

# Otimização com Colônias de Formigas

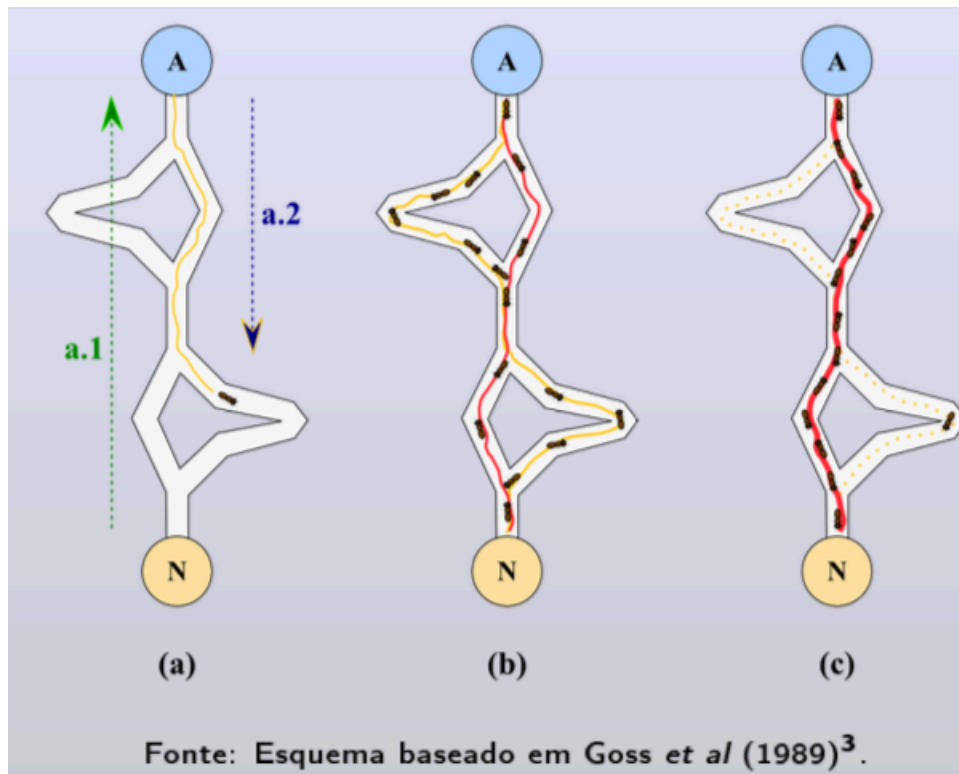
Gisele L. Pappa

# Formigas Reais



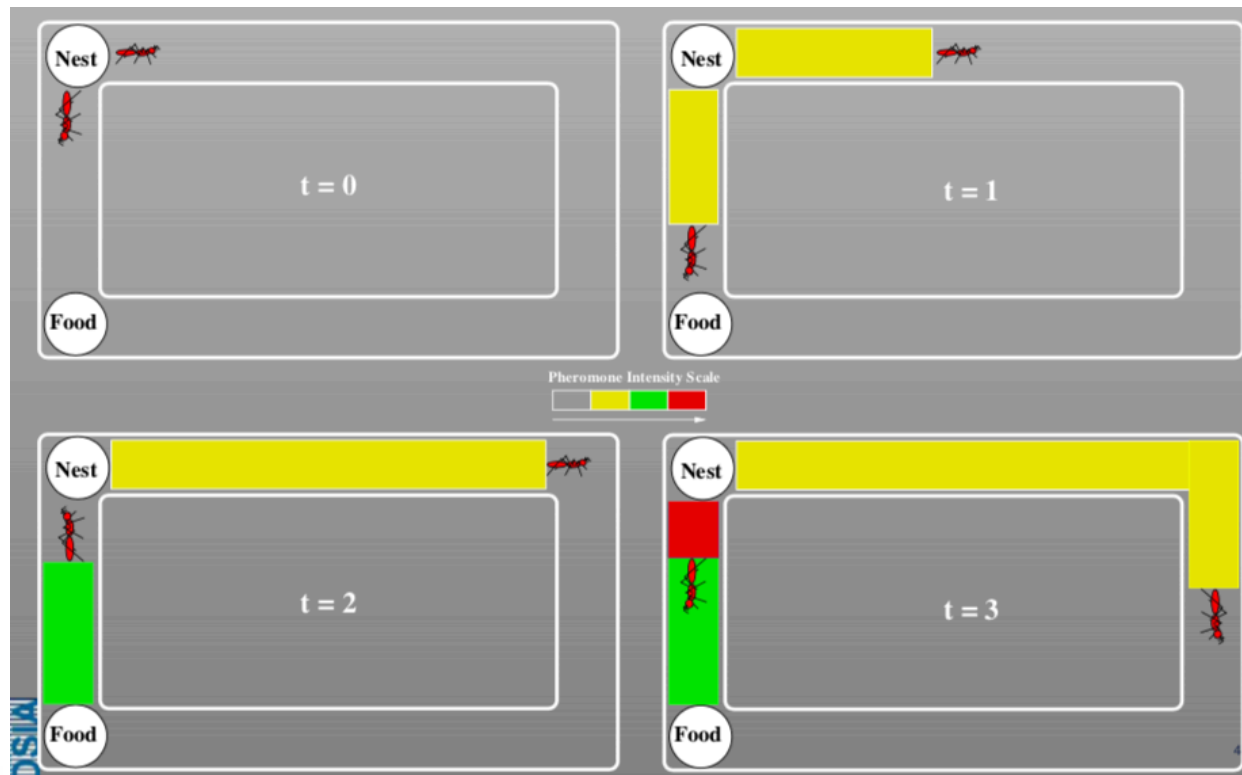
Formigas trabalhadoras seguindo uma trilha artificial de feromônio

