

4. MÉTODO DO FORMIGUEIRO (*ANT COLONY OPTIMIZATION* – ACO)

4.1. [Analogia Comportamental: a busca de alimentos em um formigueiro](#)

4.2. [A Tradução Matemática: o algoritmo básico \(versão discreta\)](#)

4.3. [A Programação do Algoritmo](#)

4.4. [Exemplo Ilustrativo: O Problema do Caixeiro Viajante \(PCV\)](#)

4. MÉTODO DO FORMIGUEIRO (*ANT COLONY OPTIMIZATION*)

4.1. **Analogia Comportamental : a busca de alimento em um formigueiro.**



Figura 1 –Foto mostrando o aspecto cooperativo do trabalho de formigas ao transportar uma folha para o formigueiro.

O método do formigueiro (*ant colony optimization*¹ – ACO) é o método de otimização que tenta imitar o comportamento coletivo de formigas quando saem à busca de alimentos. As formigas, em um formigueiro, executam tarefas coletivas e tomam decisões que requerem um alto grau de coordenação tais como: a construção do formigueiro, a alimentação da ninhada, o armazenamento e busca de alimento e assim por diante.

No experimento esquematizado na Figura 2, as formigas *descobrem* de forma coletiva qual o menor caminho que separa o formigueiro de uma fonte de alimento. Nos experimentos conduzidos com a espécie *Linepithaema humile*, um alimento é separado do formigueiro por uma trilha com

¹ um bom e pioneiro trabalho sobre o assunto é o artigo de : Bonabeau, E. , Dorigo, M. & Theraulaz, G. – “*Inspiration for optimization from social insect behaviour*” – em NATURE , vol 406, páginas 39 a 42 de 6 de julho de 2000

dois ramos. O ramo maior é r vezes maior que o menor. No final do experimento todas as formigas escolhem o menor caminho se r for suficientemente elevado (no experimento se adotou $r=2$). Isto ocorre devido ao fato das formigas, quando em movimento, secretarem no solo uma certa quantidade de feromônio. Este feromônio, por ser uma substância biologicamente muito ativa, funciona como chamariz para outras formigas. No experimento da Figura 2, as primeiras formigas a retornarem ao formigueiro com o alimento são aquelas que percorreram o caminho mais curto duas vezes (na ida e na volta). Assim, quando novas formigas se encontrarem nas duas bifurcações do caminho darão preferência ao caminho mais curto por ter uma maior concentração de feromônio.

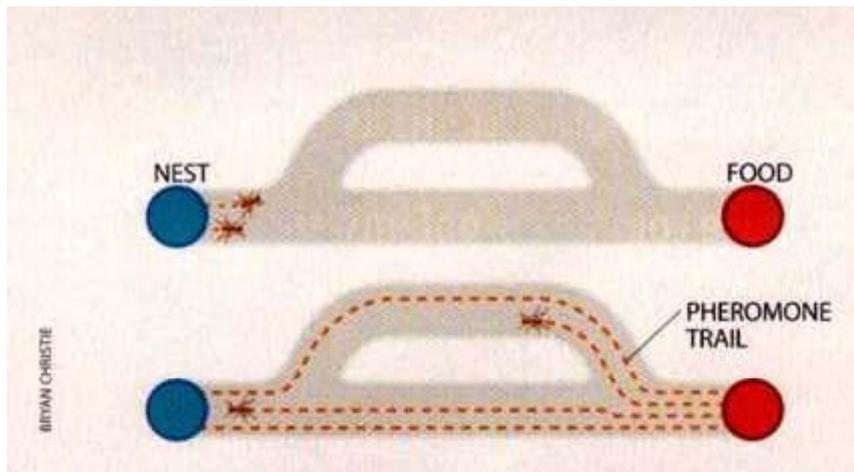


Figura 2 –Diagrama mostrando as trilhas de feromônio que permitem às formigas o armazenamento de alimentos com maior eficiência.

Duas formigas deixam o formigueiro em um mesmo instante (diagrama superior), cada uma delas tomando um caminho diferente e marcando-o com feromônio. A formiga que percorreu o caminho mais curto retorna ao formigueiro antes da outra (diagrama inferior). Assim, a trilha mais curta está, neste instante, marcada com o dobro de quantidade de feromônio do que o caminho mais longo. Este fato faz com que o caminho mais curto atraia uma maior quantidade de formiga do que o mais longo.

O mesmo experimento foi refeito, tornando disponível às formigas o caminho mais curto apenas após um determinado tempo. Neste caso, as formigas não escolheram o caminho mais curto devido ao fato do caminho mais longo já estar marcado com feromônio quando o mais curto é disponibilizado. Entretanto, tal comportamento não seria desejável em algoritmos de otimização, quando é importante que o caminho mais curto seja sempre o preferencialmente selecionado mesmo se disponibilizado após transcorrido um certo tempo. A simulação deste comportamento poderia ser efetuada através da consideração de uma taxa de evaporação do feromônio, traduzindo assim a maior dificuldade de manutenção do traço de feromônio em trilhas mais longas. Nos experimentos com a formiga *Linepithaema humile*, apesar do decaimento do traço de feromônio estar presente, ocorre a uma taxa tão baixa (ou vida média tão elevada) que não foi possível verificar tal comportamento.

