Modelos de Investigación Operativa, Ingeniería Informática
Universidad de Valladolid

Práctica 16

Daniel González Alonso

13 de mayo de 2017

Resumen

En este documento se describen los problemas y los resultados obtenidos de la práctica 16 del tema 6 de la asignatura Modelos de Investigación Operativa de Ingeniería Informática, Universidad de Valladolid.

1. Introducción

Esta práctica, al igual que la práctica anterior, trata de problemas de Rutas de Vehículos Capacitado (CVRP, *Capacited Vehicle Routing Problem*).

Tal y como se presentó en la práctica anterior, los problemas CVRP tratan determinar las rutas que una flota de vehículos ha de seguir con el fin de cubrir la demanda de todos sus clientes. Los datos del problema se representan como un grafo $G = (V, A)$, donde V son los vértices del grafo, el primer vértice es el depósito de salida de los vehículos y el resto los clientes a a abastecer, y A los arcos entre éstos, con un coste $c_{i,j}$ asociado a cada arco. Cada vértice del grafo tiene asociado una demanda $d_i \geq 0$, en el caso del depósito esta demanda es siempre $d_1 = 0$. Por otro lado, se supone que nuestra flota de vehículos es homogénea, es decir todos los vehículos tienen la misma capacidad, con un total de K vehículos. También se supone que todos los puntos de demanda pueden ser satisfechos y que la matriz de costes satisface la desigualdad triangular:

$$c_{i,k} + c_{k,j} \geq c_{i,j} \qquad \forall i, j, k \in V \tag{1}$$

Para poder resolver este problema en la presente práctica, se nos pide emplear el algoritmo de ahorros Clark-Wright, el cual consta de los siguientes pasos:

1. Calcular la matriz de ahorros (*savings*): Para cada arco i y nodo j calculamos el ahorro que supondría incluir un arco (i, j) en la ruta de la solución, el cual se calcula mediante la siguiente fórmula:

$$s(i, j) = c_{1,i} + c_{j,1} - d_{i,j} \tag{2}$$

2. Ordenamos los ahorros de mayor a menor.
3. Creamos una solución inicial: La solución inicial consta de un total de $n-1$ rutas conectando el vértice 1 (depósito) con cada vértice cliente.
4. Se recorre esta lista uniendo las rutas siempre que sea posible, es decir, siempre que cumplan las restricciones, como por ejemplo la capacidad de los vehículos.

2. Desarrollo

Para esta práctica se nos pide resolver una serie de problemas con distancias Euclídeas mediante el algoritmo Clarke-Wright y comparar la solución obtenida con la de los algoritmos exactos de la práctica anterior.

Los problemas en cuestión se encuentran en los ficheros `data/E021-04m.dat`, `data/E026-08m.dat`, `data/E051-05e.dat` y `data/E076-10e.dat`. Todos estos problemas se resolvieron mediante *Xpress Mosel* usando el modelo de flujo de redes en los ficheros `cvrp_clarke_wright_e021.mos`, `cvrp_clarke_wright_e026.mos`, `cvrp_clarke_wright_e051.mos` y `cvrp_clarke_wright_e076.mos` (el nombre indica el fichero de datos empleado).

Para la implementación de este problema lo primero que se tuvo que calcular, una vez cargados los datos, fue la matriz de costes a partir de las coordenadas empleando la distancia Euclídea redondeada al entero más cercano.

