CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA

Grupo 81

JUAN MIGUEL PULGAR ESQUIVEL 100451036
100451036@alumnos.uc3m.es

ALBA MARCO UGARTE 100451139
100451036@alumnos.uc3m.es

https://github.com/100451139/P1-100451139-100451036

OCTUBRE DE 2022

GRADO EN INGENIERÍA INFORMÁTICA UC3M

Índice

1. Introducción

En esta práctica se nos presenta un problema de transporte que podríamos decir que está mezclado con uno de asignación, en el que tenemos que asignar a ciertos autobuses para que recojan a unos estudiantes y los lleven al colegio de la manera más óptima posible, es decir, que minimice el coste de transportarlos a todos. En este documento se encuentran, en primer lugar, la implementación del primer modelo y un análisis de los resultados obtenidos al realizar el modelo en las herramientas Libreoffice Calc y GLPK. En la segunda parte del documento encontramos las modificaciones necesarias del primer modelo para crear un segundo modelo más complejo y un análisis de los resultados obtenidos en GLPK. Por último se encuentran unas conclusiones acerca de la práctica realizada.

2. Modelo parte 1.

2.1. Constantes del problema.

En el problema se nos presenta un grafo no dirigido compuesto por 3 paradas de autobús, un parking de autobuses y un colegio al que deben llegar los autobuses, que deben pasar por las paradas para recoger a los alumnos que hay en ellas.
Una de las constantes del problema, será, por lo tanto, una matriz (**P**) que contenga los pesos de las aristas, es decir, la distancia que hay entre 2 puntos del grafo. Tomando los datos del problema, la matriz será la siguiente:

Matriz de pesos de las aristas					
O\D	C	S1	S2	S3	P
C	100	100	100	100	100
S1	6	100	3	7	100
S2	7	3	100	5	100
S3	4	7	5	100	100
P	100	8	10	10	100

Como se puede observar, para aquellos puntos que no estén conectados directamente, utilizamos un valor lo suficientemente alto como para que no se tenga en cuenta a la hora de calcular la ruta de menor coste.
Otra de nuestras constantes es el número de alumnos que hay en cada parada, esto lo representamos como un vector (**V**). En los puntos en los que no hay alumnos (parking y colegio), el valor es 0.
Otras 2 constantes de este problema son el precio por kilómetro recorrido (**p/km**) y el precio fijo de utilizar un autobús (**p/bus**). Así, tendremos 2 constantes con valor de 5 y 120, respectivamente.
Por último, tomaremos como constantes el número de autobuses disponibles (**Nºbuses**) y el número de alumnos que caben como máximo en cada autobús (**plazasBus**), cuyos valores son 3 y 20.

2.2. Variables de decisión.

Nuestras variables de decisión son matrices de tamaño NxM, siendo N el número de lugares de origen posibles y M el número de lugares de destino posibles. Los lugares de destinos posibles serán las paradas {s1 ,s2, s3} y el colegio {c}, mientras que los lugares de origen serán las paradas y el parking {p}.
La matriz de tránsito (**T**) es binaria (0 , 1) y representa si pasa o no un bus por una arista del camino, es decir, tendrá un 0 si el autobús no pasa por esa arista y 1 uno en caso contrario.
La matriz de flujo (**F**) representa el número de alumnos que iría dentro del autobús cuando se pasa de una arista a otra, es decir, el acumulativo de todos los alumnos que se han recogido por el camino.

2.3. Restricciones.

- <u>Para realizar las rutas se dispone de un máximo de 3 autobuses.</u>

El número de autobuses que sale del parking será el número total de autobuses que se usan para la realización de las rutas. Este valor vendrá dado por la suma de los elementos de la fila de la matriz T donde parking es el origen, es decir:

$$\boxed{\sum_{k \in paradas} T_{p\,k} \leq N^{o} buses}$$

- <u>A cada parada llega una única ruta.</u>

Para cumplir esta premisa, necesitaremos 3 restricciones, una para cada parada {s1, s2, s3}.
El número de rutas que llega a cada parada

$$\boxed{\sum_{k \in paradas} T_{si\,k} = 1 \text{ si} \in \text{s1, s2,s3}}$$

- <u>De cada parada sale una única ruta.</u>

Para cumplir la restricción, hemos pensado que si a cada parada llega una única ruta, el número de rutas que salen de una parada debe ser el mismo número de rutas que llegan. Por lo tanto:

$$\boxed{\sum_{k \in paradas} T_{si\,k} = \sum_{j \in paradas} T_{j\,si} \quad \text{si} \in \text{s1, s2,s3}}$$

- Todas las rutas salen del parking y llegan al colegio.

