

## Práctica 12

---

Daniel González Alonso

28 de abril de 2017

### Resumen

En este documento se describen los problemas y los resultados obtenidos de la práctica 12 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 *GRASP*. El pseudocódigo de la esta metaheurística se muestra a continuación:

---

**Algoritmo 1** Metaheurística *GRASP*

---

```
1: function GRASP( $N$ )
2:    $f^* \leftarrow \infty$ 
3:   for  $i \leftarrow 1, N$  do
4:      $x \leftarrow \text{SOLUCIÓNGREEDY}()$ 
5:      $x \leftarrow \text{BÚSQUEDALOCAL}(x)$ 
6:     if  $f(x) < f^*$  then
7:        $f^* \leftarrow f(x)$ 
8:        $x^* \leftarrow x$ 
9:     end if
10:  end for
11:  return  $x^*$ 
12: end function
```

---

En nuestro caso,  $N$  es el número máximo de iteraciones,  $\text{SOLUCIÓNGREEDY}()$  es la solución de la heurística del entorno y  $\text{BÚSQUEDALOCAL}(x)$  es el algoritmo *2-opt*.

## 2. DESARROLLO

En esta práctica hay que programar la heurística *GRASP* partiendo de la solución que proporciona la heurística del entorno más cercano (comenzando en un nodo aleatoriamente escogido) y aplicarlo a los 5 ejemplos de  $n = 21$  nodos y los 6 problemas Euclídeos de las prácticas anteriores. Parámetros sugeridos:  $N = 30, 60$  y  $100$ , y  $K = 5$ .

Estos problemas se encuentran resueltos mediante *Xpress Mosel* en los ficheros `tsp_grasp_n21_1.mos`, `tsp_grasp_n21_2.mos`, `tsp_grasp_n21_3.mos`, `tsp_grasp_n21_4.mos`, `tsp_grasp_n21_5.mos` en el caso de los ficheros `n21` y por otro lado para los ficheros `tsp` Euclídeos en los ficheros `tsp_grasp_tsp_60_1.mos`, `tsp_grasp_tsp_60_2.mos`, `tsp_grasp_tsp_60_3.mos`, `tsp_grasp_tsp_100_1.mos`, `tsp_grasp_tsp_100_2.mos` y `tsp_grasp_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 ficheros la matriz se calcula mediante la distancia Euclídea redondeada al entero más cercano. En caso de la distancia de un nodo a si mismo, se introduce en esta matriz en vez de 0 un valor “infinito” (`MAX_INT`).

Una vez obtenidos los costes, lo primero que se hizo fue un bucle que itera entre los valores de  $N$  que nos pide el enunciado (30, 60, y 100). Dentro de ese bucle se ejecuta la solución *GRASP* con cada valor de  $N$ .

Para implementar la solución *GRASP* siguiendo el esquema 1, primero definí una variable llamada `distancia_minima` ( $f^*$  en el esquema superior) así como un vector `siguientes` los cuales almacenan la mejor solución encontrada por el algoritmo. Después cree un bucle que se ejecuta  $N$  veces, el cual en cada iteración hace lo siguiente:

1. Obtener una solución Greedy: Para esta parte reutilicé el código de la práctica 10 donde se obtenía una solución Greedy mediante la heurística del entorno más cercano.
2. Hacer la búsqueda local: Para hacer la búsqueda local en el entorno de la solución obtenida por el algoritmo Greedy utilizamos la mejora *2-opt*. Para esta parte utilicé el mismo código que el empleado para la práctica 11.
3. Actualizar la solución óptima: En esta parte, en caso de que la distancia total obtenida en los pasos anteriores sea mejor que `distancia_minima` actualizamos este valor así como el vector `siguientes` con la solución de la iteración actual.

