Modelos de Investigación Operativa, Ingeniería Informática
Universidad de Valladolid

Práctica 13

Daniel González Alonso

29 de abril de 2017

Resumen

En este documento se describen los problemas y los resultados obtenidos de la práctica 13 del tema 5 de la asignatura Modelos de Investigación Operativa de Ingeniería Informática, Universidad de Valladolid.

1. Introducción

Esta práctica trata de problemas TSP (*Travelling Salesman Problem*). Los problemas TSP constan de un grafo $G = (N, A)$, donde N son los nodos del grafo y A los arcos entre éstos, con un coste asociado por cada arco, y el objetivo consiste en encontrar el camino Hamiltoniano (un camino que pase por todos los nodos) de coste mínimo.

En esta práctica se nos pide implementar la solución al problema TSP mediante la metaheurística *Simulated Annealing* para TSP. El pseudocódigo de la esta metaheurística se muestra a continuación:

Algoritmo 1 Metaheurística *Simulated Annealing*

```
1: function SimulatedAnnealing(T_0, X)
2:     T ← T_0
3:     x_1 ← GenerarSolucionInicial(X)
4:     x* ← x_1, F* ← F(x_1)
5:     flag = 0
6:     while Condición de parada no se satisfaga do
7:         x ← GenerarSolucionEnEntorno(x_n)
8:         if F(x) ≤ F(x_n) then
9:             x_n ← x
10:        end if
```

```
        if F* < F(x_n) then
            x* ← x, F* ← F(x)
        else
            p ← RANDOM                          ▷ Generar número aleatorio en [0,1]
            if p ≤ p(n) then
                x_n ← x
            end if
        end if
        DISMINUIR(T)
    end while
    return x*
end function
```

En nuestro caso, para la función GenerarSolucionInicial(X) se ha empleado la heurística del entorno más cercano. Para la función GenerarSolucionEnEntorno(x_n) se elegían aleatoriamente dos nodos i y j no contiguos, y se hacía el intercambio explicado en la práctica 11. Para calcular el valor de p_n se empleaba la siguiente fórmula:

$$p_n = \exp\left(-\frac{\Delta_{i,j}}{T_n}\right) \tag{1}$$

Donde $\Delta_{i,j}$ es la mejora del intercambio de la solución x respecto a x_n. Por otro lado para la función Disminuir(T), es decir la disminución de la temperatura, se escogió la disminución lineal, la cual sigue la función

$$T_{n+1} = \alpha \cdot T_n \tag{2}$$

Siendo el valor α una constante a definir. Por último como condición de parada se eligió tanto limitar el número máximo de iteraciones, como limitar el número máximo de iteraciones sin mejora K.

2. Desarrollo

En esta práctica hay que programar la heurística *Simulated Annealing* y aplicarla a los 5 ejemplos de $n = 21$ nodos y los 6 problemas Euclídeos de las prácticas anteriores.

Estos problemas se encuentran resueltos mediante *Xpress Mosel* en los ficheros `tsp_sa_n21_1.mos`, `tsp_sa_n21_2.mos`, `tsp_sa_n21_3.mos`, `tsp_sa_n21_4.mos`, `tsp_sa_n21_5.mos` en el caso de los ficheros `n21` y por otro lado para los ficheros `tsp` Euclídeos en los ficheros `tsp_sa_tsp_60_1.mos`, `tsp_sa_tsp_60_2.mos`, `tsp_sa_tsp_60_3.mos`, `tsp_sa_tsp_100_1.mos`, `tsp_sa_tsp_100_2.mos` y `tsp_sa_tsp_100_3.mos` (el nombre indica el fichero de datos empleado).

Antes de explicar la implementación del algoritmo cabe destacar que los costes $c_{i,j}$ en nuestro caso son distancias. Para los ficheros `n21` la matriz de distancias nos viene dada en el mismo fichero. En el caso de los ficheros `tsp` solo nos vienen las coordenadas de cada nodo, por ello antes de empezar con estos últimos ficheros hay que calcular la matriz de distancias. Para estos fichero la matriz se calculo mediante la distancia Euclídea redondeada al entero más cercano. En caso de la distancia de un nodo a si mismo, se introducía en esta matriz en vez de 0 un valor "infinito" (`MAX_INT`).

Para el algoritmo *Simulated Annealing* como se ha explicado en el esquema anterior lo primero que hay que obtener es una solución inicial, y como se hay dicho se empleó la heurística del entorno más cercano, ya implementado en la práctica 10.

