

ALGORITMOS DE COLÔNIAS DE FORMIGAS

Francisco A. M. Gomes

1º sem/2009

MT852 – Tópicos em pesquisa operacional

Formigas reais

- Em geral, as formigas seguem o menor caminho entre o formigueiro e sua fonte de alimento.
- Enquanto andam, as formigas depositam no solo uma substância chamada **feromônio**.
- Na presença de feromônio, elas possuem certa tendência a seguir o caminho marcado com a substância.
- Essa tendência é baseada na quantidade de feromônio presente em cada caminho: quanto maior concentração, maior a chance da trilha ser seguida.

Como as formigas caminham

- ❑ Inicialmente, como não há feromônio nos caminhos, as formigas andam aleatoriamente em busca de alimento.
- ❑ As formigas que escolhem (ao acaso) o menor caminho voltam por ele.
- ❑ Como a ida e a volta são mais rápidas, há um pouco mais de feromônio nesse caminho.
- ❑ Como essa trilha possui mais feromônio, outras formigas tendem a segui-la.
- ❑ Com isso, o feromônio na trilha irá aumentar.
- ❑ Com o passar do tempo, os menores caminhos recebem uma carga maior de feromônio, sendo escolhidos pelas formigas.
- ❑ O feromônio também evapora com o tempo.
- ❑ Dessa forma, os caminhos menos visitados perdem feromônio, levando as formigas aos caminhos escolhidos pelas demais.

Formigas artificiais

Os algoritmos de colônias de formigas:

- São aplicados a problemas representados por um grafo $G(V, A)$.
- São algoritmos de construção: a cada iteração, cada formiga, individualmente, constrói uma solução para o problema.
- São algoritmos que, a cada iteração, trabalham com uma população de formigas (e, portanto, com uma população de soluções).

Na construção da solução, a formiga leva em conta:

- Informações heurísticas (fixas), que indicam a conveniência, para a obtenção da solução do problema, de se tomar determinado caminho.
- A quantidade de feromônio (variável), que indica quão desejável é um determinado caminho.

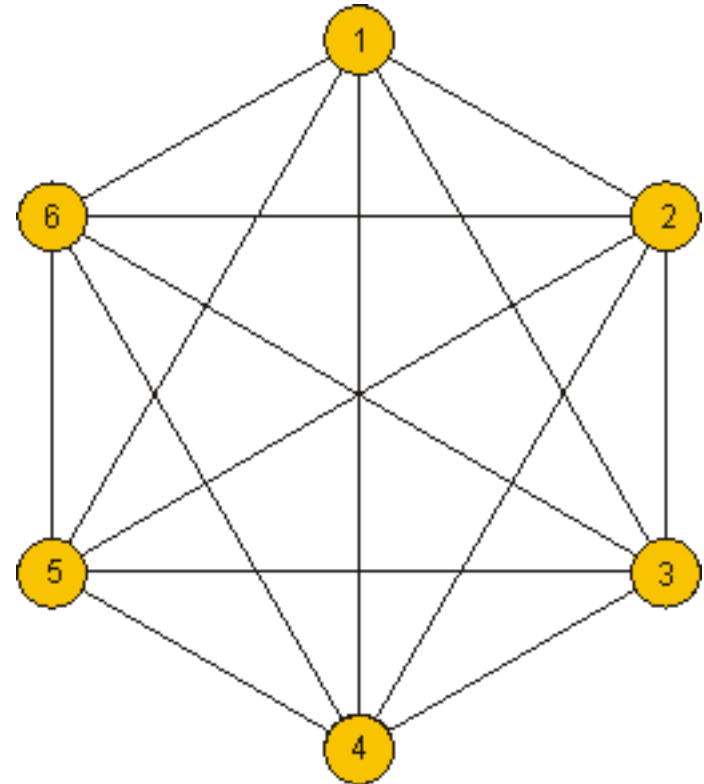
Problema do caixeiro viajante (TSP)

Dados:

- $V = \{v_1, \dots, v_m\}$ conjunto de cidades (nós)
- $A = \{(r, s) \mid r, s \in V\}$ conjunto de arestas (caminhos entre cidades)
- c_{rs} = custo associado à aresta (r, s)

Propósito:

- O caixeiro deve visitar todas as cidades
- Passando uma só vez em cada cidade
- Com o menor custo possível



TSP simétrico

Problema artificial de formigas

O problema a ser resolvido

- Contém um conjunto de restrições Ω .

