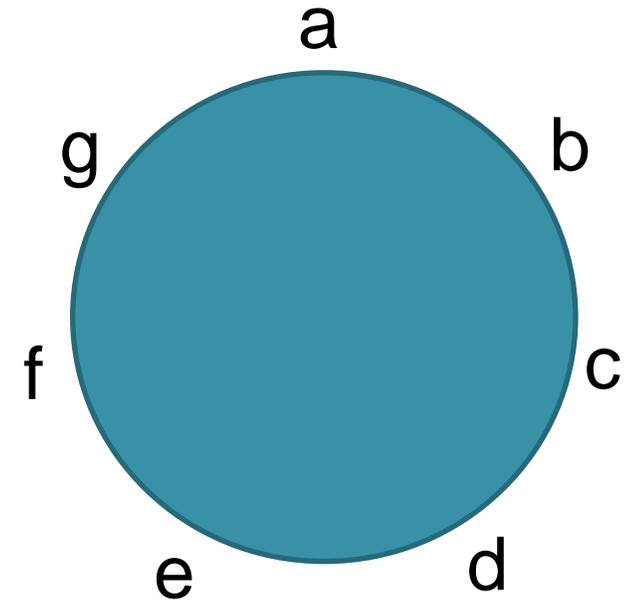
Vizinhança Global

- Tratar todas as partículas como vizinhas.
- Aparentemente boa.
 - Mas reduz chance de convergência para solução ótima em vários problemas.
- Uso de vizinhanças menores, com sobreposição, tem sido mais efetivo.

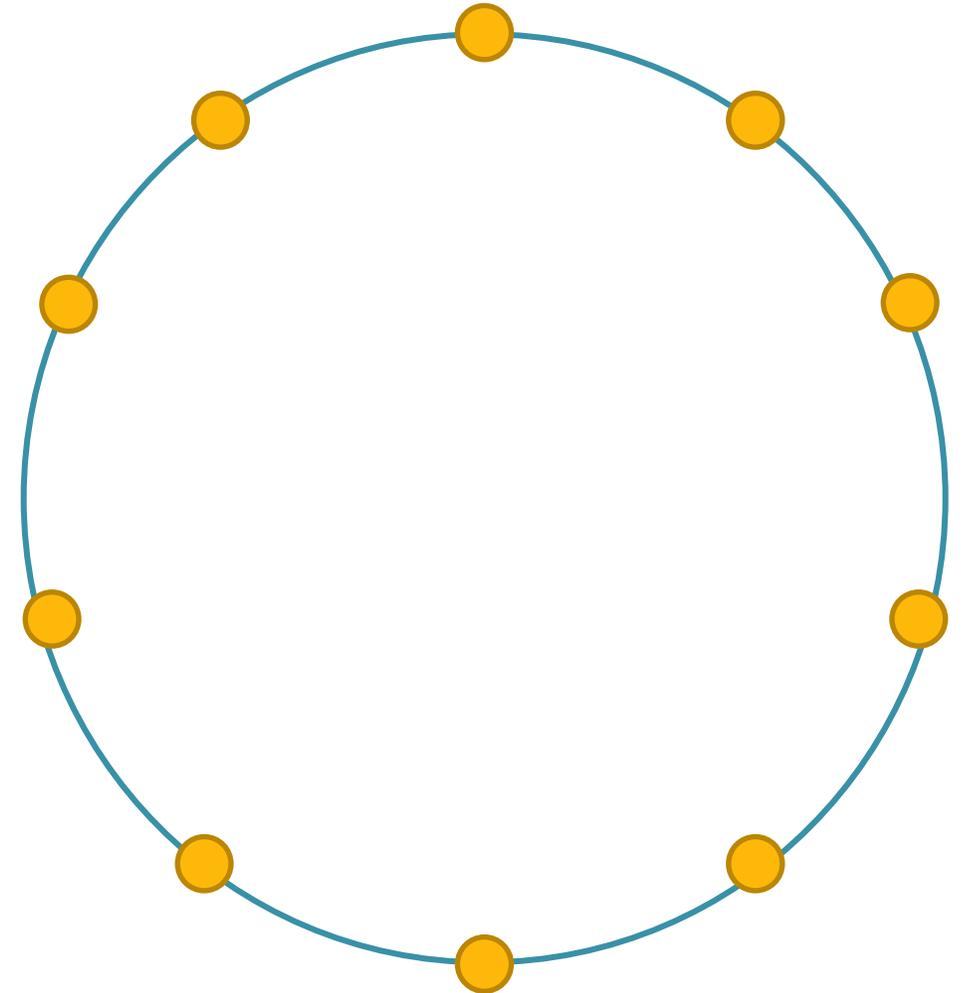
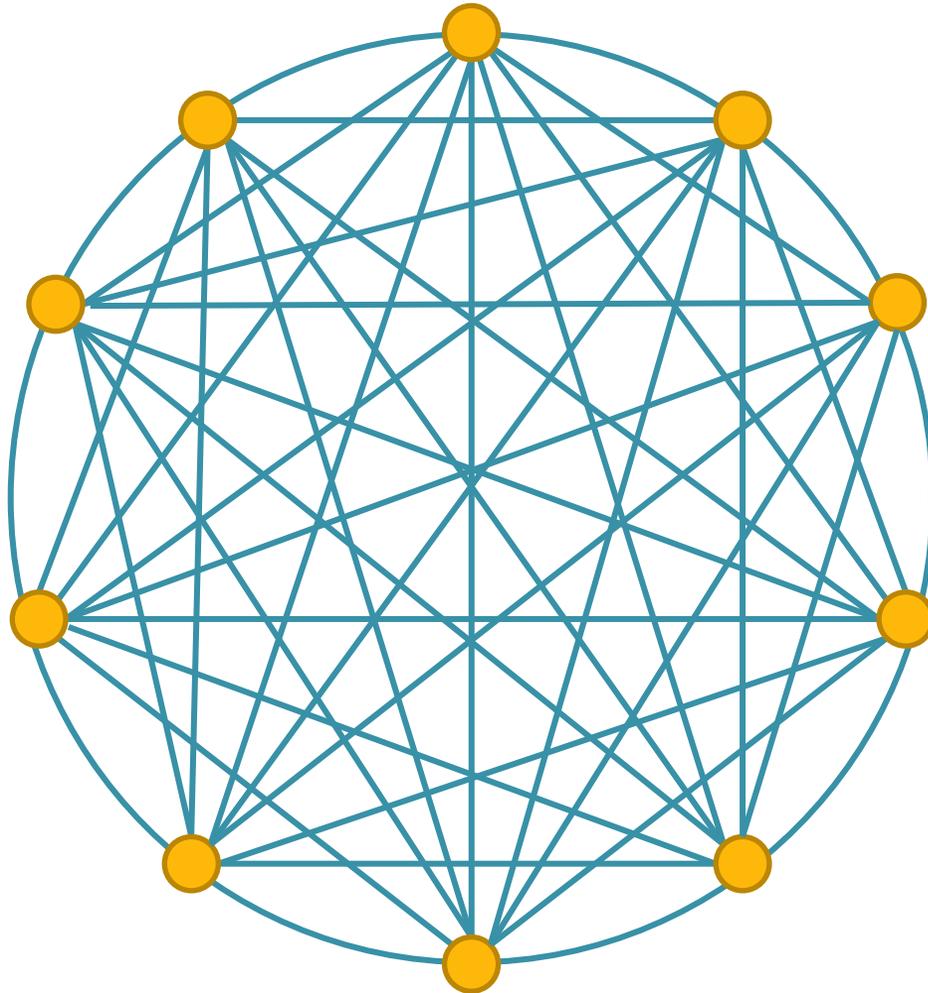


Vizinhança Local

- Supor 7 partículas: a, b, c, d, e, f e g
 - Usar as vizinhanças: gab, abc, bcd, cde, efg e fga
 - Esta topologia atrasa convergência e permite maior exploração do espaço de busca.
 - Partícula a é vizinha de g e de b .
 - Se b ou g encontrarem uma solução melhor que a , a será influenciada por esta decisão.
 - g e b , por sua vez, têm suas próprias vizinhanças e serão influenciadas por elas.



