Heurística y Optimización

Práctica: Programación Lineal

Grado de Ingeniería en Informática. Curso 2022-2023. Departamento de Informática

Jose Antonio Barrientos 100451290 , Alejandro Gonzalez Nuñez 100429135

Introducción:

En esta práctica vamos a modelar problemas de programación lineal y resolverlos con dos tipos de herramientas: hojas de cálculo y algoritmos de resolución usando técnicas de modelización (GLPK).

Para ello resolveremos 2 problemas relacionados con una tarea de programación lineal. El problema 1 consiste en un ejercicio de transporte en donde se debe encontrar una forma de desplazar *e-scooters* de localizaciones sin demanda a otras localizaciones con demanda, minimizando los gastos que generan dichos desplazamientos. Resolver esté problema no es un desafío sencillo, para ello haremos uso en la primera parte, de Libre office. Primero se deberá realizar un modelado que contemple todos los puntos establecidos en el enunciado, para luego implementar dicho modelo en una hoja calc.

Siguiendo el mismo ritmo de ideas, el problema dos tiene las características de un ejercicio de asignación, en el cual debemos decidir en función de minimizar los costes, que *e-scooters* le corresponderá a un freelancer, teniendo en cuenta una serie de restricciones dadas por el enunciado. Para la resolución de dicho problema nos aprovecharemos de la librería GNU MathProg (GLPK), al igual que en el apartado anterior se deberá llegar a un modelado a partir de el enunciado con el que posteriormente podamos implementar este modelo en GLPK.

Para finalizar, se darán una serie de justificaciones y explicaciones de las implementaciones que hemos decidido hacer junto con un análisis de los resultados que hemos obtenido.

Parte 1: Modelo básico en Calc:

Modelado:

Variables de decisión: Xij -> Número de autobuses salen de i y llegan a j

Tabla para el autobús

Origen\Destino	P	S1	S2	S3	E
P	PP	PS1	PS2	PS3	PE
S1	S1P	S1S1	S1S2	S1S3	S1E
S2	S2P	S2S1	S2S2	S2S3	S2E
S3	S3P	S3S1	S3S2	S3S3	S3E
E	EP	ES1	ES2	ES3	EE

Variables de decisión: Nij -> Número de personas salen de i y llegan a j

Origen\Destino	P	S1	S2	S3	E
P	NPP	NPS1	NPS2	NPS3	NPE
S1	NS1P	NS1S1	NS1S2	NS1S3	NS1E
S2	NS2P	NS2S1	NS2S2	NS2S3	NS2E
S3	NS3P	NS3S1	NS3S2	NS3S3	NS3E
E	NEP	NES1	NES2	NES3	NEE

Costes (Cij): Cij -> coste para desplazar un autobús de i hasta j

Origen\Destino	P	S1	S2	S3	E
P	∞	8	10	10	∞
S1	∞	∞	3	7	6
S2	∞	3	∞	5	7
S3	∞	7	5	∞	4
E	∞	∞	∞	∞	∞

Función objetivo:

$$\text{mín } z = ((\sum_{i=1}^{5} \sum_{j=1}^{5} CijXij * 5) + \sum_{k=1}^{5} Pj * 120))$$

Restricciones:

Restricción 1(El número de rutas que entra a una parada es como máximo 1):

$$\sum_{i=1}^{5} iS1 <= 1 \quad \sum_{i=1}^{5} iS2 <= 1 \quad \sum_{i=1}^{5} iS3 <= 1$$

Restricción 2(El número de rutas que sale de una parada es igual al número de rutas que entran):

$$\sum_{j=1}^{5} S1j = \sum_{i=1}^{5} iS1 \quad \sum_{j=1}^{5} S2j = \sum_{i=1}^{5} iS1 \quad \sum_{j=1}^{5} S3j = \sum_{i=1}^{5} iS1$$

