

## 09 – ACO otimização por colónia de formigas

IC 22/23

### Optimização por colónia de formigas

- Características
  - O ACO considera *swarms* com um número elevado de indivíduos, podendo existir **indivíduos de diferentes características e comportamento** (No PSO, todos os indivíduos são semelhantes)
    - Apesar das diferenças, todos os indivíduos contribuem para um **objectivo comum**
    - As formigas constituem 50% de todos os insectos “sociais”
      - Existem há 100 milhões de anos, população estimada de 10e16!
      - As colónias de formigas podem atingir vários milhões (até 30 milhões) de indivíduos.



# Conceitos

- ...
  - O “Manager” Invisível
    - Numa colónia, grupos diferentes executam tarefas distintas, que incluem:
      - Reprodução – tarefa da rainha
      - Defesa – soldados
      - Busca de alimento – trabalhadoras especializadas
      - Cuidado com as crias – trabalhadoras especializadas
      - Limpeza do formigueiro - trabalhadoras especializadas
      - Construção e manutenção do formigueiro – trabalhadoras esp.
    - A distribuição das tarefas ocorre sem a necessidade de um comando central!



# Conceitos

- Estigmergia
  - O comportamento distribuído apresentado pela colónia caracteriza-se por:
    - Não existe uma coordenação central.
    - A coordenação e comunicação são baseadas nas modificações locais ao ambiente.
    - Realimentação positiva, i.e. reforço das acções.
  - A modelação artificial da colónia baseia-se no conceito de estigmergia artificial, definida por:
    - Comunicação **indirecta** através de alterações do estado do ambiente, acessíveis **localmente** aos agentes.

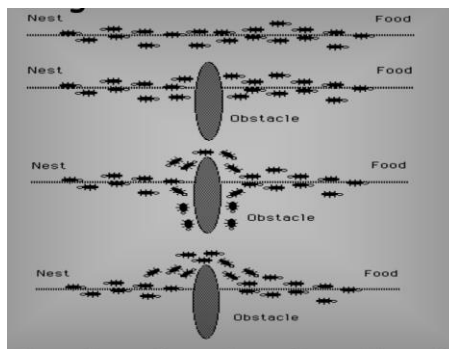
## Conceitos

- ....
  - O método de comunicação por Estigmergia existe nos Humanos?



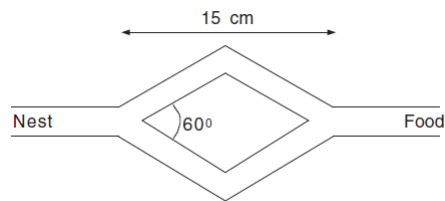
## Conceitos

- ...
  - Qual o mecanismo que permite que as formigas encontrem o caminho mais curto entre o ninho e fonte de alimento, sem nenhuma coordenação central visível?



# Estigmergia

- A Feromona
  - De facto, na procura de alimento, as formigas demonstram a capacidade de determinar o caminho mais curto entre o formigueiro e a fonte de alimento.
  - Exemplo 1 (2 caminhos com a mesma dimensão)

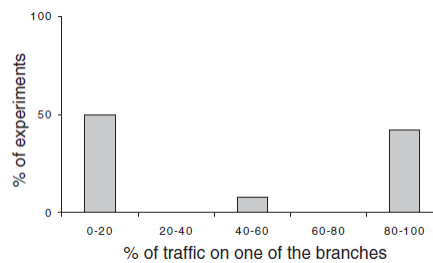


- O que acontece?

# Estigmergia

• ...

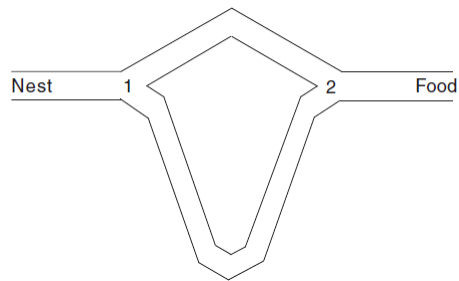
- Resultados:



- As formigas usam um ou outro caminho com igual probabilidade!