A continuación se implementó el algoritmo Clarke-Wright. En mi caso a la hora de ordenar los ahorros en el paso 2 del apartado anterior, creé una lista llamada `savings_list` con los indices i y j de los vértices que uniría el posible arco. Para obtener el ahorro solo se tiene que indexar con estos índices en la matriz de ahorros `savings`. Por otro lado, para almacenar las diferentes rutas creé un total de $n-1$ matrices, una por cada ruta inicial, con los arcos de cada ruta. Además utilicé un vector llamado `rutas_activas` para poder saber qué ruta ya había sido unida o descartada así como otro vector llamado `rutas_demandas` con la demanda total de cada ruta. Así para implementar el último paso del algoritmo, solo se tendría que buscar 2 rutas activas, las cuales una empezara con un arco $(1, j)$, llamada `ruta_i`, y otra acabara con un arco $(i, 1)$, llamada `ruta_j`. Si estas rutas existen, `ruta_i` $\neq$ `ruta_j` y además la demanda de `ruta_i` $+$ `ruta_j` $\leq$ `capacidad` entonces se unirían las rutas.

3. Resultados

Los resultados obtenidos a los anteriores problemas, junto con los gráficos IVE que representan los caminos de las soluciones (obtenidos a partir de las coordenadas de los ficheros de datos), son los siguientes:

Fichero de datos	E021-04m.dat	E026-08m.dat	E051-05e.dat	E076-10e.dat
N^o de vértices	21	26	51	76
Solución	391	626	582	889
N^o de vehículos	5	9	6	11
Valores de x (arcos)	(1,2) (1,3) (1,5) (1,14) (1,21) (2,7) (3,6) (4,10) (5,16) (6,1) (7,13) (8,1) (9,1) (10,8) (11,19) (12,20) (13,18) (14,15) (15,9) (16,4) (17,1) (18,1) (19,17) (20,11) (21,12)	(1,3) (1,4) (1,7) (1,8) (1,13) (1,16) (1,18) (1,21) (1,23) (2,24) (3,5) (4,1) (5,26) (6,1) (7,15) (8,12) (9,10) (10,1) (11,1) (12,9) (13,1) (14,17) (15,6) (16,20) (17,1) (18,2) (19,25) (20,1) (21,22) (22,14) (23,19) (24,1) (25,1) (26,11)	(1,2) (1,16) (1,18) (1,28) (1,33) (1,48) (2,23) (3,17) (4,29) (5,19) (6,13) (7,1) (8,44) (9,49) (10,11) (11,50) (12,3) (13,47) (14,26) (15,1) (16,46) (17,39) (18,38) (19,1) (20,41) (21,36) (22,30) (23,21) (24,7) (25,1) (26,15) (27,8) (28,9) (29,32) (30,51) (31,35) (32,27) (33,12) (34,40) (35,22) (36,37) (37,4) (38,45) (39,6) (40,31) (41,42) (42,14) (43,20) (44,25) (45,43) (46,34) (47,1) (48,5) (49,24) (50,1) (51,10)	(1,4) (1,7) (1,12) (1,18) (1,24) (1,48) (1,52) (1,53) (1,69) (1,73) (1,76) (2,74) (3,31) (4,45) (5,35) (6,30) (7,1) (8,27) (9,1) (10,40) (11,32) (12,66) (13,59) (14,55) (15,1) (16,58) (17,64) (18,13) (19,51) (20,36) (21,38) (22,29) (23,62) (24,57) (25,50) (26,56) (27,1) (28,16) (29,63) (30,1) (31,46) (32,26) (33,10) (34,3) (35,47) (36,9) (37,70) (38,6) (39,54) (40,41) (41,1) (42,43) (43,65) (44,42) (45,33) (46,1) (47,68) (48,37) (49,22) (50,1) (51,25) (52,17) (53,28) (54,8) (55,20) (56,19) (57,44) (58,14) (59,39) (60,15) (61,71) (62,1) (63,2) (64,34) (65,23) (66,67) (67,60) (68,1) (69,75) (70,72) (71,21) (72,61) (73,11) (74,1) (75,49) (76,5)