Restricción 3(número de rutas menor a número de autobuses):

$$\sum_{j=1}^{5} Pj <= 3 \quad \sum_{j=1}^{5} iE <= 3$$

Restricción 4 (todas las rutas salen del parking y llegan al colegio):

$$\sum_{j=1}^{5} Pj = \sum_{i=1}^{5} iE$$

Restricción 5:

$$NS1J <= 20 * S1J \quad NS2j <= 20 * S2j \quad NS3j <= 20 * S3j$$

Siendo j, 5 posibles valores: P, S1, S2, S3, E

Restricción 6 ():

$$\sum_{i=1}^{5} NS1j = \sum_{i=1}^{5} NiS1 + 15 \quad \sum_{i=1}^{5} NS2j = \sum_{i=1}^{5} NiS2 + 5$$

$$\sum_{i=1}^{5} NS3j = \sum_{i=1}^{5} NiS2 + 10$$

Restricción 7:

$Xijk \in N$

Descripción del modelo:

Hay diversos puntos a destacar del modelado que nosotros hemos empleado, a su vez una serie de adaptaciones y modificaciones que hemos hecho para poder lograr que sea: sencillo, práctico y funcione como modelo para ser utilizado en otros problemas/prácticas. Es por esto que dividiremos en puntos cada una de las decisiones tomadas, para así poder argumentarlas y dar una explicación detallada.

Valores infinito en las matriz de costes:

Los valores infinito en nuestra matriz de coste indican que no hay un coste asociado para desplazar un *e-scooters* de la localización origen a la localización destino. Este caso se puede ver claramente cuando decimos que un scooter se desplazará de la localización 1 a la localización 1, no disponemos en nuestro enunciado cuál es el coste por producirse este desplazamiento sin embargo decidimos incluirlo ya que permite la resolución correcta del ejercicio y podría implementarse en un futuro otro valor distinto.

Variables de decisión como una matriz:

Para nuestras variables de decisión hemos hecho una matriz 4x4, esto va en relación con lo descrito con el punto anterior, de esta forma tenemos en cuenta que todos los posibles caminos que podrían existir en el grafo aun cuando no existen.

Libreoffice (Calc):

Para este problema, utilizaremos una hoja de cálculo de libreoffice, en ella haremos uso de buenas prácticas, es decir, usaremos suma producto para introducir los coeficientes de las variables y a su vez agregamos distintos colores a las celdas para poder facilitar su comprensión. Por último se resolverá usando la herramienta “solver” que viene incluida en libre office.

Parte 2: Modelo avanzado en GLPK

Modelado:

Variables de decisión: Xij -> Número de autobuses salen de i y llegan a j

Origen\Destino	P	S1	S2	S3	E
P	PP	PS1	PS2	PS3	PE
S1	S1P	S1S1	S1S2	S1S3	S1E
S2	S2P	S2S1	S2S2	S2S3	S2E
S3	S3P	S3S1	S3S2	S3S3	S3E
E	EP	ES1	ES2	ES3	EE

Variables de decisión: Nij -> Número de personas salen de i y llegan a j

Origen\Destino	P	S1	S2	S3	E
P	NPP	NPS1	NPS2	NPS3	NPE
S1	NS1P	NS1S1	NS1S2	NS1S3	NS1E
S2	NS2P	NS2S1	NS2S2	NS2S3	NS2E
S3	NS3P	NS3S1	NS3S2	NS3S3	NS3E
E	NEP	NES1	NES2	NES3	NEE

Variables de decisión: Aij ->Alumno i que se asigna a la parada j

Origen\Destino	A1	A2	A3	A4	A5	A6	A7	A8
S1	A1S1	A2S1	A3S1	A4S1	A5S1	A6S1	A7S1	A8S1
S2	A1S2	A2S2	A3S2	A4S2	A5S2	A6S2	A7S2	A8S2
S3	A1S3	A2S3	A3S3	A4S3	A5S3	A6S3	A7S3	A8S3

Costes (Cij): **Cij** -> coste para desplazar un autobús de i hasta j

Origen\Destino	P	S1	S2	S3	E
P	∞	8	10	10	∞
S1	∞	∞	3	7	6
S2	∞	3	∞	5	7
S3	∞	7	5	∞	4
E	∞	∞	∞	∞	∞