Así, el número de rutas que salen del parking debe coincidir con el número de rutas que llegan al colegio. Para cumplir esta restricción, necesitamos hacer 3 restricciones individuales, una para cada parada. De esta forma nos queda:

$$\boxed{\sum_{k \in paradas} T_{k\,c} = \sum_{j \in paradas} T_{p\,j}}$$

- No se podrá superar la capacidad del autobús.

Se multiplicará la matriz de tránsito (T) por las plazas de cada bus (plazasBus), de está manera conseguiremos una matriz que indica el número máximo de alumnos que debe haber en cada trayecto. Si en el problema no se pasa por ese trayecto el número máximo de alumnos será 0. La matriz descrita deberá ser mayor o igual que la matriz de flujo.

$$\boxed{T_{i\,j} * plazasBus = F_{i\,j} \quad \text{i , j} \in \text{paradas}}$$

- El flujo de alumnos que sale de una parada debe ser exactamente el flujo de alumnos que entra en esa parada más el número de alumnos que esperan en ella.

El flujo de alumnos que sale de una parada es representado por la suma de los elemento de la fila de la parada correspondiente en la matriz de flujo (F), por ejemplo, la suma de los elemento de la fila donde s2 es el origen, es el número de alumnos que salen de s2 porque solo puede llegar un autobus. Este mismo razonamiento se hace para el flujo de alumnos que entra en una parada, pero con las filas, ya que representan qué parada es el destino. Este sumatorio de la columna más el número de alumnos que se suben en esta parada (en el ejemplo es V(s2)) tendrá que ser igual al sumatorio de la fila. Al igual que en otras restricciones tendrá que haber una para cada parada.

$$\boxed{\sum_{k \in paradas} F_{si\,k} = \sum_{j \in paradas} F_{j\,si} + V_{si} \quad \text{si} \in \text{s1, s2,s3}}$$

- Se deben recoger a todos los alumnos

En nuestro caso hemos necesitado implementar esta restricción ya que en el enunciado no se dice explícitamente que se deba recoger a todos los alumnos, pero es obvio que debe ser así ya que es de lo que trata el problema. Por lo tanto, el número de alumnos totales que llegan al colegio debe ser la suma de alumnos que hay en todas las paradas.

$$\boxed{\sum_{k \in paradas} F_{k\,c} = \sum_{i \in paradas} V_i}$$

- Las rutas que salen del parking deben ser de 0.

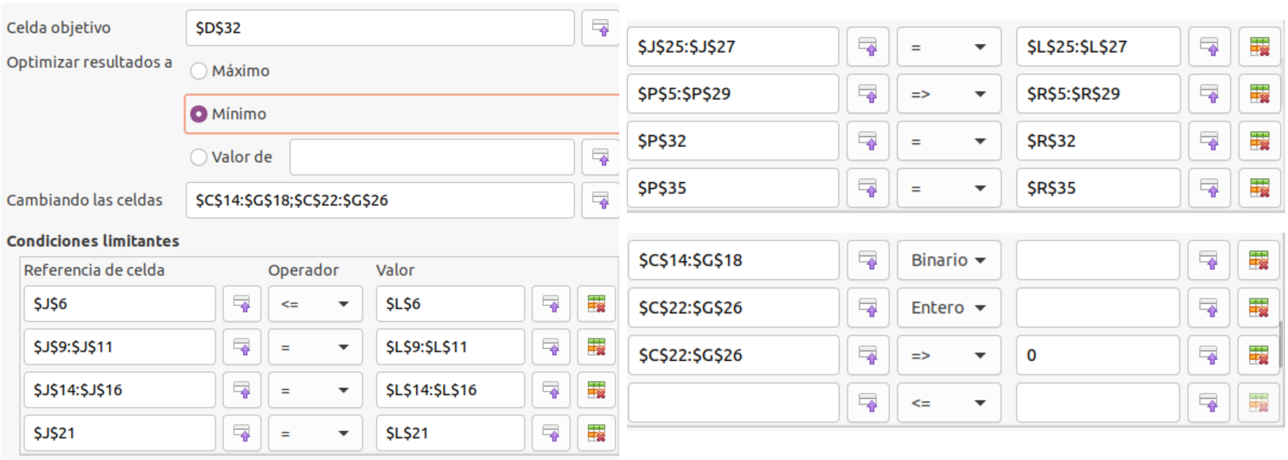
$$\boxed{\sum_{k \in paradas} T_{k\,p} = 0}$$

