Otimização com Colônias de Formigas

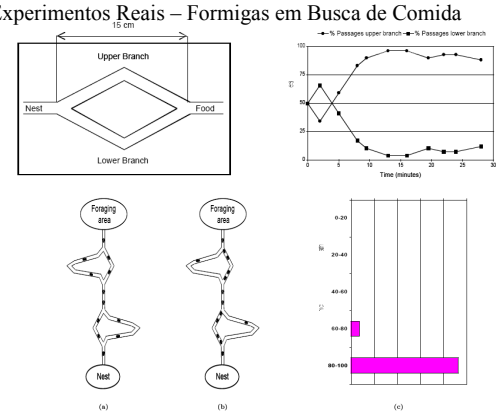
Computação Natural

Gisele L. Pappa

Introdução

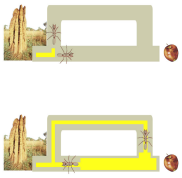
- Baseados no comportamento real dos formigueiros
- Principais características:
  - Comportamento emergente
    - Cada um contribui pouco, mas o conjunto leva a uma boa solução
  - Feedback positivo
  - Estigmergia (comunicação por feromônio)

Experimentos Reais – Formigas em Busca de Comida



Colônias de Formigas

Inicialmente, cada formiga escolhe um caminho aleatoriamente



Formigas depositam feromônio enquanto andam (inicialmente, os 2 caminhos tem mais ou menos a mesma quantidade de feromônio)

Depois de um período de tempo fixo, a formiga seguindo o caminho mais curto vai depositar mais feromônio no seu caminho (formigas se movem com a mesma velocidade). Assim, caminhos mais curtos terão uma maior concentração de feromônio

Quanto mais feromônio no caminho, maior a probabilidade das próximas formigas escolherem aquele caminho (feedback positivo)

Com o tempo… toda a colônia de formigas vai convergir para o caminho mais curto

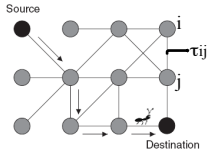
Idéias Básicas

- As soluções não são representadas pelas formigas, e sim pelos caminhos percorrido por elas
  - Uma formiga constrói o caminho (solução candidata) incrementalmente, normalmente adicionando um componente de cada vez a solução
- Conforme as formigas se movem no espaço, elas depositam uma quantidade de feromônio proporcional a qualidade da solução
- Quando as formigas têm que que escolher entre dois caminhos, a probabilidade delas seguirem o melhor caminho é dada de acordo com a concentração de feromônio

Otimização de Colônia de Formigas

Em sua versão mais simples

- Assuma um grafo conectado, onde buscamos o menor caminho de um ponto A a um ponto B
- Associado a cada aresta do grafo está uma quantidade de feromônio τ
- Cada formiga é capaz de sentir (ler) ou deixar (escrever) feromônio



Otimização de Colônia de Formigas

- Uma formiga escolhe uma aresta para continuar seu caminho de acordo com uma probabilidade, que depende da concentração de feromônio de cada aresta.
- O feromônio em cada aresta é atualizado conforme as formigas caminham
- Cada vez que a formiga caminha, o feromônio na aresta deve ser atualizado
  - Existe uma taxa de evaporação para evitar convergência

Otimização de Colônia de Formigas

- A versão mais geral do ACO inclui também uma função (η) *(desirability)* para medir a qualidade de cada componente que pode ser adicionado a uma solução candidata parcial
  - Usa informação local, e dependente do problema em questão
- Essa função é equivalente a *fitness* em algoritmos evolucionários.

ACO (maxIt, N, τ_o)

Inicializa τ_{ij} (igualmente para cada aresta)

Distribui cada uma das k formigas em um nó selecionado aleatoriamente

t = 1

while (t < maxIt) do //número de iterações

for i = 0 to N do //para cada formiga

1. Constrói uma solução aplicando uma regra de transição probabilística (*e*-1) vezes // *e* é o número de arestas do grafo
2. Avalia o custo de cada solução construída
3. Atualiza melhor solução
4. Atualiza as trilhas de feromônio

end for

t = t + 1

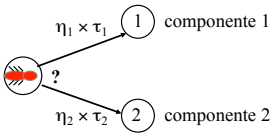
end while

Principais elementos de um algoritmo de OCF

- Representação do problema
  - Especificar os elementos que as formigas usarão para construir incrementalmente a solução para um problema
  - Guarantir a construção de soluções válidas