Función objetivo:

$$\text{mín } z = ((\sum_{i=1}^{5}\sum_{j=1}^{5} CijXij * 5) + \sum_{j=1}^{5} Pj * 120))$$

Restricciones:

Restricción 1(El número de rutas que entra a una parada es como máximo 1):

$$\sum_{i=1}^{5} iS1 <= 1 \quad \sum_{i=1}^{5} iS2 <= 1 \quad \sum_{i=1}^{5} iS3 <= 1$$

Restricción 2(El número de rutas que sale de una parada es igual al número de rutas que entran):

$$\sum_{j=1}^{5} S1j = \sum_{i=1}^{5} iS1 \quad \sum_{j=1}^{5} S2j = \sum_{i=1}^{5} iS1 \quad \sum_{j=1}^{5} S3j = \sum_{i=1}^{5} iS1$$

Restricción 3(número de rutas menor a número de autobuses):

$$\sum_{j=1}^{5} Pj <= 3$$

Restricción 4 (todas las rutas salen del parking y llegan al colegio):

$$\sum_{j=1}^{5} Pj = \sum_{i=1}^{5} iE$$

Restricción 5 (El flujo que sale de una parada no excede la capacidad máxima de el autobús si existe la ruta:

$$NS1J <= 4 * S1J \quad NS2j <= 4 * S2j \quad NS3j <= 4 * S3j$$

Siendo j, 5 posibles valores: P, S1, S2, S3, E

Restricción 6 (El flujo que llega a la escuela debe ser el total de estudiantes)

$$\sum_{i=1}^{5} \text{NiE} = 8$$

Restricción 7 (El flujo que sale de una parada es igual al que entra más el número de alumnos asignados a esa parada):

$$\sum_{i=1}^{5} NS1j = \sum_{i=1}^{5} NiS1 + \sum_{i=1}^{5} AiS1$$

$$\sum_{i=1}^{5} NS2j = \sum_{i=1}^{5} NiS2 + \sum_{i=1}^{5} AiS2$$

$$\sum_{i=1}^{5} NS3j = \sum_{i=1}^{5} NiS2 + \sum_{i=1}^{5} AiS3$$

Restricción 7 (Paradas prohibidas para los alumnos):
Estas son las que aplican para el problema sin embargo nosotros lo hemos generalizado en una matriz de alumnos por paradas donde se indican cada una de las paradas.

$$A1S2 = 0 \quad A1S3 = 0 \quad A2S2 = 0 \quad A2S3 = 0$$

$$A3S2 = 0 \quad A3S3 = 0 \quad A4S1 = 0 \quad A5S1 = 0$$

$$A6S1 = 0 \quad A6S2 = 0 \quad A7S1 = 0 \quad A7S2 = 0$$

$$A8S1 = 0 \quad A8S2 = 0$$

Restricción 8 (Si son hermanos deben ir a la misma parada :
Para esto hemos creado un set de Hermanos donde si son hermanos forzamos a que la parada asignada sea la misma.

$$A4j = \text{A5j}$$

Siendo j, 3 posibles valores: S1, S2, S3

Restricción

$Xij \in N$

Descripción del modelo:

Hay diversos puntos a destacar del modelado que nosotros hemos empleado, a su vez una serie de adaptaciones y modificaciones que hemos hecho para poder lograr que sea: sencillo, práctico y funcione como “template” para ser utilizado en otros problemas/prácticas. Es por esto que dividiremos en puntos cada una de las decisiones tomadas, para así poder argumentarlas y dar una explicación detallada.

Elección de las variables de decisión:

Para este problema, hemos escogido 3 sets de variables de decisión: rutas, flujo y paradas asignadas a los alumnos, se pueden observar en las tablas que hemos enseñado previamente.

Valores infinito en la matriz de costes:

Esto es igual que lo explicado anteriormente, para evitar que el bus tome rutas que no existan o que por ejemplo al llegar al colegio decida tomar otra ruta hemos colocado valores muy altos en la matriz de costes.