Cuadro 1: Resultados obtenidos para los ficheros *e0*

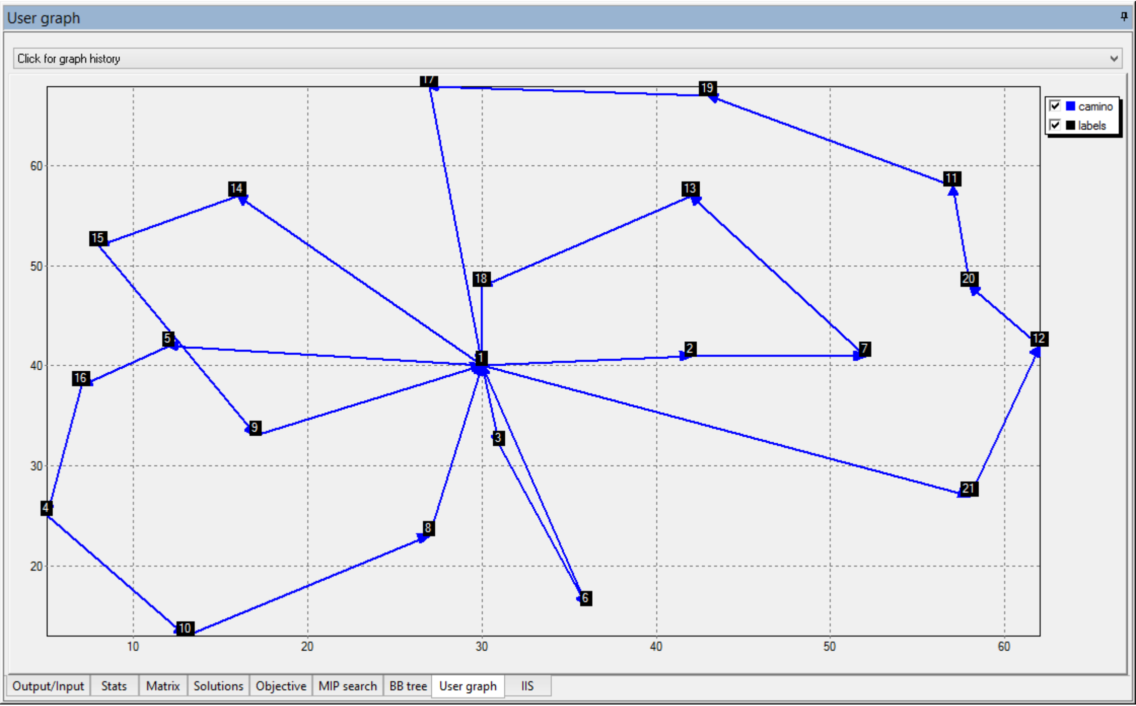

Figura 1: Solución del problema `E021-04m.dat`

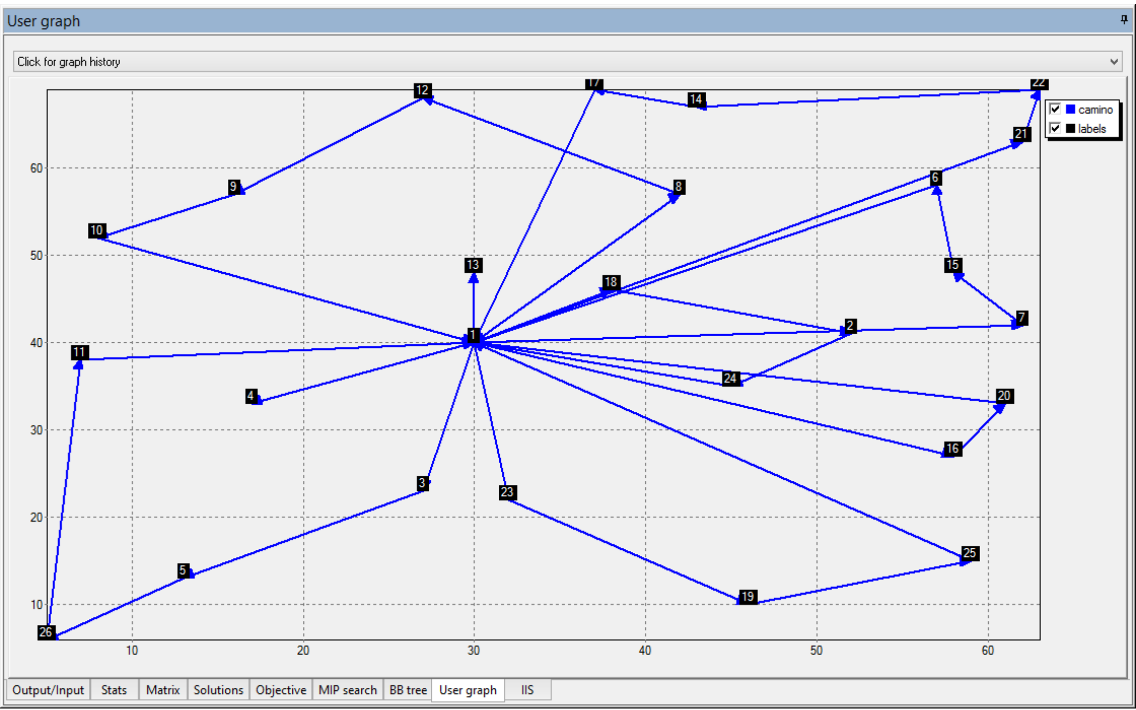

Figura 2: Solución del problema `E026-08m.dat`

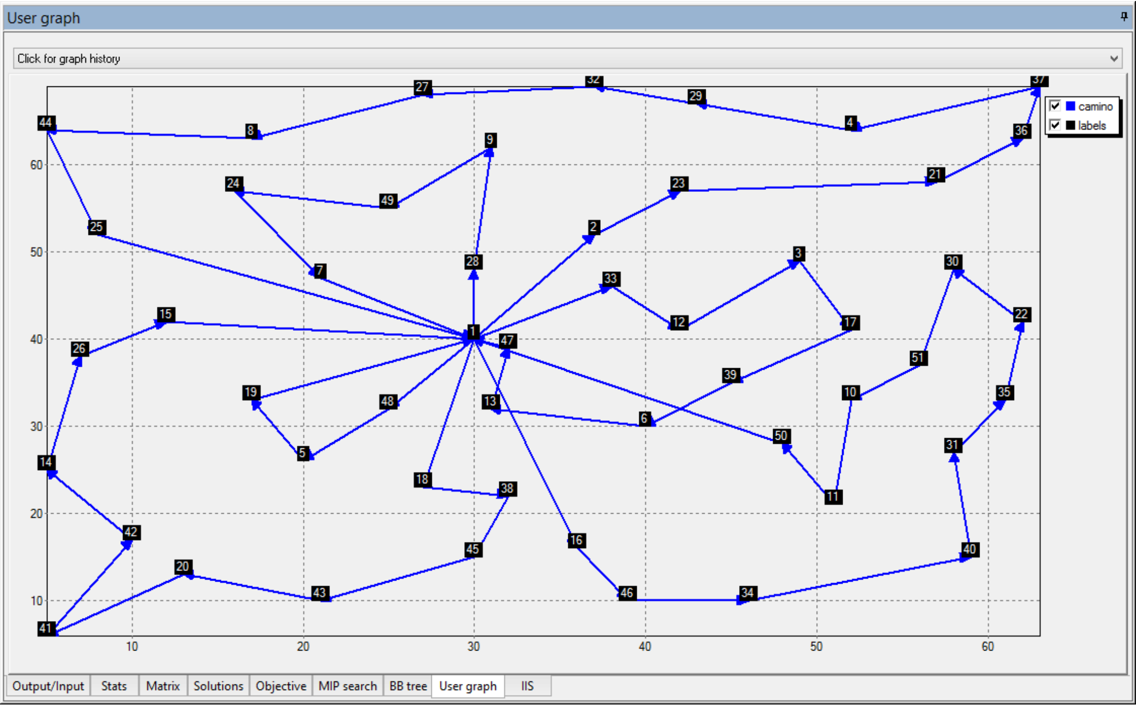

Figura 3: Solución del problema `E051-05e.dat`

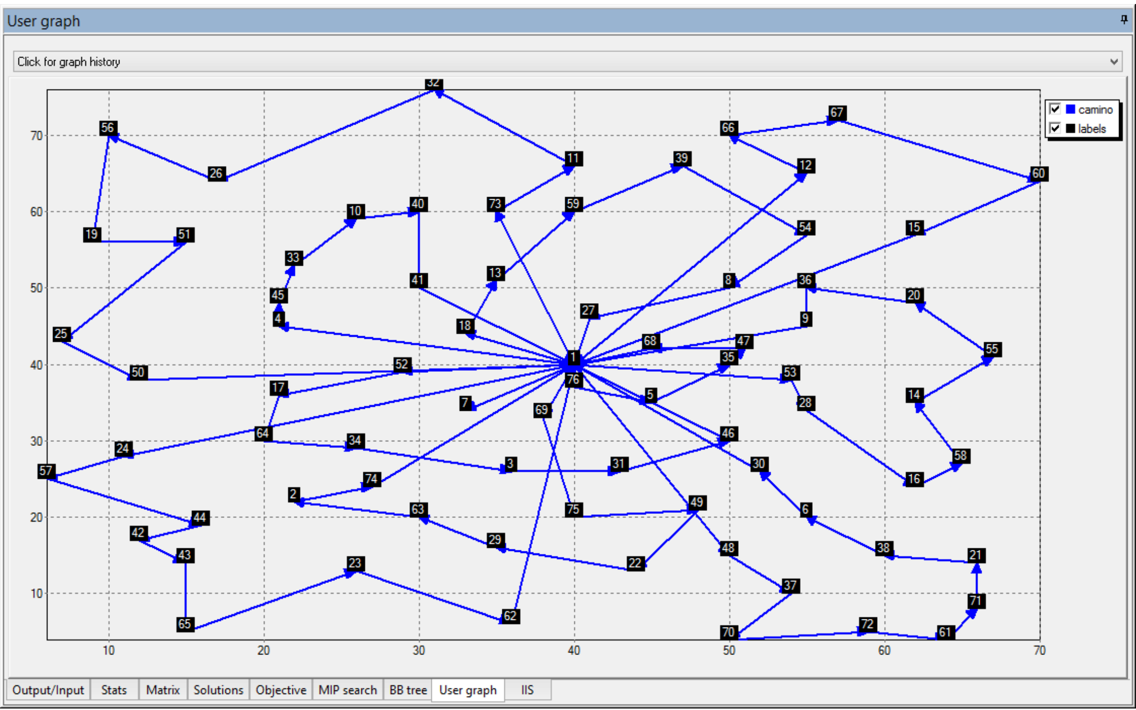

Figura 4: Solución del problema `E076-10e.dat`

Por último, en la siguiente tabla se pueden comparar los resultados obtenido con el algoritmo exacto (modelo de flujo de redes) de la práctica 15 con los de la práctica actual:

Fichero de datos		`E021-04m`	`E026-08m`	`E051-05e`	`E076-10e`
Modelo de flujo de redes	Nº de vértices	21	26	51	76
	Solución	358	606	584	918
	Nº de vehículos	5	9	7	12
Algoritmo Clarke-Wright	Nº de vértices	21	26	51	76
	Solución	391	626	582	889
	Nº de vehículos	5	9	6	11

Cuadro 2: Comparación de resultados para los ficheros *e0*

Como se puede observar, el algoritmo exacto da mejores resultados para los dos primeros ficheros de datos, mientras que el algoritmo Clarke-Wright emplea menos vehículos y un menor coste para los dos últimos.