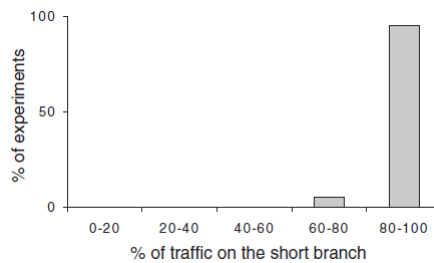
# Estigmergia

- ...
  - Experiência 2
    - Um caminho com maior comprimento que outro (neste caso, com o dobro do comprimento)



# Estigmergia

- ...
  - Resultados



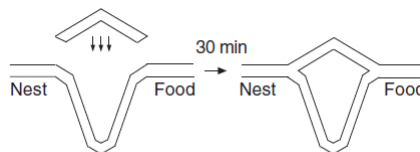
- Em todas as experiências, a maioria das formigas escolhe o caminho mais curto!

# Estigmergia

- ...
  - Análise:
    - No início, a probabilidade de seguir um dos dois caminhos é igual. Ao deslocarem-se libertam marcadores químicos (feromona).
    - Os caminhos mais curtos possuem maior concentração de feromona, pois as formigas fazem a viagem mais rapidamente.
    - As formigas escolhem o caminho com maior quantidade de feromona.
      - Considerando que estas se evaporam ao fim de algum tempo, os caminhos mais curtos são os escolhidos!

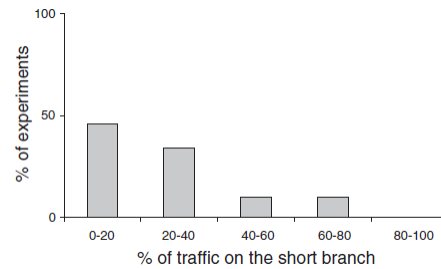
# Estigmergia

- ...
  - Experiência 3
    - Após o estabelecimento de um caminho, é adicionado um caminho mais curto ao fim de 30 minutos:



# Estigmergia

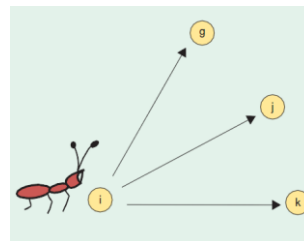
- ...
- Resultados



- Na maioria das experiências, após a inclusão do caminho mais curto, as formigas continuam a usar o caminho anteriormente estabelecido, apesar de mais longo!
  - Consegue explicar?

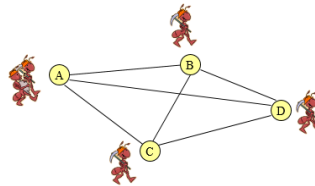
## Optimização

- Aplicação a problemas de otimização
  - A modelação de fenómeno de seguimento das feromonas permite resolver problemas de optimização, como por exemplo o bem conhecido TSP (problema do caixeiro viajante)
    - encontrar um modelo matemático que descreva com precisão o processo de estigmergia social.



# Optimização

- ...
  - TSP
    - Em cada cidade, a formiga deve decidir qual a próxima cidade a visitar com base na quantidade de feromonas depositadas em cada ligação.
    - Inicialmente a escolha deve ser feita aleatoriamente, o que é conseguido inicializando-se os pesos das ligações entre cidades com valores pequenos aleatórios;



## Optimização - TSP

- ...
  - A partir da cidade  $i$ , a probabilidade da próxima cidade a visitar pela formiga  $k$  ser a cidade  $j$ , é determinada por:

$$p_{ij,k}(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha}{\sum_{c \in C_{i,k}} \tau_{ic}(t)^\alpha} & , \quad \text{se } j \in C_{i,k} \\ 0 & , \quad \text{caso contrário} \end{cases}$$

onde:

- $C_{i,k}$  representa o conjunto de cidades ainda a visitar pela formiga;
- $\tau_{ij}(t)$  representa a quantidade de feromona entre as cidades  $i$  e  $j$  no instante  $t$ ;
- $\alpha$  representa um constante a especificar.



## Optimização - TSP

- ...