2.4.Función objetivo.

Por lo tanto, con estas restricciones, nuestra función objetivo será la que minimice el coste de transportar todos los alumnos al colegio. Por lo tanto, debe ser el número de kilómetros que se hagan en total multiplicado por el precio de recorrer un kilómetro, sumado a la cantidad de autobuses que se usan por el precio de utilizar un autobús.

$$\boxed{\min f = \left(\sum_{i \in paradas} \sum_{j \in paradas} P_{ij} * T_{ij} \right) * p/km + p/bus * \sum_{k \in paradas} T_{p\,k}}$$

2.5.Configuración del Excel.



Como la configuración de excel no se guarda, dejamos unas capturas de pantalla donde se indica las restricciones que hay que poner en el solver.

Las únicas restricciones que no vienen del todo bien indicadas en el propio excel son:

- Matriz de tránsito binaria.
- Matriz de flujo de números enteros positivos.

3. Modelo parte 2.

Para la parte dos modificaremos el modelo de la parte 1, por lo que solo nombraremos las constantes , variables y restricciones nuevas o que tengan que ser modificadas.

3.1. Constantes del modelo

Este problema es parecido al anterior, exceptuando algunas modificaciones. En primer lugar, tenemos 8 alumnos, que se han nombrado con la letra a (a1-a8). Además, en el problema se nos dice que algunos alumnos son hermanos, por lo que se ha creado una matriz que representa que alumnos son hermanos. A esta matriz la llamaremos **H** y en este caso es de la siguiente forma:

	a1	a2	a3	a4	a5	a6	a7	a8 :=
a1	0	0	0	0	0	0	0	0
a2	0	0	0	0	0	0	0	0
a3	0	0	0	0	0	0	0	0
a4	0	0	0	0	1	0	0	0
a5	0	0	0	1	0	0	0	0
a6	0	0	0	0	0	0	0	0
a7	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0;

En este ejemplo, podemos ver que solo los alumnos a4 y a5 son hermanos. Todos los alumnos que sean hermanos deben ir juntos a la parada, es decir, no podemos mandar a hermanos a paradas distintas.
A este problema además se le añade la restricción de que ciertos alumnos deben ir necesariamente a una parada, por lo tanto, se ha creado una matriz **X** que indica a qué paradas puede ir cada alumno. Esta matriz tendrá un 1 si el alumno puede ir a esa parada y un 0 en caso contrario. De esta forma, en el ejemplo que se propone la matriz nos queda de la siguiente forma:

	c	s1	s2	s3	p
a1	0	1	0	0	0
a2	0	1	0	0	0
a3	0	1	0	0	0
a4	0	1	1	1	0
a5	0	1	1	1	0
a6	0	0	0	1	0
a7	0	0	0	1	0
a8	0	0	0	1	0;

Además la capacidad del bus cambia, en esta parte la constante plazasBus será 4.

3.2. Variables de decisión.

Para este problema como cada alumno puede ir a "x" paradas en concreto se añadirá una nueva matriz (A) que asignará donde es recogido cada alumno. Esta matriz tiene tantas filas como alumnos y tantas columnas como paradas disponibles. Con esta matriz podremos saber a qué parada ha ido cada alumno y se podrá comprobar si los dos hermanos están en la misma parada.