A primeira versão do algoritmo ACO foi desenvolvida por Dorigo & Gambardella² para a resolução do Problema do Caixeiro Viajante, que é um problema de otimização combinatorial de busca em um espaço de permutação. Este algoritmo é fundamentado na distribuição de m formigas em n cidades e em permitir que cada uma destas formigas percorra um caminho fechado passando apenas uma vez por cada uma destas n cidades.

As duas características fundamentais deste algoritmo são:

- i-) Seleção da Próxima Cidade: é a forma com que cada uma das formigas estando em uma determinada cidade (cidade i) *escolhe* a próxima cidade (cidade j), excluindo em sua seleção todas as cidades já visitadas. Esta seleção é feita de uma forma aleatória em forma de roleta (ver item 3-7-a do curso!), associando-se a cada trilha de ligação com a cidade i a seguinte função densidade de probabilidade:

$$p_{ij} = \begin{cases} 0 & \text{se a cidade } j \text{ já foi visitada} \\ \frac{(\tau_{ij})^\alpha \cdot \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{k=1}^n p_{ik}} & \text{se a cidade } j \text{ não foi ainda visitada} \end{cases}$$

Onde:

τ_{ij} : intensidade da trilha de feromônio na trilha que liga a cidade i à cidade j ³(uma grandeza adimensional que expressa o valor relativo entre estas intensidades);

d_{ij} : distância entre a cidade i e a cidade j (também uma grandeza adimensional que expressa o valor relativo entre estas distâncias);

α e β parâmetros positivos que expressam a importância relativa da intensidade de feromônio e do tamanho da trilha ij , respectivamente;⁴

Para *saber* se a cidade j já foi visitada, cada formiga mantém um registro das cidades

² Dorigo, M. & Gambardella, L.M. – “*Ant Colonies for the Travelling Salesman Problem*”-Biosystems, Vol. 43, pp 73-81 (1997).

³ No início do procedimento adota-se o valor de $\tau_{ij} = \tau_0 (>0)$ para i

⁴ Note que se adotando $\alpha = 0$ a busca da próxima cidade independe totalmente da marcação de feromônio na trilha. Por outro lado, adotando-se $\beta = 0$ apenas a intensidade de marcação de feromônio orienta a seleção da próxima cidade. A recomendação na literatura é para adotar os valores de $\alpha = 1$ e $\beta = 5$.

anteriormente visitadas até o momento (este registro, que é atualizado cada vez que uma nova cidade é visitada, se chama de *lista tabu*).

Após o cálculo de p_{ij} para todos os valores de j , assim se procede:

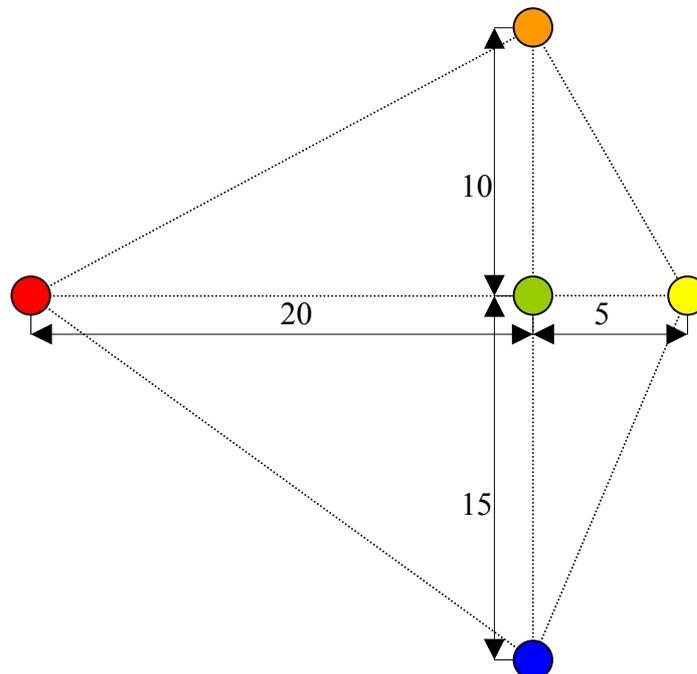
$Fatia_0 = 0$; $Fatia_J = \sum_{j=1}^J p_{ij}$ para $J=1, 2, \dots, (n-1)$ e $Fatia_n=1$; sorteia-se a seguir um número

aleatório λ (de distribuição uniforme), a próxima cidade a ser visitada é a cidade J , sendo J o valor do índice da Fatia tal que : $Fatia_{J-1} < \lambda \leq Fatia_J$.