Principais elementos de um algoritmo de OCF

- Regra de transição probabilística: baseada no valor da função η e na quantidade de feromônio τ associada a cada componente de uma solução candidata
  - Essa regra decide que componente será inserido na solução parcial
  - Normalmente, a probabilidade de escolher um componente *i* é proporcional ao produto $\eta_i \times \tau_i$



Principais elementos de um algoritmo de OCF

- Regra para atualização do feromônio: que especifica como atualizar a quantidade de feromônio (τ) associada a cada componente inserido no caminho seguido por uma formiga
  - Feromônio aumenta proporcionalmente a qualidade do caminho (solução)
  - Usa um mecanismo global, e uma estratégia de adaptação independente do problema

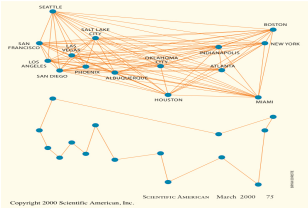
$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau,$$

onde $\rho \in (0,1]$ é a taxa de evaporação de feromônio.

ACO para o problema do Caxeiro Viajante

- Assumimos um grafo totalmente conectado (existe uma aresta entre cada par de cidade (i,j))
- Feromônio é associado com arestas

  τ_{ij} corresponde ao feromôneo deixado quando a formiga caminha da cidade i para a cidade j



ACO para o problema do Caxeiro Viajante

- Para cada formiga:
  - Escolhemos uma cidade s de início
  - Utilizamos valores de feromônio e da função para construir incrementalmente um caminho (adicionando uma cidade de cada vez ao caminho atual), considerando que:
    - A probabilidade da formiga ir da cidade i para a cidade j é proporcional a quantidade de feromônio τ_{ij} na aresta (i,j) e ao valor de η_{ij}
    - A aresta (i,j) só pode ser adicionada ao caminho se a cidade j ainda não foi visitada
    - O caminho é construído até que todas as cidades tenham sido visitadas, e no último passo uma aresta da cidade final para a inicial é inserida

ACO para o problema do Caxeiro Viajante

- A função η_{ij} incorpora informações específicas do problema
  - Calculada de maneira local, baseada apenas no custo da aresta (i,j)
  - No problema do caxeiro, utilizamos: $\eta_{ij} = 1 / d_{ij}$
- A quantidade de feromônio τ_{ij} incorpora o resultado da aprendizagem (adaptação) da colônia como um todo, tentando vários caminhos
  - O valor é atualizado a cada iteração, baseado no custo (global) do caminho

ACO para o problema do Caxeiro Viajante

- Uma formiga na cidade i se move para a cidade ainda não visitada j com probabilidade:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\Sigma_{k \in N(i)}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}}$$

α, β são pesos, e.g. $\alpha = 1$, $\beta = 2$
$N(i)$ é o conjunto de vizinhos elegíveis de i
(i.e., nós que ainda não foram vizitados)

O denominador é um fator de normalização, para que: $0 \leq p_{ij} \leq 1$

- Essa fórmula representa a regra de transição proporcional aleatória

Atualização de Feromônio

- Feromônio é atualizado depois que todas as formigas constrõem um caminho

1. Aplica-se uma regra de evaporação de feromônio, diminuindo seu valor em cada nó por uma constante:

   $$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \forall \text{aresta}(i,j) \in E,$$

   onde ρ é a taxa de evaporação de feromônio, $0 < \rho \leq 1$
2. Todas as formigas depositam uma quantidade $\Delta\tau_{ij}$ de feromônio nas arestas (i,j) pelas quais elas passaram durante seu caminho, ou seja, para cada formiga

   $$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}, \text{ onde } \Delta\tau_{ij} = \begin{cases} 1/C \text{ se a formiga passou pela aresta } (i,j) \\ 0 \text{ cc} \end{cases}$$

   onde C é o custo total do caminho

Variações do método de atualização do Feromônio

- Inicialmente, 2 versões eram muito utilizadas:
  1. Atualizar o feromônio logo após a chegada a uma nova cidade
     - A quantidade de feromônio depositada por uma formiga é inversamente proporcional ao tamanho da aresta que acabou de ser visitada
  2. Atualizar o feromônio apenas depois que uma formiga construiu o caminho completo
     - A quantidade de feromônio depositada por uma formiga é proporcional a qualidade do tour
- Hoje, a versão (2) é mais utilizada na prática, pois funciona melhor
  - Por quê?