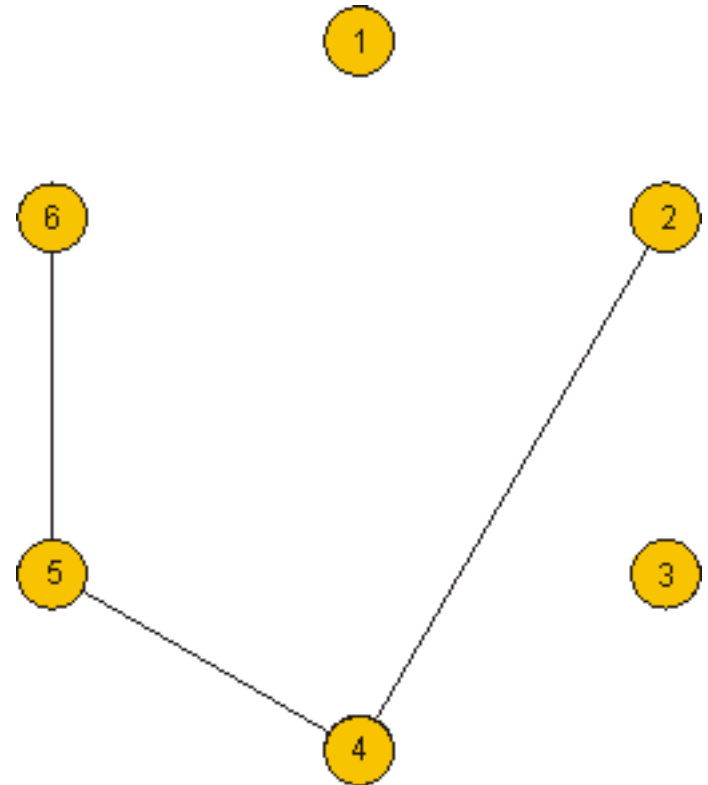
No TSP, as restrições são:

- (i) cada vértice só pode ser visitado uma vez; e
 - (ii) todos os vértices devem ser visitados.
- Contém um conjunto finito de componentes $V = \{v_1, v_2, \dots, v_m\}$

No TSP as componentes costumam ser os vértices do problema.

Problema artificial de formigas

- Cada seqüência de componentes, $\delta = \langle v_{i1}, v_{i2}, \dots, v_r \rangle$, define um estado.
No TSP, um estado é um caminho, ou seja, uma seqüência de nós (e arcos).
- O conjunto de todos os possíveis estados é representado por Δ .
- O conjunto de todos os estados factíveis é representado por Δ_f .

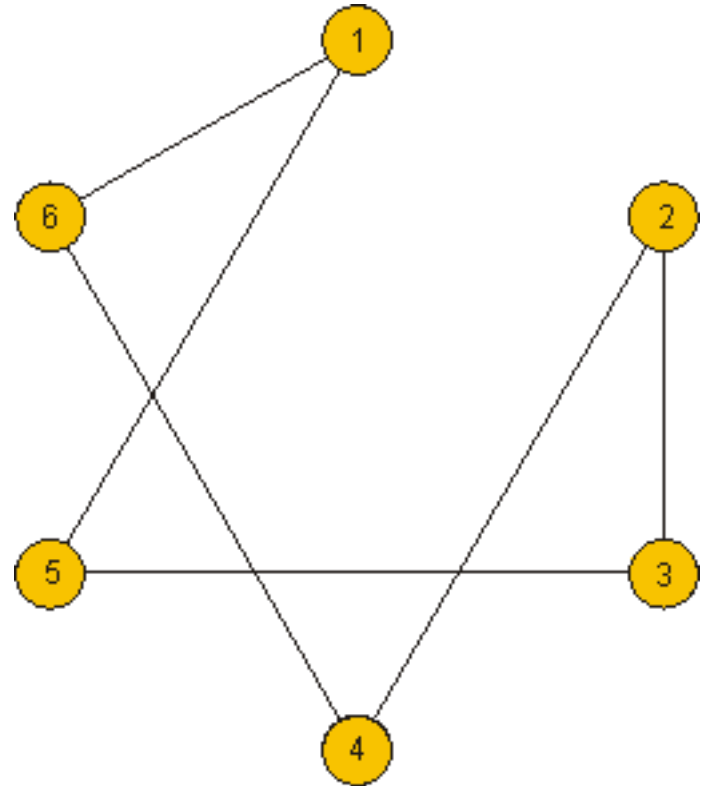


Um estado factível para o TSP

Problema artificial de formigas

- Uma solução S é um elemento de Δ_f que satisfaz todas as restrições.
- Há um custo $C(S)$ relacionado a cada solução S