Para ilustrar este procedimento considera-se o caso de 5 cidades com as seguintes coordenadas:

Cidade	1	2	3	4	5
(cor correspondente no diagrama)					
Localização (coordenadas)	(0;15)	(20;0)	(20;15)	(20;25)	(25;15)

O diagrama abaixo mostra a localização de cada uma das cidades:



Destas foram a matriz que traduz as distâncias entre as cidades é:

$$d = \begin{pmatrix} 0 & 25 & 20 & 22.361 & 25 \\ 25 & 0 & 15 & 25 & 15.811 \\ 20 & 15 & 0 & 10 & 5 \\ 22.361 & 25 & 10 & 0 & 11.18 \\ 25 & 15.811 & 5 & 11.18 & 0 \end{pmatrix}$$

e a matriz que traduz a intensidade da marcação de feromônio em cada trilha (valores hipotéticos) é:

$$\tau = \begin{pmatrix} 0 & 0.06 & 0.35 & 0.19 & 0.39 \\ 0.06 & 0 & 0.1 & 0.33 & 0.12 \\ 0.35 & 0.1 & 0 & 0.82 & 0.31 \\ 0.19 & 0.33 & 0.82 & 0 & 0.65 \\ 0.39 & 0.12 & 0.31 & 0.65 & 0 \end{pmatrix} \blacksquare$$

Uma formiga tendo como ponto de partida a primeira cidade (*cidade vermelha*), apresentaria as seguintes densidades de probabilidade de escolher a cidade j (adotando-se : $\alpha = 1$ e $\beta = 5$):

$$p = \begin{pmatrix} 0 \\ 0.031 \\ 0.577 \\ 0.182 \\ 0.211 \end{pmatrix} \blacksquare \quad \text{resultando nas fatias:} \quad \text{fatia} = \begin{pmatrix} 0 \\ 0 \\ 0.031 \\ 0.607 \\ 0.789 \\ 1 \end{pmatrix} \blacksquare$$

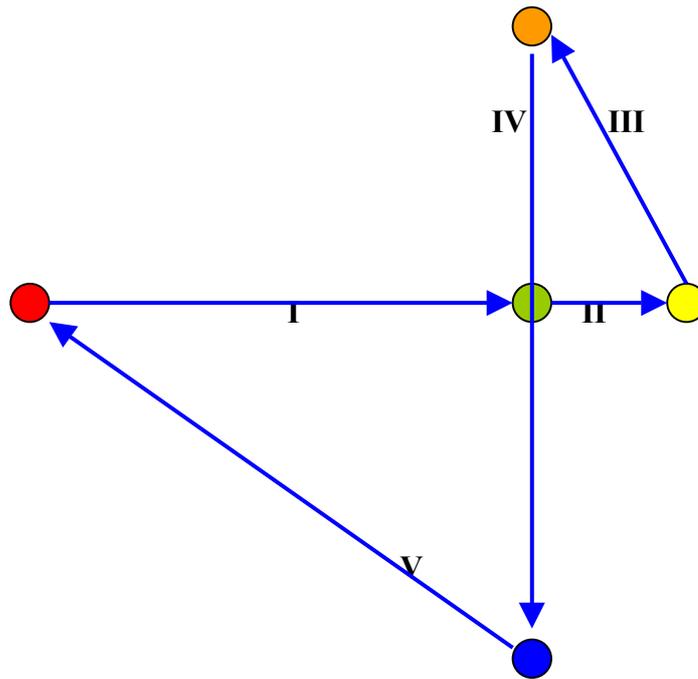
O número λ sorteado deu o valor: 0.28, fazendo assim com que a cidade 3 (*cidade verde*) seja a escolhida. A mesma formiga partindo da cidade 3 apresentaria as seguintes densidades de probabilidade de escolher a cidade j :

$$p = \begin{pmatrix} 0 \\ 0.001 \\ 0 \\ 0.076 \\ 0.923 \end{pmatrix} \blacksquare \quad \text{resultando nas fatias:} \quad \text{fatia} = \begin{pmatrix} 0 \\ 0 \\ 0.001 \\ 0.001 \\ 0.077 \\ 1 \end{pmatrix} \blacksquare$$

O número λ sorteado deu o valor: 0.682, fazendo assim com que a cidade 5 (*cidade amarela*) seja a escolhida. A mesma formiga partindo da cidade 5 apresentaria as seguintes densidades de probabilidade de escolher a cidade j :

$$p = \begin{pmatrix} 0 \\ 0.032 \\ 0 \\ 0.968 \\ 0 \end{pmatrix} \blacksquare \quad \text{resultando nas fatias:} \quad \text{fatia} = \begin{pmatrix} 0 \\ 0 \\ 0.032 \\ 0.032 \\ 1 \\ 1 \end{pmatrix} \blacksquare$$

O número λ sorteado deu o valor: 0.722, fazendo assim com que a cidade 4 (*cidade laranja*) seja a escolhida. Como, neste ponto, restou apenas uma cidade não visitada pela formiga e sendo a mesma a cidade 2 (*cidade azul*), esta é a última cidade visitada antes da formiga retornar a seu ponto de partida. Desta forma o roteiro percorrido pela formiga é : $1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1$, roteiro este esquematizado no diagrama a seguir:



- ii-) Vaporização e Reforço da Marcação de Feromônio. A intensidade de marcação de feromônio em todas as trilhas ligando a cidade i às cidades j passa por dois processos transformadores: (a) enfraquecimento da intensidade da marcação por vaporização, este processo é traduzido matematicamente pela simples expressão: $\tau_{i,j}^{NEW} \leftarrow \rho \cdot \tau_{i,j}^{OLD}$, onde ρ é um parâmetro entre 0 e 1 (um) escolhido pelo usuário, e os sobrescritos OLD e NEW representam a intensidade da marcação antes e depois da vaporização, respectivamente; (b) reforço da intensidade da marcação por formigas que percorreram a trilha ij , isto é feito da seguinte forma:

$$\Delta\tau_{i,j}^{\text{reforço}} = \begin{cases} 0 & \text{se a trilha } ij \text{ não faz parte do roteiro total percorrido pela formiga} \\ \frac{Q}{L_{\text{total}}} & \text{se a trilha } ij \text{ faz parte do roteiro total percorrido pela formiga} \end{cases}$$

sendo L_{total} o tamanho total do roteiro percorrido pela formiga (até retornar a seu ponto de partida!) e Q : um parâmetro positivo escolhido pelo usuário. Os novos valores das intensidades de marcação em cada trilha ij será a composição destes dois fatores, resultando assim em:

$$\tau_{i,j}^{NEW} \leftarrow \rho \cdot \tau_{i,j}^{OLD} + \Delta\tau_{i,j}^{\text{reforço}}$$

No exemplo ilustrativo acima adotando $\rho = 0.9$ e $Q=1$, tem-se: $L_{\text{total}} =$

$20+5+11.18+25+25=86.18$, assim: $\frac{Q}{L_{\text{total}}} = 0.012$, resultando no reforço:

$$\Delta\tau = \begin{pmatrix} 0 & 0.012 & 0.012 & 0 & 0 \\ 0.012 & 0 & 0 & 0.012 & 0 \\ 0.012 & 0 & 0 & 0 & 0.012 \\ 0 & 0.012 & 0 & 0 & 0.012 \\ 0 & 0 & 0.012 & 0.012 & 0 \end{pmatrix}$$

resultando, finalmente, no novo valor de τ :

$$\tau = \begin{pmatrix} 0 & 0.063 & 0.323 & 0.172 & 0.348 \\ 0.063 & 0 & 0.092 & 0.307 & 0.109 \\ 0.323 & 0.092 & 0 & 0.735 & 0.292 \\ 0.172 & 0.307 & 0.735 & 0 & 0.601 \\ 0.348 & 0.109 & 0.292 & 0.601 & 0 \end{pmatrix} \blacksquare$$

Há diferentes formas distintas de atualizar os valores dos componentes de τ ao longo do caminho percorrido por cada formiga, algumas destas formas são apresentadas no próximo item em que se detalham aspectos implementacionais do algoritmo.

[voltar para MÉTODO DO FORMIGUEIRO](#)

4.2. A Programação do Algoritmo (versão discreta)

Formalmente o algoritmo de busca do melhor caminho no problema do caixeiro viajante, usando o método do formigueiro (*ant-colony*) pode ser resumido pelas etapas:

Etapa Inicial (Iteração zero): Entre com n (número de cidades), m (número de formigas) e N (número de iterações) ; especifique os valores dos parâmetros: α , β , ρ e Q (explicados no item anterior).

Especifique as coordenadas de localização de cada uma das n cidades [armazenadas em uma matriz C de n linhas e 2 colunas; sendo as coordenadas da cidade i são $(C_{i,0}, C_{i,1})$]; calcule a matriz D , matriz $n \times n$ em que:

$D_{i,j} = \sqrt{(C_{i,0} - C_{j,0})^2 + (C_{i,1} - C_{j,1})^2}$ para $i, j = 1, \dots, n$. Especifique os valores iniciais da matriz τ cujo termo geral $\tau_{i,j} = \tau_{j,i}$ é a intensidade de marcação de feromônio na trilha que liga a cidade i à cidade j , no início do processo iterativo se adota: $\tau_{i,j} = \begin{cases} 0 & \text{para } i = j \\ \tau_0 & \text{para } i \neq j \end{cases}$, sendo τ_0 um valor não nulo (pequeno!) escolhido pelo usuário.

Calcule o valor: $L_{\min} = \sum_{i=1}^n \sum_{j=1}^n D_{i,j}$.

Etapa Principal Para $iter=1, 2, \dots, N$ (número de iterações especificado) execute o procedimento:

(a)-Para $k=1, \dots, m$ (m : número total de formigas)-*Etapa de Busca do Melhor Percurso*:

busque o *melhor* caminho fechado para a formiga k (segundo o procedimento descrito separadamente abaixo!) e armazene os resultados na matriz V , sendo: $V_{i,k} \Rightarrow$ o índice

designativo da cidade visitada pela formiga k em seu i 'ésimo trecho de seu percurso fechado (note que i varia de 1 a $n!$) e $V_{n+1,k} = L_{total}^{(k)}$ que é o tamanho total do percurso fechado escolhido pela formiga k .