Vizinhança Global X Local



Algoritmo PSO



- Atualização de velocidade:

$$\mathbf{v}_i(t + 1) = \mathbf{v}_i(t) + \boldsymbol{\varphi}_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \boldsymbol{\varphi}_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$

onde

\mathbf{v}_i é a velocidade da partícula i

\mathbf{x}_i é a posição da partícula i

t é o tempo

\mathbf{p}_i é a melhor posição encontrada até o momento pela partícula i

\mathbf{p}_g é a melhor posição encontrada até o momento pela vizinhança de i

$\boldsymbol{\varphi}_1$ é um vetor de números aleatórios em uma distribuição uniforme entre 0 e AC_1

$\boldsymbol{\varphi}_2$ é um vetor de números aleatórios em uma distribuição uniforme entre 0 e AC_2

AC_1 e AC_2 são constantes de aceleração. Geralmente $AC_1 = AC_2 = 2$

Kennedy define:

$AC_1 + AC_2 = 4,1$ (normalmente $AC_1 = AC_2 = 2,05$)

Algoritmo PSO



Novos vetores de números aleatórios devem ser gerados em cada iteração

Multiplicação de elemento por elemento

$$\mathbf{v}_i(t + 1) = \mathbf{v}_i(t) + \varphi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \varphi_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$

A notação $\Delta \mathbf{x}_i$ também é usada

Inércia

Distância entre a melhor posição encontrada pela partícula e a posição atual
Comportamento Cognitivo

Distância entre a melhor posição encontrada pela vizinhança e a posição atual
Comportamento Social

Algoritmo PSO



- Atualização de posição:

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1)$$

onde:

\mathbf{v}_i é a velocidade da partícula i (calculada no slide anterior)

t é o tempo

\mathbf{x}_i é a posição da partícula i

Treinamento

- Permite a propagação lenta para o ótimo global por meio de todas as vizinhanças.
 - Durante a propagação:
 - Partículas das vizinhanças ainda não afetadas continuarão sua busca no espaço de soluções.
 - Seguindo o caminho que elas julgam melhor.
 - Aumenta as chances de encontrar uma nova solução global melhor, próxima a elas ou entre elas.
 - Se existir.



Esta Foto de Autor Desconhecido está licenciado em [CC BY-SA](#)

Avaliação da função de aptidão

- Partícula não precisa saber qual é o seu valor de aptidão.
- Precisa apenas saber quão boa é sua aptidão em relação aos vizinhos.
- Também pode saber quão boa é sua aptidão em relação ao ótimo.
 - Valor do ótimo é definido pelo projetista.
- Valor de aptidão é usado para comparar aptidão de duas partículas quaisquer.
 - Para definir melhor posição das partículas e das vizinhanças.

Limitar movimentação das partículas

- Para limitar a mudança de posição das partículas é possível estabelecer limites de velocidade:
 - Se $v_{id} > v_{\max}$ então $v_{id} = v_{\max}$
 - Se $v_{id} < v_{\min}$ então $v_{id} = v_{\min}$
 - v_{id} é a dimensão d do vetor \mathbf{v}_i

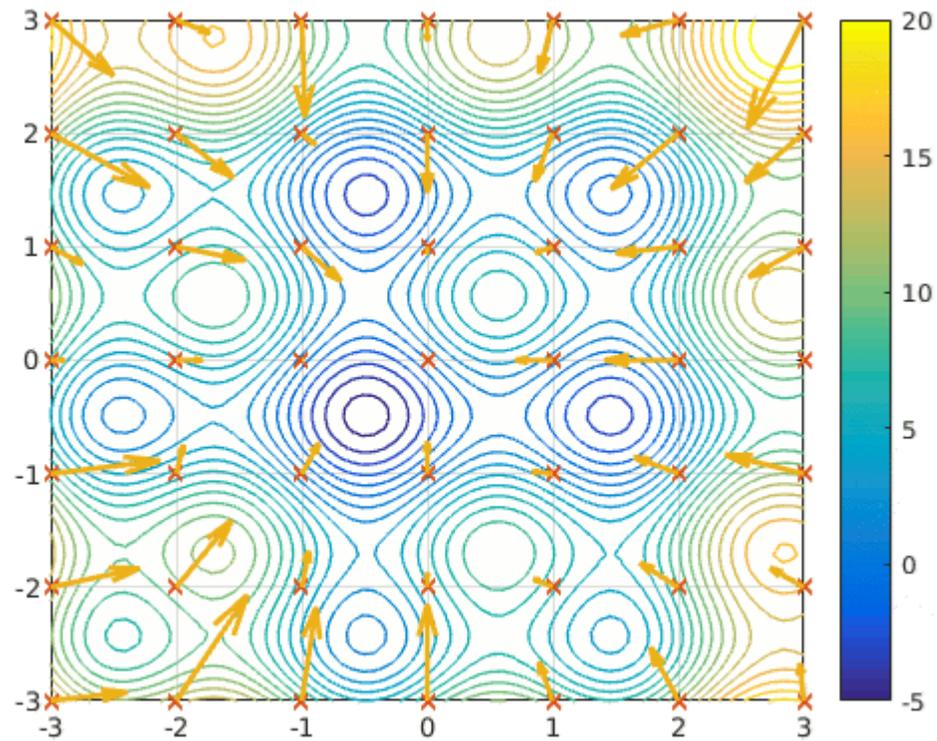


Algoritmo



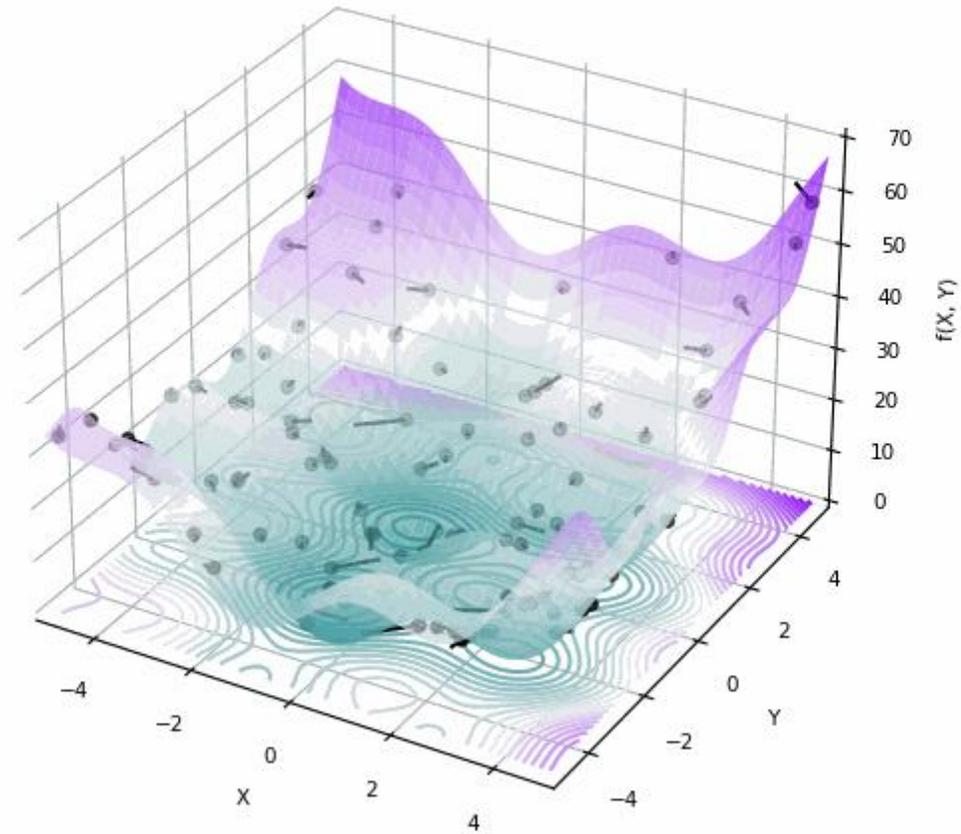
```
procedimento [X] = PS(max_it, AC1, AC2, Vmax, Vmin)
inicializar X // normalmente  $\mathbf{x}_i$ ,  $\mathbf{v}_i$ , é inicializado aleatoriamente
inicializar v // aleatoriamente,  $\mathbf{v}_i \in [v_{\min}, v_{\max}]$ 
t ← 1;
enquanto t ≤ max_it faça
  para i de 1 até N faça // para cada partícula
    se  $g(\mathbf{x}_i) > g(\mathbf{p}_i)$ 
      então  $\mathbf{p}_i = \mathbf{x}_i$  // melhor desempenho individual
    fim-se
     $g = i$  // arbitrário
    para j = índice dos vizinhos // para todos os vizinhos
      se  $g(\mathbf{p}_j) > g(\mathbf{p}_g)$ 
        então  $g = j$  // índice do melhor vizinho
      fim-se
    fim-para
     $\mathbf{v}_i \leftarrow \mathbf{v}_i + \varphi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i) + \varphi_2 \otimes (\mathbf{p}_g - \mathbf{x}_i)$ 
     $\mathbf{v}_i \in [v_{\min}, v_{\max}]$ 
     $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ 
  fim-para
  t ← t + 1
fim-enquanto
fim-procedimento
```