Variações do método de atualização do Feromônio

- *Elitista* – adiciona mais feromônio às arestas do melhor caminho encontrado até o momento (desde a primeira iteração)
- *Baseado em Rank*– a quantidade de feromônio depositada por cada formiga diminui de acordo com sua posição no rank;
- *Max-Min :*
  - Apenas uma formiga atualiza o feromônio a cada iteração: ou a melhor formiga da iteração ou a melhor desde o princípio
  - Porém, nesse caso os valores de feromônio em cada aresta são limitados entre $[\tau_{min}, \tau_{max}]$, evitando que uma aresta se torna muito forte ou muita fraca

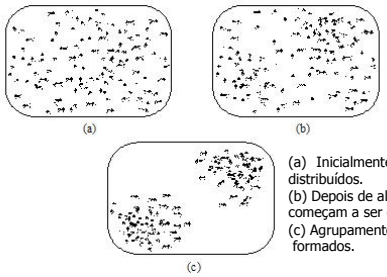
Parâmetros

- Número máximo de iterações
- Número de formigas
  - Normalmente igual ao número de arestas do grafo
- Taxa de feromônio inicial τ_0
- Pesos para concentração de feromônio (α) e qualidade da função (β) quando calculando a probabilidade de uma formiga escolher um caminho ou outro
- Taxa de evaporação

ACO inspirados em outras atividades

- ACO visto até agora considerava como as formigas buscavam por comida
- Exitem outras atividades que podem ser utilizadas como inspiração
  - Organização dos cemitérios
  - Organização de larvas
    - Agrupamento

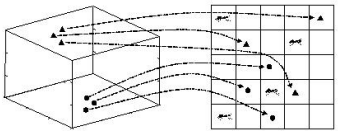
ACO para Agrupamento



(a) Inicialmente, corpos aleatoriamente distribuídos.
(b) Depois de algum tempo, corpos começam a ser empilhados.
(c) Agrupamentos de corpos são formados.

OCF para Agrupamento

- Idéias gerais
  - Formigas andam em uma matriz 2D
  - Máximo de uma formiga por célula



  - Formigas movem objetos a fim de agrupá-los

OCF para Agrupamento

- Itens isolados devem ser recolhidos e depositados em um novo local, com probabilidade

$$p_p = \left(\frac{k_1}{k_1 + f}\right)^2$$ onde f é a fração de itens próximos a formiga, e k_1 é uma constante.

– f >> k1, probabilidade de pegar um objeto numa região com muitos outros objetos é pequena, e vice-versa

- A probabilidade de uma formiga depositar um item sendo carregado é

$$p_d = \left(\frac{f}{k_2 + f}\right)^2$$ onde k_2 é uma outra constante.

– f << k2, probabilidade de depositar um objeto numa região com muitos outros objetos é pequena, e vice-versa

Algoritmo de Agrupamento baseado em Formigas

- Algoritmo de Agrupamento baseado em Formigas (ACA)
  - Aplicação: análise de dados.
  - As formigas se movem em uma matriz 2D e consideram uma região de vizinhança de área s^2 , que corresponde a um quadrado Neigh($s \times s$) de $s \times s$ células ao redor da célula atual r

Algoritmo de Agrupamento baseado em Formigas

- Função f que calcula a densidade local de itens na vizinhança do objeto $\mathbf{x}_i$ situado no site r:

$$f(\mathbf{x}_i) = \begin{cases} \frac{1}{s^2} \sum_{\mathbf{x}_j \in \mathrm{Neigh}_{(s \times s)}(r)} \left[1 - \frac{d(\mathbf{x}_i, \mathbf{x}_j)}{\alpha} \right] & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases}$$

onde $f(\mathbf{x}_i)$ é uma medida da similaridade média de um objeto $\mathbf{x}_i$ com um outro objeto $\mathbf{x}_j$ na vizinhança de $\mathbf{x}_i$, α define a escala de dissimilaridade, e $d(\mathbf{x}_i, \mathbf{x}_j)$ é a distância euclidiana entre duas instâncias de dados no seu espaço de dados original (L-dimensional).

