Universidade Federal do Ceará
Departamento de Computação

Meta-heurísticas para otimização: Otimização por Enxame de Partículas *Particle Swarm Optimization* - PSO

César Lincoln Cavalcante Mattos

Junho 2018

Objetivos de Aprendizagem

- Revisar os conceitos básicos de otimização meta-heurística.
- Motivar o uso da cooperação como estratégia de busca.
- Descrever e analisar o algoritmo de Otimização por Enxame de Partículas (PSO) original e suas principais variantes.
- Exemplificar o uso do método PSO em problemas de otimização numérica.

Conteúdo

Revisão de Conceitos Básicos

Otimização Meta-heurística

- Busca usualmente estocástica para problemas de otimização.
  - → Presença de números aleatórios na geração/modificação de soluções candidatas e em tomadas de decisão do algoritmo.
- Obtém "boas soluções" em tempo hábil para problemas complexos a partir de pouca informação *a priori*.
  - → Não necessita de gradientes da função objetivo.
  - → Não necessita conhecer a forma analítica da função objetivo.
- Pode ser de estado único (ou de trajetória) ou populacional.

Revisão de Conceitos Básicos

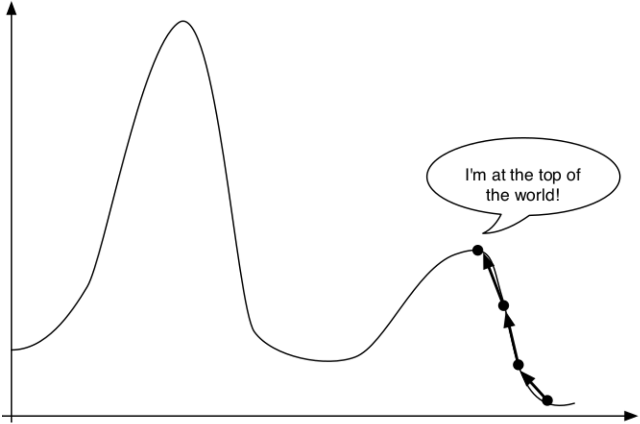
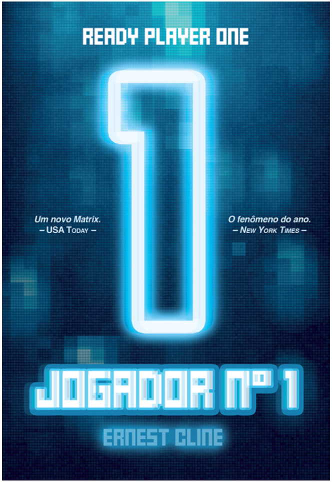


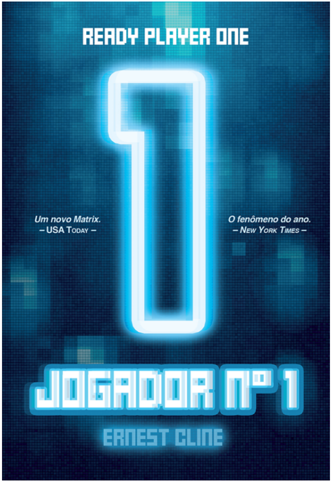
Figura 1: Exemplo de máximo local obtido via gradiente ascendente.

Estratégias de busca



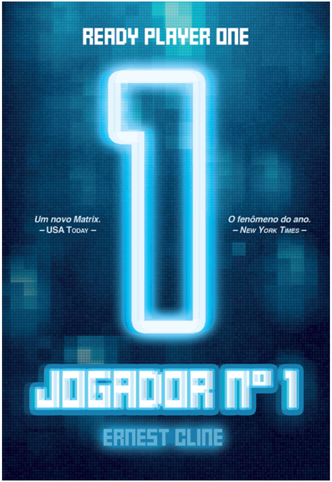
- "Jogador nº 1" é um livro de ficção científica de Ernest Cline.
- Na trama, o criador de um jogo online de realidade virtual esconde um *easter egg* dentro do jogo, prometendo a herança de toda sua fortuna para quem encontrá-lo.
- Como os jogadores devem agir para encontrar o *easter egg*?

Estratégias de busca



- "Jogador nº 1" é um livro de ficção científica de Ernest Cline.
- Na trama, o criador de um jogo online de realidade virtual esconde um *easter egg* dentro do jogo, prometendo a herança de toda sua fortuna para quem encontrá-lo.
- Como os jogadores devem agir para encontrar o *easter egg*?
  - → Cada jogador inicialmente possui **pouca informação**.