- A regra anterior pode ser melhorada se considerarmos também como critério a distância entre as cidades:

$$\phi_{ij,k}(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \eta_{ij}^\beta}{\sum_{c \in C_{i,k}} \tau_{ic}(t)^\alpha \eta_{ic}^\beta}, & \text{se } j \in C_{i,k} \\ 0, & \text{caso contrário} \end{cases}$$

onde:

- $C_{i,k}$  representa o conjunto de cidades ainda a visitar pela formiga
- $\tau_{ij}(t)$  representa a quantidade de feromona entre as cidades  $i$  e  $j$  no instante  $t$
- $\alpha$  e  $\beta$  representam constantes
- $\eta_{ij} = \frac{1}{d_{ij}}$ ,  $d_{ij}$  representa a distância entre as cidades  $i$  e  $j$

## Optimização - TSP

- ...

- ...

- Note-se que os valores de probabilidade podem ser diferentes para diferentes formigas na mesma cidade, dado que podem ter optado por caminhos distintos para a mesma cidade.
- Depois de todas as formigas percorrerem à vez o grafo, os valores das feromonas para cada aresta são atualizados.
  - Evaporação – permite “esquecer” decisões mais antigas!
  - Acumulação com base nos novos caminhos.

# Optimização - TSP

• ...

- Atualização dos valores de feromona entre cidades  $i$  e  $j$ :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

onde  $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij,k}(t)$ , representa a soma de depósitos de feromonas da formiga  $K$ , determinado por:

$$\Delta\tau_{ij,k}(t) \begin{cases} \frac{Q}{L_k(t)} & \text{se } (i,j) \in T_k(t) \\ 0 & \text{caso contrário} \end{cases}$$

$m$  representa o número total de formigas

O parâmetro  $Q$  tem o valor da mesma ordem de grandeza do comprimento que se suspeita ser a melhor rota;

$L_k(t)$  Representa o comprimento da rota percorrida pela formiga  $k$  na iteração  $t$

$T_k(t)$  Representa o conjunto de arestas da rota percorrida pela formiga  $k$  na iteração  $t$

# Optimização - TSP

• ...

- Algoritmo

1. Inicializar os depósitos de feromonas com valores pequenos aleatórios,  $\tau_{ij}(0) \sim U(0, \max)$

2. Colocar todas as formigas  $k \in 1, \dots, m$  na cidade de origem

3. Seja  $T^+$  a rota mais curta de comprimento  $L^+$ , e  $n$  o número de cidades

4. Para  $t = 1$  to  $t_{\max}$ :

4.1 Para cada formiga, construir a rota  $T_k(t)$  escolhendo a próxima cidade  $n-1$  vezes com probabilidade dada por  $\phi_{ij,k}(t)$

4.2 Calcular o comprimento da rota de cada formiga  $L_k(t)$

4.3 Se for encontrada uma rota melhor, actualizar  $T^+$  e  $L^+$

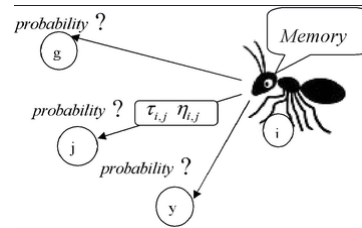
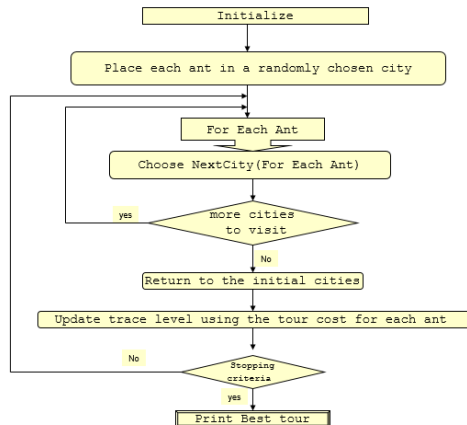
4.4 Actualizar os depósitos de feromonas:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

5. Devolver a melhor rota  $T^+$

## Optimização - TSP

• ...



## Optimização - TSP

• ...