# Formigas Reais: Experiência das “2 pontes”

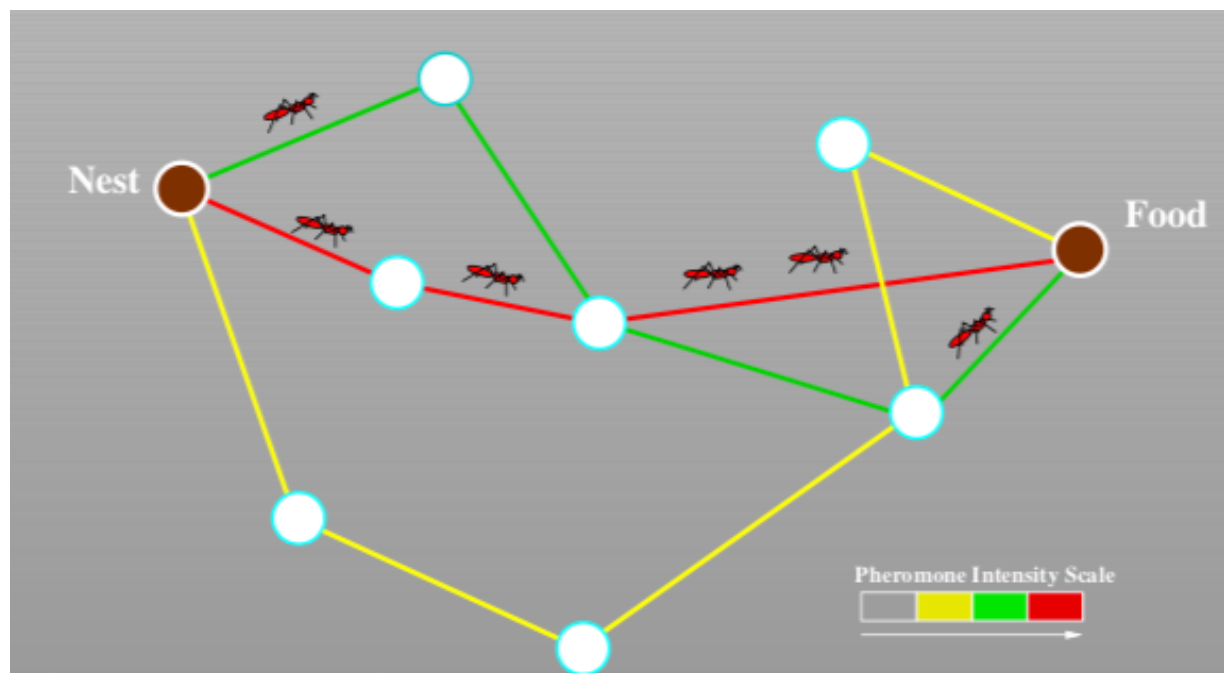


a.1 – Exploração inicial  
a.2 - Feromônio

# Exemplo de 2 caminhos



# Exemplo de vários caminhos

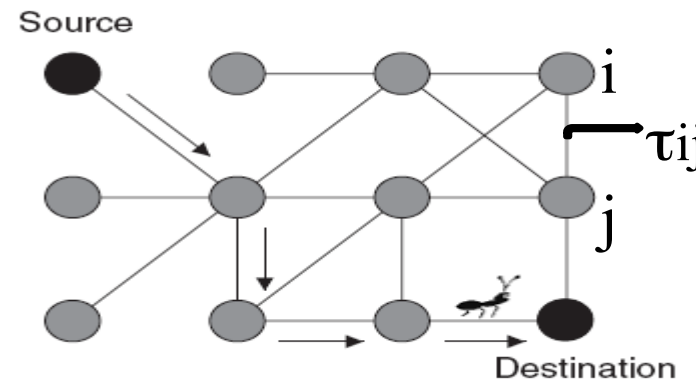


# Otimização por Colônia de Formigas

- Método de busca onde um conjunto de agentes (formigas) constroem incrementalmente soluções candidatas para um problema
- Contrução é probabilística e guiada por:
  1. Informações heurísticas sobre o problema
  2. Por um memória compartilhada por todos os agentes
    - Trilha artificial de feromônio

# Otimização por Colônia de Formigas

- Problema modelado como um grafo, onde buscamos um caminho de um ponto A a um ponto B
- Associado a cada aresta do grafo está uma quantidade de feromônio  $\tau$
- Cada formiga é capaz de sentir (ler) ou deixar (escrever) feromônio



# Otimização por Colônia de Formigas

- Cada **caminho** representa uma possível solução