3.3. Restricciones.

- <u>El flujo de alumnos que sale de una parada debe ser exactamente el flujo de alumnos que entra en esa parada más el número de alumnos que esperan en ella.</u>

Esta restricción es muy similar, pero ya no tenemos el vector que representa el número de alumnos que hay en cada parada, ahora tenemos la matriz de asignación de alumnos (A) donde al sumar los elementos de las columnas, te dará el total de alumnos de esa parada.

$$\boxed{\sum_{k \in paradas} F_{si\,k} = \sum_{k \in alumnos} \sum_{j \in paradas} F_{j\,si} + A_{k\,si}}$$

- Un alumno solo puede estar en una parada

Las filas de la matriz de asignación de alumnos debe ser igual a 1, de no ser así el alumno se estaría "duplicando" y estaría en dos paradas a la vez.

$$\sum_{j \in paradas} A_{ai\,j} = 1$$

- La parada asignada al alumno tiene que ser una de las posibles

Esta restricción es igual a la anterior, la única diferencia es que ahora la multiplicamos por la matriz de paradas posibles, para obligar al programa a elegir una parada que el alumno tenga asignada.

$$\sum_{j \in paradas} A_{ai\,j} * X_{ai\,j} = 1$$

- Se deben recoger a todos los alumnos

En este problema, el número total de alumnos será la suma de todas las casillas de la matriz de asignación (A), ya que esta matriz tendrá exactamente 8 unos que son el número total de alumnos.

$$\sum_{k \in paradas} F_{k\,c} = \sum_{i \in paradas} \sum_{\in j\, paradas} A_{i\,j}$$

- Todos los hermanos deben ir juntos

Para ver si un hermano va a la misma parada que otro multiplicaremos la matriz de hermanos por la matriz de asignación así nos dará un uno en la parada donde este este hermano, hacemos lo mismo para el hermano y lo igualamos.

$$H_{i\,j} * A_{i\,k} = H_{i\,j} * A_{j\,k} \quad \text{i ,j} \in \text{alumnos ; k} \in \text{paradas}$$

4. Análisis de resultados.

4.1. Parte 1.

4.1.1. Análisis del resultado obtenido.

Camino realizado: Bus 1: P-S1-S2-C Bus 2: P-S3-C.

Valor de función objetivo: 400.

El valor de la función objetivo es correcto (coincide con el valor que ha comentado el profesor en sus clases). Como el objetivo es minimizar, el programa debe buscar los trayectos más cortos y debe intentar usar el mínimo número de buses. El mínimo número de buses posibles va a ser de 2, porque hay un límite de 20 plazas por bus y tenemos 30 alumnos, tal y como hace la solución. Por otro lado, un bus no puede ir a S1 y a S3 a la vez, porque superaría las plazas del autobús, dicho esto el programa debe buscar el camino más corto cumpliendo estas restricciones. La arista más corta es de S1 a S2, por lo que es lógico que haga ese recorrido.

El resultado obtenido cumple con todas las restricciones:

- Hay como máximo 3 autobuses: Se usan 2 autobuses.
- A cada parada llega una única ruta y de cada par sale una única ruta: Se cumple porque ninguno de los autobuses repite parada.
- Todas las rutas que salen del parking, llegan al colegio: Los dos autobuses comienzan en el parking y terminan en el colegio.
- No se podrá superar la capacidad del autobús: En el bus 1 hay alumnos y en el bus 2 hay , por lo tanto ninguno de los dos buses tiene más de 20 alumnos por bus.
- El número de alumnos que llegan a la parada más el número de alumnos que se suben en ella ha de ser igual al número de alumnos que llegan a la parada destino. Como se observa en la tabla , se cumple está restricción. Por ejemplo, si nos fijamos en el trayecto de S1 a S2, hay 15 personas, en la parada S2 se suben 5, es decir, 20 personas en total y en la parada destino S2-C hay justo 20 personas, por lo que se cumple correctamente esta restricción.

Matriz de flujo nuevo					
O\D	C	S1	S2	S3	P
C	0	0	0	0	0
S1	0	0	15	0	0
S2	20	0	0	0	0
S3	10	0	0	0	0
P	0	0	0	0	0

- Se deben recoger a todos los alumnos: El total de alumnos es 30, por lo que en la columna donde C es el destino debería de sumar 30, como se observa en la tabla anterior este valor es correcto, todos los alumnos son recogidos.
- Las rutas del parking deben ser de 0: La suma de la fila donde P es el origen da 0, por lo que todos los buses salen del parking y ninguno vuelve al parking.