Posteriormente, ya dentro del bucle principal, los pasos que se hicieron siguiendo el esquema 1 fueron los siguientes:

1. Generar solución en el entorno de la solución actual: Para esta parte, simplemente elegía dos nodos aleatoriamente, y después al igual que se hizo en la práctica 11 con el algoritmo *2-opt*, se intercambia los arcos $(i, s(i))$ y $(j, s(j))$ por $(i, j$ y $(s(i), s(j))$, y se invertía de dirección el camino entre ellos. Además también se calculaba el valor $\Delta_{i,j}$ para obtener la distancia total de la solución actual.

2. Actualizar la solución óptima: en este caso simplemente reemplazaríamos el vector `siguientes` y la `distancia_minima` por la de la solución actual en caso de que su distancia fuese inferior.

3. Actualizar la solución x_n: en este caso, al igual que en el paso anterior si la solución actual fuese mejor o igual reemplazaríamos su vector de siguientes y la distancia. Si no fuese menor o igual, lo reemplazaríamos con probabilidad p basándonos en la fórmula 1.

4. Actualizar T_n: Para esta parte simplemente aplicamos la fórmula 2. En el caso de los ficheros `n21` empleó un valor de $\alpha = 0{,}961$, $\alpha = 0{,}931$ para los ficheros `tsp_60` y $\alpha = 0{,}91$ para los fichero `tsp_100`.

5. Actualizar la condición de parada: para la condición de parada utilizamos una bandera, la cual pararía el bucle en caso de que se alcanzase un máximo número de iteraciones sin mejorar ($K = 58000$ en el caso de los ficheros `n21`, $K = 65000$ para `tsp_60` y $K = 2000000$ para `tsp_100`) la solución o un máximo número de iteraciones.

3. Resultados

Los resultados obtenidos para los ficheros de datos de esta práctica fueron los siguientes:

Problema TSP	`n21_1`	`n21_2`	`n21_3`	`n21_4`	`n21_5`
Distancia Total	243	234	227	231	253
Conexiones	$1 \to 8 \to 14 \to 4 \to 15 \to 3 \to 16 \to 2 \to 6 \to 12 \to 9 \to 11 \to 13 \to 19 \to 18 \to 20 \to 10 \to 17 \to 7 \to 21 \to 5$	$1 \to 2 \to 11 \to 10 \to 20 \to 8 \to 4 \to 21 \to 13 \to 18 \to 16 \to 12 \to 17 \to 7 \to 6 \to 5 \to 9 \to 19 \to 3 \to 15 \to 14$	$1 \to 7 \to 6 \to 20 \to 19 \to 13 \to 12 \to 18 \to 15 \to 16 \to 8 \to 3 \to 5 \to 17 \to 10 \to 9 \to 11 \to 2 \to 21 \to 4 \to 14$	$1 \to 4 \to 13 \to 21 \to 12 \to 5 \to 17 \to 18 \to 19 \to 16 \to 10 \to 8 \to 3 \to 20 \to 11 \to 15 \to 14 \to 7 \to 6 \to 2 \to 9$	$1 \to 7 \to 10 \to 16 \to 15 \to 5 \to 11 \to 6 \to 3 \to 18 \to 13 \to 19 \to 21 \to 4 \to 2 \to 9 \to 17 \to 8 \to 20 \to 14 \to 12$

Cuadro 1: Resultados obtenidos para los ficheros `n21`

Problema TSP	`tsp_60_1`	`tsp_60_2`	`tsp_60_3`
Distancia Total	837	709	732
Conexiones	$1 \to 35 \to 53 \to 27 \to 23 \to 32 \to 8 \to 12 \to 20 \to 51 \to 54 \to 39 \to 31 \to 50 \to 58 \to 49 \to 40 \to 60 \to 14 \to 37 \to 18 \to 29 \to 4 \to 34 \to 16 \to 25 \to 19 \to 9 \to 43 \to 47 \to 55 \to 28 \to 56 \to 5 \to 42 \to 48 \to 36 \to 11 \to 44 \to 59 \to 3 \to 33 \to 45 \to 52 \to 24 \to 41 \to 38 \to 17 \to 2 \to 7 \to 21 \to 15 \to 26 \to 6 \to 57 \to 46 \to 30 \to 13 \to 22 \to 10$	$1 \to 30 \to 9 \to 28 \to 4 \to 44 \to 29 \to 13 \to 5 \to 27 \to 8 \to 40 \to 50 \to 56 \to 15 \to 54 \to 7 \to 26 \to 34 \to 60 \to 10 \to 14 \to 47 \to 39 \to 19 \to 25 \to 49 \to 58 \to 59 \to 41 \to 31 \to 23 \to 48 \to 42 \to 12 \to 35 \to 11 \to 22 \to 24 \to 32 \to 37 \to 18 \to 51 \to 43 \to 16 \to 21 \to 33 \to 6 \to 45 \to 46 \to 52 \to 38 \to 55 \to 17 \to 2 \to 53 \to 20 \to 36 \to 3 \to 57$	$1 \to 9 \to 27 \to 4 \to 38 \to 49 \to 40 \to 32 \to 13 \to 35 \to 36 \to 8 \to 37 \to 59 \to 52 \to 18 \to 23 \to 56 \to 48 \to 58 \to 14 \to 16 \to 44 \to 45 \to 42 \to 17 \to 47 \to 55 \to 3 \to 19 \to 51 \to 5 \to 50 \to 22 \to 20 \to 28 \to 12 \to 6 \to 41 \to 26 \to 33 \to 46 \to 43 \to 39 \to 53 \to 29 \to 24 \to 57 \to 34 \to 2 \to 31 \to 60 \to 54 \to 15 \to 10 \to 25 \to 11 \to 21 \to 7 \to 30$

