Inteligência Computacional

Luís A. Alexandre

UBI

Ano lectivo 2019-20

Conteúdo

peterpanama.wordpress.com/2009/05/16/army-ants/

Otimização por colónia de formigas

Introdução

- A OCF difere da OEP no sentido em que esta última lidava com um conjunto de indivíduos idênticos.
- Na OCF temos indivíduos distintos, tanto em termos de morfologia como de funções.
- Na natureza os insetos sociais são todas as espécies de térmitas e de formigas e algumas espécies de vespas e de abelhas.
- As formigas têm mais de 12.000 espécies e ocupam todos os continentes exceto a Antártida, Gronelândia, Islândia, partes da Polinésia, o Hawai, e outras pequenas ilhas remotas que não têm espécies indígenas.
- Estas colónias de formigas são constituídas por entre 30 a vários milhões de indivíduos.

Tarefas numa colónia de formigas

- Dado serem seres tão bem sucedidos, o seu comportamento deve ser altamente otimizado.
- Uma colónia de formigas exige a execução de várias tarefas distintas.
- Estas tarefas são executadas por grupos de formigas distintos:
  - reprodução: rainha
  - defesa: formigas soldado
  - recolha de comida: formigas trabalhadoras especializadas
  - cuidado das crias: formigas trabalhadoras especializadas
  - limpeza do formigueiro: formigas trabalhadoras especializadas
  - construção e manutenção do formigueiro: formigas trabalhadoras especializadas

Estigmergia

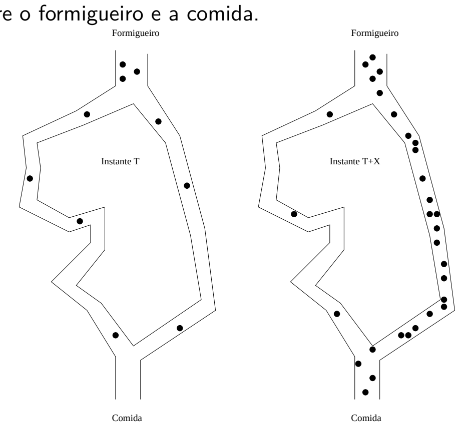
- A estigmergia é um termo inventado pelo biólogo Pierre-Paul Grassé em 1959 no âmbito do estudo do comportamento das térmitas.
- Definiu-o como: 'Estimulação dos trabalhadores através do desempenho que alcançaram'.
- A estigmergia na natureza é caracterizada por:
  - A falta de coordenação centralizada;
  - A comunicação e coordenação entre os indivíduos duma colónia é baseada nas modificações locais do ambiente;
  - Reforço positivo.

Estigmergia

- A modelação artificial das colónias de formigas é baseada no conceito de estigmergia artificial, definido como: ‘comunicação indireta através de alterações numéricas no estado do ambiente que são acessíveis apenas localmente aos agentes’.
- Desta forma, a essência da modelação de aspetos das colónias de formigas reside na determinação de um modelo que permita descrever as características de estigmergia desses aspetos a modelar.

Feromonas

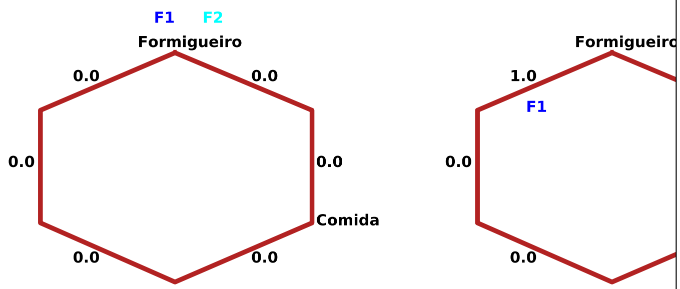
- As formigas têm a capacidade de encontrar sempre o caminho mais curto entre o formigueiro e a comida.



Feromonas

- Inicialmente (figura da esquerda) ambos os caminhos são igualmente prováveis.
- As formigas largam feromonas (são marcadores químicos) ao deslocarem-se.
- As feromonas evaporam-se ao fim de algum tempo: se um caminho não é usado durante muito tempo, fica sem feromonas.
- As formigas escolhem deslocar-se no caminho que contém maior concentração de feromonas.
- Imaginemos que o caminho mais curto tem metade do comprimento do mais longo.
- Num dado intervalo de tempo, enquanto uma formiga parte do formigueiro e chega à comida pelo percurso mais longo, outra que use o mais curto consegue ir e voltar ao formigueiro nesse intervalo de tempo.
- Deste modo, o caminho mais curto fica com o dobro da concentração de feromonas relativamente ao mais longo.

