

1 Objetivo

El objetivo de esta práctica es que el alumno aprenda a modelar problemas de programación lineal y a resolverlos con dos tipos de herramientas: hojas de cálculo y algoritmos de resolución utilizando técnicas de modelización.

2 Enunciado del problema

El servicio de ambulancias de una ciudad cuenta con parkings de ambulancias desde los cuales se atienden las emergencias que se produzcan. Con el objetivo de hacer el servicio más eficiente, los responsables han estudiado el número de llamadas que llegan desde cada uno de los distritos en los que se divide la ciudad a lo largo de un año, y se han planteado varios problemas. Junto con los responsables de infraestructuras del ayuntamiento, se han puesto en contacto con los alumnos de Heurística y Optimización para resolverlos óptimamente. Concretamente, estos problemas están relacionados con la forma de determinar qué parking de ambulancias debe atender las llamadas de qué distritos, y también con la localización alternativa de otros parkings en la ciudad.

2.1 Parte 1: Modelo básico en Calc

El primer problema consiste en determinar el número de llamadas de cada distrito que debe atender cada parking de ambulancias. Esto les ayudará a elegir más rápidamente el parking del que tienen que enviar las ambulancias cuando se produzca una emergencia.

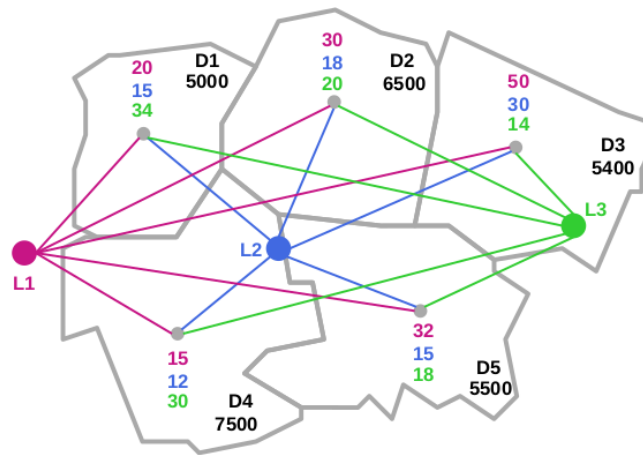


Figure 1: Problema con 5 distritos y 3 parkings de ambulancias.

Inicialmente se plantea una versión simplificada del problema en la que se consideran únicamente cinco distritos (D1, D2, D3, D4 y D5) y tres localizaciones de parkings de ambulancias (L1, L2 y L3). La Figura 1 muestra el problema a resolver. El número que aparece debajo del identificador del distrito representa el número de llamadas que se realizaron desde ese distrito. Por ejemplo, del distrito D1 se realizaron 5000 llamadas, 6500 desde D2, etc.

Los responsables del servicio de emergencias consideran que una ambulancia situada en un determinado parking tarda siempre el mismo tiempo en atender una llamada desde cualquier punto del mismo distrito. Este tiempo se estima como el tiempo medio que tardaron las ambulancias del mismo parking en atender las llamadas del distrito el año anterior, y se muestran en la Figura 1 con un código de colores: L1 (rosa), L2 (azul) y L3 (verde). Así, para el distrito D1, el tiempo que tarda una ambulancia del parking L1 es 20 minutos, 15 minutos si es desde el parking L2, y hasta 34 si sale del parking L3.

A continuación se indica una posible distribución de la atención de llamadas:

| Localización | D1 | D2 | D3 | D4 | D5 |
|--------------|------|------|------|------|------|
| L1 | 2500 | -- | -- | 3000 | -- |
| L2 | 2500 | 6500 | 1400 | 1000 | 5000 |
| L3 | -- | -- | 4000 | 3500 | 500 |

Es condición importante que todas las llamadas de todos los distritos sean atendidas en algún parking de ambulancias. Sin embargo, no todas las posibles distribuciones son aceptables, puesto que existen requisitos adicionales que se deben cumplir:

- Un parking de ambulancias no puede atender más de un determinado número de llamadas en total, que en este caso es 10.000 llamadas.
- Se debe garantizar que una ambulancia no tardará nunca más de 35 minutos en llegar al lugar donde se produce la emergencia.
- Para balancear el esfuerzo, el número total de llamadas asignado a un parking no puede exceder en más del 50% el número total de llamadas asignado a cualquier otro parking de ambulancias.

Además, la distribución de las llamadas debe ser tal que el tiempo total empleado en atender todas las llamadas del año sea el mínimo posible.

Se pide:

1. Modelar el problema como un problema de Programación Lineal entera.
2. Implementar el modelo en una hoja de cálculo (LibreOffice).

2.2 Parte 2: Modelo avanzado en GLPK

El ayuntamiento de la ciudad junto con los responsables de emergencias se están planteando ahora modificar las localizaciones de los parkings de ambulancias. Para ello han incorporado nuevas localizaciones posibles. El problema que quieren resolver es el siguiente: dado un conjunto de localizaciones candidatas, donde algunas existían previamente y otras son nuevas, se debe determinar cuáles son las más adecuadas para situar los parkings, así como la distribución de llamadas que deberían tener las nuevas localizaciones seleccionadas.