Estratégias de busca



- "Jogador nº 1" é um livro de ficção científica de Ernest Cline.
- Na trama, o criador de um jogo online de realidade virtual esconde um *easter egg* dentro do jogo, prometendo a herança de toda sua fortuna para quem encontrá-lo.
- Como os jogadores devem agir para encontrar o *easter egg*?
  - → Cada jogador inicialmente possui **pouca informação**.
  - → Eles devem **cooperar** entre si!

Conteúdo

Intuição

Figura 2: Ilustração de cardumes de peixes e grupos de pássaros.

Inteligência de Enxame (*Swarm Intelligence*)

- Propriedade em que comportamentos pouco sofisticados em conjunto resultam em **ordenamento global emergente**.
- Comportamento de resolução de problemas baseado na **interação entre agentes**.
- **Métodos computacionais**: Algoritmos PSO e ACO.

Algoritmo PSO

Otimização por Enxame de Partículas
PSO - Particles Swarm Optimization

- Inspiração no comportamento social e na auto-organização de grupos de pássaros migratórios e cardumes de peixes[1].
- **Partícula**: Representa uma possível solução.
  - → Sua posição contém os valores das componente da solução.
  - → Sua velocidade indica a tendência de movimento pelo espaço.
- **Enxame**: Grupo de partículas que buscam otimizar a mesma função.
- **Ideias básicas**
  - → Um indivíduo tende a revisitar a melhor posição encontrada por ele.
  - → Um indivíduo tende a se mover na direção da melhor posição encontrada por seus informantes.
- Equilíbrio entre **explotação** e **exploração**.

[1] Kennedy e Eberhart, 1995.

Algoritmo PSO

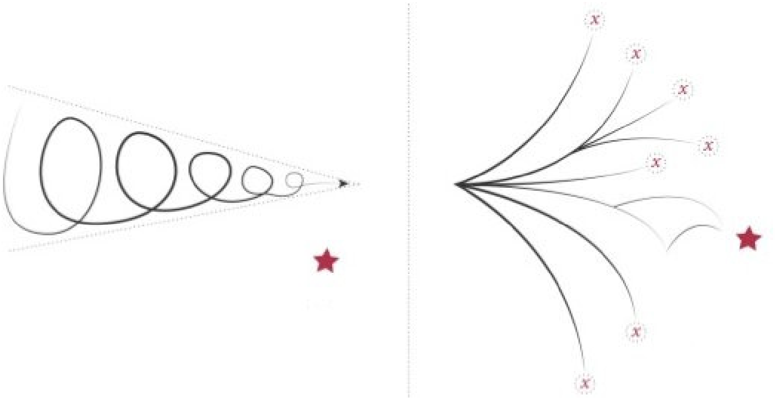


Figura 3: Ilustração da necessidade de equilibrar explotação e exploração. O excesso de explotação (à esquerda) ou exploração (à direita) resultam na perda de boas soluções (estrela).

Algoritmo PSO

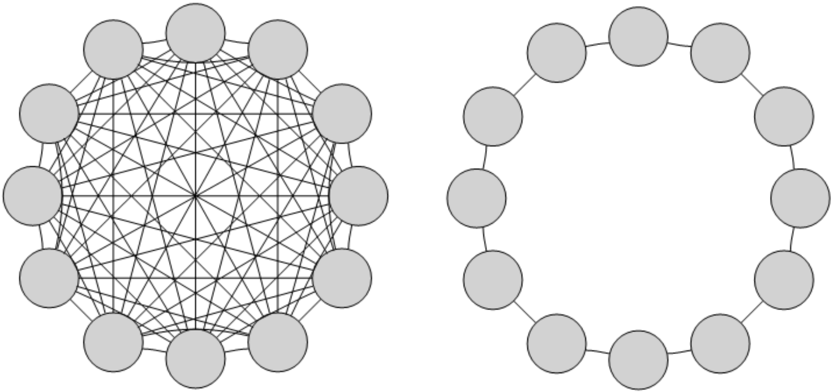


Figura 4: Topologias global (à esquerda) e local (à direita) de enxames de partículas.

