

Inteligência Computacional

Luís A. Alexandre

UBI

Ano lectivo 2019-20

Conteúdo

Inteligência de enxame

Otimização por enxame de partículas

Estrutura de rede social

Algoritmo OEP

Parâmetros dum sistema OEP

OEP versus CE

Exemplos

Otimização cooperativa por enxame de partículas

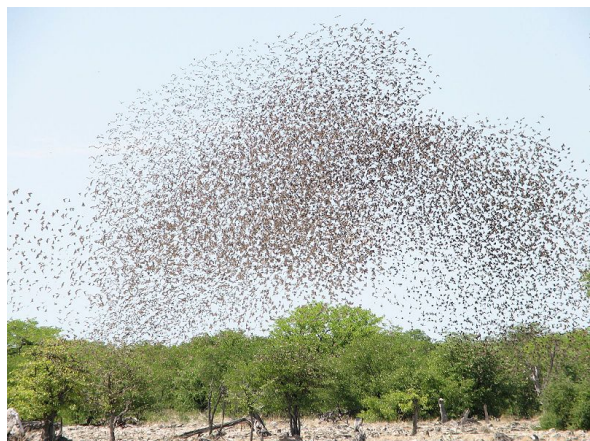
Leitura recomendada

Inteligência de enxame

Inteligência de enxame

Inteligência de enxame

Introdução



Inteligência de enxame

Introdução

- ▶ Definimos para os efeitos da disciplina, um **enxame** como sendo uma coleção estruturada de organismos que interagem.
- ▶ Esta **interação** pode ser definida de forma genética ou social.
- ▶ Exemplos de alguns indivíduos já estudados no âmbito da Inteligência de Enxame são as vespas, as abelhas, as térmitas, as formigas, os pássaros e os peixes.

Otimização por enxame de partículas

Otimização por enxame de partículas

Introdução

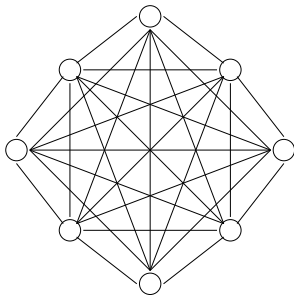
- ▶ A OEP é um **algoritmo de otimização** baseado na simulação do comportamento social de pássaros num bando.
- ▶ Aqui chamamos **partículas** aos indivíduos (pássaros).
- ▶ As partículas deslocam-se, em princípio à semelhança dos pássaros no bando, mas num espaço n -dimensional de pesquisa.
- ▶ A mudança de posição das partículas no espaço de pesquisa é baseada na tendência dos indivíduos copiarem outros indivíduos mais bem sucedidos.
- ▶ A mudança duma partícula depende das partículas vizinhas.
- ▶ A OEP é normalmente usada para determinar os extremos (máximos e mínimos) de funções não-lineares.

Estrutura de rede social

- ▶ A característica fundamental da OEP é a **interação social** entre as partículas.
- ▶ As partículas aprendem com as suas vizinhas e tendem a tornar-se mais parecidas com as mais bem sucedidas.
- ▶ A estrutura social é formada através de **vizinhanças**.
- ▶ As partículas numa mesma vizinhança podem comunicar entre si.
- ▶ Existem vários tipos de vizinhança, alguns dos quais iremos ver de seguida.
- ▶ De notar que as vizinhanças se definem com base num índice atribuído a cada partícula (não com base em medidas geométricas, como por exemplo, distância entre partículas, embora também já tenha sido usado tal critério).

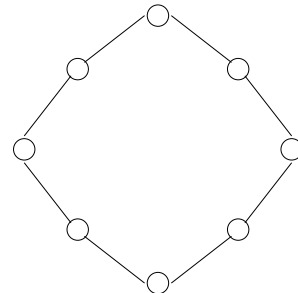
Topologia em estrela

- ▶ Cada partícula pode comunicar com todas as outras.
- ▶ Cada partícula é atraída para a melhor global (dentro do enxame).



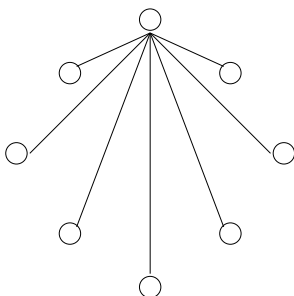
Topologia em anel

- ▶ Neste caso cada partícula comunica com os seus n vizinhos mais próximos.
- ▶ Cada partícula é atraída para a melhor partícula da sua vizinhança.
- ▶ O caso $n = 2$ encontra-se representado abaixo.