$g()$ é a função de aptidão
 g é o índice do melhor vizinho
 \mathbf{p}_i é o melhor desempenho da partícula i
 \mathbf{p}_g é o melhor desempenho do melhor vizinho de i

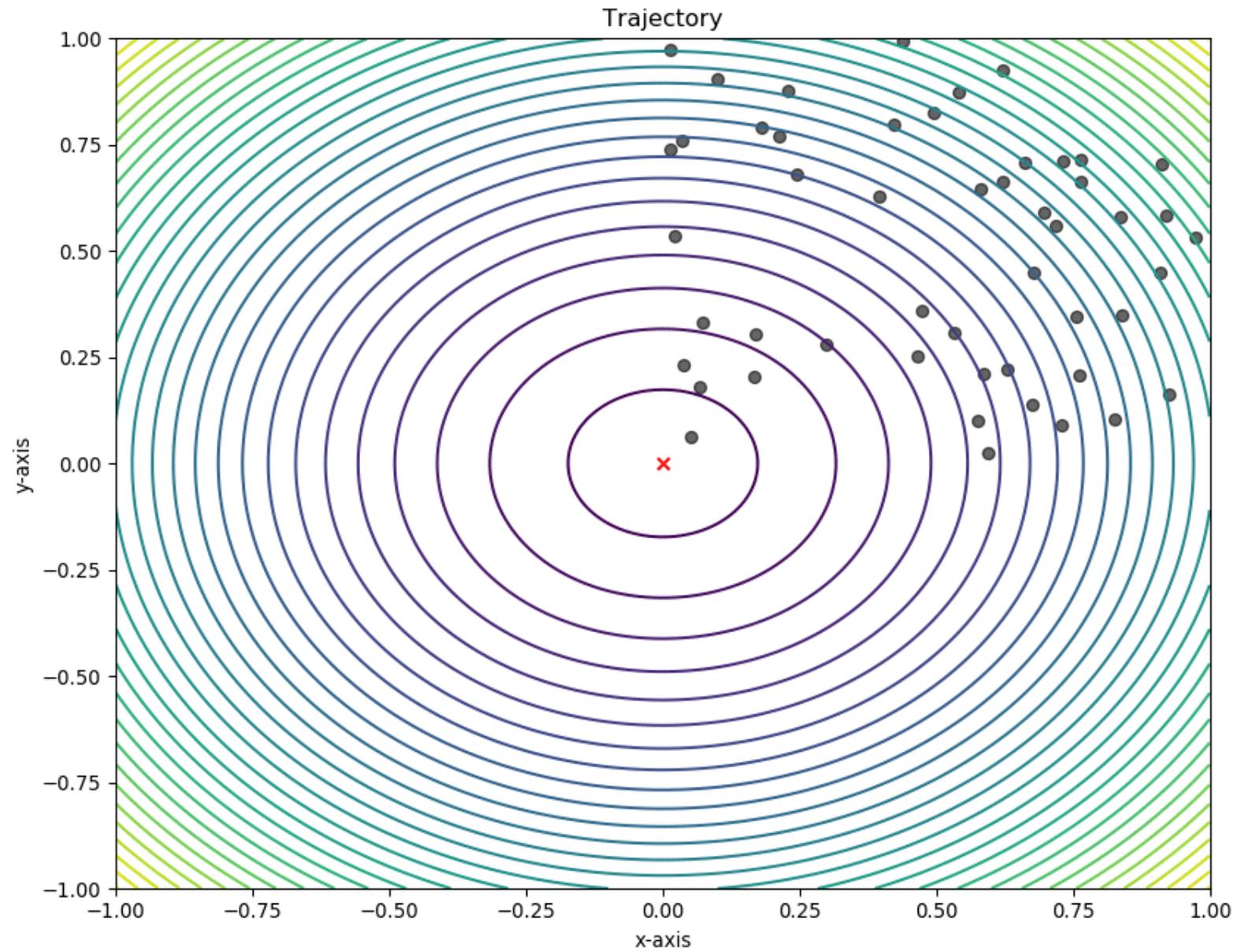


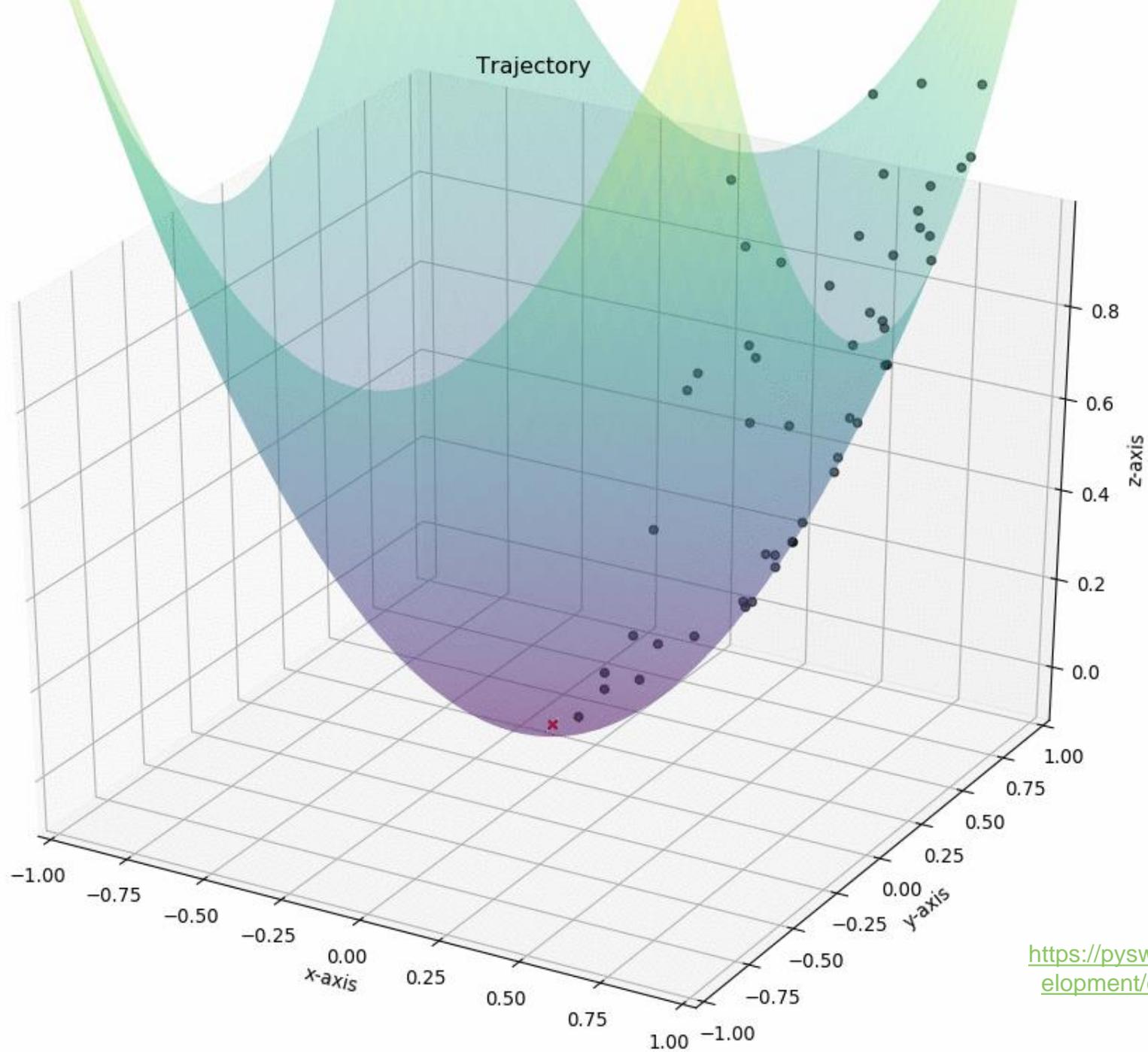
By Ephramac - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=54975083>