Algoritmo de Agrupamento baseado em Formigas

- As probabilidade de pegar e soltar um item são definidas pelas seguintes fórmulas:

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

$$p_d(\mathbf{x}_i) = \begin{cases} 2f(\mathbf{x}_i) & \text{if } f(\mathbf{x}_i) < k_2 \\ 1 & \text{otherwise} \end{cases}$$

- Aplicação:
  - Apropriados para análise de dados exploratória, onde existe um conjunto de dados cujos "grupos" (classes) são desconhecidos, e alguma informação deve ser extraída desses dados

Resultados

- Observou-se que o algoritmo apresentado apresenta mais grupos no espaço projetado que no espaço real.
- Seguintes mudanças foram propostas:
  - Uso de formigas se movendo a velocidades diferentes
  - Memória
  - "Distúrbios" de comportamento

Mudanças sugeridas

- Formigas com diferentes velocidades
  - Formigas que se movem mais rápido são menos seletivas, e isso se refletirá no cálculo da densidade local dos itens
  - Velocidade influencia as probabilidades de pegar ou deixar um item
- Memória
  - Formiga guarda os últimos m itens deixados e suas posições
  - Cada vez que a formiga pega um novo ítem, ele é comparado com os itens na memória
  - Levado para a célula mais similar a ele

Mudanças sugeridas

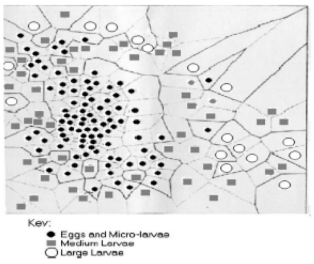
- "Distúrbio de comportamento"
  - Formiga tem a capacidade de destruir grupos que não tenham recebido nenhum item por um determinando período de tempo

Brood Sorting (2007)

Alguns tipos de formigas ordenam seus ovos de acordo com "sua idade" em uma estrutura anelar

A maneira como essa ordenação é feita pelas formigas não é totalmente conhecida, mas acredita-se que é uma mistura de um sistema aleatório e depósito de feromôneo

Brood sorting



Franks,N.R. and Sendova-Franks, A.B. (1992) Brood sorting by ants: distributing the workload over the work-surface. *Behavioural Ecology and Sociobiology* **30**, 109-123

Ant annular sorting Algorithm

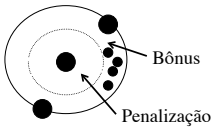
Baseado nas seguintes idéias:

- Formigas pegam e soltam items probabilisticamente, dependendo de um *score* (baseado na sua posição)
- Formigas se movem aleatoriamente no espaço
- Objetos tem um '*placement score*' (usa uma função dependente do perímetro para atrair ou repelir)
- Baseado nesse '*score*' as formigas decidem pegar ou não um objeto
- Quando uma formiga carrega um objeto, ela calcula um 'placement score' e decide, baseado nele, se vai soltar o objeto ou não

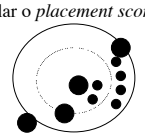
Objetos

• Cada objeto possui um tamanho, um perímetro mínimo (age como uma força repulsiva) e um perímetro máximo (age como uma força atrativa)

•Os perímetros são utilizados para calcular o *placement score*

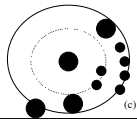


(a) Good placement



(b) Placement score of central object drops as a result of 3 new objects placed in the perimeter

Perimetro min e max



(c) higher placement score

Modelo

```
while ants exist
  for each ant

    if out of energy, remove ant
    else
    if ant unladen && hits object || ant laden && in free space

        set placement score = 0
        apply penalty for each object within minimum radius
        give bonus for each object within attraction radius

        if placement score > random[0..1]
            pick up/drop object
            remove energy

    move ant in random direction

  end for
end while
```

Resultados

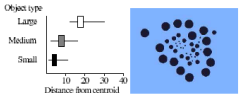


Fig. 4. Equal numbers of objects. (Left) Radial displacement. Minimum and maximum distances are represented by the line, and the interquartile range by the box. (Right) Typical pattern obtained.

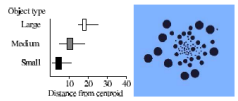


Fig. 5. Unequal numbers of objects. (Left) Radial displacement. (Right) Typical pattern obtained.

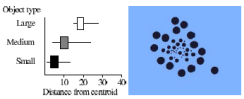


Fig. 6. Pre-sorted objects. (Left) Radial displacement. (Right) Typical pattern obtained.

Bibliografia

- **Leitura essencial**

E. Bonabeau and G. Theraulaz. Swarm Ants. Scientific American. 2000.

- **Leitura recomendada**

M. Dorigo, V. Maniezzo e A. Colorni, The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, 1996, pp.1-13

M. Dorigo, G. Di Caro and L. M. Gambardella. Ant Algorithms for Discrete Optimization Artificial Life, 5(2):137-172.