Algoritmo PSO Global

Notação do algoritmo

- Seja um enxame de N partículas com posições $\boldsymbol{x}_i$ e velocidades $\boldsymbol{v}_i$, $i = 1, \cdots, N$:

$$\boldsymbol{x}_i = [x_1, \cdots, x_D]^\top, \quad (1)$$

$$\boldsymbol{v}_i = [v_1, \cdots, v_D]^\top. \quad (2)$$

$\rightarrow$ D: Quantidade de variáveis em cada solução.
$\rightarrow$ $\boldsymbol{x}_i \in \mathbb{R}^D$: Posição da i-ésima partícula.
$\rightarrow$ $\boldsymbol{v}_i \in \mathbb{R}^D$: Velocidade da i-ésima partícula.
$\rightarrow$ $\boldsymbol{p}_i \in \mathbb{R}^D$: Melhor solução encontrada pela i-ésima partícula.
$\rightarrow$ $\boldsymbol{g} \in \mathbb{R}^D$: Melhor solução global encontrada pelo enxame.

Algoritmo PSO Global

Resumo do algoritmo

0. Iniciar as variáveis do algoritmo.
   - Inicializar $\boldsymbol{x}_i$ e $\boldsymbol{v}_i$ de acordo com algum critério.
   - Calcular o valor da função objetivo $f(\cdot)$ para o enxame inicial.
   - Iniciar $\boldsymbol{p}_i$ e $\boldsymbol{g}$ com as melhores soluções iniciais.
1. Atualizar $\boldsymbol{v}_i$ e $\boldsymbol{x}_i$ aplicando uma regra de atualização das partículas.
2. Avaliar $f(\cdot)$ para todas as partículas.
3. Atualizar $\boldsymbol{p}_i$, caso a nova posição da partícula i seja a sua melhor.
4. Atualizar $\boldsymbol{g}$, caso uma melhor solução global tenha sido encontrada pelo enxame.
5. Repetir o procedimento a partir do Passo 1 até que uma condição de parada seja encontrada.

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \quad (3)$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \quad (4)$$

  - → w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
  - → c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
  - → $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
  - → $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \quad (3)$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \quad (4)$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

Interpretação dos termos de velocidade

- $w\boldsymbol{v}_i^{(k)}$: Componente de inércia, evita mudanças drásticas.
- $c_1\boldsymbol{r}_1 \circ (\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)})$: Componente cognitivo da experiência individual.
- $c_2\boldsymbol{r}_2 \circ (\boldsymbol{g} - \boldsymbol{x}_i^{(k)})$: Componente social, referente à busca pela norma social.

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{5}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{6}$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- $c_1 > 0$ e $c_2 = 0$?

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{5}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{6}$$

  - $\rightarrow$ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
  - $\rightarrow$ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
  - $\rightarrow$ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
  - $\rightarrow$ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- $c_1 > 0$ e $c_2 = 0$? Múltiplos hill-climbings estocásticos independentes.

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{5}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{6}$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- $c_1 > 0$ e $c_2 = 0$? Múltiplos hill-climbings estocásticos independentes.
- $c_1 = 0$ e $c_2 > 0$?

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{5}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{6}$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- $c_1 > 0$ e $c_2 = 0$? Múltiplos hill-climbings estocásticos independentes.
- $c_1 = 0$ e $c_2 > 0$? Convergência em um só hill-climbing estocástico.

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \quad (7)$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \quad (8)$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- O valor de w for baixo?

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{7}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{8}$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).

→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).

→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.

→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- O valor de w for baixo? Ênfase em explotação.

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{7}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{8}$$

→ w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
→ c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
→ $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
→ $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- O valor de w for baixo? Ênfase em explotação.
- O valor de w for alto?

Algoritmo PSO Global

Regra de atualização das partículas

- A cada iteração k atualiza-se as velocidades e posições do enxame:

$$\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)} + c_1\boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + c_2\boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right), \tag{7}$$

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}. \tag{8}$$

  - → w: Parâmetro de inércia (usualmente $0{,}4 \leq w \leq 0{,}9$).
  - → c_1 e c_2: Coeficientes aceleradores (usualmente $c_1 = c_2 = 2{,}05$).
  - → $\boldsymbol{r}_1$ e $\boldsymbol{r}_2$: Vetores D-dimensionais de números aleatórios uniformemente distribuídos entre 0 e 1.
  - → $\circ$: Notação do produto de Hadamard (produto elemento a elemento).