Para los nuevos parkings, dado que no hay datos históricos, se estima que el tiempo necesario para atender una llamada de un distrito es el tiempo que se tarda en llegar desde el parking hasta el centro del distrito.

Por ejemplo, considérense tres nuevas localizaciones candidatas L4, L5 y L6 al caso descrito en el apartado anterior. Para estas nuevas localizaciones se podría contar con la siguiente estimación de tiempos de llegada a los diferentes distritos:

| Localización | D1 | D2 | D3 | D4 | D5 |
|--------------|----|----|----|----|----|
| L4 | 12 | 15 | 40 | 10 | 20 |
| L5 | 20 | 18 | 20 | 17 | 10 |
| L6 | 55 | 35 | 20 | 30 | 30 |

Al igual que en el problema descrito en el apartado anterior, el tiempo máximo en atender una llamada está limitado, así como el número máximo de llamadas que puede atender un parking de ambulancias. Igualmente, el número total de llamadas asignado a un parking de ambulancias no puede exceder en más del 50% el número total de llamadas asignado a cualquier otro parking, aunque ahora esto sólo se tiene que cumplir para los parkings situados en una localización seleccionada. Lógicamente, una localización seleccionada debe estar asignada al menos a un distrito.

Adicionalmente, es necesario que se cumplan los siguientes requisitos:

- Si el número de llamadas de un distrito es mayor o igual que el 75% del número máximo de llamadas que pueden atender los parkings de ambulancias, sus llamadas se tienen que distribuir necesariamente entre dos o más parkings de ambulancias. En el problema descrito en la parte 1, dado que el número máximo de llamadas que atiende un parking es 10.000, esto se tendría que cumplir para los distritos con al menos 7.500 llamadas.
- Si un parking de ambulancias cubre alguna llamada de un distrito, no puede cubrir menos del 10% del total de las llamadas de ese distrito.

Construir y poner en marcha un nuevo parking de ambulancias tiene un coste fijo de 50.000 euros, así que es preciso asumir este coste para cada una de las nuevas localizaciones seleccionadas. Por otro lado, el gasto anual en combustible depende de la distribución de las llamadas, y del tiempo que tardan las ambulancias en atenderlas. Concretamente, se estima que el gasto en combustible por atender una llamada tiene un coste de 2 euros por minuto. En este problema, los responsables están interesados en que el gasto total sea el mínimo posible.

Se pide:

1. Modelar el problema descrito en la parte 2, tomando como base el modelo de la parte 1 y haciendo las modificaciones oportunas.
2. Implementar el modelo de la parte 1 de la práctica en un lenguaje de modelado más sofisticado, MathProg. El modelo de la parte 1 permitirá practicar con la sintaxis de este lenguaje antes de abordar la implementación del modelo de la parte 2. El modelo de la parte 1 debe ser lo suficientemente general para permitir solucionar problemas diferentes del presentado en la Figura 1 con las mínimas modificaciones posibles. De hecho, estas modificaciones deberían llevarse a cabo en el fichero de datos .dat, dejando tal cual el fichero con el modelo .mod.
3. Una vez implementado el modelo de la parte 1, se harán las modificaciones necesarias a éste para generar un nuevo modelo que permita resolver el problema de optimización descrito en la parte 2. Nuevamente el modelo de la parte 2 debe ser lo suficientemente general para permitir solucionar problemas diferentes.

De esta parte resultarán dos modelos en MathProg, uno para el tipo de problemas definido en la parte 1 y otro para el tipo de problemas definido en la parte 2.

2.3 Parte 3: Análisis de Resultados

En este apartado se deben analizar todos los resultados obtenidos, describir la solución obtenida (comprobando que cumple con todas las restricciones del enunciado) y analizar qué restricciones están limitando la solución del problema.

También se debe realizar un análisis de la complejidad del problema: ¿cuántas variables y restricciones has definido? Inventar tus propios problemas tanto para la parte 1 como para la parte 2. Explica las nuevas soluciones que se generan en estos problemas y cómo las modificaciones afectan a la dificultad de resolución.

Por último, se deben discutir las ventajas y desventajas de las dos herramientas utilizadas en la asignatura: LibreOffice y GLPK.

3 Directrices

La práctica se realizará en grupos de dos estudiantes. Se utilizará Github Campus Education, por lo que es indispensable que ambos estudiantes estén dados de alta.

La memoria debe entregarse en formato .pdf y tener un máximo de 10 páginas en total, incluyendo la portada, contraportada e índice. Al menos, debe contener:

1. Una portada con los nombres y NIAs de los autores.
2. Breve introducción explicando los contenidos del documento.
3. Descripción formal de los modelos, argumentando las decisiones tomadas, y mostrando algebraicamente los modelos implementados.
4. Análisis de los resultados de acuerdo a las indicaciones de la sección 2.3
5. Conclusiones acerca de la práctica.

La memoria no debe incluir código fuente en ningún caso.