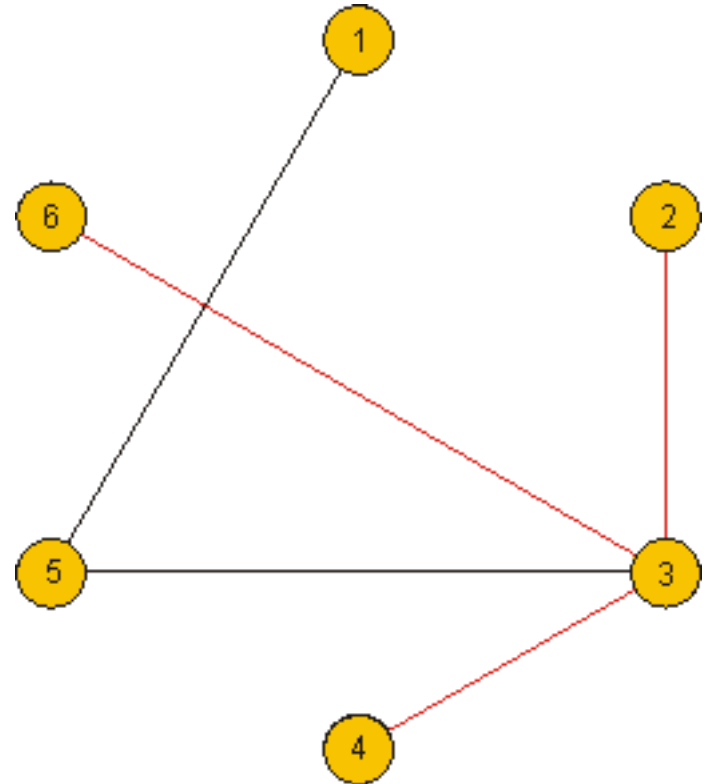
No TSP, o custo de uma solução é a soma dos custos das arestas percorridas.



Uma solução para o TSP

Problema artificial de formigas

- Para cada estado δ , há uma vizinhança.
- δ_2 pertence à vizinhança de δ_1 se podemos obter δ_2 a partir de δ_1 em um único movimento.
- δ_2 pertence à vizinhança factível N_{δ_1} de δ_1 se $\delta_2 \in \Delta_f$.
- No TSP, a vizinhança factível é definida pelas arestas que ligam o último nó de δ_1 a outro nó não visitado do grafo.
- No exemplo ao lado, a vizinhança factível de $\langle 1, 5, 3 \rangle$ é formada por $\langle 1, 5, 3, 2 \rangle$, $\langle 1, 5, 3, 4 \rangle$ e $\langle 1, 5, 3, 6 \rangle$



Vizinhança factível de $\langle 1, 5, 3 \rangle$

Voltando às formigas artificiais

No algoritmo de colônia de formigas:

- Cada formiga possui um estado inicial, do qual ela parte.
Geralmente, esse estado inicial é composto apenas pelo nó no qual a formiga está quando começa a andar.
- As formigas se comunicam apenas através do feromônio deixado pelo caminho.
- Cada formiga constrói uma sequência de estados, incluindo um nó por vez.
- A formiga tem uma memória L que a permite reconstruir o caminho percorrido até o momento.
- Quando está em um estado δ_r , a formiga pode mover-se para qualquer nó de sua vizinhança factível.
- A formiga possui condições de parada, que estão associadas à obtenção de uma solução factível para o problema.

Voltando às formigas artificiais

- Na hora de escolher qual caminho seguir, a formiga leva em consideração duas informações diferentes:
 - A informação heurística, η_{rs} , que mede o impacto na função objetivo de se atravessar a aresta (r, s) .
 - A informação vinda do feromônio artificial, τ_{rs} , que mede quão desejável é a aresta (r, s) .
- A formiga pode atualizar o feromônio depositado em uma aresta (r, s) :
 - Assim que passa por essa aresta;
 - Depois que termina seu caminho, se (r, s) pertence à solução obtida.
- Geralmente, a quantidade de feromônio depositada é proporcional à qualidade da solução obtida.
- O feromônio tem uma taxa de evaporação.

