Práctica 2

Yimin Pan

January 2020

1 Introducción

En esta práctica se nos presenta un problema de vehicle routing, dónde cada alumno tiene asignado un grafo dirigido diferente.
En los problemas de VRP tenemos un nodo que hace de `Depot`, y un conjunto de nodos que son los clientes. Cada cliente tiene una demanda de mercancias, y desde el `Depot` han de salir k vehículos con una capacidad de transporte limitado para llevar las mercancias a todos los clientes. De este modo, el problema se transforma en diseñar k rutas (tours) que salen desde el Depot y han de pasar exactamente una vez por cada node cliente, teniendo en cuenta la restricción de capacidad.
El siguiente es el grafo que tengo asignado, donde por la paridad de DNI, el nodo A es un cliente mientras que el B el Depot.

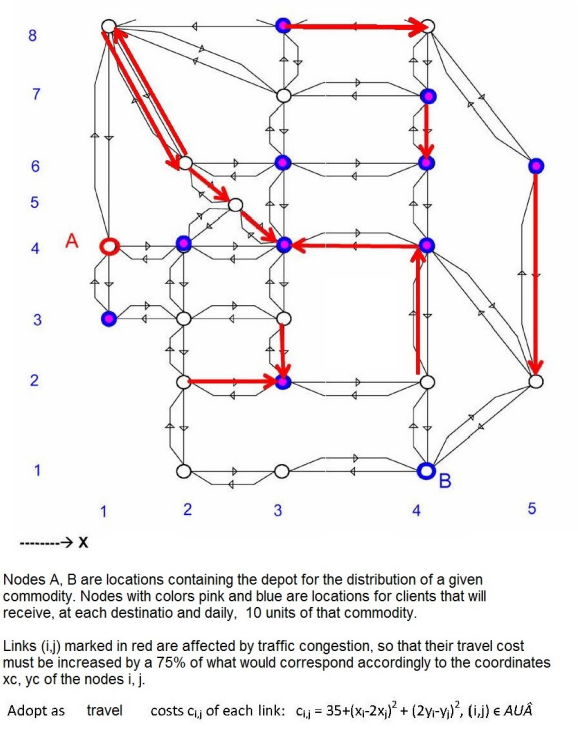

Figure 1: Gráfo

Como se puede observar, no todos los arcos tiene un coste igual, sino que se calcula a partir de la fórmula

de la imagen, cosa que tenemos que tener en cuenta a la hora de diseñr las rutas.
Dicho esto, podemos plantear el problema en programación lineal con la formulación MTZ, teniendo los costes mínimos para cada pareja de nodos c_{ij} previamente calculados con el algoritmo Floyd-Warshall:

$$
\begin{array}{lll}
Min_{x,\,u} & \sum_{(i,j)\in A} c_{i,j}x_{i,j} & \\
 & \sum_{i\in N} x_{i,j} = 1, & j \in N \\
 & \sum_{j\in N} x_{i,j} = 1, & i \in N \\
 & \sum_{j\in N} x_{0,j} = K, & \\
 & \sum_{j\in N} x_{i,0} = K, & \\
 & u_j \geq u_i + a_j - C(1 - x_{i,j}) & i \neq 0,\, j \neq 0 \\
 & a_i \leq u_i \leq C & i \in N \\
 & x_{i,j} \in \{0, 1\} &
\end{array}
$$

Figure 2: Formulación mtz

El modelo que usaremos en AMPL se basará básicamente en esta formulación, añadiendo el Floyd-Warshall. Las variables de decisión son $x_{ij} \in (i, j) \in Arcs$, donde `Arcs` son todas las aristas (i,j) tal se define sobre $(Client \cup Depot) \times (Client \cup Depot)$. Podemos obviar el resto de nodos y quedar con estos que son los que en realidad nos interesa, ya que tenemos calculado el coste de la ruta para ir de un lugar a otro por Floyd-Warshall. De este modo, $x_{ij} \in 0, 1$ es un binario que nos indicará si decidimos incluir la arista (i,j) en el tour o no.

2 Experimentos

Realizaremos diferentes experimentos modificando el valor de k, es decir el número de vehículos (o rutas). Para $k = 3, 4, 5$ como se pide en el enunciado, el resultado deberá ser diferente.
Los otros parámetros ya vienen dados del enunciado, y son fijos para el experimento.