[1/100] w:0.800 - c_1 :3.500 - c_2 :0.500



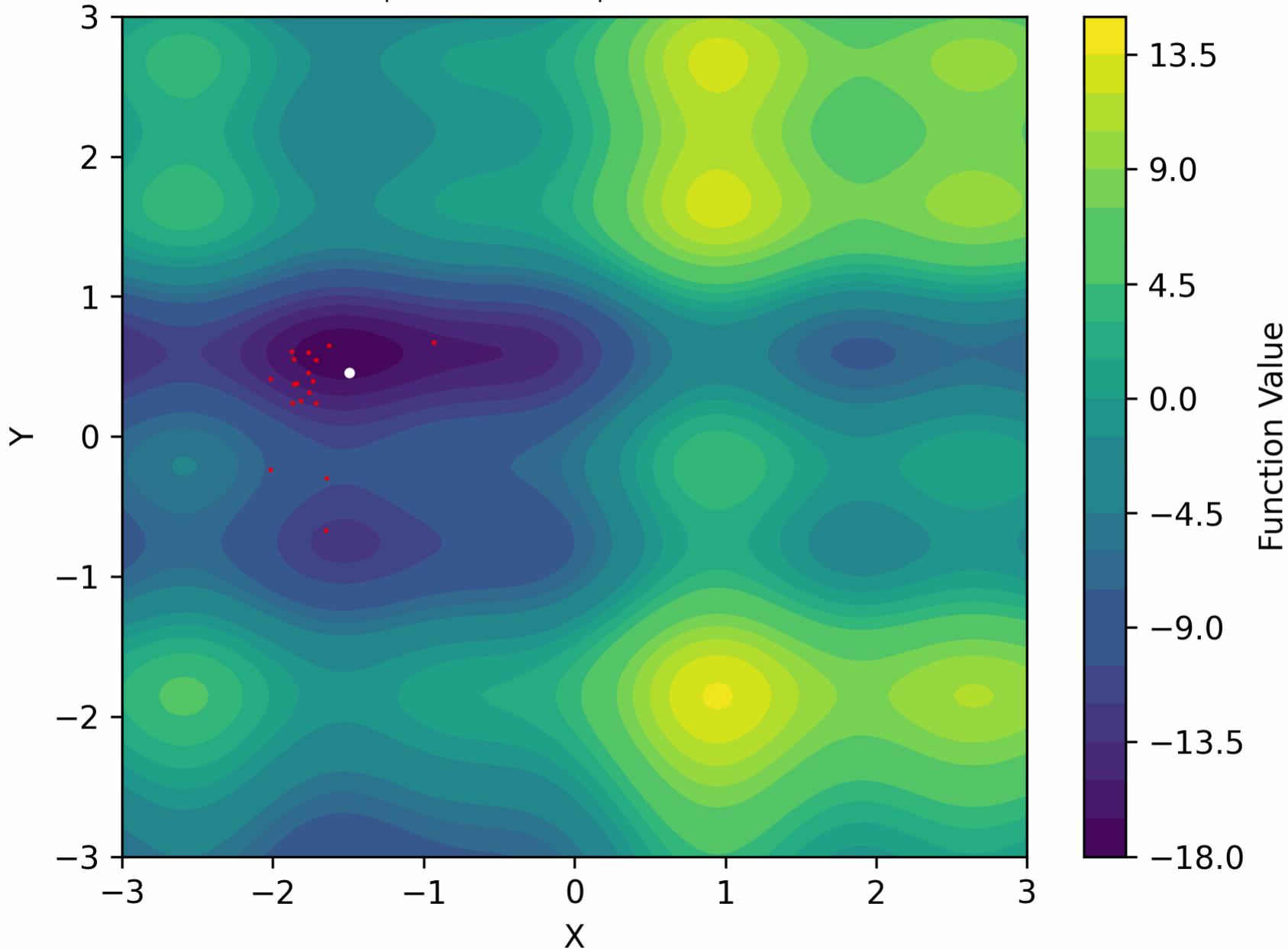
<https://towardsdatascience.com/particle-swarm-optimization-visually-explained-46289eeb2e14>