Algoritmo de colônia de formigas

1. Enquanto o critério de parada não é satisfeito,
 - 1.1. Para cada uma das formigas (em paralelo),
 - 1.1.1. Encontrar uma solução. Opcionalmente, atualizar, passo a passo, o feromônio das arestas.
 - 1.1.2. Para cada solução gerada, atualizar o feromônio do caminho percorrido.
 - 1.2. Evaporar certa quantidade de feromônio nas arestas.
 - 1.3. Eventualmente, executar ações *daemon*.

Ações *daemon*

- Ações (sem contrapartida natural) que adicionam uma perspectiva global à visão local das formigas, quando estas não conseguem obter boas soluções.
- Exemplos:
 - ▣ Ação de diversificação: evaporar feromônio em alguns nós específicos (que fazem parte de soluções boas).
 - ▣ Ações de intensificação:
 - Aplicar um algoritmo de busca local a cada solução obtida por uma formiga.
 - Aplicar feromônio em algumas arestas selecionadas.

Versões do algoritmo

- Principal diferença está na atualização do feromônio (muitas vezes, para permitir a diversificação das soluções).
- Principais algoritmos:
 - Ant System (AS)
 - Ant Colony System (ACS)
 - Max-Min Ant System
 - Rank-based Ant System

Ant system (AS)

- Criado em 1991 por Dorigo, Maniezzo e Colorni.
- A escolha do caminho é feita com base na probabilidade de se escolher uma aresta. Essa probabilidade pondera a informação heurística e aquela vinda do feromônio:

$$p_{rs}^k = \frac{(\tau_{rs})^\alpha \cdot (\eta_{rs})^\beta}{\sum_{u \in N_k^r} (\tau_{ru})^\alpha \cdot (\eta_{ru})^\beta}, \text{ se } s \in N_k(r)$$

A probabilidade da formiga percorrer uma aresta (r, s) corresponde à razão entre o peso do uso da aresta $(\tau_{rs})^\alpha \cdot (\eta_{rs})^\beta$ e a soma de todos os pesos dos caminhos que pertencem à vizinhança factível $N_k(r)$.

No TSP, o termo η_{rs} é definido por $\eta_{rs} = 1/c_{rs}$.

- Os parâmetros α e β determinam a proporção entre a informação heurística e a informação dada pelo feromônio considerada no peso da aresta (r, s) .

Ant system (AS)

- A evaporação ocorre logo antes do depósito de feromônio, depois de todas as formigas terem obtido sua solução.
- A evaporação é feita em todas as arestas (r, s) do grafo e obedece à seguinte fórmula:

$$\tau_{rs} \leftarrow (1 - p) \cdot \tau_{rs}$$

onde p é um número entre $(0, 1]$ que corresponde à taxa de evaporação.

- Em seguida, é feito o depósito de feromônio apenas nas arestas percorridas pelas formigas.
- A quantidade a ser depositada é função da qualidade da solução, sendo dada por

$$\tau_{rs} \leftarrow \tau_{rs} + \Delta^k \tau_{rs}$$

- onde $\Delta^k \tau_{rs} = f(C(S_k))$, ou seja, uma função da qualidade da solução S_k , como, por exemplo $f(C(S_k)) = Q/C(S_k)$, onde Q é uma constante.

Ant colony system (ACS)

Modificações principais com relação ao algoritmo AS.

- Dado um parâmetro q entre $(0, 1]$ e um número aleatório q_0 também entre $(0, 1]$, a escolha da aresta a ser seguida pela formiga é feita da seguinte maneira:

- Se $q < q_0$,

$$p_{rs}^k = \begin{cases} 1, & \text{se } s = \arg \max_{u \in N_k(r)} \{\tau_{ru} \cdot \eta_{ru}^\beta\} \\ 0, & \text{caso contrário} \end{cases}$$

- Se $q > q_0$,

$$p_{rs}^k = \begin{cases} \frac{(\tau_{rs})^\alpha \cdot (\eta_{rs})^\beta}{\sum_{u \in N_k(r)} (\tau_{ru})^\alpha \cdot (\eta_{ru})^\beta}, & \text{se } s \in N_k(r) \\ 0, & \text{caso contrário} \end{cases}$$

- Pode-se aplicar um algoritmo de busca local para melhorar as soluções das formigas antes de atualizar o feromônio no final da iteração.