O que acontece se:

- O valor de w for baixo? Ênfase em explotação.
- O valor de w for alto? Ênfase em exploração.

Versão local do algoritmo PSO

Algoritmo PSO Local

- Limita a comunicação de cada partícula à sua vizinhança.
- **Vizinhança**: Subgrupo de partículas que compartilham informações.
  - $\rightarrow$ É definida no início do algoritmo.
  - $\rightarrow$ Normalmente baseia-se somente nos índices das partículas.
  - $\rightarrow$ Sobrepõe-se a outras vizinhanças.
- Substituímos a melhor solução global $\boldsymbol{g}$ por $\boldsymbol{l}_i \in \mathbb{R}^D$, que possui a melhor solução encontrada pela vizinhança da i-ésima partícula.
- Os demais passos são idênticos à versão global.

Versão local do algoritmo PSO

Algoritmo PSO Local

- Limita a comunicação de cada partícula à sua vizinhança.
- **Vizinhança**: Subgrupo de partículas que compartilham informações.
  - $\rightarrow$ É definida no início do algoritmo.
  - $\rightarrow$ Normalmente baseia-se somente nos índices das partículas.
  - $\rightarrow$ Sobrepõe-se a outras vizinhanças.
- Substituímos a melhor solução global $\boldsymbol{g}$ por $\boldsymbol{l}_i \in \mathbb{R}^D$, que possui a melhor solução encontrada pela vizinhança da i-ésima partícula.
- Os demais passos são idênticos à versão global.

Local ou Global?

- A versão Global apresenta convergência mais rápida.
- A versão Local apresenta maior diversidade entre as partículas, o que normalmente implica em melhor exploração.

Interpretação visual do algoritmo PSO

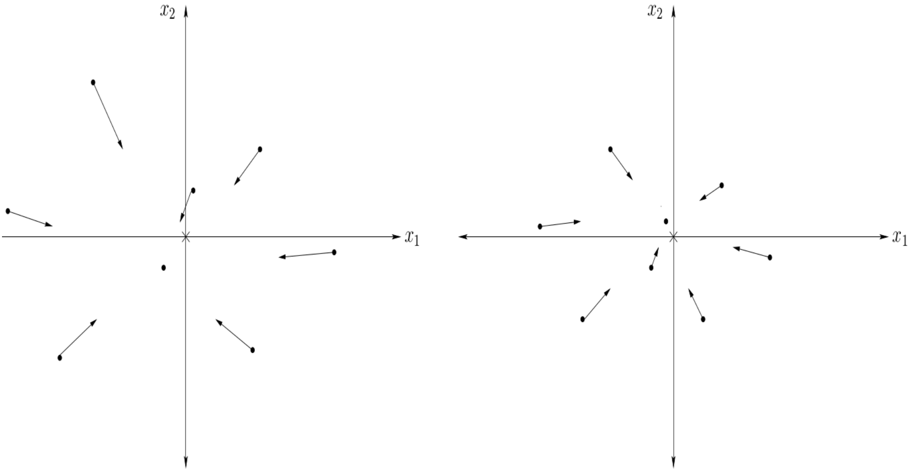


Figura 5: Ilustração do enxame inicial (à esquerda) e após a primeira atualização (à direita) no algoritmo PSO Global.

Interpretação visual do algoritmo PSO

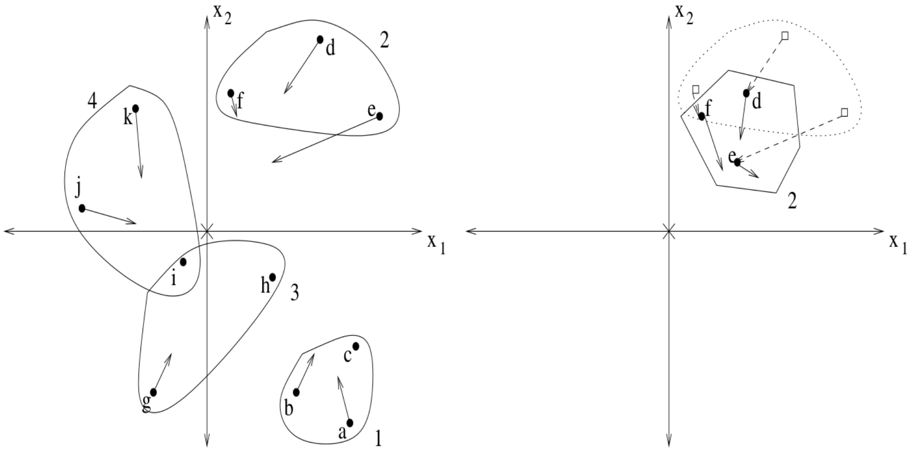


Figura 6: Ilustração do enxame inicial (à esquerda) e após a primeira atualização (à direita) no algoritmo PSO Local.

