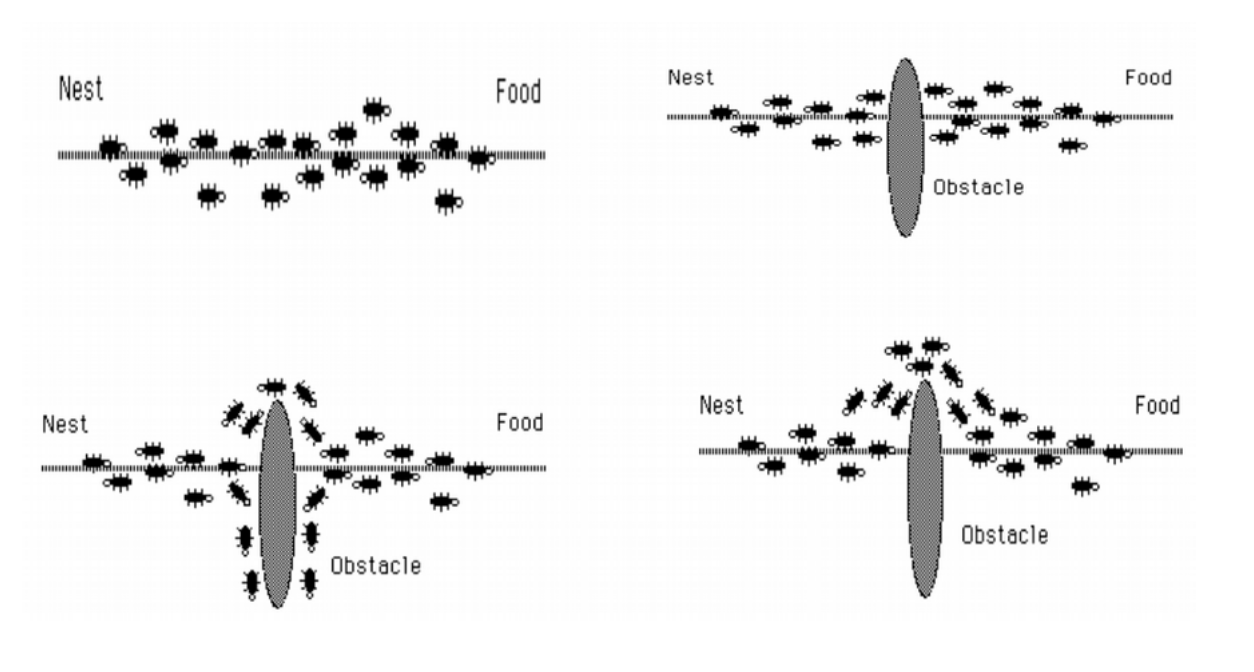
1. *El problema*

La Optimización de Colonia de Hormiga, agrupa un conjunto de algoritmos que se inspira en el comportamiento de las colonias de hormigas para resolver problemas discretos de optimización. El pionero en esta área fue Marco Dorigo, quien en 1992 introdujo la primera propuesta de algoritmo.

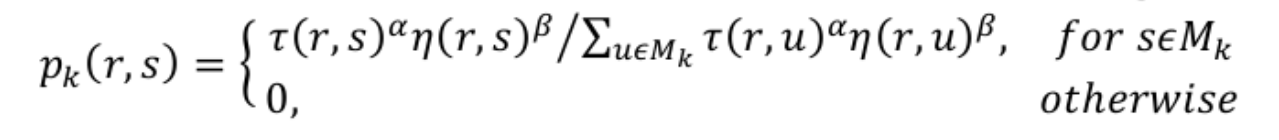


Al observar el diagrama anterior, la pregunta a responder es: ¿Cómo hacen las hormigas para terminar recorriendo el camino más corto entre el nido y la comida? El secreto es que las hormigas marcan su trayectoria con feromonas, lo que le indica a sus compañeras un rastro que es más fuerte en la medida en que la trayectoria es más frecuentada. Las feromonas se acumulan más en trayectorias cortas que en las largas, por lo que, después de un tiempo, todas las hormigas terminan transitando la trayectoria más corta, al ser esta la más atractiva. En este comportamiento de las hormigas se han inspirado muchos algoritmos de optimización, aplicados a problemas de enrutamiento de redes y en los sistemas de transportes urbanos. La primera propuesta (Sistema Hormiga o Ant System) tenía como objetivo resolver el problema del viajante que consiste en encontrar el camino hamiltoniano más corto en un grafo completo. El problema del viajero (TSP o travelling salesman problem, por sus siglas en inglés) puede ser modelado como un grafo ponderado no dirigido, en el cual los vértices son las ciudades y los caminos entre ellas son las aristas que en este caso están ponderados, pues cada arista tendrá al menos un valor que corresponde a la distancia entre las dos ciudades que conecta. Tome en cuenta que una arista puede ser multiponderada.