Ant colony system (ACS)

- A atualização do feromônio ao fim da iteração é feita por uma única formiga (que representa a melhor solução já obtida pelo algoritmo (S^*)).

- Primeiramente, o feromônio é evaporado segundo a regra

$$\tau_{rs} \leftarrow (1-\rho) \cdot \tau_{rs}, \quad \forall (r,s) \in S^*$$

- Em seguida, o feromônio é depositado segundo

$$\tau_{rs} \leftarrow \tau_{rs} + \mu \cdot f(C(S^*)), \quad \forall (r,s) \in S^*$$

- Cada formiga também atualiza o feromônio cada vez que passa por uma aresta, segundo a regra

$$\tau_{rs} \leftarrow (1-\varphi) \cdot \tau_{rs} + \varphi \cdot \tau_0, \quad \varphi \in (0,1]$$

(como τ_0 é muito pequeno, isso costuma reduzir o feromônio nas arestas)

Max-min ant system

Características básicas:

- Como no ACS, ao final da iteração, atualiza-se o feromônio apenas de s^* (que pode ser a melhor solução já encontrada ou a melhor solução da iteração).
- Como no ACS, aplica-se um algoritmo de busca local para melhorar as soluções das formigas antes de atualizar o feromônio no final da iteração.
- As possíveis quantidades de feromônio depositadas possuem um mínimo τ_{\min} e um máximo τ_{\max} (o que evita que uma aresta seja ignorada).
- Teoricamente τ_{\max} corresponde à quantidade de feromônio a ser depositada na melhor solução S^* . No entanto, como S^* não é conhecida, uma aproximação é feita utilizando a melhor solução encontrada até o momento.
- O valor de τ_{\min} é definido como alguma constante menor que τ_{\max} .
- No início do algoritmo, a carga de feromônio em cada nó é igual a uma estimativa de τ_{\max} , calculada através de heurísticas gulosas.

Rank-based ant system

- Também atualiza o feromônio ao final da iteração.
- Essa atualização é feita da seguinte maneira:
 - As m formigas são classificadas na ordem decrescente da qualidade de sua solução.
 - As $\sigma - 1$ melhores formigas depositam quantidades de feromônio de acordo com a sua classificação e a qualidade da solução.

$$\tau_{rs} \leftarrow \tau_{rs} + \sum_{\mu=1}^{\sigma-1} (\sigma - \mu) \cdot f(C(s_{\mu})), \quad \forall (r,s) \in S_{\mu}$$

- A melhor solução encontrada pelo algoritmo também recebe uma carga de feromônio baseada na sua qualidade. Esse depósito é considerado o mais importante, uma vez que ele recebe um peso σ .

$$\tau_{rs} \leftarrow \tau_{rs} + \sigma \cdot f(C(s^*)), \quad \forall (r,s) \in S^*$$

Resultados teóricos

- Teorema 1 (para o ACS com algumas modificações e tomando $\tau_{\min} > 0$):
 - seja $p^*(t)$ a probabilidade de o algoritmo encontrar uma solução ótima ao menos uma vez nas primeiras t iterações. Então, para $\varepsilon > 0$ arbitrariamente pequeno, existe t suficientemente grande tal que $p^*(t) \geq 1 - \varepsilon$.
 - Assintoticamente $\lim_{t \rightarrow \infty} p(t) = 1$.
- Teorema 2 (para o ACS (com algumas modificações e permitindo que τ_{\min} decresça a zero, mas de forma suficientemente lenta) :
 - seja $p^*(t)$ a probabilidade de o algoritmo encontrar uma solução ótima ao menos uma vez nas primeiras t iterações. Então, $\lim_{t \rightarrow \infty} p(t) = 1$.
 - Seja t^* a iteração em que a primeira solução ótima s^* foi encontrada e $p(s^*, t, k)$ a probabilidade de que uma formiga k construa a solução ótima s^* na iteração $t > t^*$. Então, $\lim_{t \rightarrow \infty} p(s^*, t, k) = 1$.