(b)-Faça : $\tau \leftarrow \rho \cdot \tau$ (vaporização do feromônio depositado em todas as trilhas possíveis). *Etapa da Vaporização da Marcação de Feromônio.*

(c)-Para $k = 1, \dots, m$ (m : número total de formigas). *Etapa de Reforço da Marcação de Feromônio:*

Faça o reforço da intensidade da marcação de feromônio em todas as trilhas percorridas pela formiga k segundo o procedimento: para $i = 1, 2, \dots, n-1$ faça : $\tau_{I,J} \leftarrow \tau_{I,J} + \frac{Q}{L_{total}^{(k)}}$ onde $I =$

$V_{i,k}$ e $J = V_{i+1,k}$; além de $I = V_{1,k}$ e $J = V_{n,k}$ (este último par de valores de I e J para assegurar o fechamento do percurso!).

(d)-Para $k = 1, \dots, m$ (m : número total de formigas) . *Etapa de Seleção do Melhor Percurso:*

Se $L_{total}^{(k)} < L_{min}$ faça então : $L_{min} \leftarrow L_{total}^{(k)}$ e para i de 1 a n faça $city_i \leftarrow V_{i,k}$; vá para

Etapa (e)

Se $L_{total}^{(k)} \geq L_{min}$ mantenha o valor de L_{min} vá para **Etapa (e)**.

(e)-Armazenamento dos Melhores Percursos em Cada Iteração.

Armazene na matriz $R_{i,iter} = city_i$ para i de 1 a n e $Melhor_{iter} = L_{min}$.

Etapa de Saída: Escreva os valores da matriz R e do vetor $Melhor$.

FIM

Este procedimento foi utilizado no desenvolvimento do programa em MATHCAD®, mostrado abaixo:

```

ACO_Globa(N,m) :=
  for i ∈ 0..n - 1
    for j ∈ 0..n - 1
      τi,j ← τ0
  d ← distancias
  trip ← ∑i=0n-1 ∑j=0n-1 di,j
  for iter ∈ 0..N
    for k ∈ 0..m - 1
      v(k) ← Rota(τ,d)
      τ ← ρ · τ
      for k ∈ 0..m - 1
        Δ ←  $\frac{Q}{v_{n,k}}$ 
        for i ∈ 0..n - 2
          I ← vi,k
          J ← vi+1,k
          τI,J ← τI,J + Δ
          τJ,I ← τI,J
        I ← v0,k
        J ← vn-1,k
        τI,J ← τI,J + Δ
        τJ,I ← τI,J
        if vn,k < trip
          trip ← vn,k
          for j ∈ 0..n - 1
            cityj ← vj,k
        for i ∈ 0..n - 1
          Ri,iter ← cityi
        Rn,iter ← trip
  R

```

A construção do melhor percurso para cada formiga **k** é feita de acordo com o procedimento:

(a) Inicialização do Percurso:

Apague todos os registros existentes no Carnê em que se anota as cidades percorridas no percurso.

Sorteie aleatoriamente a cidade inicial do percurso e anote o índice designativo a esta cidade no topo do Carnê, isto é $Tabu_1 \leftarrow \text{rnd}(n)$;

(b) Etapa Principal: para s de 1(um) a $n-1$, faça:

(b-1) – Para $i=1$ a n , **igual** $\leftarrow 0$ e $p_i \leftarrow 0$;

para cada valor de i faça

(b-1-1) para j de 1 a $s-1$, se i for igual a $Tabu_j$ faça **igual** $\leftarrow 1$ e vá para **(b-1-2)**;

se i não for igual a $Tabu_j$ e $j = s-1$ e vá para **(b-1-2)**; se i não for igual a $Tabu_j$ e $j < s-1$ vá para **(b-1-1)** com o valor seguinte de j (se $j < s-1$).

(b-1-2) se **igual** for igual a 0 faça: $p_i \leftarrow \frac{(\tau_{i, Tabu_{s-1}})^\alpha}{(d_{i, Tabu_{s-1}})^\beta}$ e volte para **(b-1)** com o valor

seguinte de i (se $i < n$); se **igual** não for igual a 0 volte para **(b-1)** com o valor seguinte de i (se $i < n$);

(b-2) – *Escolha da Próxima Cidade* .

Faça : $S \leftarrow \sum_{j=1}^n p_j$ e a seguir $p_i \leftarrow \frac{p_i}{S}$ para $i = 1, 2, \dots, n$, a seguir faça : $fatia_0 \leftarrow 0$;

$fatia_n \leftarrow 1$ e $fatia_i \leftarrow \sum_{j=1}^i p_j$ para $i = 1, 2, \dots, n-1$; sorteie a seguir um número real

positivo λ entre 0 e 1 [$\lambda \leftarrow \text{rnd}(1)$], a próxima cidade é a cidade i^* em que ocorre : $fatia_{i-1} < \lambda < fatia_i$; escreva o valor desta cidade no carnê, isto é: $Tabu_s \leftarrow i^*$.

(b-3) – *Escolha da Última Cidade a ser Visitada* .

Verifique em seu carnê qual a cidade que está faltando entre todas que devem ser visitadas, sendo esta $i_{última}$ deve se registrar: $Tabu_n \leftarrow i_{última}$

(c) Etapa Final: cálculo do tamanho total do percurso.