Feromonas: exemplo



OCF

- Vejamos como usar a abordagem das feromonas na resolução de um problema de otimização: o problema do caixeiro viajante.
- O problema representa-se num grafo não dirigido, pesado, em que cada vértice representa uma cidade e cada aresta a ligação entre um par de cidades. O peso de cada aresta é a quantidade de feromona na ligação respetiva.
- Em cada cidade a tarefa da formiga é escolher a próxima cidade a visitar, baseada numa regra probabilística que depende da quantidade de feromonas depositadas nos diferentes caminhos.
- Inicialmente essa escolha é aleatória, o que é conseguido inicializando a quantidade de feromonas em cada caminho com um valor pequeno, aleatoriamente.

OCF

- A probabilidade de a próxima cidade a ser visitada pela formiga k que se encontra na cidade i, ser a cidade j, é dada por

$$\Phi_{ij,k}(t) = \frac{\tau_{ij}(t)^{\alpha}\eta_{ij}^{\beta}}{\sum_{c\in C_{i,k}} \tau_{ic}(t)^{\alpha}\eta_{ic}^{\beta}} \quad (1)$$

se $j \in C_{i,k}$. Caso contrário, $\Phi_{ij,k}(t) = 0$.

- Os componentes desta expressão são os seguintes:
    - $\tau_{ij}(t)$ é a intensidade da feromona na aresta (i,j), na iteração t
    - α e β são constantes
    - $C_{i,k}$ é o conjunto de cidades adjacentes a i que a formiga k pode visitar partindo da cidade i
    - $\eta_{ij} = 1/d_{ij}$ e d_{ij} é a distância entre as cidades i e j

OCF

- α vai permitir controlar a importância da intensidade das feromonas na escolha da próxima cidade
- β serve para controlar a importância de η_{ij}
- Por sua vez, η_{ij} é informação local sobre o interesse em visitar a cidade j partindo da i: quanto mais próxima j se encontra de i, maior é o interesse em visitá-la de seguida.
- Cada formiga percorre à vez o grafo. Após todas o terem percorrido, os valores da intensidade da feromona nas arestas pelas quais cada formiga passou são atualizados usando a seguinte expressão:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

- A constante $\rho \in [0,1]$ é o fator de esquecimento que modela a evaporação das feromonas.

OCF

- $\Delta\tau_{ij,k}(t)$ é o depósito de feromona da formiga k na aresta que liga as cidades i e j, na iteração t, e é dado por

$$\Delta\tau_{ij,k}(t) = \begin{cases} Q/L_k(t) & \text{se} \quad (i,j) \in T_k(t) \\ 0 & \text{caso contrário} \end{cases} \tag{3}$$

- O parâmetro Q tem um valor da mesma ordem de grandeza do comprimento que suspeitamos terá a melhor rota.
- $L_k(t)$ é o comprimento da rota percorrida pela formiga k na iteração t.
- Definimos $T_k(t)$ como o conjunto das arestas do caminho percorrido na iteração t pela formiga k.
- A soma dos depósitos $\Delta\tau_{ij}(t)$ de todas as formigas é

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij,k}(t) \tag{4}$$

onde m é o número total de formigas.

Parâmetros para a OCF

- Vejamos algumas considerações relativas a parâmetros da OCF.
- Se m for elevado teremos um custo computacional alto.
- Se m for pequeno, teremos convergência para rotas sub-ótimas.
- Se $\beta = 0$ só será usada informação das feromonas, o que pode levar a que se obtenham rotas sub-ótimas.
- Se $\alpha = 0$ não será usada informação das feromonas, tornando o algoritmo numa pesquisa estocástica.

Algoritmo OCF

- Seja T^+ a melhor rota e L^+ o seu comprimento.
- Seja n o número de cidades.

