

Universidad Tecnológica de la Mixteca

Instituto de Computación



Metaheurísticas

Reporte:

Algoritmo de colonia de hormigas aplicado a TCP.

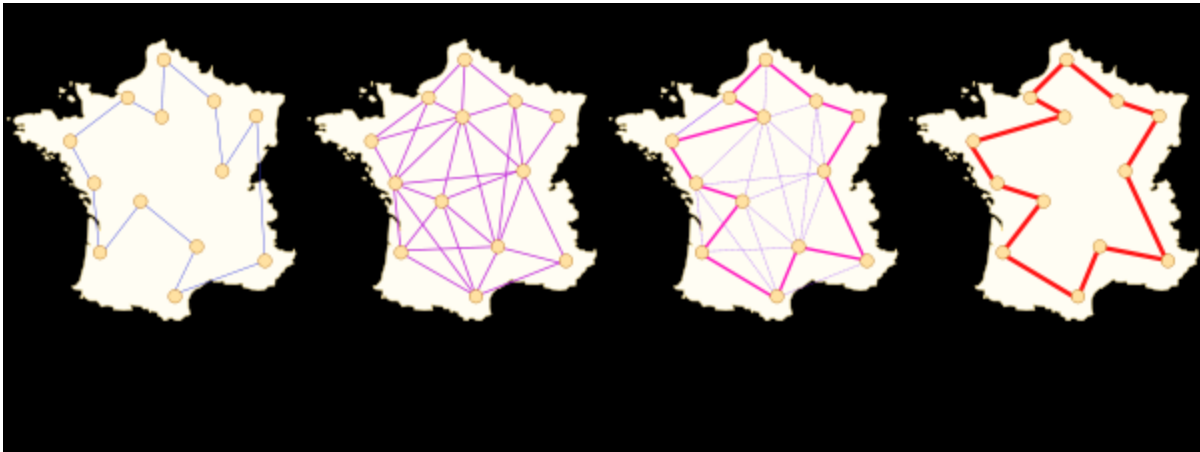
Dra. Luvia Carolina Morales Reynaga
Profesora-Investigadora

Fernando Torres Bautista
Alumno

16 de enero de 2018

1. TCP (Travelling Salesman Problem)

El problema del viajante consiste en encontrar la ruta más corta que debe llevar a cabo un vendedor que, comenzando por un ciudad de origen visite un determinado y preestablecido conjunto de ciudades y vuelva a la ciudad original, con la restricción de que por cada ciudad sólo pase una vez.



2. Optimización por colonia de hormigas

Los algoritmos de OCH se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales, esto es, unos agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales.

Los algoritmos de OCH son esencialmente algoritmos constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arista del grafo, que representa los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga:

- Información heurística, que mide la preferencia heurística de moverse desde el nodo r hasta el nodo s , o sea, de recorrer la arista Ars . Las hormigas no modifican esta información durante la ejecución del algoritmo.
- Información de los rastros de feromona artificiales, que mide lo deseable del movimiento de r a s . Imita a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas.

2.1 Parámetros

- #Hormigas en la colonia.

- Valores de Alpha y Beta.
- Rho o constante de evaporación entre 0 y 1.
- #Máximo de iteraciones.
- #Ciudades seleccionadas por su probabilidad para seleccionar el siguiente estado.

3. Implementación

Se implemento el OCH con la estructura siguiente:

```
Mientras #iteraciones < #MaxIteraciones
    Desde ant=1:#Hormigas
        // Encontrar una solución
        Mientras ruta( ant(i) ) .size() != #Ciudades
            ruta( ant(i) ).push_back( Siguiente_estado() )
        //Actualizar Feromonas
        ActualizarFeromonas( ruta( ant(i) ) )
        //Actualizar mejor estado
        ActualizarMejorRuta( Xbest, ruta( ant(i) ) )
    #iteraciones++
FinMientras
```

Para la implementación del siguiente estado se hace en base a la probabilidad de las mejores ciudades eligiendo al azar entre un número de las mejores.

- La probabilidad de que las hormigas de un nodo particular i elijan la ruta del nodo i al nodo j , viene dada por:

$$p_{ij} = \frac{\phi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i,j=1}^n \phi_{ij}^{\alpha} d_{ij}^{\beta}}$$

Donde $\alpha > 0$ y $\beta > 0$ son los parámetros de influencia.

Sus valores típicos son $\alpha \approx \beta \approx 2$.

ϕ_{ij} es la concentración de feromona en la ruta entre i y j .

d_{ij} es la atracción de la ruta entre i y j . En este caso se toma la inversa de la distancia entre el nodo i y el nodo j .