- Quantidade de feromônio depositada é proporcional a **qualidade** da solução
- Se formigas têm que escolher entre dois caminhos, a probabilidade delas seguirem o caminho melhor é maior



# Otimização de Colônia de Formigas

$t = 0$

while  $t < \text{max\_it}$  do

    for each ant do

        build\_solution();

    endfor

    update\_pheromone();

endwhile

max\_it = Número máximo de iterações

# Otimização de Colônia de Formigas

$t = 0$

while  $t < \text{max\_it}$  do

    for each ant do

        build\_solution();

    endfor

    update\_pheromone();

endwhile

max\_it = Número máximo de iterações

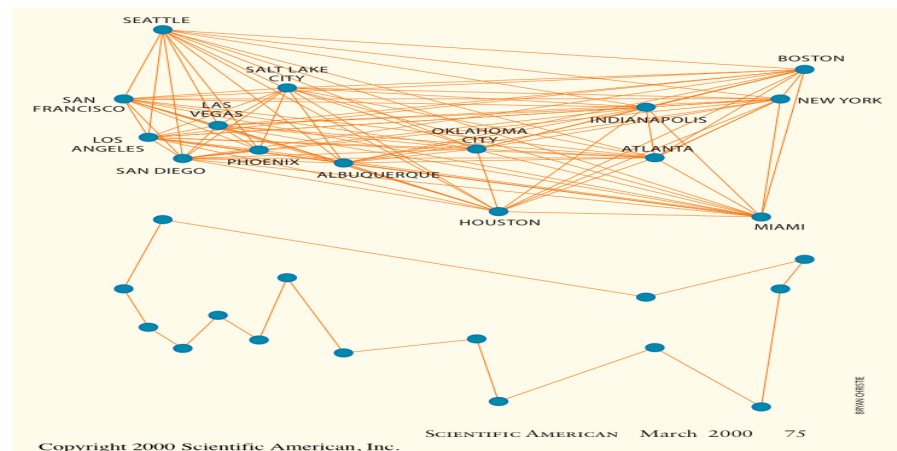
# Principais elementos de um ACO

- Representação do problema
  - Especificar os elementos que as formigas usarão para construir **incrementalmente** a solução para um problema (normalmente pensado em forma de um grafo)
  - Garantir a construção de soluções válidas
- Construção incremental da solução