### 3. RESULTADOS

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

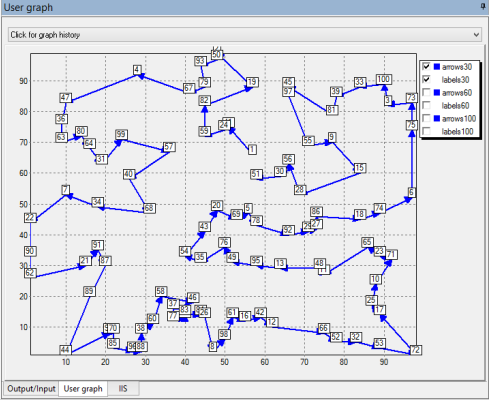
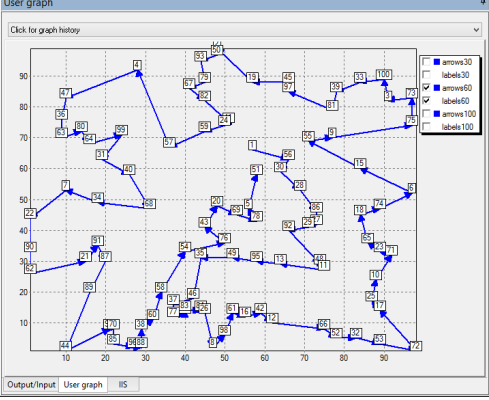
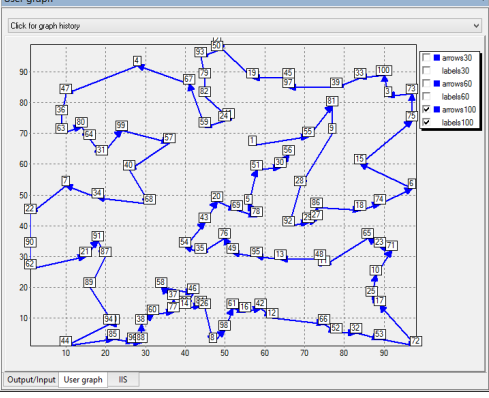
Valor de N	Problema TSP				
	n21_1	n21_2	n21_3	n21_4	n21_5
30	198	174	213	189	193
60	198	174	213	189	193
100	198	174	213	189	193

Cuadro 1: Comparación de las distancias totales obtenidas con distintos valores de  $N$  para los ficheros **n21**

Valor de N	Problema TSP					
	tsp_60_1	tsp_60_2	tsp_60_3	tsp_100_1	tsp_100_2	tsp_100_3
30	633	611	592	779	787	815
60	628	607	589	751	762	810
100	632	614	586	760	777	789

Cuadro 2: Comparación de las distancias totales obtenidas con distintos valores de  $N$  para los ficheros **tsp**

También obtuve los gráficos IVE para los ficheros **tsp**. En este caso aquí se muestra una comparación de los resultados obtenidos para **tsp\_100\_1** con distintos valores de  $N$ :

Valor de $N$	Solución <i>GRASP</i>
30	 <p>The screenshot shows the 'User graph' window of the GRASP software. The graph is plotted on a grid with x and y axes ranging from 10 to 90. Numerous nodes are labeled with numbers. A path is highlighted in blue, starting from node 1 and ending at node 100. The path consists of several segments, some of which are labeled with '30', indicating the value of N. The legend on the right shows that 'arrows30' is selected, and 'labels30' is also selected. The 'Output/Input' tab is active at the bottom.</p>
60	 <p>This screenshot is identical to the one for N=30, showing the same graph and path. The path is highlighted in blue, and the legend indicates that 'arrows30' and 'labels30' are selected. The 'Output/Input' tab is active at the bottom.</p>
100	 <p>This screenshot is identical to the one for N=30, showing the same graph and path. The path is highlighted in blue, and the legend indicates that 'arrows30' and 'labels30' are selected. The 'Output/Input' tab is active at the bottom.</p>

Cuadro 3: Comparación de los caminos obtenidos con distintos valores de  $N$  para el ficheros `tsp_100_1.txt`