- Qual o número ideal de formigas?
  - Se o número de formigas for elevado, o custo computacional é elevado
  - Se o número é reduzido, existe convergência para rotas sub-ótimas
- Qual o melhor valor dos parâmetros?
  - Se  $\alpha=0$ , não será usada informação sobre as feromonas!
  - Se  $\beta=0$ , será apenas usada a informação sobre as feromonas - pode conduzir à convergência para rotas sub-ótimas

# Otimização - TSP

## • Variantes

### • Elitismo

- Apenas as “melhores formigas” adicionam feromona de forma proporcional à qualidade dos seus caminhos.
- Tem como consequência condicionar a pesquisa de todas as formigas para a construção de uma solução contendo as ligações das melhores rotas atuais:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) + n_e \Delta\tau_{ij}^e(t)$$

$$\Delta\tau_{ij}^e(t) = \begin{cases} \frac{Q}{f(\tilde{x}(t))} & \text{if } (i,j) \in \tilde{x}(t) \\ 0 & \text{otherwise} \end{cases}$$

e: número de formigas elite

melhor rota atual

# Otimização - TSP

## • ...

### • ‘Antabu’

- Adapta o algoritmo para incluir uma pesquisa local com base na pesquisa tabu (movimentos proibidos)
- Modifica método de atualização de feromona:
  - O depósito em cada ligação do caminho é proporcional à qualidade deste.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \left(\frac{\rho}{f(x^k(t))}\right) \left(\frac{f(x^-(t)) - f(x^k(t))}{f(\tilde{x}(t))}\right)$$

Custo do pior caminho até agora encontrado

Custo do caminho encontrado pela formiga k

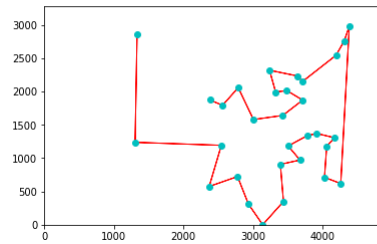
Custo do melhor caminho encontrado até agora

# Optimização - TSP

- Implementação

- <https://github.com/ppoffice/ant-colony-tsp/blob/master/aco.py>

cost: 16003.03696235567, path: [13, 11, 12, 10, 22, 15, 4, 5, 6, 1, 3, 7, 8, 9, 21, 20, 17, 2, 16, 18, 23, 19, 24, 25, 27, 26, 29, 30, 28, 0, 14]

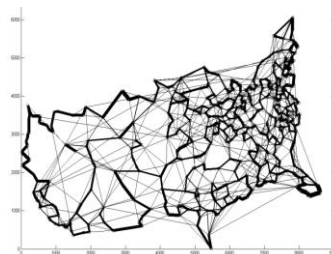


# Optimização - TSP

- ...

- Tsplib95

- Biblioteca para utilizar diversas instâncias do problema TSP
    - <https://pypi.org/project/tsplib95/>
    - <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>



# Clustering

- Adaptação do comportamento das formigas a um algoritmo de clustering:
  - Verifica-se que várias espécies de formigas agrupam diferentes objetos, ainda que inicialmente estes de encontrem distribuídos de forma aleatória.
  - Assumimos que:
    - As formigas deslocam-se aleatoriamente e observam uma área circundante – Vizinhança;
    - Podem “pegar” ou “largar” objetos de acordo com uma determinada probabilidade;
    - O espaço de pesquisa é definido por uma grelha, e em cada posição pode conter apenas um objeto (e uma formiga).

# Clustering

- ...
  - Considere-se que no instante “t”, existe uma formiga na posição “r” onde se encontra o objecto ou vector de dados “z”
  - Na vizinhança desta formiga, a densidade local do vector de dados é determinada por:

$$f(z_i) = \frac{1}{s^2} \sum_{z_j \in V_{sxs}(r)} \left( 1 - \frac{d(z_i, z_j)}{\alpha} \right)$$

onde  $d(z_i, z_j)$  define a semelhança entre os objectos  $z_i$  e  $z_j$

A constante  $\alpha$  determina a escala de semelhança

Define a vizinhança  $sxs$  da formiga que se encontra na posição  $\alpha$ , como  $V_{sxs}(r)$

# Clustering

- ...

- A probabilidade de pegar num objecto é dada por:

$$p_p(z_i) = \left( \frac{k_1}{k_1 + f(z_i)} \right)^2$$

onde  $k_1$  representa uma constante a definir

- Qual a valor desta probabilidade quando a vizinhança se encontra densamente povoada?
  - Será pequena

# Clustering

- ...