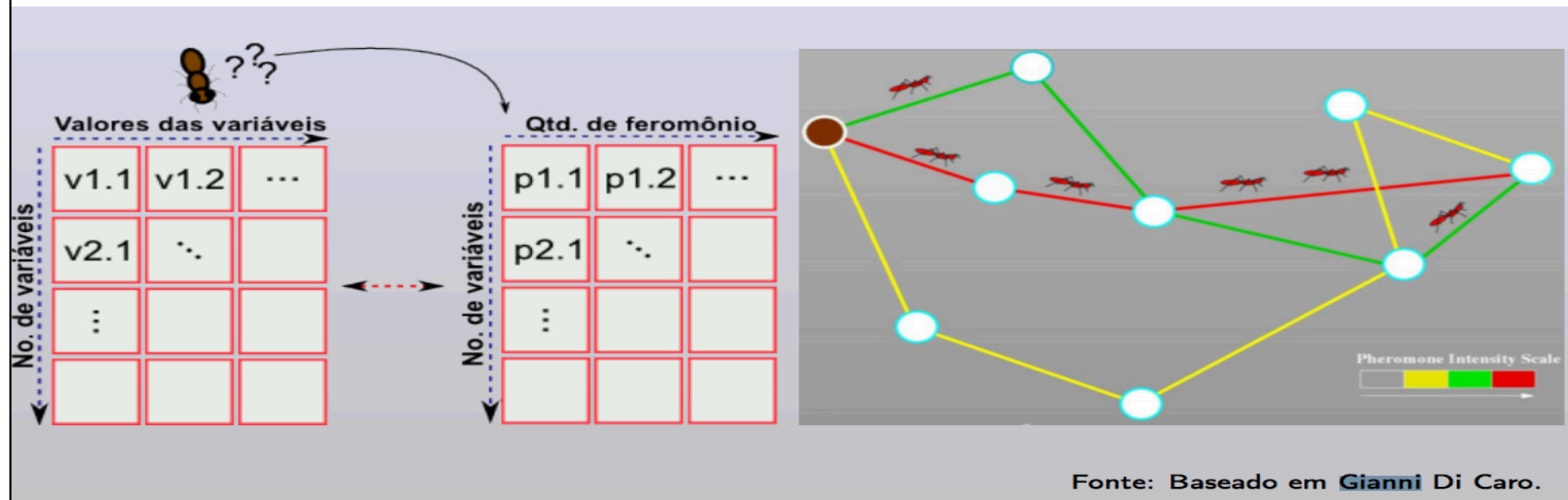
# ACO para o Caixeiro Viajante

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

$\tau_{ij}$  corresponde ao feromônio deixado quando a formiga caminha da cidade  $i$  para a cidade  $j$



# ACO para o Caixeiro Viajante



# Principais elementos de um ACO

- Representação do problema
- Construção incremental da solução
  - Depende de uma função probabilística baseada:
    - No feromônio ( $\tau$ )
    - Na “qualidade da aresta” a ser incluída - função ( $\eta$ )  
(*desirability*)

# Otimização de Colônia de Formigas

- A função  $\eta$  (*desirability*) mede 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.

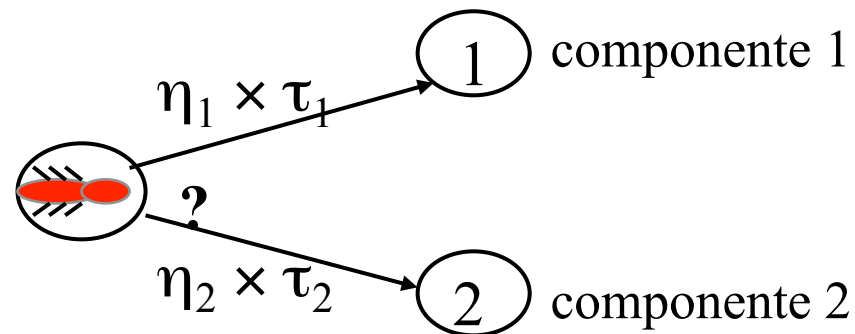
# ACO para o Caixeiro Viajante

- A função  $\eta_{ij}$  incorpora informações do problema
  - Calculada de maneira local, baseada no custo da aresta (i,j)
  - No problema do caixeiro, utilizamos:  $\eta_{ij} = 1 / d_{ij}$
- A quantidade de feromônio  $\tau_{ij}$  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