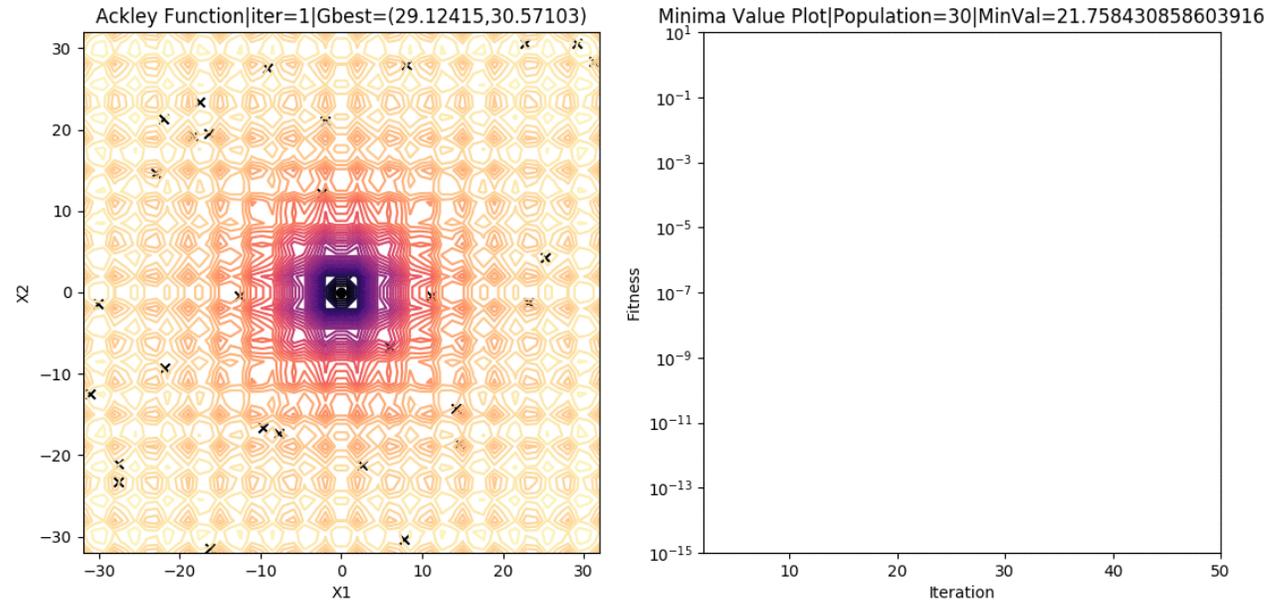


<https://pyswarms.readthedocs.io/en/dev/development/examples/visualization.html>

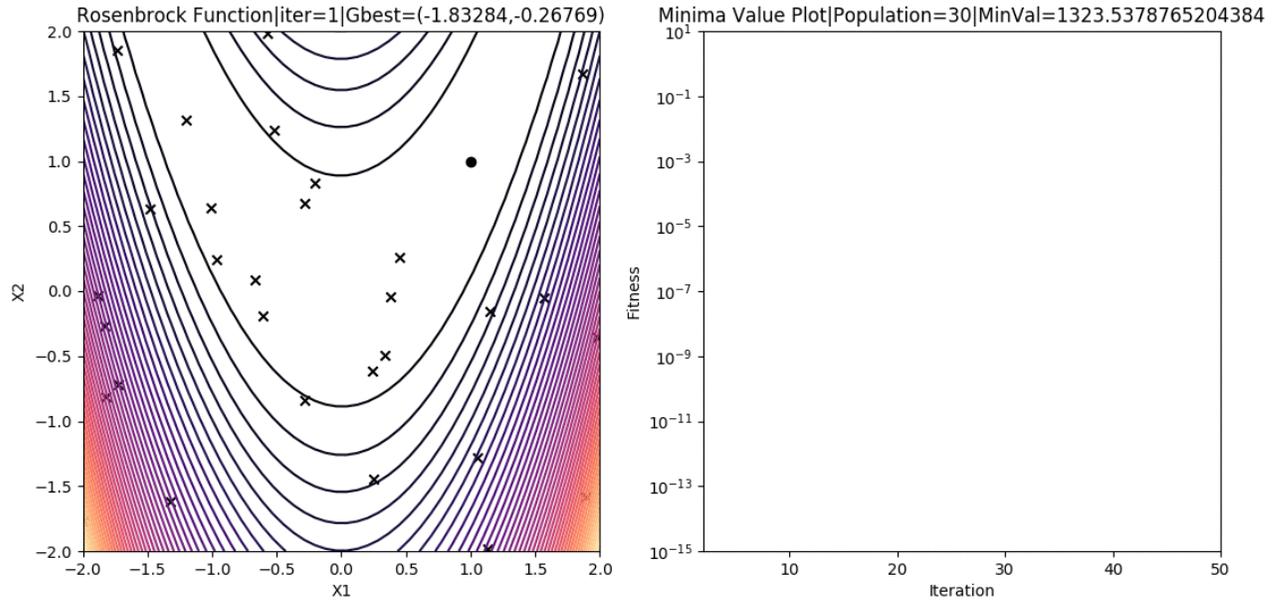
Best value: -17.029597231295877
seed=67, speed=0.001, decay=0.05
exploitation=0.01, exploration=0.0001



Optimizing of Ackley function by PSO, $f_{\min}(0,0)=0.0$



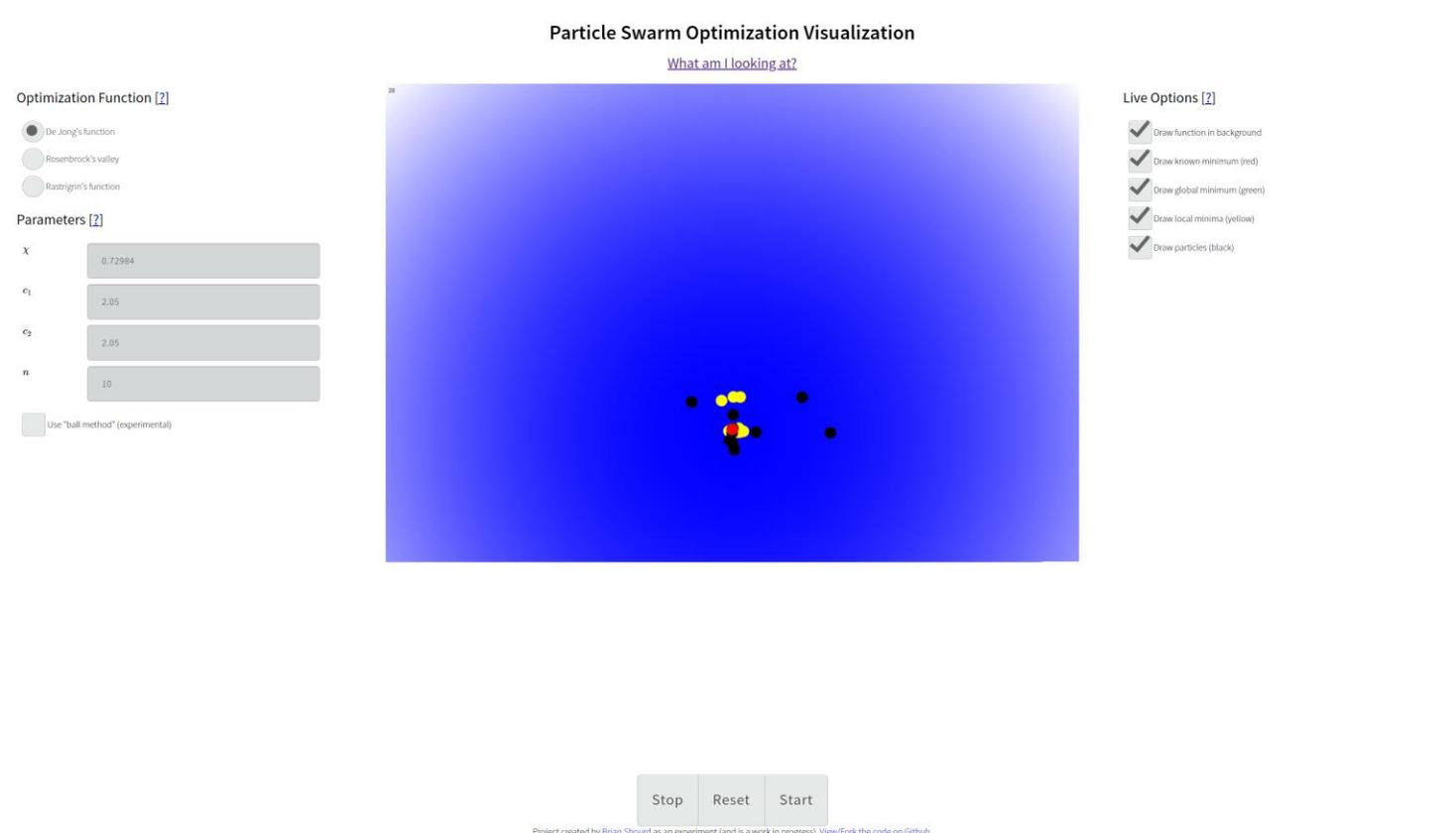
Optimizing of Rosenbrock function by PSO, $f_{\min}(1,1)=0$



https://github.com/Keishilshihara/ps0_with_python

Visualizador de PSO em JavaScript

- <https://github.com/brianshourd/PSO-visualization>



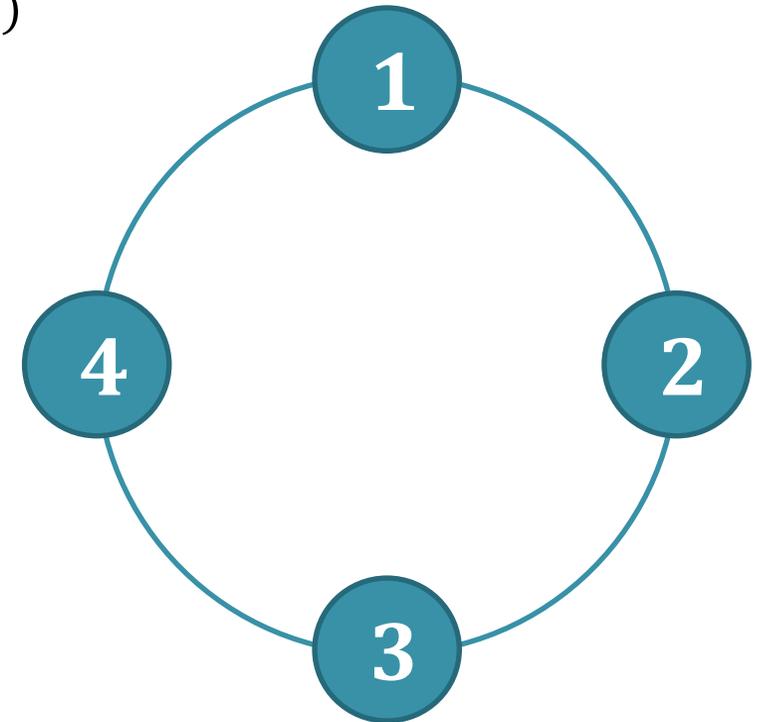
Parâmetros

- Tamanho do enxame $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
 - Quantidade de partículas
 - Normalmente entre 10 e 50.
 - $N \in [10,50]$
 - Dimensão do espaço de busca L
 - Número de parâmetros livres do problema.
 - Escolha de representação
 - Números reais.
 - Escolha da função de aptidão
 - Depende do problema.