- Al momento de actualizar las feromonas para cada arista del grafo si se encuentra en la ruta solución generada por la hormiga se sigue la siguiente forma de actualización de la feromona:

If(Arista(i,j).inRuta(ruta(ant(i))))

Arista(i, j) = Arista(i, j)*(1-evaporación) + invDistancia(i, j)

else

Arista(i, j) = Arista(i, j)*(1-evaporación)

4. Resultados

Se utilizaron tres archivos de prueba que se describen a continuación:

Nombre archivo	Número de ciudades	Costo de la solución óptima
1.tsp	436	1443
2.tsp	662	2513
380.txt	380	1621

Se hizo uso de la misma combinación de parámetros para cada conjunto de soluciones obtenidas mostradas a continuación.

Parámetro\Conjuntos Solucion	S1	S2	S3	S4	S5
# Hormigas	10				20
Alpha	2				2
Beta	2		1		1
Rho(%evaporación)	0.1				0.1
# Iteraciones	10	15	20	10	20
# NCxSPA	5	10	10	1	3

NcxSPA es la constante en el código para el número de ciudades seleccionadas por probabilidad para escoger el siguiente estado, si éste parámetro es igual a 1 se selecciona la mejor probabilidad de las obtenidas en la función de selección del siguiente estado.

Combinaciones de parámetros para cada solución:

S1: 10 2 2 0.1 10 5

S2: 10 2 2 0.1 15 10

S3: 10 2 1 0.1 20 10

S4: 10 2 1 0.1 10 1

S5: 20 2 1 0.1 20 3

Se hace una variación simple a beta para que tenga más peso en la ecuación la cantidad de feromonas que contenga la arista.

La solución S4 solo contempla a la mejor probabilidad del nodo actual al siguiente.

Las soluciones obtenidas se muestran a continuación por cada conjunto de combinación de parámetros para los tres archivos de prueba.

S1

Archivo	Peor tiempo ejecución por iteración	Mejor tiempo de ejecución por iteración	Tiempo promedio	Costo de ruta
1.tsp	12072.4 ms	9154.85 ms	10235.1 ms	3198.44
2.tsp	27359.2 ms	27097.7 ms	27212.2 ms	5511.95
380.txt	6917.29 ms	6693.82 ms	6779.1 ms	3556.88

S2

Archivo	Peor tiempo ejecución por iteración	Mejor tiempo de ejecución por iteración	Tiempo promedio	Costo de ruta
1.tsp	10427.1 ms	9166.53 ms	9355.9 ms	4870.62
2.tsp	30922.7 ms	27111.6 ms	27931.2 ms	8566.74
380.txt	6634.37 ms	6602.81 ms	6625.27 ms	5497.97

La solución 1 y 2 solo varían en el número de ciudades de las cuales obtener la siguiente, de 5 a 10, y un número de iteraciones de 10 a 15.

La solución obtenida en cada ruta para S2 es peor por la cantidad de ciudades seleccionadas para su siguiente estado.

S3

Archivo	Peor tiempo ejecución por iteración	Mejor tiempo de ejecución por iteración	Tiempo promedio	Costo de ruta
1.tsp	11324.9 ms	9154.56 ms	9402.98 ms	4766.11
2.tsp	30178.1 ms	27202.1 ms	27418.1 ms	8657.94
380.txt	6617.55 ms	6585.92 ms	6638.72 ms	5589.4

Con la solución S3 dándole más valor a las feromonas contenidas en las rutas haciendo $\beta=1$ para considerar menos a las distancias, y en lugar de 15 ahora 20 iteraciones, la solución no mejora en comparación con S2.

S4

Para ésta solución en particular se escogió siempre el mejor candidato dado por su probabilidad.

Solo con 10 hormigas y 10 iteraciones bastaron para acercarse de mejor forma a la solución.

Archivo	Peor tiempo ejecución por iteración	Mejor tiempo de ejecución por iteración	Tiempo promedio	Costo de ruta	Diferencia con el óptimo
1.tsp	9286.56 ms	9142.01 ms	9194.15 ms	1794.09	351.09

2.tsp	27503.3 ms	26954.2 ms	27100.4 ms	3054.36	541.36
380.txt	7512.11 ms	6544.06 ms	6733.69 ms	1962.87	341.87

S5

Con el doble de hormigas y el doble de iteraciones conservando los parámetros anteriores no se encuentran soluciones mejores que en S4, sin embargo mejora en relación con las 3 primeras pruebas.

Archivo	Peor tiempo ejecución por iteración	Mejor tiempo de ejecución por iteración	Tiempo promedio	Costo de ruta
1.tsp	19865.2 ms	19392.7 ms	19520.9 ms	2288.6
2.tsp	59120.9 ms	57134.7 ms	57733 ms	4086.66
380.txt	14345.1 ms	13923.4 ms	14052.7 ms	2641.78

La variación más grande que se encontró en la realización de las pruebas es en la forma de tomar el siguiente vecino de un conjunto de los mejores, mientras menos el número de los cuales seleccionar al mejor mejor la solución, sin embargo, el número de iteraciones y la relación entre el parámetro Alpha y Beta para dar peso a la distancia entre ciudades o a las feromonas que existen en la arista cambia el valor de las soluciones obtenidas.

El valor más próximo al óptimo está dado por la solución S4.