Optimización II

Mathias Felipe Badilla Salamanca

Octubre 2020

1. Modelamiento matemático

A continuación se encuentra el modelo matemático correspondiente para la tarea 1 del curso de optimización II. El modelo cuenta con 3 variables binarias y una entera, las cuales son las siguientes.

$$x_{i,j} = \begin{cases} 1; & \text{Si el arco } (i,j) \text{ es parte de la ruta} \\ 0; & \text{En otro caso} \end{cases}$$

$$y_{i,j} = \begin{cases} 1; & \text{Si el cliente } i \text{ es abastecido por la instalación } j \\ 0; & \text{En otro caso} \end{cases}$$

$$z_i = \begin{cases} 1; & \text{Si el vertice } i \text{ pertenece a la ruta} \\ 0; & \text{En otro caso} \end{cases}$$

$$G_{i,j} = \begin{cases} 0, 1, ..., n; & \text{Orden en que se atraviesan los arcos} \end{cases}$$

Presentadas las variables, nos encontramos con los siguientes parámetros.

$$\begin{aligned} I &= \text{Conjunto de instalaciones, incluyendo el deposito} \\ C &= \text{Conjunto de los clientes} \\ t_{ij} &= \text{Distancia asociada a ir desde } i \text{ a } j \\ n &= \text{Tamaño del conjunto } I \end{aligned}$$

Revisando ahora la función objetivo y las restricciones.

$$min\ z = \sum_{i \in I} \sum_{j \in I} t_{ij}\ x_{ij} \tag{1}$$

$$s.a: \quad \sum_{i \in I} \sum_{j \in I} y_{ij} \geq p \tag{2}$$

$$\sum_{j \in C} y_{ij} \leq 1; \quad \forall i \in C \tag{3}$$

$$\sum_{i \in C} y_{ij} \leq z_j; \quad \forall j \in I \tag{4}$$

$$\sum_{i \in I} x_{ij} = z_j; \quad \forall j \in I \tag{5}$$

$$\sum_{j \in I} x_{ij} = z_i; \quad \forall i \in I \tag{6}$$

$$\sum_{j \in I} G_{0j} = 0 \tag{7}$$

$$\sum_{j \in I} G_{ij} - G_{ji} = z_i\ ; \quad \forall i \in I - \{0\} \tag{8}$$

$$0 \leq G_{ij} \leq n x_{ij}: \quad \forall (i,j) \in I \tag{9}$$

La función objetivo (1) minimiza el costo de la ruta. Las restricciones (2) aseguran que el beneficio asociado a la ruta, sea mayor que el P solicitado. Las restricciones (3) limitan a que los clientes puedan tener maximo un abastecedor. Las restricciones (4) establece que cuando un vertice z_j pertenece a la ruta, considera abastecidos los clientes conectados a él. Las restricciones (5) y (6) limitan la posibilidad de una entrada y una salida como máximo para cada instalación. Las restricciones (7), (8) y (9) fueron sacadas de la presentación de la clase 6, sobre el problema del vendedor viajante. Fueron debidamente adaptadas para el caso tal como incluir desigualdades o cambiar constantes por parametros evitando así tener soluciones que incluyan todos los nodos.

2. Resultados computacionales

El modelo propuesto es resuelto con **CPLEX 12.10.0** y es implementado en Python usando la biblioteca **docplex**.Las pruebas se realizaron en un Intel(R) Core(TM) i7-8550U CON 1.8Ghz y 8GB de RAM (Ejecutado en un hilo), en el sistema operativo Windows 10 Home Single Language 64 bits. El codigo de la implementación se encuentra disponible y con los ejemplos desplegados en Link Repositorio. En la tabla siguiente se presentan los resultados de las 5 instancias entregadas mediante un archivo de texto plano.

Resultados del modelo					
Instancia	Instalaciones	Clientes	Arco	Costo optimo	Tiempo(Seg.)
3.txt	15	35	240	172.355	0.22
6.txt	25	25	650	38.487	0.39
13.txt	21	30	462	3910.037	2.94
16.txt	26	25	702	2958.779	15.72
26.txt	23	48	552	147.891	14.31

Tabla 1

3. Conclusiones

Podemos notar las variaciones en el tiempo que se van dando en los distintos casos. A priori podemos notar una relación entre la cantidad de instalaciones y el tiempo, pero tambien notamos como la cantidad de clientes o arcos, no son realmente influyentes a simple vista en en el tiempo. Finalmente como conclusion personal, creo que la complejidad operacional de un problema, tiene relacion con las variables, sí. Pero no creo que sea el factor determinante final.