# Principais elementos de um ACO

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



# ACO para o Caixeiro Viajante

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

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

$\alpha, \beta$  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 visitados)

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

# ACO para o Caixeiro 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  $\tau_{ij}$  na aresta  $(i,j)$  e ao valor de  $\eta_{ij}$
    - 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

# Otimização de Colônia de Formigas

t = 0

while t < max\_it do

  for each ant do

    build\_solution();

  endfor

  update\_pheromone();

endwhile

max\_it = Número máximo de iterações

# Principais elementos de um ACO

- **Regra para atualização do feromônio:** especifica como atualizar a quantidade de feromônio ( $\tau$ ) associada a cada componente inserido no caminho seguido por uma formiga
  - Feromônio aumenta proporcionalmente a qualidade do caminho
  - Usa um mecanismo **global**, e uma estratégia de adaptação **independente** do problema  $\tau_{ij}(t+1) \leftarrow (1-\rho) \tau_{ij}(t) + \Delta\tau$ ,  
onde  $\rho \in (0,1]$  é a taxa de evaporação de feromônio.

# Atualização de Feromônio

- Feromônio é atualizado depois que todas as formigas constroem uma solução S
- 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  $\rho$  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

# Atualização de Feromônio

1. Aplica-se uma regra de evaporação de feromônio, diminuindo seu valor em cada nó por uma constante
2. Todas as formigas depositam uma quantidade  $\Delta\tau_{ij}$  de feromônio nas arestas  $(i,j)$  pelas quais elas passaram, ou seja, para cada formiga

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

onde  $f(S)$  é o custo total da solução  $S$

# Métodos de atualização do feromônio

- Elitista
  - Utiliza a melhor solução encontrada até o momento para atualizar o feromônio.
  - Favorece *exploitation* ao invés de *exploration*
- Baseado em Rank
  - Soluções contruídas são ranqueadas, e um peso de atualização é atribuído a cada formiga de acordo com sua posição no ranking



# Métodos de atualização do feromônio

- *Max-Min* (uma das mais populares) :
  - Os valores de feromônio em cada aresta são limitados entre  $[\tau_{\min}, \tau_{\max}]$ , evitando que uma aresta se torne muito forte/fraca
  - Apenas uma formiga atualiza o feromônio a cada iteração: ou a melhor formiga da iteração ou a melhor desde o princípio
    - Favorece *exploitation*
  - Inicialmente, o feromônio de todas as arestas recebe valor  $\tau_{\max}$ , o que leva a exploração das soluções nas primeiras iterações do algoritmo

ACO ( $\text{maxIt}$ ,  $N$ ,  $\tau_0$  )

Inicializa  $\tau_{ij}$  com  $\tau_0$  (igualmente para cada aresta)

Distribui cada uma das  $N$  formigas em um nó selecionado aleatoriamente

$t = 1$

while ( $t < \text{maxIt}$ ) do      //número de iterações

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

1. Constrói uma solução aplicando uma regra de transição probabilística
2. Avalia o custo de cada solução construída
3. Atualiza melhor solução

    Atualiza as trilhas de feromônio

$t = t + 1$

# Parâmetros

- Número máximo de iterações
- Número de formigas
  - Regra do dedão: igual ao número de arestas do grafo
- Taxa de feromônio inicial  $\tau_0$
- Pesos para concentração de feromônio ( $\alpha$ ) e qualidade da função ( $\beta$ ) quando calculando a probabilidade de transição
- Taxa de evaporação

# Leitura Recomendada

- E. Bonabeau and G. Theraulaz. Swarm Ants. Scientific American. 2000.
- **Ant Algorithms for Discrete Optimization** Marco Dorigo, Gianni Di Caro and Luca M. Gambardella, Artificial Life, 5, 1999, 137-172
- **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, Marco Dorigo, Vittorio Maniezzo and Alberto Colorni