Topologia em roda

- ▶ Apenas uma partícula se encontra ligada a todas as restantes.
- ▶ Apenas esta partícula se ajusta relativamente à melhor.
- ▶ Se o resultado do ajuste for de facto um melhoramento, este será transmitido às restantes partículas.



Algoritmo OEP

- ▶ Chamamos um **enxame** a um conjunto de partículas.
- ▶ A **posição** de cada partícula representa uma **solução**.
- ▶ As partículas deslocam-se no espaço de soluções e a posição das mesmas é alterada de acordo com a sua experiência e a das suas vizinhas.
- ▶ Seja $x_i(t)$ a posição da partícula i , no instante t . Esta posição é alterada adicionando uma **velocidade** $v_i(t)$ (vezes uma unidade de tempo):

$$x_i(t) = x_i(t-1) + v_i(t) \quad (1)$$

- ▶ Esta velocidade é o motor do processo de otimização e **reflete a troca de informação** entre as partículas.
- ▶ De seguida veremos 3 formas de obter a velocidade.

Melhor partícula individual

- ▶ Neste caso, cada partícula compara a sua posição atual com a sua melhor posição até ao momento. **Não são realizadas comparações entre partículas.**
 1. Inicializar o enxame $P(t)$, sendo que a posição de cada partícula no espaço, $x_i(t)$, e a sua velocidade inicial, $v_i(t)$, são aleatórias e $t = 0$.
 2. Avaliar o desempenho F de cada partícula usando a sua posição atual, $x_i(t)$.
 3. Comparar o desempenho de cada partícula com o seu melhor desempenho até ao momento ($pbest_i$). Se $F(x_i(t)) < pbest_i$,
 - 3.1 $pbest_i = F(x_i(t))$
 - 3.2 $x_{pbest_i} = x_i(t)$
 4. Mudar a velocidade de cada partícula usando

$$v_i(t) = v_i(t-1) + \rho(x_{pbest_i} - x_i(t))$$

onde ρ é um número positivo aleatório.

5. Mover cada partícula para a sua nova posição usando a expressão (1).
6. Fazer $t = t + 1$ e voltar a 2 até convergir.

Melhor partícula individual

- ▶ Quanto mais afastada uma partícula se encontrar da posição em que obteve o seu melhor desempenho, maior será a alteração na sua velocidade de forma a que volte à sua melhor posição.
- ▶ O limite máximo para a variável ρ é um parâmetro do sistema definido pelo utilizador.
- ▶ Quanto maior for esse limite, maior será a oscilação nas trajetórias das partículas.
- ▶ Pequenos valores de ρ implicam trajetórias suaves.

Melhor partícula global

- ▶ Esta versão reflete uma **vizinhança em estrela**.
- ▶ É usada a informação da melhor partícula do enxame assim como a melhor solução individual até ao momento.
 1. Inicializar o enxame, $P(t)$, sendo que a posição de cada partícula no espaço, $x_i(t)$, é aleatória, $v_i(t) = 0$, e $t = 0$.
 2. Avaliar o desempenho F de cada partícula usando a sua posição atual, $x_i(t)$.
 3. Comparar o desempenho de cada partícula com o seu melhor desempenho até ao momento ($pbest_i$). Se $F(x_i(t)) < pbest_i$,
 - 3.1 $pbest_i = F(x_i(t))$
 - 3.2 $x_{pbest_i} = x_i(t)$
 4. Comparar o desempenho de cada partícula com o melhor desempenho global ($gbest$). Se $F(x_i(t)) < gbest$,
 - 4.1 $gbest = F(x_i(t))$
 - 4.2 $x_{gbest} = x_i(t)$

Melhor partícula global

- ▶
 5. Mudar a velocidade de cada partícula usando

$$v_i(t) = v_i(t-1) + \rho_1(x_{pbest_i} - x_i(t)) + \rho_2(x_{gbest} - x_i(t))$$
 onde ρ_1 e ρ_2 são números positivos aleatórios.
 6. Mover cada partícula para a sua nova posição usando a expressão (1).
 7. Fazer $t = t + 1$ e voltar a 2 até convergir.
- ▶ Quanto mais afastada uma partícula estiver da melhor posição global e da sua melhor posição até ao momento, maior será a mudança do seu vetor de velocidade, para a fazer aproximar das melhores soluções.