Calcule: $L_{\text{total}}^{(k)} \leftarrow D_{Tabu_1, Tabu_n} + \sum_{i=1}^{n-1} D_{Tabu_i, Tabu_{i+1}}$

Este procedimento foi utilizado no desenvolvimento do programa em MATHCAD®, mostrado abaixo:

```

Rota( $\tau, d$ ) :=
  lista0  $\leftarrow$  floor( $\frac{\text{rnd}(100 \cdot n)}{100}$ )
  v0  $\leftarrow$  lista0
  N  $\leftarrow$  n - 1
  listan  $\leftarrow$  0
  for s  $\in$  1..n - 2
    for i  $\in$  0..N
      igual  $\leftarrow$  0
      pi  $\leftarrow$  0
      for k  $\in$  0..s - 1
        if  $|i - \text{lista}_k| < 10^{-6}$ 
          igual  $\leftarrow$  1
          break
      pi  $\leftarrow$  ( $\tau_{i, \text{lista}_{s-1}}$ ) $\alpha$  · ( $\frac{1}{d_{i, \text{lista}_{s-1}}}$ ) $\beta$  if igual = 0
      p  $\leftarrow$   $\frac{1}{\sum_{i=0}^N p_i}$  · p
      fatian  $\leftarrow$  1
      for i  $\in$  0..N - 1
        fatiai+1  $\leftarrow$  fatiai + pi
       $\lambda \leftarrow \frac{\text{rnd}(100)}{100}$ 
       $\delta \leftarrow \lambda$ 
      for i  $\in$  0..N
        a  $\leftarrow$   $\lambda - \text{fatia}_{i+1}$ 
        crit  $\leftarrow$  a ·  $\delta$ 
         $\delta \leftarrow$  a
        if crit < 0
          listas  $\leftarrow$  i
          break
      vs  $\leftarrow$  listas
  v  $\leftarrow$  sort(v)
  listaN  $\leftarrow$  N
  for i  $\in$  0..N - 1
    if  $|i - v_i| > 10^{-6}$ 
      listaN  $\leftarrow$  i
      break
  listan  $\leftarrow$  d(lista0, listan-1) +  $\sum_{k=0}^{n-2} d(\text{lista}_k, \text{lista}_{k+1})$ 
  lista

```

Há outras formas de atualizar (reforçar e evaporar!) a intensidade da marcação de feromônio nas trilhas, uma delas é apresentada a seguir:

Atualização Local e Global: neste procedimento ocorre uma *Atualização Local* ao término de

cada caminho fechado que uma formiga k percorra. A intensidade de feromônio em toda a trilha ij é então atualizada segundo:

$$\tau_{i,j}^{(k)} \leftarrow \begin{cases} (1 - \rho_{\text{local}}) \cdot \tau_{i,j}^{(k)} & \text{se o trecho } ij \text{ não faz parte do caminho fechado} \\ (1 - \rho_{\text{local}}) \cdot \tau_{i,j}^{(k)} + \rho_{\text{local}} \cdot \Delta\tau_{i,j}^{(k)} & \text{se o trecho } ij \text{ faz parte do caminho fechado} \end{cases}$$

onde há duas formas de calcular $\Delta\tau_{i,j}^{(k)}$ ⁵:

(a) Modelo de *densidade de formiga (ant-density)*: $\Delta\tau_{i,j}^{(k)} = Q_{\text{local}}$ (constante);

(b) Modelo de *quantidade de formiga (ant-quantity)*: $\Delta\tau_{i,j}^{(k)} = \frac{Q_{\text{local}}}{d_{i,j}^{(k)}}$

Após todas as formigas terem completado suas rotas, há também uma atualização da intensidade da marcação de feromônio com vaporização geral e o reforço apenas na *melhor rota (Atualização Global)* segundo:

$$\tau_{i,j} \leftarrow \begin{cases} (1 - \rho_{\text{global}}) \cdot \tau_{i,j} & \text{se o trecho } ij \text{ não faz parte da melhor rota} \\ (1 - \rho_{\text{global}}) \cdot \tau_{i,j} + \rho_{\text{global}} \cdot \frac{Q_{\text{global}}}{L_{\text{min}}} & \text{se o trecho } ij \text{ faz parte da melhor rota} \end{cases}$$

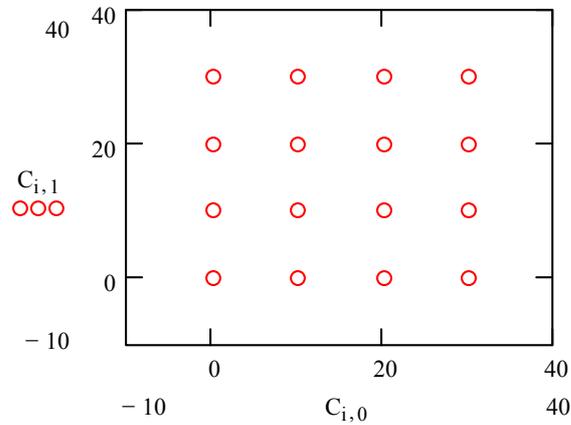
Onde os parâmetros: Q_{local} , Q_{global} , ρ_{local} e ρ_{global} são especificados pelo usuário.