En cuanto a la complejidad del problema, esta parte del trabajo es más simple, y viene reflejado en el número de restricciones que en nuestro caso son 34. Viendo las variables de decisión son solo dos matrices 5x5, es decir 25 variables de decisión.

4.1.2. Batería de pruebas.

A la hora de hacer el modelo se ha tenido en cuenta que el modelo debe ser lo más general posible, es decir, que aún cambiando los datos debe dar una solución correcta. Para comprobarlo se harán varias modificaciones en el modelo y comprobaremos si los datos son correctos.

Caso	Valor esperado	Valor obtenido
Igual pero de S1 a S2 hay 20 km.	Se espera que ya no pase por la arista S1 -S2 , por lo que el bus que va a S1 no debe ser el mismo que el que va a S2.	Bus 1: P-S1-C Bus 2: P-S2-S3-C No pasan por la arista S1-S2
Igual que en el caso de ejemplo pero en la parada S2 hay 10 alumnos	Se espera que el bus 1, ya no pase por S1 y S2 porque superaría el límite de plazas.	Bus 1: P-S1-C Bus 2: P-S2-S3-C El bus dos va a s2 y s3 sin superar el límite de plazas.
Igual que en el caso de la parte1 pero solo hay 1 autobús	Sin solución, porque se supera el límite de plazas.	Sin solución
De S1 a S2 hay 20km y de S2 a S3 también. Además solo hay 2 buses.	Se espera que aumente la función objetivo, ya que ahora hay más kilómetros.	Función objetivo = 480. Bus 1: P-S1-C Bus 2: P-S2-S3-C
Hay 20 alumnos en S1, 15 en S2 y 10 en S3.	Se espera que se use un bus para cada parada y que aumente la función objetivo.	Función objetivo: 585 Bus 1:P-S1-C Bus 2: P-S2-C Bus 3: P-S3-C
Hay 15 buses disponibles	Se espera que no cambie el resultado, porque solo se llegaban a usar 2 buses.	Función objetivo = 400 Bus 1: P-S1-S2-C Bus 2: P-S3-C
El precio por kilómetro son 200 euros y aumentan los kilómetros entre paradas.	Como el precio por kilómetro es mayor que el del bus y las distancias entre paradas han aumentado, el programa intentará usar un bus por parada.	Bus 1:P-S1-C Bus 2: P-S2-C Bus 3: P-S3-C

Como podemos ver, todas las soluciones son las esperadas y el problema es en gran medida generalizable, ya que podemos cambiar cualquier constante del problema y el resultado sigue siendo correcto.

4.2. Parte 2.

4.2.1. Análisis del resultado obtenido.

Camino realizado: Bus 1: P-S1-C Bus 2: P-S2-C. Bus 3: P-S3-C
Asignación de paradas: S1 { A1, A2, A3}, S2{ A4, A5} , S3{A6, A7, A8}.
Valor de función objetivo: 585.
El valor de la función objetivo es correcto (coincide con la solución dada por el profesor). En primer lugar hay 8 alumnos, por lo que podríamos pensar que debe haber solo dos autobuses, pero esto no es realmente así. Para a la parada s1 pueden ir A1, A2 y A3 y a la parada s3 pueden ir A6, A7, A8, por lo que solo cabe un alumno más en el bus, por ello como A4 y A5 deben permanecer juntos, se necesitará un nuevo autobús.
Las restricciones del apartado anterior se cumplen y se comprueban de la misma manera, por lo que solo me voy a fijar en las nuevas restricciones.

- Un alumno solo puede estar en una parada: Si nos fijamos en la asignación de paradas (S1 { A1, A2, A3}, S2{ A4, A5} , S3{A6, A7, A8}) ningún alumno repite parada, ni está duplicado.
- La parada asignada al alumno tiene que ser una de las posibles: Los alumnos A1 , A2 y A3 solo tienen asignada la parada 1 , los alumnos A6, A7 y A8 solo tienen asignada la parada 3 y los alumnos A4 y A5 pueden ir a todas las paradas, sin embargo como son hermanos los dos "deciden" ir a la parada 2, por lo que todos los alumnos van a una parada de las posibles.
- Todos los hermanos deben ir juntos: A4 y A5 van juntos a la parada 2.