Melhor partícula global

- ▶ As variáveis ρ_1 e ρ_2 definem-se como:

$$\rho_1 = r_1 c_1 \text{ e } \rho_2 = r_2 c_2$$

com $r_1, r_2 \sim U(0, 1)$.

- ▶ c_1 e c_2 são constantes positivas (de aceleração).
- ▶ Para que as velocidade e posições das partículas não divirjam, estas constantes têm de verificar

$$c_1 + c_2 \leq 4$$

Melhor partícula local

- ▶ A versão local da OEP, $lbest$, reflete a **vizinhança em anel**.
- ▶ As partículas continuam a ser influenciadas pela melhor partícula na sua vizinhança assim como pela sua melhor posição até ao momento.
- ▶ Em relação ao algoritmo anterior, apenas os passos 4 e 5 sofrem a alteração de $gbest$ para $lbest$.
- ▶ Esta abordagem é mais lenta que a anterior mas consegue obter melhores soluções e efetua a pesquisa numa zona maior do espaço.

Cálculo da aptidão

- ▶ O passo 2 dos algoritmos anteriores implica uma **avaliação do desempenho das partículas**, ou seja, da sua **aptidão**.
- ▶ O que se usa é uma função que mede a distância das partículas relativamente ao ótimo.
- ▶ No caso de se estar a tentar achar o mínimo dada função, usa-se essa mesma função para achar o desempenho/aptidão.
- ▶ Exemplo: se pretendermos achar um máximo da função

$$f(x, y) = \sin(x) \cos(y)$$

então a função de aptidão a usar é a própria $f(x, y)$.

Convergência

- ▶ Normalmente estes algoritmos correm um número fixo de iterações.
- ▶ Uma outra forma de avaliar se se está próximo da convergência consiste em verificar se os valores das velocidades praticamente não se alteram entre duas iterações consecutivas.

Velocidade máxima

- ▶ Normalmente define-se uma **velocidade máxima**, V_{max} , para evitar que as partículas se desloquem muito rapidamente.
- ▶ Se a velocidade da partícula segundo qualquer direção for superior a V_{max} ela é colocada igual a V_{max} e de forma simétrica se a velocidade for inferior a $-V_{max}$ é colocada em $-V_{max}$.
- ▶ De notar que a imposição deste limite na velocidade não limita a zona do espaço do problema que é percorrida.

Tamanho da vizinhança

- ▶ A versão g_{best} é simplesmente a l_{best} com a vizinhança definida como todo o enxame.
- ▶ A g_{best} é mais suscetível de ficar presa em mínimos locais visto todos os indivíduos serem atraídos para o mesmo mínimo.
- ▶ Quanto menores forem as vizinhanças e maior o seu número, menos suscetível o algoritmo fica de ficar preso num mínimo local.
- ▶ Neste caso uma maior região do espaço é percorrida e nenhuma solução encontrada consegue influenciar todas as partículas do enxame.

Peso inercial

- ▶ O desempenho do algoritmo pode ser melhorado se for usada uma expressão de atualização das velocidades modificada pela introdução dum termo a que chamamos **peso inercial**, ϕ :

$$v_i(t) = \phi v_i(t-1) + \rho_1(x_{pbest_i} - x_i(t)) + \rho_2(x_{gbest} - x_i(t))$$

- ▶ Este parâmetro controla a influência das velocidades anteriores na nova velocidade.
- ▶ Valores elevados de ϕ fazem com que uma maior zona do espaço do problema seja percorrida.
- ▶ Pequenos valores concentram o algoritmo na exploração duma pequena zona.
- ▶ Normalmente o que se faz é inicializar este parâmetro com um valor elevado que vai sendo reduzido no decorrer das iterações.

Parâmetros

- ▶ Os dois parâmetros mais influentes na OEP são o peso inercial ϕ e as constantes de aceleração c_1 e c_2 .
- ▶ Para que o algoritmo OEP convirja é necessário que se verifique a seguinte relação:

$$\phi > 0.5(c_1 + c_2) - 1$$

onde $\phi \leq 1$.