[voltar para MÉTODO DO FORMIGUEIRO](#)

⁵ Dorigo, M., Maniezzo, V. & Colomo, a _ “ *Ant System: Optimization by a Colony of Cooperating Agents* “IEEE Transactions on Systems, Man, and Cybernetics, Vol. 26, N° 1, pp. 29-41 (1996)

4.3. Exemplo Ilustrativo – O Problema do Caixeiro Viajante (PCV)

Na forma clássica apresentada acima o ACO foi testado para o seguinte PCV, 16 cidades uniformemente distribuídas segundo o diagrama:



Sendo as coordenadas de cada uma destas cidades armazenadas na matriz:

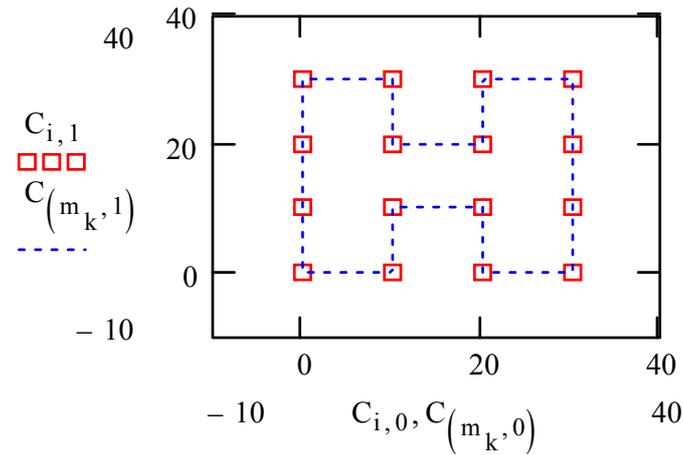
$$C := \begin{pmatrix} 0 & 0 \\ 0 & 10 \\ 0 & 20 \\ 0 & 30 \\ 10 & 0 \\ 10 & 10 \\ 10 & 20 \\ 10 & 30 \\ 20 & 0 \\ 20 & 10 \\ 20 & 20 \\ 20 & 30 \\ 30 & 0 \\ 30 & 10 \\ 30 & 20 \\ 30 & 30 \end{pmatrix}$$

Os seguintes valores dos parâmetros do algoritmo foram adotados:

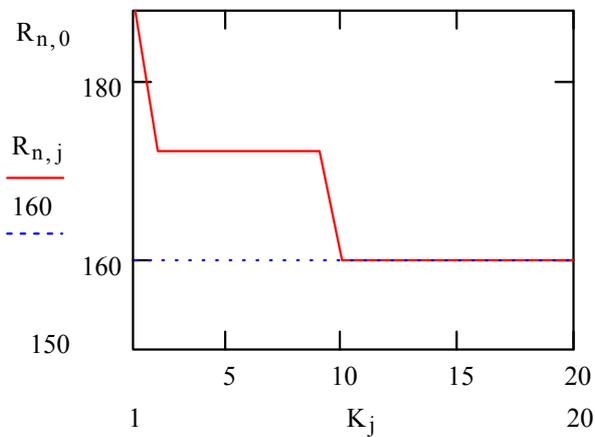
$$n := 16 \quad m := 8 \quad N := 20 \quad \alpha := 1 \quad \beta := 5 \quad \rho := .5 \quad \tau_0 := 10^{-6} \quad Q := 100$$

Verificando-se visualmente a convergência ao percurso ótimo após 10 iterações sendo este percurso o seguinte: $10 \rightarrow 9 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 12 \rightarrow 11 \rightarrow 7 \rightarrow 8 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow 10$, o que resulta em um comprimento total de $L_{\text{ótimo}} = 160$

Esquematizado na figura abaixo:

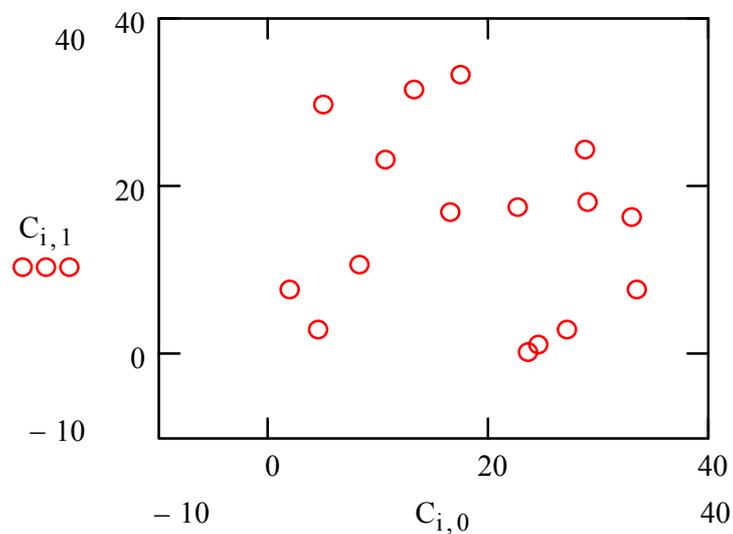


Na figura abaixo, é mostrada a variação do melhor comprimento total *versus* o índice designativo da iteração:



Uma animação deste procedimento é mostrada no programa AVI : *PCV_ACO_Global*.