1. Inicializar $\tau_{ij}(0) \sim U(0, max)$, com max pequeno.
2. Para t de 1 até t_{max} fazer:
   - 2.1 Para cada formiga k, fazer:
     - 2.1.1 Colocá-la na cidade de origem.
     - 2.1.2 Construir a rota $T_k(t)$ escolhendo a próxima cidade $(n-1)$ vezes com probabilidade dada por $\Phi_{ij,k}(t)$.
     - 2.1.3 Calcular o comprimento da sua rota, $L_k(t)$.
     - 2.1.4 Se for encontrada uma rota melhor, atualizar T^+ e L^+.
   - 2.2 Atualizar os depósitos de feromonas usando a equação (2).
3. Devolver a melhor rota T^+

Clustering por colónia de formigas

Clustering por colónia de formigas

- Vejamos agora como alguns dos comportamentos das formigas podem permitir a criação de algoritmos de agrupamento (clustering) de dados.
- Várias espécies de formigas guardam os cadáveres em cemitérios de forma a manterem limpos os formigueiros.
- Estudos verificaram que as formigas agrupam os cadáveres ao fim de algumas horas, cadáveres estes que se encontravam inicialmente aleatoriamente distribuídos.
- Embora não se compreenda ainda totalmente este comportamento, é possível modelá-lo para criar um algoritmo de clustering.

Clustering por colónia de formigas

- A ideia é que as formigas possam percorrer o espaço do problema, pegando ou largando itens de acordo com uma dada probabilidade.
- Iremos assumir para simplificar a abordagem que existe apenas um tipo de objetos.
- Iremos ainda assumir que o espaço do problema é uma grelha, e que em cada posição da grelha podemos ter apenas uma formiga e apenas um objeto (embora possam existir simultaneamente uma formiga e um objeto na mesma posição da grelha).
- Os objetos serão vetores de dados z_i.

Definição do algoritmo de clustering

- Primeiro é necessário definir uma função de dissemelhança $d(z_i, z_j)$ entre dois vetores. Podemos usar, por exemplo, a distância euclidiana.
- Vamos usar esta medida para efetuar um clustering que obedeça às seguintes propriedades:
  - Distâncias intra-cluster devem ser pequenas. A distância entre 2 pontos do mesmo cluster deve ser pequena.
  - Distâncias inter-cluster devem ser grandes. A distância entre 2 pontos de clusters diferentes deve ser grande.
- As formigas deslocam-se de forma aleatória no espaço, observando uma área circundante (vizinhança) de $s \times s$ posições.
- Definimos a vizinhança $s \times s$ da formiga que se encontra na posição r como $V(s, r)$.

Definição do algoritmo de clustering

- Consideremos que no instante de tempo t, a formiga se encontra na posição r onde se encontra o vetor de dados z_i.
- A densidade local do vetor de dados na vizinhança da formiga é dada por
$$f(z_i) = \frac{1}{s^2} \sum_{z_j \in V(s,r)} \left(1 - \frac{d(z_i, z_j)}{\alpha}\right) \tag{5}$$
quando $f(z_i) > 0$, caso contrário $f(z_i) = 0$.
- $f(z_i)$ vai medir a semelhança entre z_i e os restantes vetores na vizinhança.
- A constante α controla a escala da semelhança permitindo definir quando é que 2 vetores são agrupados.

Probabilidades de pegar e largar objetos

- A probabilidade de pegar num objeto é dada por
$$p_p(z_i) = \left(\frac{k_1}{k_1 + f(z_i)}\right)^2 \tag{6}$$
onde k_1 é uma constante positiva não nula.
- Quando a vizinhança se encontra densamente povoada, $f(z_i)$ é grande e $p_p(z_i)$ pequena.
- A probabilidade de largar um objeto é dada por
$$p_d(z_i) = \begin{cases} 2f(z_i) & \text{se } f(z_i) < k_2 \\ 1 & \text{se } f(z_i) \geq k_2 \end{cases} \tag{7}$$
onde k_2 é uma constante.
- Quando a vizinhança se encontra densamente povoada, $f(z_i)$ é grande e $p_d(z_i)$ é também elevada.