OEP versus CE

- ▶ Vejamos uma breve comparação entre a OEP e a CE.
- ▶ São ambos algoritmos de otimização baseados em populações de indivíduos.
- ▶ Em ambos os casos o espaço de soluções do problema é percorrido tendo como base regras probabilísticas.
- ▶ A OEP tem memória enquanto que a CE não.
- ▶ Na OEP as mudanças são obtidas através da aprendizagem social enquanto que na CE as mudanças derivam de cross-over e mutação dos indivíduos.

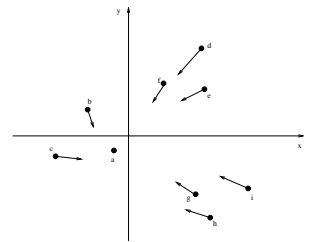
Exemplos

Exemplo 1

- ▶ Vejamos como exemplo algumas figuras relativas à aplicação da OEP à minimização da função $f(x, y) = x^2 + y^2$.
- ▶ O mínimo da função encontra-se na origem dos eixos.
- ▶ O espaço onde as partículas se deslocam é bi-dimensional (só existem 2 variáveis).
- ▶ Não usamos peso inercial.

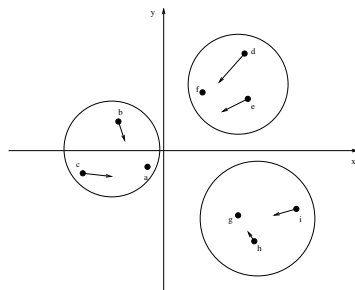
Exemplo 1

- ▶ A figura ao lado mostra o enxame inicial.
- ▶ As setas representam os deslocamentos das partículas para o caso da vizinhança em estrela (ótimo global).
- ▶ Inicialmente o *pbest* de cada partícula é o ponto inicial, logo apenas o ótimo global afeta o seu movimento.
- ▶ A partícula *a* é o ótimo global, influenciando assim todas as outras.
- ▶ As setas representam a direção e a amplitude do deslocamento que as partículas irão sofrer em direção a *a*.



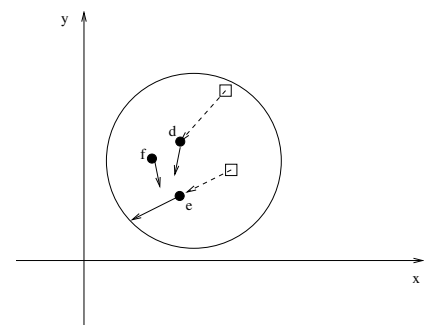
Exemplo 2

- ▶ A figura ao lado mostra o enxame inicial, para o caso duma vizinhança local.
- ▶ As partículas *b* e *c* deslocam-se em direção à *a*; as *d* e *e* em direção à *f*; e as *h* e *i* em direção à *g*.
- ▶ Porque é que não se deslocam exatamente na direção da melhor?



Exemplo 2

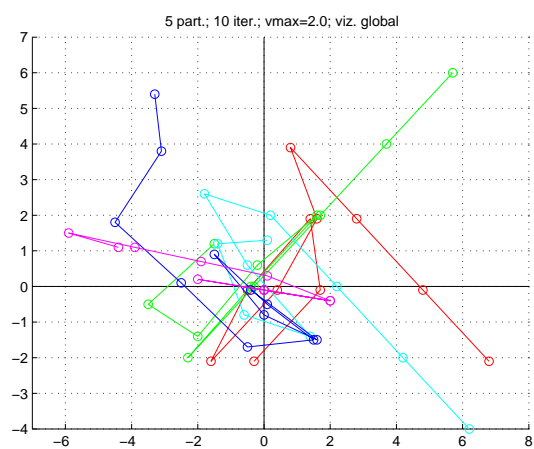
- ▶ Ao lado está representada a iteração seguinte apenas para uma das vizinhanças.
- ▶ A melhor partícula passou a ser a *e*.
- ▶ Os quadrados representam as posições anteriores.
- ▶ Porque é que a velocidade da partícula *e* não foi alterada?