En cuanto a la complejidad del problema nos fijamos en las restricciones y las variables de decisión. Esta parte del problema es bastante más compleja, ya que se añade el problema de asignación de alumnos a paradas y de que hay alumnos que son hermanos y por ello deben ir juntos. En esta parte tenemos 330 restricciones y 90 variables. Si nos fijamos en las variables tendremos una matriz 5x5 para indicar el recorrido del bus (25 variables) , otra matriz 5x8 para a donde va cada alumno (40 variables) y otra matriz 5x5 indicando el número de alumnos que hay en cada trayecto (25 variables).

4.2.2. Batería de pruebas.

Igual que en la parte 1, hemos intentado que sea lo más general posible, por lo que si hacemos pequeños cambios en el problema debería de dar una solución correcta.

Caso	Valor esperado	Resultado obtenido
Añado otros dos hermanos (A1 Y A8) y modificó la matriz de paradas posibles para que puedan ir juntos.	Se espera que esos dos hermanos vayan juntos.	Los alumnos A1 y A8 van juntos. Y se cumplen todas las restricciones.
Permito que todos los alumnos vayan a todas las paradas y aumentó el tamaño de bus a 8	Se espera que todos los alumnos se asignen a la parada más cercana. Por lo que se usará un solo bus.	Todos los alumnos van a la parada s3 que es la más cercana. Y se usa un solo bus.
Hago que a un alumno no se le asigne a ninguna parada.	Sin solución	Sin solución
Asigno a todos los alumnos a la misma parada.	Sin solución, porque no saben todos los alumnos en un bus de 4 plazas.	Sin solución
Hago que A1 y A2 sean hermanos, los dos pueden ir a s1 además A1 puede ir a s2 y A2 a s3.	Se espera que los alumnos A1 y A2 vayan a la misma parada, es decir, a s1.	Los dos alumnos van a s1.
Dos hermanos que no tienen asignado ninguna parada igual	Sin solución, porque no cumpliria las dos restricciones (hermanos juntos y paradas disponibles)	Sin solución.
Hago que haya 3 hermanos , los tres tienen una parada en común (s2).	Todos los hermanos deben ir a s2.	Los tres hermanos van a s3.

4.3. LibreOffice vs GLPK.

En la práctica usamos LibreOffice para la primera parte, que es más sencilla. Esto es lógico, porque GLPK es más potente y es más sencillo poner las restricciones, por ejemplo, en LibreOffice hemos hecho 25 restricciones una por una para indicar que no puede haber más de 20 alumnos por bus, sin embargo, en GLPK estas 25 restricciones se pueden poner en una sola línea de código, por lo que ahorra mucho tiempo.

LibreOffice es más visual, porque te sale el resultado en las tablas que tu hayas puesto y te da la opción de poner colores y diseñar el archivo para que sea más fácil de visualizar. Por otro lado, GLPK te devuelve un archivo con todas las restricciones y con todas las variables una detrás de otra, por lo que es más costoso visualizar bien la solución.
Para la parte 2, usar LibreOffice habría sido una pérdida de tiempo, porque tendríamos que escribir más de 300 restricciones una a una.
Por último, estamos más habituados a usar LibreOffice o Excel en vez de GLPK, por lo que a lo largo de la práctica, ha sido más costoso usar GLPK que LibreOffice.

5. Conclusión.

En esta práctica hemos aprendido a relacionarnos con las herramientas libreoffice Calc y GLPK para la resolución de problemas de programación lineal. Además, hemos aprendido a modelar problemas más complejos a los propuestos en los ejercicios en clase. Pensamos que en cuanto a utilidad, GLPK es mucho mejor que Calc, ya que nos permite hacer ciertas cosas con mucha mayor facilidad y velocidad. Pero si lo que buscamos es entender el modelo de forma visual, Calc es una mejor herramienta porque GLPK es muy poco visual.