- Los nodos y arcos son los que se muestran en la imagen.
- El coste de transportar por un arco se calcula apartir de la fórmula.
- El coste mínimo de un nodo a otro se calcula por Floy-Warshall.
- Todos los clientes tienen la misma demanda, 10.
- Todos los vehículos tienen una capacidad máxima C = 40.

Para facilitar la comprensión del resultado, representaremos las rutas encontradas por el solver sobre este grafo de clientes-depot.

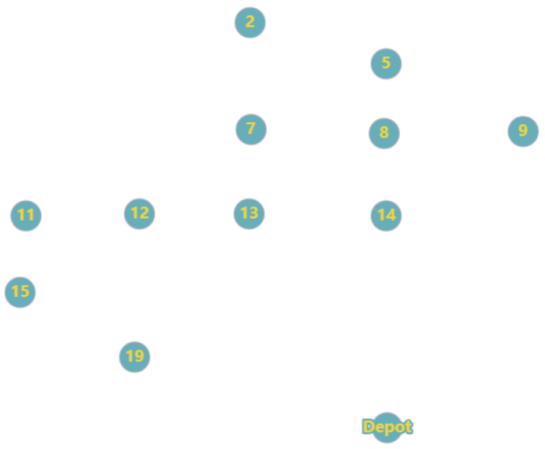

Figure 3: Nodos Cliente-Depot

2.1 k = 3

función objetivo: 1838.25.

Figure 4: vrp k=3

2.2 k = 4

función objetivo: 2124.25.

Figure 5: vrp k=4

2.3 k = 5

función objetivo: 2390.25.

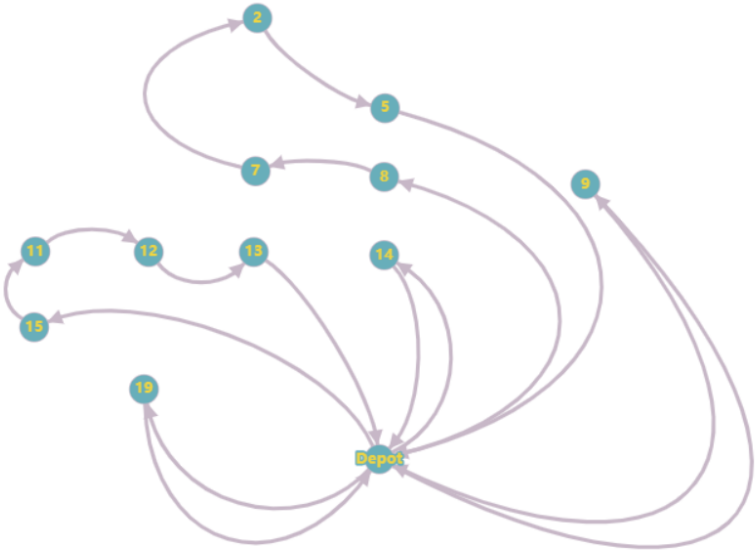

Figure 6: vrp k=5

3 Conclusión

Durante los experimentos vemos una tendencia muy clara, a medida que aumentamos el número de vehículos, la función objetivo (coste total) es cada vez mayor. Pensamos que esto se debe a que cada vez que añadimos un vehículo más, forzosamente habrá que crear un tour para éste. Por lo que en compración, mínimo habrá que introducir una arista más, con esto nos referimos a pasar por esa arista ya que no creamos ninguno inexistente en el grafo original, y solamente estamos decidiendo por dónde pasar los vehículos.

Dado que los costes de pasar de un nodo a otro son calculados por Floyd-Warshall, son mínimos. Esto hace que los resultados entre diferentes k son parecidos, al intentar minimizar el coste total, el solver preferentemente buscará pasar por las aristas de coste pequeño, por ejemplo el `2-5` que aparece en todos los experimentos.
También vemos una situación muy habitual en los resultados, cuando aumentamos k, hay tours que solamente consta del depot y un nodo. Esto es que hemos arrancado una arista del tour de k-1, y en k le hemos añadido una arista más para que forme un tour indenpendiente extra con el depot, para satisfacer la restricción de tener k tours.

En conclusión, con menos número de tours (menor k), menos coste total obtendremos. En nuestro caso k como mínimo ha de ser 3, ya que sinó estamos violando las restricciones de capacidad de transporte, recordando que C = 40 y cada cliente pide 10, teniendo 11 clientes 11*10/40 por lo mínimo hemos de usar 3 vehículos.