Exemplo

- Problema: encontrar valor de x que minimiza $f(x) = x^2, x \in [-7, +7]$
 - População inicial \Rightarrow aleatória
 - População de tamanho 4
 - Função de aptidão $\Rightarrow f(x)$
 - Melhor aptidão = posição de x que gera melhor $f(x)$
 - $AC_1 = AC_2 = 2,05$
 - $v_{max} = 2$
 - $v_{min} = -2$
 - Vizinhaça local circular:
 - 1: 2 e 3
 - 2: 1 e 3
 - 3: 2 e 4
 - 4: 3 e 1



Exemplo

Inicializando:

$$\mathbf{x}_1 = [2]$$

$$\mathbf{x}_2 = [3]$$

$$\mathbf{x}_3 = [-4]$$

$$\mathbf{x}_4 = [-2]$$

$$\mathbf{v}_1 = [1]$$

$$\mathbf{v}_2 = [-2]$$

$$\mathbf{v}_3 = [2]$$

$$\mathbf{v}_4 = [-1]$$

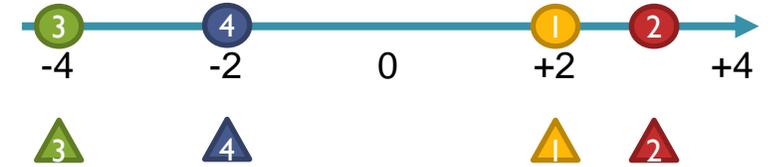
Aptidão:

$$g(\mathbf{x}_1) = 4$$

$$g(\mathbf{x}_2) = 9$$

$$g(\mathbf{x}_3) = 16$$

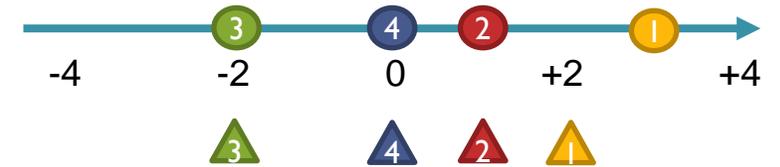
$$g(\mathbf{x}_4) = 4$$



Melhores posições

$$p_1 = 2$$
$$p_2 = 3$$
$$p_3 = -4$$
$$p_4 = -2$$
$$p_{GERAL} = 2$$

Exemplo – Passo I



$$\mathbf{v}_i(t + 1) = \mathbf{v}_i(t) + \boldsymbol{\varphi}_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \boldsymbol{\varphi}_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$
$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1)$$

- $v_1 = 1 + 1,6 (2 - 2) + 1,8 (2 - 2) = 1$
- $x_1 = 2 + 1 = 3$
- $v_2 = -2 + 0,3 (3 - 3) + 1,9 (2 - 3) = -3,9 = -2$
- $x_2 = 3 - 2 = 1$
- $v_3 = 2 + 1,3 (-4 - (-4)) + 0,2 (-2 - (-4)) = 2,4 = 2$
- $x_3 = -4 + 2 = -2$
- $v_4 = -1 + 0,6 (-2 - (-2)) + 1,1 (2 - (-2)) = 3,4 = 2$
- $x_4 = -2 + 2 = 0$

Aptidão

$$g(x_1) = 9$$

$$g(x_2) = 1$$

$$g(x_3) = 4$$

$$g(x_4) = 0$$

Melhores
Posições

$$p_1 = 2$$

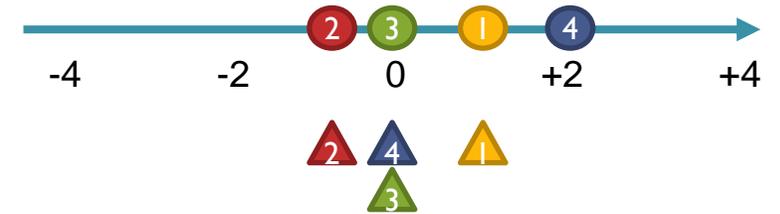
$$p_2 = 1$$

$$p_3 = -2$$

$$p_4 = 0$$

$$p_{GERAL} = 0$$

Exemplo – Passo 2



$$\mathbf{v}_i(t + 1) = \mathbf{v}_i(t) + \boldsymbol{\varphi}_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \boldsymbol{\varphi}_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$
$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1)$$

- $v_1 = 1 + 2,0 (2 - 3) + 1,0 (0 - 3) = -4 = -2,0$
- $x_1 = 3 - 2 = 1$
- $v_2 = -2 + 1,6 (1 - 1) + 0,3 (2 - 1) = -1,7$
- $x_2 = 1 - 1,7 = -0,7$
- $v_3 = 2 + 0,9 (-2 - (-2)) + 1,9 (0 - (-2)) = 5,8 = 2,0$
- $x_3 = -2 + 2 = 0$
- $v_4 = 2 + 1,6 (0 - 0) + 2,0 (0 - 0) = 2$
- $x_4 = 0 + 2 = 2$

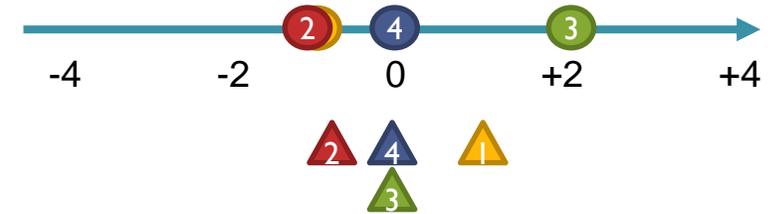
Aptidão

$$g(x_1) = 1$$
$$g(x_2) = 0,5$$
$$g(x_3) = 0$$
$$g(x_4) = 4$$

Melhores
Posições

$$p_1 = 1$$
$$p_2 = -0,7$$
$$p_3 = 0$$
$$p_4 = 0$$
$$p_{GERAL} = 0$$

Exemplo – Passo 3



$$\mathbf{v}_i(t + 1) = \mathbf{v}_i(t) + \boldsymbol{\varphi}_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \boldsymbol{\varphi}_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1)$$

- $v_1 = -2 + 1,3 (1 - 1) + 0,1 (0 - 1) = -2,1 = -2,0$
- $x_1 = 1 - 2 = -1$
- $v_2 = -1,7 + 1,7 (-0,7 - (-0,7)) + 1,9 (0 - (-0,7)) = -0,4$
- $x_2 = -0,7 - 0,4 = -1,1$
- $v_3 = 2 + 1,4 (0 - 0) + 1,6 (0 - 0) = 2$
- $x_3 = 0 + 2 = 2$
- $v_4 = 2 + 1,5 (0 - 2) + 0,8 (0 - 2) = -2,6 = -2,0$
- $x_4 = 2 - 2 = 0$