Cuadro 2: Comparación de los resultados de los ficheros `tsp_60`

Problema TSP	tsp_100_1	tsp_100_2	tsp_100_3
Distancia Total	957	937	1023
Conexiones	1 → 24 → 41 → 59 → 57 → 40 → 31 → 64 → 80 → 63 → 36 → 47 → 99 → 4 → 67 → 79 → 82 → 19 → 45 → 97 → 50 → 2 → 93 → 68 → 98 → 8 → 53 → 72 → 6 → 28 → 30 → 56 → 55 → 9 → 81 → 39 → 33 → 100 → 3 → 73 → 75 → 15 → 74 → 18 → 65 → 23 → 71 → 10 → 25 → 17 → 32 → 52 → 66 → 12 → 42 → 16 → 61 → 26 → 84 → 14 → 83 → 37 → 77 → 60 → 38 → 88 → 96 → 85 → 70 → 94 → 44 → 89 → 21 → 87 → 91 → 34 → 7 → 22 → 90 → 62 → 58 → 46 → 35 → 54 → 43 → 76 → 49 → 95 → 13 → 48 → 11 → 27 → 29 → 86 → 92 → 78 → 5 → 69 → 20 → 51	1 → 22 → 45 → 82 → 11 → 62 → 92 → 4 → 35 → 44 → 48 → 94 → 34 → 42 → 68 → 56 → 32 → 85 → 61 → 25 → 99 → 38 → 66 → 73 → 19 → 43 → 74 → 40 → 7 → 64 → 83 → 41 → 60 → 15 → 90 → 36 → 30 → 17 → 10 → 58 → 29 → 72 → 18 → 33 → 70 → 88 → 91 → 100 → 69 → 95 → 2 → 49 → 46 → 12 → 80 → 93 → 50 → 3 → 71 → 14 → 65 → 52 → 96 → 89 → 97 → 79 → 28 → 63 → 98 → 78 → 77 → 86 → 9 → 55 → 53 → 51 → 26 → 6 → 59 → 57 → 8 → 5 → 13 → 81 → 84 → 24 → 39 → 54 → 67 → 27 → 75 → 47 → 37 → 21 → 76 → 23 → 16 → 20 → 31 → 87	1 → 49 → 88 → 67 → 68 → 27 → 90 → 58 → 39 → 80 → 83 → 60 → 22 → 46 → 71 → 95 → 93 → 19 → 69 → 98 → 23 → 100 → 51 → 76 → 91 → 74 → 65 → 29 → 38 → 44 → 79 → 36 → 33 → 89 → 84 → 41 → 75 → 47 → 24 → 2 → 72 → 42 → 4 → 55 → 18 → 57 → 99 → 66 → 94 → 10 → 12 → 86 → 85 → 11 → 53 → 32 → 5 → 50 → 8 → 9 → 59 → 87 → 43 → 31 → 20 → 30 → 25 → 21 → 97 → 73 → 34 → 63 → 92 → 14 → 96 → 3 → 45 → 78 → 28 → 82 → 56 → 52 → 54 → 13 → 48 → 40 → 81 → 7 → 62 → 70 → 17 → 6 → 64 → 77 → 15 → 26 → 61 → 35 → 37 → 16

Cuadro 3: Comparación de los resultados de los ficheros tsp_100

2 También obtuve los gráficos IVE para los ficheros `tsp`. En este caso aquí se muestra el resultado obtenido para el fichero `tsp_100_1`: