Aplicaciones Computacionales en Negocios: Trabajo Práctico 2

Agustina Glusman, Nadia Guadalupe Molina, Jazmín Sneider

Noviembre/Diciembre 2024

Introducción:

En este informe, detallaremos cómo resolvimos el problema de optimización relacionado con la conexión de 56 oficinas distribuidas en el micro y macrocentro a través de un máximo de 10 centrales operativas. El objetivo es minimizar los costos totales, que incluyen tanto la apertura de las centrales como el costo de los cables de conexión, respetando restricciones de capacidad y demanda. Para ello, planteamos y resolvimos un modelo de programación lineal entera que nos permitió asignar eficientemente las oficinas a las centrales, garantizando que se cumplieran todas las restricciones operativas y económicas. En adición, queremos mencionar que para este trabajo hicimos un solver que usa SCIP en Python ya que no pudimos instalar el solver directamente en nuestros sistemas.

1 Ejercicio 1: Planteo del modelo

1.1 Cargado de datos:

Primeramente, leímos los datos de los archivos *distancias.txt* y *oficinas.txt*. El primero contenía las distancias entre cada oficina y las 10 centrales operativas por lo que decidimos almacenar los datos en una matriz donde las filas representaban las oficinas y cada columna una central operativa. El segundo archivo tenía la demanda de cada oficina y también las coordenadas de ubicación de cada una de ellas. Sólo nos interesaba el primer dato (porque las distancias ya las habíamos guardado) por lo que lo guardamos en una lista. Con los datos almacenados de esta manera, ya podíamos empezar a pensar en cómo plantear el problema para que un algoritmo de programación lineal pudiera resolverlo.

1.2 Establecer variables, restricciones y función objetivo:

1.2.1 Variables:

Creamos las siguientes variables donde:

- **C:** Son las centrales del modelo
- **O:** Son las oficinas del modelo
- **d:** Es una matriz con las distancias (en metros) de cada oficina a cada central
- **D:** Demandas de las oficinas

$$d_{ij} : \text{Distancia (en metros) entre la oficina } i \text{ y la central } j.$$

Luego definimos las siguientes variables binarias:

$$x_{ij} = \begin{cases} 1 & \text{si la oficina } i \text{ está atendida por la central } j, \\ 0 & \text{en caso contrario.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{si la central } j \text{ está activada,} \\ 0 & \text{en caso contrario.} \end{cases}$$

$$i \in O, \quad j \in C$$

1.2.2 Función Objetivo:

La función objetivo del modelo tiene como objetivo minimizar los costos totales asociados con la conexión de las oficinas a las centrales operativas. Esta función se compone de dos términos:

$$\text{Min} \sum_{i \in O} \sum_{j \in C} 0.017\, d_{ij}\, x_{ij} + \sum_{j \in C} 5700\, y_j$$

El primer término de la función objetivo es:

$$\sum_{i \in O} \sum_{j \in C} 0.017\, d_{ij}\, x_{ij}$$

Este término representa el costo asociado con la conexión de las oficinas a las centrales operativas. Aquí, d_{ij} es la distancia en metros entre la oficina i y la central j, y x_{ij} es una variable binaria que indica si la oficina i está conectada a la central j (con valor 1) o no (con valor 0). El factor 0.017 es el costo por cada metro de cable que se necesita para realizar la conexión. Al multiplicar por x_{ij}, solo sumamos el costo cuando realmente se conecta una oficina a una central, lo que asegura que solo se cuente el costo de las conexiones activas. El segundo término de la función objetivo es:

$$\sum_{j \in C} 5700\, y_j$$

Este término representa el costo de apertura de las centrales operativas. Aquí, y_j es una variable binaria que indica si la central j está abierta (con valor 1) o no (con valor 0). El valor 5700 es el costo de abrir una central operativa.

Al multiplicar por y_j, se asegura que solo se sume el costo de apertura para las centrales que realmente se abran. Por lo tanto, la función objetivo busca minimizar el costo total, que incluye tanto el costo de las conexiones de cable como el costo de apertura de las centrales, mientras se cumplen las restricciones del modelo que detallaremos a continuación.

1.2.3 Restricciones:

El modelo de programación lineal está sujeto a las siguientes restricciones, las cuales garantizan el cumplimiento de las condiciones operativas y logísticas de la empresa.

1. **Restricción 1: Cada oficina debe ser atendida por exactamente una central.** La primera restricción asegura que cada oficina debe estar conectada y atendida por una única central operativa. Esta restricción se expresa de la siguiente manera:

   $$\sum_{j \in C} x_{ij} = 1 \quad \forall i \in O$$

   Aquí, x_{ij} es una variable binaria que indica si la oficina i está conectada a la central j. La restricción garantiza que la suma de las variables x_{ij} para cada oficina i sea igual a 1, lo que significa que cada oficina debe ser atendida por exactamente una central j.

2. **Restricción 2: Una central no puede atender más de su capacidad máxima.** La segunda restricción garantiza que ninguna central operativa se sobrecargue. Es decir, la suma de las demandas de las oficinas conectadas a una central no debe superar la capacidad máxima de 15,000 operaciones por hora. Esta restricción se expresa como:

   $$\sum_{i \in O} x_{ij} D_i \leq 15000 \quad \forall j \in C$$

   Aquí, D_i es la demanda de operaciones por hora de la oficina i, y x_{ij} es la variable binaria que indica si la oficina i está conectada a la central j. La restricción asegura que la suma de las demandas de las oficinas asignadas a una central no supere los 15,000 operaciones por hora.

3. **Restricción 3: Si una oficina es atendida por una central, entonces esa central debe estar activada.** La tercera restricción asegura que si una oficina está conectada a una central, esa central debe estar activada. Esta restricción se expresa de la siguiente manera:

   $$x_{ij} \leq y_j \quad \forall i \in O, \forall j \in C$$

   Aquí, y_j es una variable binaria que indica si la central j está abierta (con valor 1) o no (con valor 0). La restricción establece que si una oficina i está conectada a una central j (es decir, si $x_{ij} = 1$), entonces la central j debe estar activada (es decir, $y_j = 1$).

1.3 Resultados:

El solver dió como costo mínimo **$34549.843** con las centrales 1, 2, 3, 5, 7 y 9 activadas. La siguiente tabla muestra qué oficinas están siendo atendidas por cada una de las centrales operativas, según el resultado del solver.

Central Operativa	**Oficinas Atendidas**
Central 1	15, 46, 47, 48, 49, 50, 51, 52, 53, 54
Central 2	1, 2, 3, 14, 16, 29, 30, 42, 43, 44
Central 3	10, 11, 18, 19, 21, 31, 32, 33
Central 5	8, 9, 17, 20, 34, 35, 36, 37, 40, 41
Central 7	4, 5, 6, 7, 26, 28, 38, 45, 55, 56
Central 9	12, 13, 22, 23, 24, 25, 27, 39

Table 1: Oficinas atendidas por cada central operativa

2 Ejercicio 2: Agregar una restricción:

Para este inciso se nos pidió agregar la restricción extra de que una central no podía atender a más de 10 oficinas, por lo que se la pasamos de esta manera al solver.

$$\sum_{i \in O} x_{ij} \leq 10 \quad \forall j \in C$$

La solución proporcionada por el solver no modificó a la del ejercicio anterior. Esto seguramente se debe a que en la solución original ninguna central atendía a más de 10 oficinas, por lo que la restricción no la afectó.

3 Ejercicio 3: Mínima capacidad para la cual el problema es factible

3.1 Planteo de un nuevo modelo:

Para este propósito, decidimos reformular el problema como otro problema de programación lineal, donde minimizamos una variable *capacidad* que está relacionada con la restricción de capacidad máxima del problema original. De este modo, creamos un nuevo modelo con una nueva función objetivo y mantuvimos las mismas restricciones que el problema original, así como sus variables de decisión. La nueva formulación tiene la siguiente función objetivo:

$$\text{Min} \quad \text{capacidad}$$

Las restricciones de este nuevo modelo son las mismas que en el modelo original, con la diferencia de que la segunda restricción ahora depende de la variable *capacidad*, reemplazando el valor fijo de 15000 por la variable *capacidad*.

$$\sum_{j \in C} x_{ij} D_i \leq \text{capacidad} \quad \forall j \in C$$

3.2 Resultados:

Al resolver el modelo con el solver, se obtuvo el siguiente valor óptimo de la función objetivo:

$$\text{Valor óptimo de la función objetivo} = 8800.0$$

Este valor representa la mínima capacidad necesaria para que el problema sea factible, según la formulación planteada.

4 Ejercicio 4: Generación de nuevas instancias

Para evaluar el comportamiento del modelo con diferentes configuraciones, diseñamos la función **generar_matriz_distancias**. Esta función permite generar instancias a partir de la cantidad de oficinas y centrales especificadas, además de un rango de distancias posibles entre ellas. El resultado es una matriz donde cada fila representa una oficina y cada columna una central, rellenando las distancias de forma aleatoria con *np.random.randint* dentro del rango definido. Después usamos esta función para crear 7 instancias con diferentes combinaciones de oficinas y centrales. La información de cada instancia fue almacenada en archivos .txt, siguiendo el formato proporcionado por la cátedra.

4.1 50 oficinas, 10 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2095.68
- Las demandas promedio son: 1562.84
- Valor óptimo de la función objetivo: 34778.305999999524
- **El modelo tardó 0.7453 segundos en ejecutarse.**

Esta primera instancia era muy parecida a la proporcionada por la cátedra por lo que no nos ofreció tanta información con respecto a la performance.

4.2 100 oficinas, 10 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2136.12
- Las demandas promedio son: 1611.49

- No se encontró una solución óptima
- **El modelo tardó 0.0490 segundos en ejecutarse.**

Esta instancia nos dió una buena pista: al incrementar significativamente la cantidad de oficinas pero mientras mantenemos bajo el número de centrales, el tiempo de ejecución se mantiene aún en niveles aceptables, sin alcanzar valores excesivos.

4.3 150 oficinas, 18 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2124.485925925926
- Las demandas promedio son: 1522.1
- Valor óptimo de la función objetivo: 92238.7
- **El modelo tardó 45.9897 segundos en ejecutarse.**

Ya en esta instancia con solamente 8 centrales más, podemos ver que sube muchísimo el tiempo de resolución del modelo, llegando a más de 45 segundos. Igualmente, sigue siendo un tiempo que vale la pena esperar.

4.4 150 oficinas, 18 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2082.495185185185
- Las demandas promedio son: 1705.0666666666666
- Valor óptimo de la función objetivo: 103646.724
- **El modelo tardó 3.7541 segundos en ejecutarse.**

En este caso, hicimos otro modelo con la misma cantidad de centrales y oficinas que el modelo pasado. Hay una diferencia amplia entre 45.9897 y 3.7541 segundos, pero sospechamos que el hecho de que se generen distancias aleatorias puede afectar el tiempo de resolución según como éstas estén distribuidas.

4.5 75 oficinas, 13 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2117.2851282051283
- Las demandas promedio son: 1574.64
- Valor óptimo de la función objetivo: 46315.785

- **El modelo tardó 15.1125 segundos en ejecutarse.**

En este caso, también probamos con bastantes oficinas y 13 centrales. Ya podemos ver que pasándonos un poco de 10 centrales empieza a aumentar el tiempo de resolución.

4.6 50 oficinas, 22 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2062.44454545
- Las demandas promedio son: 1575.16
- Modelo con 50 oficinas y 22 centrales
- Valor óptimo de la función objetivo: 23297.369
- **El modelo tardó 6287.9814 segundos en ejecutarse (104 minutos).**

Como era de esperar, con 22 centrales (y aunque tengamos pocas oficinas) podemos ver que el tiempo de resolución es muy alto.

4.7 50 oficinas, 25 centrales

Al ejecutar este modelo nos encontramos con la siguiente salida:

- Las distancias promedio son: 2080.4696
- Las demandas promedio son: 1668.28
- **El modelo tardó en correr más de 5 horas, asi que lo interrumpimos**

En este último caso, con 25 centrales, como el tiempo de resolución era demasiado alto (pasaba las 5 horas) decidimos cortar la ejecución del modelo.

4.8 Conclusiones:

Basándonos en los resultados que obtuvimos con estas nuevas instancias, podemos concluir que el tiempo de resolución del modelo está principalmente influenciado por la cantidad de centrales, más que por la cantidad de oficinas. Aunque la variación en las distancias aleatorias puede introducir diferencias en los tiempos, a partir de 20 centrales los tiempos de ejecución tienden a volverse inaceptables, comprometiendo la practicidad del modelo.