Aptidão

$$g(x_1) = 1$$

$$g(x_2) = 1,2$$

$$g(x_3) = 4$$

$$g(x_4) = 0$$

Melhores

Posições

$$p_1 = 1$$

$$p_2 = -0,7$$

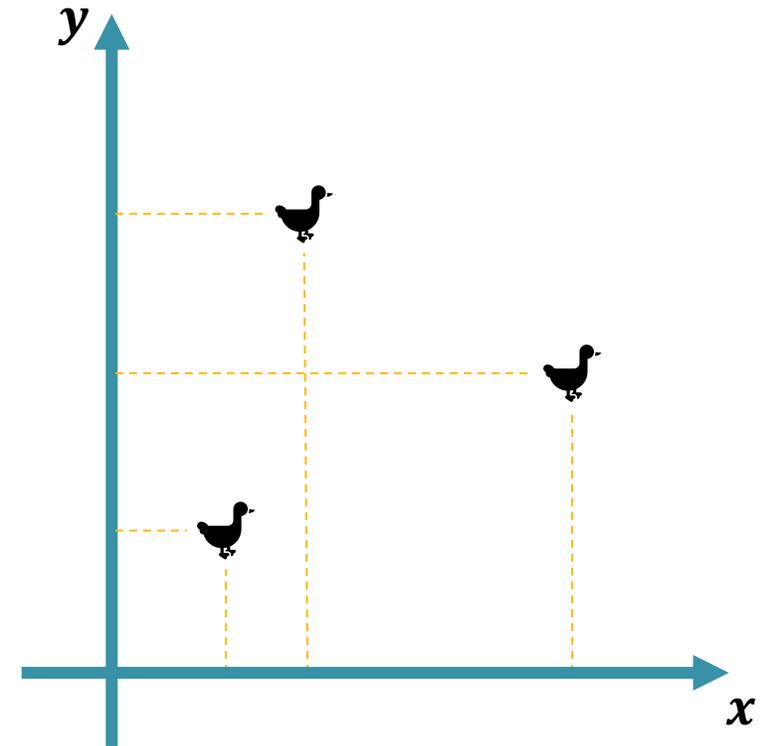
$$p_3 = 0$$

$$p_4 = 0$$

$$p_{GERAL} = 0$$

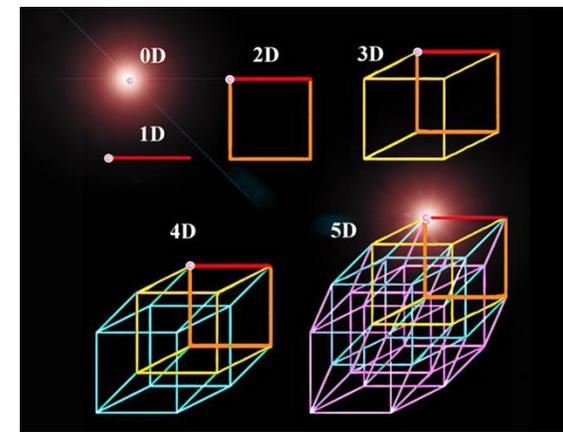
Espaço de Soluções

- Número de dimensões = número de parâmetros livres do problema.
 - Variáveis de valores desconhecidos.
- Exemplo: $x^2 - 3y + 7$
 - Tem 2 dimensões (x, y) .
 - Localização da partícula no espaço de soluções é definido por 2 coordenadas.



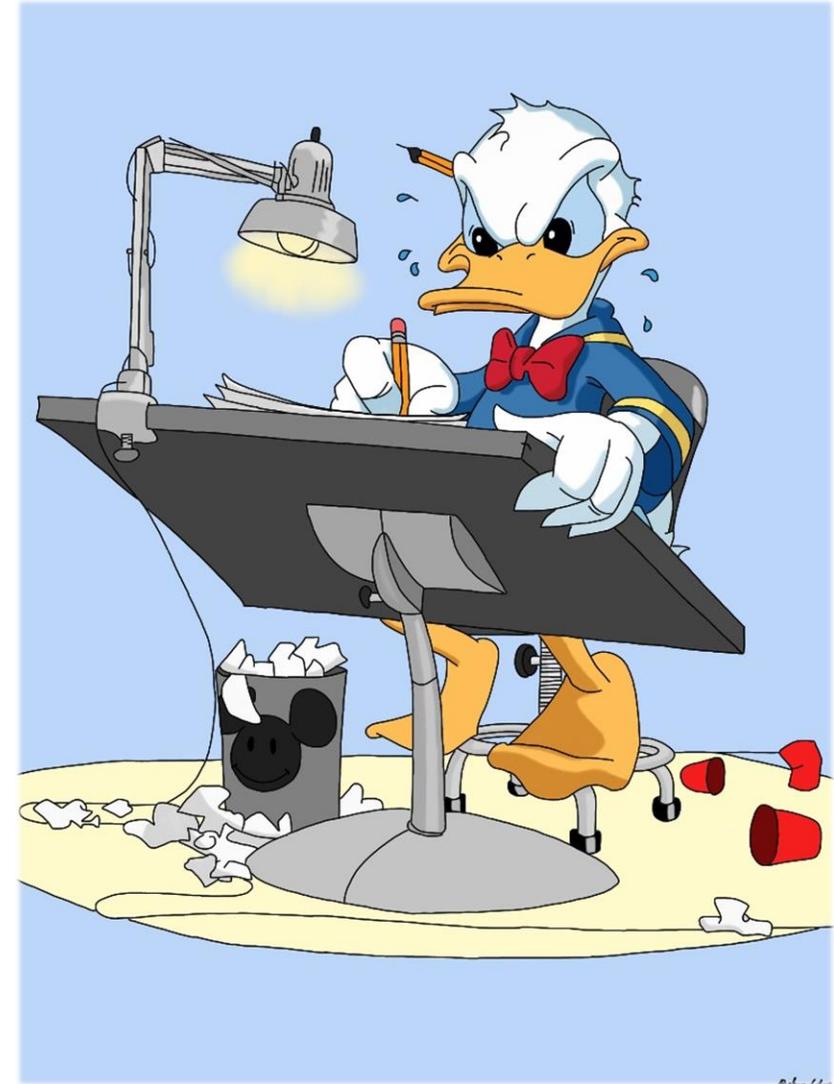
Espaço de Soluções

- Exemplo: $4x^3 - 3y^2 + 4(w - z)^2$
 - Tem 4 dimensões (x, y, z e w).
 - Localização da partícula no espaço de soluções é definido por 4 coordenadas.
- PSO não tem dificuldades para trabalhar com 4 ou mais dimensões.
 - Otimização de uma Rede Neural com 50 pesos teria 50 dimensões.



Exercício

- Seja a função
$$f(x, y, z) = 3x^2 - 5y + 2z^3 - 4$$
- Encontrar valores de x , y e z que fazem com que $f(x, y, z) = 0$
 - Partícula a
 - Coordenadas: 2,1,1
 - Partícula b
 - Coordenadas: 3,7,2



Exploração X Exploração

- É necessário um balanço entre *exploração* e *exploração*.
 - Exploração:
 - Procurar por uma boa solução no espaço de busca, visitando pontos desconhecidos.
 - Pouca *exploração*: algoritmo converge para a primeira boa solução encontrada.
 - Exploração:
 - Extrair o máximo de informação das soluções encontradas e usá-las para obter as próximas soluções.
 - Pouca *exploração*: algoritmo nunca converge.

Sociabilidade X Individualidade

- Balanço *exploração X exploração* pode ser visto como de outra forma:
 - Sociabilidade X Individualidade
 - Indivíduos devem ter individualidade.
 - Pássaros não querem chocar-se com outro(s).
 - Também devem ser sociáveis.
 - Saber onde estão as boas soluções encontradas por outros, podendo aprender com eles.



Lago Negro, Tres Pinheiros, Gramado – Rio Grande do Sul – por Fabricio Breve



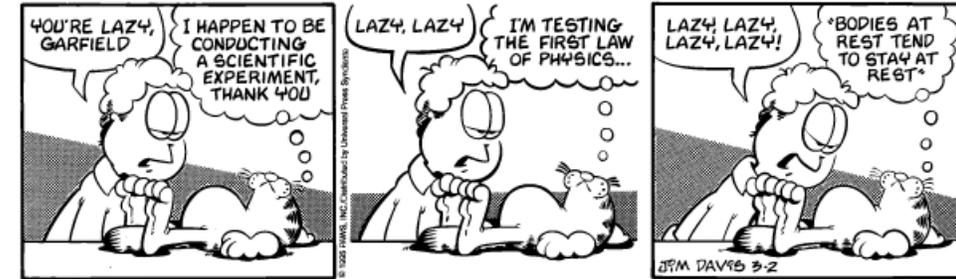
William Hawrelak Park - Edmonton - Alberta - Canadá – por Fabricio Breve

Modificações do PSO

- Peso de inércia w :

$$\mathbf{v}_i(t + 1) = w \cdot \mathbf{v}_i(t) + \phi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \phi_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t))$$

- w tem um valor inicial que pode ser reduzido durante o processo de adaptação.
 - Pode ser considerado uma temperatura.
 - Vai resfriando durante o processo.
 - Exemplo: inicialmente $w = 0,9$, decrescendo linearmente até $w = 0,4$.
 - Da mesma forma que no *Simulated Annealing* (Recozimento Simulado).