Clustering por colónia de formigas: algoritmo

1. Inicialização:
   - 1.1 Colocar os dados z_i aleatoriamente na grelha
   - 1.2 Colocar as formigas aleatoriamente na grelha
   - 1.3 Escolher valores para k_1, k_2, α, s e para o número máximo de instantes de tempo t_{max}.
2. Para $t = 1$ até t_{max} e para cada formiga, fazer:
   - 2.1 Se a formiga não tiver carga, e a sua posição estiver ocupada por um item z_i:
     - 2.1.1 Achar $f(z_i)$ e $p_p(z_i)$
     - 2.1.2 Se $U(0,1) \leq p_p(z_i)$, apanhar z_i
   - 2.2 Senão, se a formiga estiver a carregar um vetor z_i e o local estiver vazio:
     - 2.2.1 Achar $f(z_i)$ e $p_d(z_i)$
     - 2.2.2 Se $U(0,1) \leq p_d(z_i)$, largar z_i
   - 2.3 Mover aleatoriamente a formiga para um local vizinho não ocupado por outra formiga.

Comentários ao algoritmo anterior

- A grelha tem de ter mais posições que o número de formigas a usar.
- A grelha tem de ter mais posições que vetores de dados.
- O algoritmo tem tendência a criar mais clusters dos que os que normalmente são usados, fazendo um overfit aos dados.
- Uma forma de resolver este problema é fazer com que cada formiga se lembre dos últimos m vetores que largou e em que posições isso ocorreu. Ao apanhar um vetor semelhante a um dos que já apanhou anteriormente, deve deslocar-se na direção desse vetor anteriormente largado e semelhante ao atual.
- Isto fará com que a probabilidade de largar o atual elemento próximo do outro que lhe era semelhante aumente, fazendo assim com que existam menos clusters (que, naturalmente, serão maiores).

Aplicações da OCF

Aplicações da OCF

- Os algoritmos baseados em colónias de formigas já foram usados na resolução de muitos problemas reais.
- Um dos primeiros problemas foi o do caixeiro viajante (TSP).
- Os algoritmos de OCF podem ser aplicados a qualquer problema em que se possam definir os seguintes aspetos:
  1. Uma **representação** na forma de grafo que represente o espaço de pesquisa discreto.
  2. Uma **heurística** para a escolha da próxima aresta da solução.
  3. Um método de **satisfação de restrições** que garanta que apenas são geradas soluções realistas.
  4. Um método de **construção de soluções** que defina a forma de construção das mesmas.

Problema do caixeiro viajante

- Este problema é NP-hard, logo não é fácil obter soluções.
- **Definição**: dado um conjunto de n cidades o objetivo é encontrar o menor caminho que visite todas as cidades apenas uma vez.
- Seja v uma sequência de nomes de cidades (uma solução), e $v(i)$ seja a i-ésima cidade visitada. Então $P(n)$ é o conjunto de todas as permutações de $\{1, \ldots, n\}$, que é o nosso espaço de pesquisa.
- O objetivo é então encontrar a permutação ótima:

  $$v^* = \arg \min_{v \in P(n)} f(v)$$

  onde

  $$f(v) = \sum_{i=1}^{n} d_i$$

  é a função objectivo (o comprimento do percurso), com d_i a representar a distância entre as cidades $v(i)$ e $v(i+1)$. No caso de $i = n$, temos d_i como sendo a distância entre a última cidade ($v(n)$) e a primeira ($v(1)$).

Problema do caixeiro viajante

- Vejamos então os aspetos citados atrás, neste problema concreto.
- A **representação** sob a forma dum grafo é feita considerando cada cidade um nodo, cada ligação entre 2 cidades uma aresta e a respetiva distância como sendo o peso da aresta.
- A **heurística** para o interesse em colocar a cidade j após a i na solução é dada por:

  $$\eta_{ij}(t) = \frac{1}{d_{ij}(t)}$$

  onde $d_{ij}(t)$ representa a distância entre as cidades i e j no instante t.

Problema do caixeiro viajante

- **Satisfação de restrições**. Temos duas restrições neste problema:
  - todas as cidades têm de ser visitadas;
  - cada cidade só pode ser visitada uma vez.
- Para garantir a verificação da segunda restrição, só é adicionada uma cidade a uma solução se ela ainda não estiver na solução.
- Para garantir a verificação da primeira restrição basta exigir que a solução contenha n (o número total de) cidades (o que em conjunto com a segunda restrição garante a sua verificação).
- **Construção da solução**: as formigas são colocadas em cidades aleatórias (o algoritmo descrito atrás é um caso particular onde colocamos as formigas sempre a partir da cidade de origem) e cada uma vai construindo uma solução de forma incremental, selecionando a próxima cidade usando as probabilidades de transição.

Leitura recomendada

- Engelbrecht, cap. 17.