En el algoritmo de optimización llamado Sistema Hormiga, se realizan los siguientes pasos:

1. Se inicializan los valores de la simulación. Al principio cada camino entre las ciudades tiene una cantidad de feromona inicial. Además, las hormigas son ubicadas en la ciudad inicial (nido). Debe tomarse en cuenta que cada hormiga debe irse moviendo hasta alcanzar la ciudad final (comida). El valor inicial de la cantidad de feromonas en los caminos puede ser calculado de la siguiente manera: ***τ=1/m***, donde ***τ*** es la cantidad inicial de feromonas y ***m*** es la cantidad de nodos o ciudades.

2. Selección del camino a seguir. Por cada hormiga, debe determinarse la siguiente ciudad a la cual va a moverse la hormiga. Esto se logra de manera probabilística a través de la siguiente expresión:

Donde:

***α*** es el grado de importancia de la feromona. Se recomienda un valor de 1.

***β*** representa el grado de visibilidad de la ciudad. (**α** y **β** son parámetros propios del algoritmo). Se recomienda un valor de 2.

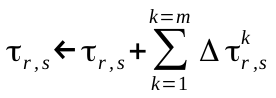
***u*** representa a una de las ciudades que pertenecen al conjunto de las ciudades adyacentes a la ciudad ***r***en la que la hormiga ***k*** puede visitar.

***η*** es un factor que usualmente se calcula con la expresión ***η=Q/drs*** , donde ***drs*** es la distancia entre las ciudades ***r*** y ***s***. ***Q*** es, en algunos casos, considerada como el aprendizaje que usualmente es 1. En otras palabras ***η*** es el inverso de la distancia ***drs***, y es conocido como el factor de visibilidad del camino.

***Mk*** es el conjunto de las ciudades que pueden ser visitadas por la hormiga ***k***. Una ciudad no puede ser visitada dos veces por una hormiga en una misma iteración.

Tome en cuenta que una vez calculadas las probabilidades de cada camino disponible, es necesario generar un número aleatorio para saber cuál camino seguirá la hormiga.

3. Actualización de la cantidad de feromonas por incremento. Una vez que una hormiga pase por una arista que conecta a dos ciudades, deja su rastro de feromonas. La actualización de la cantidad de feromonas en una arista se calcula de la siguiente manera:



Donde:

***m*** es el número de hormigas.

r,sk=QLk Es calculado para cada hormiga k que ha transitado ese camino. Si una hormiga ha transitado el camino cr,sk entre las ciudades ***r*** y ***s***, entonces *Q* es una constante (usualmente igual a 1) y ***Lk*** es la longitud del recorrido realizado por la hormiga.

4. Actualización de la cantidad de feromonas por evaporación: una vez que todas las hormigas han culminado su viaje (es decir, han llegado a la ciudad destino o alimento), se actualiza la cantidad de feromonas de todos los caminos entre las ciudades mediante la siguiente ecuación:

**τr,s ←(1−ρ)∗τr,s**

Donde ρ ⊆ (0,1] es el factor de evaporación. Usualmente es 0.5. Es en este momento cuando termina la iteración; es decir, cada hormiga debe irse moviendo hasta alcanzar el alimento y una vez que lo alcanza, no se sigue moviendo, pues no se está considerando el camino de regreso como una forma de simplificar el problema. Es importante considerar la posibilidad de que una hormiga llegue a una calle ciega, es decir que no pueda moverse debido a que ya ha visitado las ciudades que se comunican con la ciudad en la que está ubicada.

1. *Requerimientos funcionales:*

**Nueva simulación**: el usuario podrá iniciar otra, manteniéndose la estructura del grafo en el caso de que no se haya realizado una simulación anterior; sin embargo, debe ser opcional el agregar o eliminar ciudades, pero únicamente antes de iniciar una simulación. Se deberán indicar los otros valores iniciales a saber:

1. *Número de iteraciones a realizar*. Por cada iteración, cada hormiga debe haber terminado su recorrido, es decir, la hormiga debe haber llegado a su destino.

2. Número de hormigas que formarán parte de la simulación.

3. Valores de ***α***, ***β*** y ***ρ***. El usuario podrá utilizar valores por defecto o indicar otros.

**Agregar ciudad**. Tome en cuenta que una ciudad debe conectarse con otras ciudades y que dicho camino corresponde a una arista no dirigida. Cuando se agrega una ciudad, se debe indicar las distancias a las otras ciudades. Las simulaciones se podrán realizar con un mínimo de 4 ciudades y hasta un máximo de 20 ciudades.

**Eliminar ciudad**. Lo que implica que se deben eliminar los caminos conectados a esta. Esto solo puede ocurrir antes de comenzar las iteraciones.

**Guardar grafo**. El programa debe poder guardar la estructura del grafo en un archivo de texto, de manera que al salir del programa, dicha información no se pierda y pueda ser cargada nuevamente si lo desea al usuario.

**Cargar grafo**. El usuario podrá cargar un grafo desde un archivo de texto, antes de comenzar una simulación.

Por otra parte, se requiere que la interfaz muestre:

a. Cantidad de hormigas que están en la simulación.

b. Por cada iteración indicar el camino identificado como el más óptimo y por cada hormiga, el recorrido realizado y su distancia correspondiente.

c. Por cada iteración, el valor de ***τr***,s de cada camino (arco entre cada ciudad).

d. Debe presentarse grafo correspondiente y se debe indicar el camino óptimo entre la ciudad de inicio y la de destino.