Algorítmica II. Algoritmos Metaheurísticos

Joaquín Roiz Pagador[1]

Universidad Pablo de Olavide, Carretera Utrera s/n, Sevilla, España,
quiniroiz@gmail.com

Abstract. El siguiente documento pertenece a una serie de documentos que pretende servir a modo de resúmen para el temario de Algorítmica II para el Grado en Ingeniería Informática en Sistemas de Información.

Keywords: algoritmos, metaheurística, búsqueda, tabú

Optimización Colonia de Hormigas.

Hormiga Natural

Las hormigas son insectos sociales que viven en colonias. Tienen un comportamiento dirigido al desarrollo de la colonia. Encuentran el camino más corto entre hormiguero y la comida.

Las hormigas son ciegas. Segregan feromonas al moverse que permiten volver a su hormiguero desde la comida.

Si llegan a una intersección, decide el camino de un modo probabilístico. Las bifurcaciones más prometedoras **acumulan feromonas** por más hormigas que hayan pasado. Las menos prometedoras pierden feromonas por **evaporación**, pero aún así, puede influir.

La visita continuada a un lugar deja un rastro alto de feromonas, haciendo posible un camino cada vez más corto desde el hormiguero a la comida.

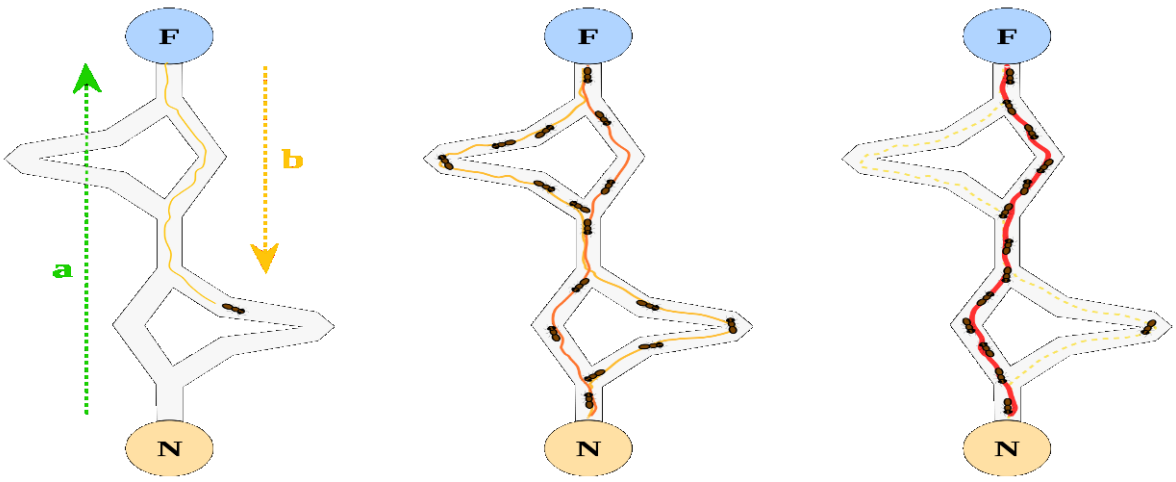


Fig. 1. Experimento del doble puente.

La Hormiga Artificial.

Idea principal.

- Agente que recuerda los nodos que se han recorrido. Listado de nodos en un grafo. Solución construida por la hormiga (ciclo sin repeticiones).
- Por cada paso, en la ciudad **r** se elige hacia qué ciudad **s** moverse de entre las vecinas no visitadas, según una **regla probabilística de transición**.
- Deja feromona a su paso por cada camino recorrido. Con el tiempo, la feromona se **evapora**.

En cada iteración se restará proporcionalmente parte de las feromonas.

La Regla Probabilística.

$$P_k(r,s) = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum\limits_{u \in J_k(r)} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{si } S \in J_k(r) \\ 0, & \text{en otro caso} \end{cases}$$

- $[\eta_{rs}]^\beta \rightarrow$ Cantidad de feromonas y visibilidad.
- $[\eta_{ru}]^\beta \rightarrow$ Cantidad de feromonas y visibilidad de los demás.
- si $S \in J_k(r) \rightarrow$Si es visitable (adyacente, existe arista).
- $\tau_{rs} \rightarrow$ Feromona existente entre las ciudades r y s.
- $\eta_{rs} \rightarrow$ Información Heurística del arco r-s (visibilidad). Normalmente es $\frac{1}{D(r,s)}$.
- $J_k(r) \rightarrow$ Conjunto de nodos que puede visitar la hormiga k, y que no han sido ya visitados en ese recorrido.
- $\alpha, \beta \rightarrow$ Parámetros ajustables $[0, 1]$.
- Si $\alpha = 0$, las ciudades más cercanas(voraz clásico).
- Si $\beta = 0$, sólo interviene la feromona $\rightarrow$ recorridos no muy buenos y sin probabilidad de mejora.

La feromona se actualiza en cada iteración. La evaporación de la feromona se utiliza para evitar un incremento ilimitado de rastros de feromonas. Se olvidan malas soluciones.

Misma evaporación para cada rastro, porcentaje: $0 \leq p \leq 1$. Evitamos óptimos locales con el mecanismo de evaporación más activo que el natural.