MathProg:

Debido a que no es posible incluir en esta parte código, consideramos que sería bueno que se analizará el .dat de nuestra entrega, ya que se puede ver claramente cuales son los sets y parámetros utilizados. Hemos usado todos los datos como template teniendo que hacer los siguientes parámetros:
número de autobuses, capacidad del autobús, coste del autobús, coste del kilómetro y número de alumnos en una parada. Esto es principalmente para no tener estos números sueltos en las restricciones y que se puedan editar fácilmente en el .dat.

Las restricciones son muy generales y permiten que adaptes el problema a más paradas y alumnos con solo modificar los datos del fichero. hemos seguido cada restricción en función de lo planteado en el modelado anteriormente.

Parte 3: Análisis de Resultados

Para esta parte, vamos a poner en marcha las diferentes herramientas para solucionar estos dos problemas. Para la primera parte, vamos a usar tanto LibreOffice Calc como MathProg. Para la segunda parte usamos tan solo MathProg.

Problema 1:

Ahora, mostraremos los valores de las diferentes variables de decisión y su comparación respecto a las diferentes restricciones, para comprobar si se cumplen.

1. **Valor de la función objetivo: 400**
2. **Variables de decisión:**
3. **Restricciones:**

Origen\ Destino	P	S1	S2	S3	E
P	PP=0	PS1=1	PS2=0	PS3=1	PE=0
S1	S1P=0	S1S1=0	S1S2=1	S1S3=0	S1E=0
S2	S2P=0	S2S1=0	S2S2=0	S2S3=0	S2E=1
S3	S3P=0	S3S1=0	S3S2=0	S3S3=0	S3E=1
E	EP=0	ES1=0	ES2=0	ES3=0	EE=0

Cumple todas las restricciones:

1	<=	1
1	<=	1
1	<=	1
0	=	0
0	=	0
0	=	0
2	<=	3
0	=	0
15	=	15
5	=	5
10	=	10
30	=	30
-20	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
-5	<=	0
0	<=	0
0	<=	0
0	<=	0
-20	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
0	<=	0
-10	<=	0
0	<=	0

Se ve que la ruta que toma es:

Bus 1: P -> S1 -> S2 -> E

Bus 2: P -> S3 -> E

4. **Conclusión:**
Podemos concluir que los resultados concuerdan con las diferentes restricciones. Por lo que el modelo de LibreOffice es correcto y nos genera una solución factible y óptima.

5. **Modelo MathProg:**
Para MathProg, tenemos exactamente los mismos valores en las variables de decisión, Los valores que tomas las variables de decisión estarán incluidas en la entrega en formato txt, por lo que podemos concluir que el modelado para esta herramienta es correcto, y como tal, nos ofrece una solución óptima y factible.

```
TOTAL = 400 (MINimum)
```

Problema 2

Para este problema, solo hemos incluido el modelo en MathProg. Por lo que no podemos hacer comparaciones respecto a otros modelos.

1. **Valor de la función objetivo:** 585
2. **Variables de decisión:** Para este problema, al haber demasiadas variables, solo vamos a decir qué variables tienen valor 1. y el flujo

Variable	Valor	Flujo N
XPS1	1	NPS1=0
XPS2	1	NPS2=0
XPS3	1	NPS3=0
XS1E	1	NS1E=3
XS2E	1	NS2E=2
XS3E	1	NS3E=3
A1S1	1	NS1E=0
A2S1	1	-
A3S1	1	-
A4S2	1	-
A5S2	1	-
A6S3	1	-
A7S3	1	-
A8S3	1	-

3. **Restricciones:**

Consideramos demasiado complicado colocar una imagen donde se pueda ver todas las restricciones y los valores que toman. todos las restricciones se cumplen y se pueden observar en el TXT donde están las salidas.

4. **Conclusión:**

Podemos concluir que los resultados concuerdan con las diferentes restricciones. Por lo que el modelo es correcto y nos genera una solución factible y óptima. También queremos destacar un pequeño detalle.