Shi, Y., & Eberhart, R. C. (1999, July). **Empirical study of particle swarm optimization**. In *Proceedings of the 1999 congress on evolutionary computation-CEC99* (Cat. No. 99TH8406) (Vol. 3, pp. 1945-1950). IEEE.

Modificações do PSO



Bando de Pássaros em Brisbane, Queensland, Austrália – por Fabricio Breve

- Coeficiente de restrição χ :

$$\mathbf{v}_i(t + 1) = \chi \left(\mathbf{v}_i(t) + \phi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i(t)) + \phi_2 \otimes (\mathbf{p}_g - \mathbf{x}_i(t)) \right)$$

- com $\chi \approx 0,729$

Clerc, M., & Kennedy, J. (2002). **The particle swarm-explosion, stability, and convergence in a multidimensional complex space.** *IEEE transactions on Evolutionary Computation*, 6(1), 58-73.

- valor calculado para $AC_1 + AC_2 = 4,1$
- $v_{max} = x_{max}$

- Modificação conhecida hoje como o PSO padrão.



Fulica em Amsterdã, Holanda – por Fabricio Breve

Kennedy, J. (2005). **Particle swarms: optimization based on sociocognition.** In *Recent developments in biologically inspired computing* (pp. 235-269). IGI Global.

Poli, R., Kennedy, J., & Blackwell, T. (2007). **Particle swarm optimization.** *Swarm intelligence*, 1(1), 33-57.

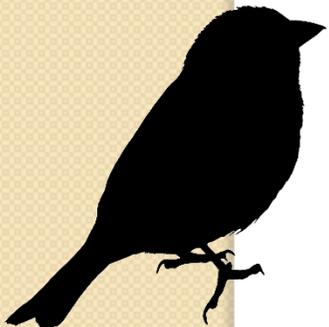
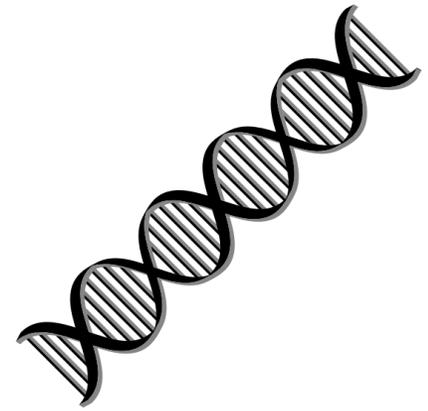
Modificações do PSO

Kennedy, J., & Mendes, R. (2002, May). **Population structure and particle swarm performance.** In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)* (Vol. 2, pp. 1671-1676). IEEE.

- **Enxame de Partículas Totalmente Informado:**
 - *Fully Informed Particle Swarm (FIPS)*
 - **PSO padrão:** fontes efetivas de influência são apenas duas: a própria partícula e a melhor vizinha. Informações dos vizinhos restantes não são utilizadas.
 - **FIPS:** a partícula é afetada por todos os seus vizinhos, às vezes sem influência de seu próprio sucesso anterior.
 - Com bons parâmetros, o FIPS parece encontrar melhores soluções em menos iterações do que o algoritmo canônico, mas é muito mais dependente da topologia da população.

PSO X AGs

- Similaridades:
 - Metaheurísticas baseadas em populações.
 - Inicializadas com uma população aleatória de indivíduos.
 - Utilizam função de aptidão para avaliar cada indivíduo.
 - Buscam por ótimo global em várias gerações.
 - Critério de parada.
 - Não garantem sucesso.



PSO



- Vôo no espaço de busca.
 - Velocidade interna, memória.
- Mais fácil de implementar.
- Menos parâmetros para ajustar.
- Apenas melhor indivíduo (da vizinhança) transmite informações.
- Utiliza valores reais (padrão).

AG



- Operadores genéticos.
 - Seleção, *Crossover*, etc...
- Mais difícil de implementar.
- Mais parâmetros para ajustar.
- Cromossomos compartilham informações.
- Utiliza valores binários (padrão).

PSO X AGs: diferenças

PSO: Resumo

- Baseado em comportamento social.
- Algoritmo muito simples.
- Poucas linhas de código.
- Além de rápido, utiliza pouca memória.
- Utiliza operadores matemáticos simples.
- Tem obtido bons resultados em várias aplicações.



PSO: Aplicações



- Otimização de parâmetros de Redes Neurais.
 - Rede neural para análise de tremores humanos.
 - Mal de Parkinson, etc.
 - Rede neural para estimativa de estado da carga de baterias.

Wu, D., Warwick, K., Ma, Z., Gasson, M. N., Burgess, J. G., Pan, S., & Aziz, T. Z. (2010). **Prediction of Parkinson's disease tremor onset using a radial basis function neural network based on particle swarm optimization.** *International journal of neural systems*, 20(02), 109-116.

Ghanad, N. K., & Ahmadi, S. (2015). **Combination of PSO algorithm and naive Bayesian classification for Parkinson disease diagnosis.** *Advances in Computer Science: an International Journal*, 4(4), 119-125.

Hossain Lipu, M. S., Hannan, M. A., Hussain, A., & Saad, M. H. M. (2017). **Optimal BP neural network algorithm for state of charge estimation of lithium-ion battery using PSO with PCA feature selection.** *Journal of Renewable and Sustainable Energy*, 9(6), 064102.

Li, R., Xu, S., Li, S., Zhou, Y., Zhou, K., Liu, X., & Yao, J. (2020). **State of charge prediction algorithm of lithium-ion battery based on PSO-SVR cross validation.** *IEEE Access*, 8, 10234-10242.

PSO: Aplicações

- Otimização de parâmetros em máquinas industriais.
- Controle de tensão e energia reativa.
 - Otimização de parâmetros de transformadores, etc.
- Composição de música.



GirirajKumar, S. M., Jayaraj, D., & Kishan, A. R. (2010). **PSO based tuning of a PID controller for a high performance drilling machine.** *International Journal of Computer Applications*, 1(19), 12-18.

Pappala, V. S., Wilch, M., Singh, S. N., & Erlich, I. (2007, November). **Reactive power management in offshore wind farms by adaptive PSO.** In *2007 International Conference on Intelligent Systems Applications to Power Systems* (pp. 1-8). IEEE.

Mahapatra, S., Badi, M., & Raj, S. (2019, October). **Implementation of PSO, it's variants and Hybrid GWO-PSO for improving Reactive Power Planning.** In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-6). IEEE.

Liu, C. H., & Ting, C. K. (2016). **Computational intelligence in music composition: A survey.** *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(1), 2-15.

Bibliografia

- CASTRO, Leandro Nunes. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, And Applications*. CRC Press, 2006.
- CARVALHO, André Ponce de Leon F. de. *Notas de Aula*, 2007.
- BROWNLEE, Jason. *Clever Algorithms: Nature-Inspired Programming Recipes*. Jason Brownlee, 2011.
- BONABEAU, Eric; DORIGO, Marco; THERAULAZ, Guy. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- DORIGO, Marco; STÜTZLE, Thomas. *Ant Colony Optimization*. Bradford Books, 2004.
- BREVE, Fabricio; ZHAO, Liang; QUILES, Marcos G.; PEDRYCZ, Witold; LIU, Jimming. **Particle competition and cooperation in networks for semi-supervised learning**. *Knowledge and Data Engineering, IEEE Transactions on*, 2012.
- BREVE, Fabricio Aparecido. *Aprendizado de Máquina em Redes Complexas*. 165 páginas. Tese. São Carlos: Universidade de São Paulo, 2010.