Detalhes de implementação do algoritmo PSO

- Normalmente são usadas entre 10 e 50 partículas no enxame.
- O enxame inicial pode ser inicializado da seguinte forma:

$$\boldsymbol{x}_i^{(0)} = \boldsymbol{x}_{\text{min}} + \boldsymbol{r} \circ (\boldsymbol{x}_{\text{max}} - \boldsymbol{x}_{\text{min}}), \tag{9}$$

$$\boldsymbol{v}_i^{(0)} = \boldsymbol{0}. \tag{10}$$

  → $\boldsymbol{r}$: Vetores de números uniformemente aleatórios entre 0 e 1.
  → $\boldsymbol{x}_{\text{max}}, \boldsymbol{x}_{\text{min}}$: Vetores de máximos e mínimos valores das componentes.
- O parâmetro de inércia w pode ser decaído ao longo das iterações, priorizando exploração no início e explotação ao final.

Limitações do algoritmo PSO

- **Explosão do enxame**
  - → Algumas atualizações podem resultar em velocidades muito altas para as partículas.
  - → As partículas afetadas podem sair da região de busca e nunca voltarem.

[2] CLERC e KENNEDY, 2002.

Limitações do algoritmo PSO

- **Explosão do enxame**
  - → Algumas atualizações podem resultar em velocidades muito altas para as partículas.
  - → As partículas afetadas podem sair da região de busca e nunca voltarem.
  - → **Alternativa**: Limitar a velocidade máxima das partículas; usar um fator de constrição ($\chi \approx 0{,}7298$) para as novas velocidades[2]:

$$\boldsymbol{v}_i^{(k+1)} = \chi \left[\boldsymbol{v}_i^{(k)} + c_1 \boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)} \right) + c_2 \boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)} \right) \right]. \quad (11)$$

[2] CLERC e KENNEDY, 2002.

Limitações do algoritmo PSO

- **Explosão do enxame**
  - → Algumas atualizações podem resultar em velocidades muito altas para as partículas.
  - → As partículas afetadas podem sair da região de busca e nunca voltarem.
  - → **Alternativa**: Limitar a velocidade máxima das partículas; usar um fator de constrição ($\chi \approx 0{,}7298$) para as novas velocidades[2]:

$$\boldsymbol{v}_i^{(k+1)} = \chi \left[\boldsymbol{v}_i^{(k)} + c_1 \boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)} \right) + c_2 \boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)} \right) \right]. \quad (11)$$

- **Convergência prematura**
  - → Caso uma partícula atinja a posição da melhor partícula do exame, ou seja, $\boldsymbol{x}_i = \boldsymbol{g}$, sua velocidade será atualizada por $\boldsymbol{v}_i^{(k+1)} = w\boldsymbol{v}_i^{(k)}$.
  - → Persistindo a condição anterior, temos $\boldsymbol{v}_i \to 0$.
  - → Esse efeito pode ocorrer em todas as partículas, paralisando o enxame.

[2] CLERC e KENNEDY, 2002.

Limitações do algoritmo PSO

- **Explosão do enxame**
  - → Algumas atualizações podem resultar em velocidades muito altas para as partículas.
  - → As partículas afetadas podem sair da região de busca e nunca voltarem.
  - → **Alternativa**: Limitar a velocidade máxima das partículas; usar um fator de constrição ($\chi \approx 0{,}7298$) para as novas velocidades[2]:

$$\boldsymbol{v}_i^{(k+1)} = \chi \left[\boldsymbol{v}_i^{(k)} + c_1 \boldsymbol{r}_1 \circ \left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)} \right) + c_2 \boldsymbol{r}_2 \circ \left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)} \right) \right]. \quad (11)$$

- **Convergência prematura**
  - → Caso uma partícula atinja a posição da melhor partícula do exame, ou seja, $\boldsymbol{x}_i = \boldsymbol{g}$, sua velocidade será atualizada por $\boldsymbol{v}_i^{(k+1)} = w \boldsymbol{v}_i^{(k)}$.
  - → Persistindo a condição anterior, temos $\boldsymbol{v}_i \to 0$.
  - → Esse efeito pode ocorrer em todas as partículas, paralisando o enxame.
  - → **Alternativa**: Escolha cuidadosa dos parâmetros do algoritmo; execução de busca local em torno de $\boldsymbol{g}$.