- A probabilidade de largar um objecto é dada por:

$$p_d(z_i) = \begin{cases} 2f(z_i) & \text{se } f(z_i) < k_2 \\ 1 & \text{se } f(z_i) \geq k_2 \end{cases}$$

onde  $k_2$  representa uma constante a definir

- Qual a valor desta probabilidade quando a vizinhança se encontra densamente povoada?
  - Será elevada

# Clustering

- Algoritmo

1. Inicialização

- 1.1 Colocar aleatoriamente os vectores de dados  $z_i$  na grelha, de forma aleatória
- 1.2 Colocar aleatoriamente as formigas na grelha
- 1.2 Definir os valores dos parâmetros  $k_1, k_2, \alpha, s$  e número máximo de iterações  $t_{max}$

# Clustering

- ...

2. Para  $t = 1$  to  $t_{max}$ , para cada formiga:

- 2.1 Se a formiga não tiver carga e a posição está ocupada por um item  $z_i$

- 2.1.1 Determinar  $f(z_i)$  e  $p_p(z_i)$

- 2.1.2 Se  $U(0,1) \leq p_p(z_i)$ , apanhar o objecto  $z_i$

- 2.2 Senão, Se a formiga transportar  $z_i$  e o local está vazio

- 2.2.1 Determinar  $f(z_i)$  e  $p_d(z_i)$

- 2.1.2 Se  $U(0,1) \leq p_d(z_i)$ , largar o objecto  $z_i$

- 2.3 Mover aleatoriamente a formiga para um local da vizinhança não ocupado por outra formiga

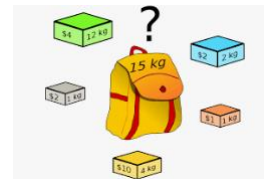


# Clustering

- ...
  - **Análise do algoritmo**
    - A grelha deve possuir mais nós do que formigas
    - A grelha deve possuir mais nós do que objectos
    - O algoritmo tende a criar mais clusters do que é habitualmente necessário. Este problema pode ser resolvido se:
      - Definir velocidades diferentes para as formigas – formigas velozes produzem clusters menos precisos.
      - Usar memória de curto prazo, de forma a que as formigas se lembrem dos últimos objectos que largou e em que posições. Ao encontrar um objecto semelhante a um objecto anterior, desloca-se na direcção desse objecto.

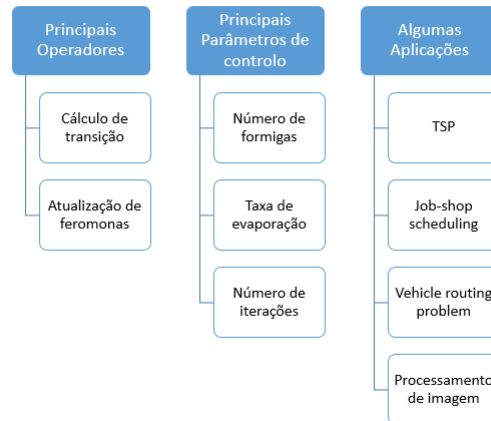
# Aplicações

- Podemos aplicar o 'ACO' a qualquer problema onde se possa definir:
  - Uma **representação do problema em forma de grafo**
    - Espaço de pesquisa discreto
  - Uma **heurística** para a escolha da próxima aresta da solução
  - Um método para **satisfação de restrições**
    - Garante que são geradas apenas soluções válidas
  - Um método de **construção de soluções**



# Aplicações

• ...



# Referências

• ...

- Engelbrecht, Andries P. *Computational intelligence: an introduction*. John Wiley & Sons, 2007. Cap. 17.
- <http://www.aco-metaheuristic.org/>
- Dorigo, Marco, and Thomas Stützle. "Ant colony optimization: overview and recent advances." *Handbook of metaheuristics*. Springer, Cham, 2019. 311-351.
- <https://github.com/ppoffice/ant-colony-tsp/blob/master/aco.py>.
- <https://github.com/madvn/ACO>