Actualización de la feromona.

$$\tau(t) = (1-\rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^{m} \Delta\tau_{rs}^k$$

- $\sum\limits_{k=1}^{m} \Delta\tau_{rs}^k \rightarrow$ Sumar si ha visitado el arco.
- $\rho \rightarrow$ Porcentaje de evaporación.
- $\tau_{rs} \rightarrow$ Feromona.
- t $\rightarrow$ Iteración.

$$\sum_{k=1}^{m} \Delta\tau_{rs}^k = \begin{cases} \frac{1}{C(S_k)}, & \text{si la hormiga k ha visitado el arco } a_{rs}. \\ 0, & \text{en otro caso.} \end{cases}$$

- $C(S_k)$ es el coste de la solución generada por la hormiga k, es decir, la longitud del circuito S_k.
- m es el número total de hormigas.

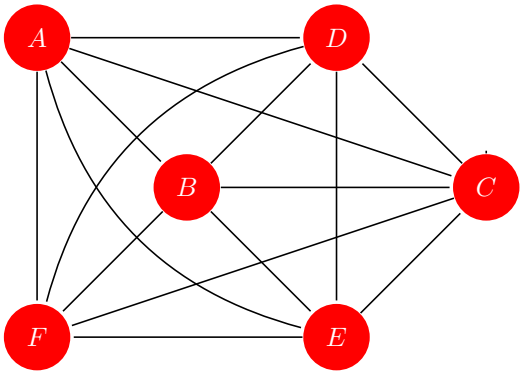
Pseudocódigo.

Inicializar parametros
Asignar cantidad inicial a los rastros de feromonas $\tau[i][j] \leftarrow T_0$
nIter $\leftarrow$ n iteraciones y NIVEL EVAPORACION.
$\alpha\beta$
$m \leftarrow$ n hormigas
while $(it < nIter)$ **do**
  `/* Paso 1: Colocar hormigas en nodos */`
  while $(k < m)$ **do**
    $K[k][1] \leftarrow nodoinicial$
  end
  `/* Paso 2: construir soluciones por hormiga */`
  while $(i < N^o Nodos)$ **do**
    while $(k < N^o Hormigas)$ **do**
      $L[k][i] \leftarrow ReglaTransicion(L[k], \tau, n)$
    end
  end
  `/* Paso 3: Mejor solucion encontrada */`
  while $(k < m)$ **do**
    $Coste[k] \leftarrow C(L[k])$
    $MejorActual \leftarrow Mejor(L[k])$
  end
  `/* Paso 4: Actualizamos feromonas */`
  while $(i < N^o Nodos)$ **do**
    while $(j < N^o Nodos)$ **do**
      $ActualizacionFerom(\tau[i][j], L, C[L], P)$
    end
  end
  `/* Paso 5 */`
  if $C(MejorActual) mejorque C(MejorGlobal)$ **then**
    $MejorGlobal \leftarrow MejorActual$
  end
end

Algorithm 1: Algoritmo Optimización Colonia de Hormigas.

1. Decidir desde qué nodo va a comenzar su ciclo cada hormiga. Puede generarse aleatoriamente.
2. Cada hormiga realiza un ciclo completo sin repetir el nodo. Se aplica la **Regla Probabilística de Transición**.
3. Existen nuevas posibles soluciones a evaluar. Se calcula el coste de cada ciclo.
4. Se actualiza la feromona de las m hormigas del paso 2.
5. Se determina si el mejor camino encontrado en el Paso 3 es el mejor de todos los encontrados hasta ahora.

Ejemplo.



$10\% P(E,F)^{\rightarrow \tau=4}_{\rightarrow \eta=16} = [0, 0.1]$

$20\% P(E,A)^{\rightarrow \tau=18}_{\rightarrow \eta=64} = (0.1, 0.3]$

$40\% P(E,B)^{\rightarrow \tau=26}_{\rightarrow \eta=3} = (0.4, 0.7]$

$10\% P(E,A)^{\rightarrow \tau=6}_{\rightarrow \eta=35} = (0.7, 0.8]$

$20\% P(E,A)^{\rightarrow \tau=4}_{\rightarrow \eta=7} = (0.8, 1)$

Suponemos dado $\rightarrow 0.63$. Por tanto, se tomará $P(E, B)$.

Nota: Valorar objeto nodo, lista/cola prioridad con Comparable $\Rightarrow$ Feromonas + probabilidad. **NO SE REVISITAN NODOS**.

$$\mathrm{L} = \begin{array}{c} \\ H_0 \\ \cdots \\ H_m \end{array} \begin{array}{c} \begin{array}{ccc} N_0 & \cdots & N_n \end{array} \\ \begin{pmatrix} \alpha_0 & \cdots & \alpha_n \\ \beta_0 & \cdots & \beta_n \\ \gamma & \cdots & \gamma \end{pmatrix} \end{array} = \begin{array}{c} coste \\ \begin{pmatrix} \sum \alpha_{H_0} \\ \cdots \\ \sum \gamma_{H_m} \end{pmatrix} \end{array}$$

- L será nuestra matriz de recorridos.
- N representa cada uno de los nodos visitados.
- H representa cada una de las hormigas que realiza un recorrido.

Si leemos horizontalmente, cada hormiga realizará un recorrido hacia un nodo, en cada iteración, hasta completar sin repeticiones el número de nodos a visitar.

References

1. Material asignatura, *Metaheurísticas*, UGR. `http://sci2s.ugr.es/graduateCourses/Metaheuristicas`
2. E.-G. Talbi, Metaheuristics. From design to implementation, Wiley, 2009 Material.
3. C. Blum, A Roli, Metaheuristics in Combinatorial Optimization: overview and conceptual comparison. ACM Computing Surveys, 35(3), 2003, 268-308.
4. B. Melián, F. Glover. Introducción a la búqueda tabú, 2006. `http://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS_Spanish%20w%20Belen(11-9-06).pdf`