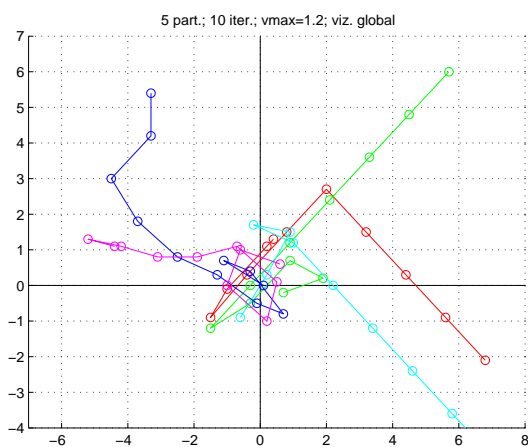
Exemplo 3

- ▶ As figuras seguintes são duma simulação com a função referida atrás, logo o mínimo está em $(0,0)$.
- ▶ Usam-se sempre 5 partículas e vizinhança em estrela, sem uso de peso inercial e $c_1 = c_2 = 2$.
- ▶ A trajetória de cada partícula está representada com uma cor diferente.
- ▶ A escala em todas as figuras é a mesma para facilitar comparações.
- ▶ Notar a diferença com as mudanças na velocidade máxima e no número de iterações.

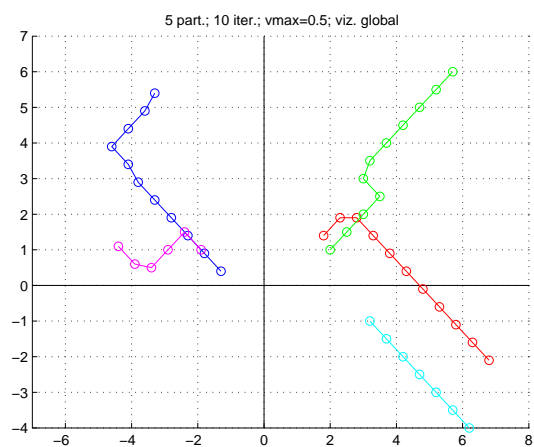
Exemplo 3



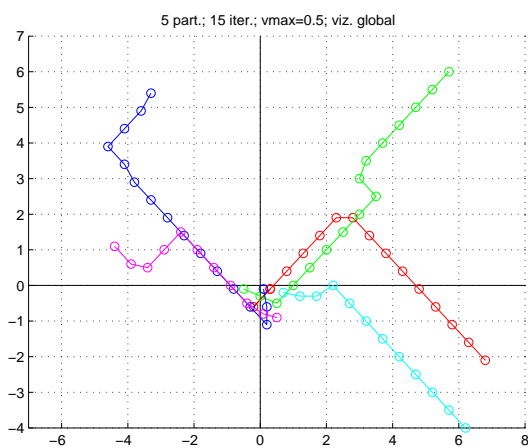
Exemplo 3



Exemplo 3



Exemplo 3



Otimização cooperativa por enxame de partículas

Otimização cooperativa por enxame de partículas (OCEP)

- ▶ O algoritmo de OEP que vimos até agora usa partículas num espaço de dimensão n igual ao número de parâmetros a otimizar.
- ▶ O enxame tem então como objetivo encontrar esses n parâmetros.
- ▶ Uma abordagem alternativa consiste em definir n enxames onde cada um otimiza apenas um parâmetro: os enxames trabalham agora num espaço 1D.
- ▶ O processo de otimização dentro de cada enxame usa a abordagem standard referida atrás.
- ▶ O problema com a OCEP é a definição da função de aptidão.
- ▶ A aptidão de cada partícula do enxame E_i não pode ser obtida de forma isolada relativamente aos outros enxames visto que representa apenas uma parte da solução n -dimensional.

Otimização cooperativa por enxame de partículas (OCEP)

- ▶ O que se faz para resolver este problema é construir um **vetor de contexto** com a informação da melhor partícula dos restantes $n - 1$ enxames e usando a informação de cada partícula do enxame E_i na posição restante.
- ▶ Esta abordagem promove a cooperação entre os diferentes enxames pois a solução é obtida a partir da melhor partícula em cada enxame.
- ▶ Esta abordagem é aconselhada nos casos em que os parâmetros a otimizar sejam independentes uns dos outros.
- ▶ A vantagem principal desta abordagem consiste na pesquisa mais fina pelos melhores parâmetros: quando tratamos de otimizar todos os parâmetros simultaneamente, uma melhor posição para uma partícula pode significar que até se piorou nalgum dos componentes da solução, embora em geral se tenha melhorado; procurando o melhor segundo cada parâmetro permitirá atingir em princípio melhores resultados.

Leitura recomendada

- ▶ Engelbrecht, cap. 16 com exceção da sec. 16.4.