O mesmo problema é resolvido para uma distribuição aleatória das 16 cidades (os parâmetros do algoritmo considerados são os mesmos do exemplo anterior), Esta distribuição é mostrada abaixo:

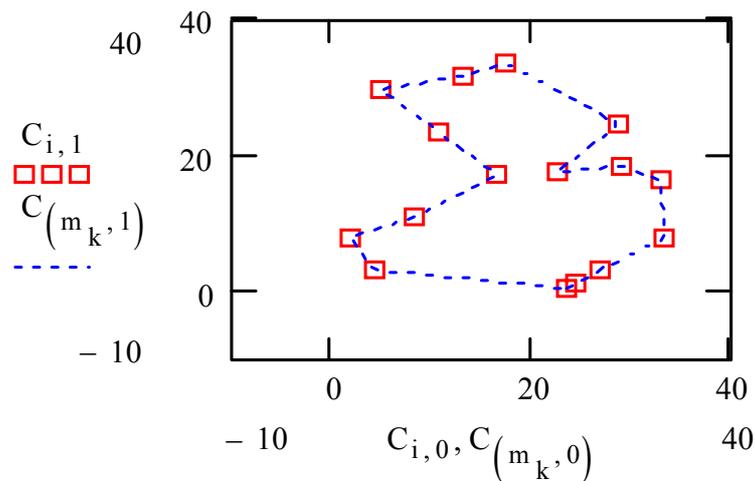


Cujas coordenadas são:

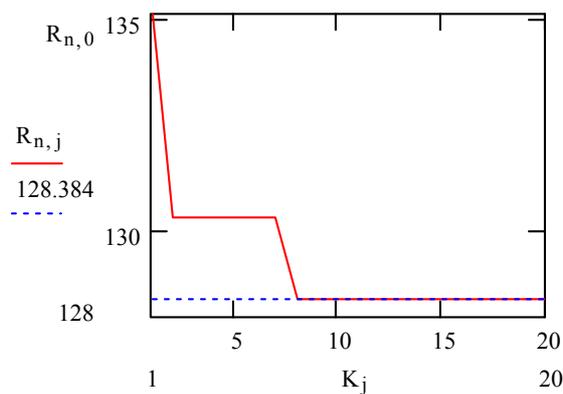
cidade	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x	22.48	33.17	32.81	26.84	16.39	8.05	4.26	12.97	22.32	10.43	28.66	1.63	4.66	28.85	17.23	24.25
y	17.59	7.73	16.39	2.99	17.08	10.71	2.92	31.43	0.16	23.21	24.43	7.63	29.78	18.21	33.42	1.23

Verificando-se visualmente a convergência ao percurso ótimo após 8 iterações sendo este percurso o seguinte: $9 \rightarrow 16 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 14 \rightarrow 1 \rightarrow 11 \rightarrow 15 \rightarrow 8 \rightarrow 13 \rightarrow 10 \rightarrow 5 \rightarrow 6 \rightarrow 12 \rightarrow 7 \rightarrow 9$, o que resulta em um comprimento total de $L_{\text{ótimo}} = 128.384$

Esquematizado na figura abaixo:



Na figura abaixo, é mostrada a variação do melhor comprimento total *versus* o índice designativo da iteração:



Uma animação deste procedimento é mostrada (com distribuição das cidades diferente da apresentada acima!) no programa AVI : *PCV_Random_ACO_Global*.

Em relação à esta aula os seguintes arquivos estão também disponíveis para *Download*:

Nome do Arquivo	Especificação
Aula6.pdf	Texto em pdf da aula virtual sobre o algoritmo do formigueiro.
Lista6.pdf	Enunciados da Lista de Exercícios em pdf relativos à aula virtual sobre o algoritmo do formigueiro
ACO_Lenzi.exe	Programa executável codificado em DELPHI® do algoritmo do formigueiro aplicado ao problema do caixeiro viajante (PCV) -
Manual_ACO_Lenzi.pdf	Manual do Usuário em pdf do programa: ACO_Lenzi.exe
PCV_ACO_Global.avi	Animação do PCV com as cidades dispostas geometricamente.
PCV_Random_ACO_Global.avi	Animação do PCV com as cidades dispostas aleatoriamente.
ACO_PCV_primeira versão.mcd	Programa em MATHCAD® do PCV com apenas atualização global do feromônio – 16 cidades dispostas geometricamente.
ACO_PCV_primeira versão random.mcd	Programa em MATHCAD® do PCV com apenas atualização global do feromônio – 16 cidades dispostas aleatoriamente.
ACO_PCV_segunda versão.mcd	Programa em MATHCAD® do PCV com atualizações local e global do feromônio – 16 cidades dispostas geometricamente.
ACO_PCV_segunda versão random.mcd	Programa em MATHCAD® do PCV com atualizações local e global do feromônio – 16 cidades dispostas aleatoriamente.

[voltar para MÉTODO DO FORMIGUEIRO](#)