4 Evaluación

La evaluación de la práctica se realizará sobre 10 puntos. Para que la práctica sea evaluada deberán realizarse, al menos el apartado 1 y la parte correspondiente de la memoria. La distribución de puntos es la siguiente:

1. Parte 1 (3 puntos)
 - Modelización del problema (1 puntos)
 - Implementación del modelo (2 puntos)
2. Parte 2 (5 puntos)
 - Modelización del problema (2 puntos)
 - Implementación de los modelos (3 puntos)
3. Parte 3 (2 puntos)

En la evaluación de la modelización del problema, un modelo correcto supondrá la mitad de los puntos. Para obtenerse el resto de puntos, la modelización del problema deberá:

- Ser formalizada correctamente en la memoria.
- Ser, preferiblemente, sencilla y concisa.
- Estar bien explicada (debe quedar claro cuál es la utilidad de cada variable/restricción).
- Justificarse en la memoria todas las decisiones de diseño tomadas.

En la evaluación de la implementación del modelo, un modelo correcto supondrá la mitad de los puntos. Para obtenerse el resto de puntos, la implementación del problema deberá:

- Hacer uso (en la implementación) de las capacidades que ofrecen las herramientas para que hacer/actualizar el modelo sea lo más sencillo posible (por ejemplo, utilizar sumaproducto si es posible en el caso de la hoja de cálculo o el uso de sets en MathProg).
- Mantener el código (hoja de cálculo o ficheros de MathProg) correctamente organizado y comentado. Los nombres deben ser descriptivos. Deberán añadirse comentarios en los casos en que sea necesario para mejorar la comprensión del trabajo realizado.

Al puntuar el análisis de resultados, se valorará positivamente el hecho de incluir en la memoria conclusiones personales acerca de la dificultad de la práctica y de lo aprendido durante su elaboración.

Importante: Los modelos implementados en la hoja de cálculo y GLPK deben ser correctos. Esto es, deben de funcionar y obtener soluciones óptimas al problema que se solicita. En ningún caso se obtendrá una calificación superior a 1 punto por un modelo que no es correcto. Por tanto, si la parte 1 no está correctamente acabada, la nota máxima será de 1 punto y, si la parte 2 no está correctamente acabada, como mucho se obtendrá una calificación de 4 puntos.

Se comprobarán los push al repositorio y que ambos miembros del equipo han contribuido.

5 Entrega

Se tiene de plazo para entregar la práctica hasta el 27 de Octubre a las 23:55. Este límite es fijo y no se extenderá de ningún modo.

Sólo un miembro de cada pareja debe subir los siguientes ficheros:

- Un único fichero .zip a través del punto de entrega de 'Aula Global' llamado ``Primera práctica''. El fichero debe nombrarse p1-NIA1-NIA2.zip, donde NIA1 y NIA2 son los últimos 6 dígitos del NIA (rellenando con 0s por la izquierda si fuera preciso) de cada miembro de la pareja.

Ejemplo: p1-054000-671342.zip.

- La memoria debe subirse separadamente a través del punto de entrega Turnitin de 'Aula Global' llamado ``Primera práctica (sólo pdf)''.

La memoria debe llamarse NIA1-NIA2.pdf ---después de sustituir adecuadamente los NIAs de cada estudiante como en el caso anterior.

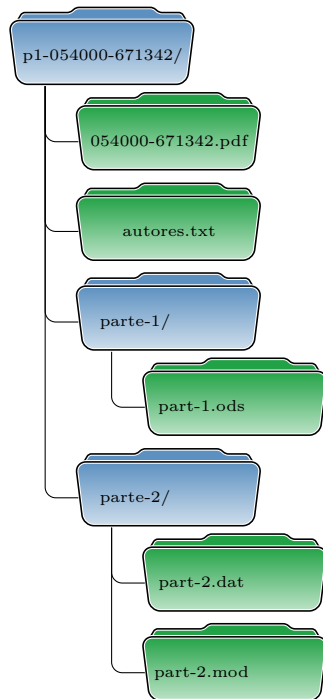
Ejemplo: 054000-671342.pdf.

La descompresión del fichero .zip descrito en el primer punto debe producir un directorio llamado p1-NIA1-NIA2, donde NIA1 y NIA2 son los últimos 6 dígitos del NIA de cada estudiante rellenando con 0s si fuera preciso. Este directorio debe contener: primero, la misma memoria entregada a través del segundo enlace descrito anteriormente que debe nombrarse como NIA1-NIA2.pdf ---después de sustituir convenientemente los NIAs de cada estudiante; segundo, un fichero llamado autores.txt que identifica a los miembros de cada pareja, con una línea por cada uno con el siguiente formato: NIA Surname, Name. Por ejemplo:

054000 Von Neumann, John
671342 Turing, Alan

Además, este directorio debe contener un directorio por cada parte realizada llamados ``parte-1'' y ``parte-2''. Las soluciones de cada parte (tanto el código fuente como la hoja de cálculo generada) deben incluirse en sus respectivos directorios.

La siguiente figura muestra una distribución posible de los ficheros después de descomprimir el fichero .zip:



Importante: no seguir las normas de entrega puede suponer una pérdida de hasta 1 punto en la calificación final de la práctica.