---
[2] CLERC e KENNEDY, 2002.

Variantes do algoritmo PSO

- **Diferentes estruturas para o enxame**
  - → Diferentes topologias, estáticas ou dinâmicas.
  - → Adição sequencial de partículas.
  - → Múltiplos enxames, com cooperação ou competição entre enxames.
- **Diferentes formas de troca de informação**
  - → Regras de atualização alternativa para as partículas.
- **Hibridização**
  - → Etapas de intensificação via algoritmos locais.
  - → Uso de outras meta-heurísticas para definir parâmetros do PSO.

Conteúdo

Exemplo - Função de Rastrigin

- Função de Rastrigin:

$$f(\boldsymbol{x}) = 10D + \sum_{d=1}^{D}[x_d^2 - 10\cos(2\pi x_d)], \tag{12}$$

$$x_d \in [-5.12, 5.12], \quad d \in \{1, \ldots, D\}.$$

→ Mínimo global: $f(\mathbf{0}) = 0$.

→ Exemplos de solução: PSO Global e PSO Local.

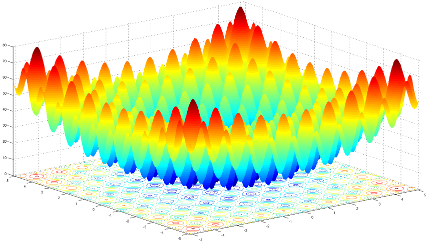


Figura 7: Função de Rastrigin bidimensional.

Exemplo - Função de Schwefel

- Função de Schwefel:

$$f(\boldsymbol{x}) = 418.9829D - \sum_{d=1}^{D} x_d \sin(\sqrt{|x_d|}), \tag{13}$$

$$x_d \in [-500, 500], \quad d \in \{1, \ldots, D\}.$$

→ Mínimo global: $f([420.9687, \cdots, 420.9687]) = 0$.

→ Exemplos de solução: PSO Global e PSO Local.

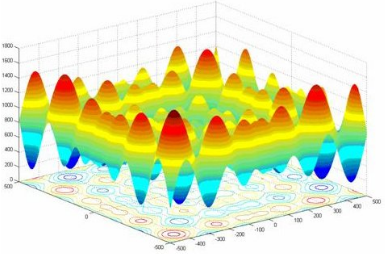


Figura 8: Função de Schwefel bidimensional.

Conteúdo

Revisão

- Cooperação como estratégia de busca.
- Descrição do algoritmo PSO: uma meta-heurística populacional baseada em cooperação.
- Busca de equilíbrio entre explotação e exploração.
- Limitações e variantes do algoritmo PSO.
- Exemplos de otimização numérica estocástica.

Bibliografia

Básica

- ENGELBRECHT, A. P. **Computational intelligence: an introduction**. West Sussex: John Wiley & Sons, 2007.
- LUKE, S. **Essentials of metaheuristics**. Raleigh: Lulu, 2013. Disponível em http://cs.gmu.edu/~sean/book/metaheuristics

Adicional

- BONYADI, M. R. *et al*. Particle swarm optimization for single objective continuous space problems: a review. **Evolutionary Computation**, v. 25, n. 1, p. 1-54, 2017.
- CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. **IEEE transactions on Evolutionary Computation**, v. 6, n. 1, p. 58-73, 2002.
- CLERC, M. **Standard Particle Swarm Optimisation**, 2012. Disponível em http://clerc.maurice.free.fr/pso/SPSO_descriptions.pdf
- JOHNSON, S. **Emergência: A dinâmica de rede em formigas, cérebros, cidades e softwares**. Rio de Janeiro: Zahar, 2003.

Tarefa

1. Implementar o algoritmo **Standard PSO 2011**, proposto por Maurice Clerc.
2. Compará-lo com as versões apresentadas em sala de aula (PSO Global e PSO Local) nos problemas de otimização das funções de Rastrigin e Schwefel, com $D = 2$.

Contato

César Lincoln Cavalcante Mattos
cesarlincoln@terra.com.br