Nuestros problemas:

Problema 1:

Para este nuevo problema hemos hecho las siguientes modificaciones: la capacidad del autobús es 30

Resultados obtenidos: Valor función objetivo = 220
Variables de decisión:

PP	PS1	PS2	PS3	PE	NPP	NPS1	NPS2	NPS3	NPE
S1P	S1S1	S1S2	S1S3	S1E	NS1P	NS1S1	NS1S2	NS1S3	NS1E
S2P	S2S1	S2S2	S2S3	S2E	NS2P	NS2S1	NS2S2	NS2S3	NS2E
S3P	S3S1	S3S2	S3S3	S3E	NS3P	NS3S1	NS3S2	NS3S3	NS3E
EP	ES1	ES2	ES3	EE	NEP	NES1	NES2	NES3	NEE

0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	15	0	0
0	0	0	1	0	0	0	0	20	0
0	0	0	0	1	0	0	0	0	30
0	0	0	0	0	0	0	0	0	0

Problema 2:

Para este nuevo problema hemos hecho las siguientes modificaciones: agregamos una 5ta localizacion (S4),

Los nuevos costes son:

Origen\Destino	P	S1	S2	S3	s4	E
P	∞	8	10	10	3	∞
S1	∞	∞	3	7	3	6
S2	∞	3	∞	5	3	7
S3	∞	7	5	∞	3	4
s4	3	∞	∞	∞	∞	10
E	∞	∞	∞	∞	∞	∞

Resultados obtenidos: están en un txt llamado output en la entrega

Ventajas y desventajas de las herramientas utilizadas:

LibreOffice:

Ventajas: existen muchas ventajas al utilizar libreoffice, entre ellas destacan: su distribución es gratuita y está disponible en cualquier sistema operativo lo que la hace más accesible. Es una herramienta con muchas funciones con lo que permite hacer muchas más operaciones. Permite personalizar el modelado y hacer que sea mucho más entendible esto se debe a que es una interfaz gráfica y no una interfaz de consola. Es más sencillo de utilizar y no requiere de mucho conocimiento para resolver los problemas.

Desventajas: para resolver problemas que tienen muchas variables y muchas restricciones es mucho más complicado y tedioso. No se puede utilizar bucles para recorrer las variables.

GLPK:

Ventajas: Es una librería gratuita y está disponible para su distribución en GNU sin dificultades. Permite resolver problemas muy grandes con muchísimas variables de una forma mucho más fácil y eficiente. Te da la oportunidad de organizar todos los datos en distintos ficheros para que se puedan editar los datos más fácilmente. Puedes utilizar bucles y organizar las variables en estructuras de datos.

Desventajas: No dispone de una interfaz gráfica lo cual limita la comprensión de los datos. requiere más conocimientos de programación por parte del usuario para poder utilizar esta herramienta.

Conclusión:

Para finalizar está práctica podemos afirmar que la programación lineal tiene la capacidad de resolver una inmensa cantidad de problemas que se presentan en el mundo actual, generando soluciones óptimas que reducen o aumentan en la mayoría de los problemas los costes de una tarea, haciendo que sea fundamental invertir en ello.

En un problema de programación lineal a día de hoy suelen existir miles de variables que influyen en nuestra función objetivo, es por esta razón que es importante aprender más de una herramienta que permita resolver las tareas de programación lineal como lo son libreoffice y GLPK.

Modelar un problema es el paso más importante para resolver un problema de programación lineal, en otras palabras lograr traducir un problema en lenguaje natural a un lenguaje matemático con variables, función objetivo y restricciones.

Personalmente podemos decir que nos ha gustado mucho trabajar en esta práctica, hemos podido aprender mucho acerca de la utilidad de la programación lineal en el mundo actual. Para ello logramos comprender 2 (GLPK y Libreoffice) herramientas que probablemente nos serán de mucha utilidad en un futuro para poder resolver problemas que se nos presenten. En especial la herramienta de MathProg, que tiene un gran potencial para problemas con un gran número de variables.