Memoria de la Práctica 1 de Heurística y Optimización

## 1. Introducción

En esta memoria se cubrirán los aspectos técnicos para la resolución de tres tareas de programación lineal para una empresa de autobuses.

Para cada tarea, se explicará el modelo utilizado para resolver la tarea correspondiente, y se justificarán las decisiones que se han tomado para construirlo.

## 2. Modelos

### 2.1. Modelo básico en hoja de cálculo

Este primer modelo se encarga de resolver una tarea de programación lineal (de aquí en adelante P.L) conocida como **problema de asignación**, en la que se tiene dispone de cinco autobuses y cinco talleres, y se quiere lograr **asignar cada bús a un taller**, minimizando la distancia total de los autobuses con los talleres.

Modelado

Para modelar este problema, se han definido los siguientes campos:

##### Variables de decisión

Como variables de decisión, hemos utilizado las siguientes variables binarias:

si el taller recibe al autobús ; en caso contrario.

##### Restricciones

- Un bus solamente puede estar asignado a un taller

*(Para todo taller, la suma de los buses asignados a él es 1)*

- Un taller tiene que recibir un autobús siempre

*(Para todo autobús, la suma de los talleres a los que está asignado es 1)*

##### Vector de recursos ()

Como vector de recursos tenemos la “disponibilidad” de cada taller y cada autobús

Los primeros cinco elementos de b corresponden a los talleres; los demás, a los buses.

= Número de autobuses que puede recibir el taller i

= Número de talleres a los que puede ir el bus i

##### Matriz de coeficientes tecnológicos ()

Contiene las distancias de cada taller con cada autobús.

Distancia entre el taller y el autobús

Función objetivo ()

Se quiere minimizar la distancia total entre los talleres y los buses

Para implementar este modelo en una hoja de cálculo, se ha usado un documento de LibreOffice Calc, y se ha conseguido resolver el problema planteado con el solver integrado que el programa ofrece.

### Análisis de resultados

La solución óptima para esta tarea de P.L. es **573**. Esto significa que la distancia total mínima que tiene que ser recorrida por los autobuses es de 573 unidades métricas.

En cuanto a las asignaciones, observamos que:

* Al taller se le asigna el bus
* Al taller se le asigna el bus
* Al taller se le asigna el bus
* Al taller se le asigna el bus
* Al taller se le asigna el bus

2.2. Modelado en GLPK

Una vez hemos definido el modelo para un problema particular, generalizarlo es tarea fácil.  
Simplemente tenemos que dejar de utilizar las dimensiones fijas () que se nos propusieron para el anterior ejercicio, y con esto tendremos un modelo que resuelve este tipo tareas de P.L. para **cualquier número de autobuses y talleres** (no negativo).

Para las siguientes tareas no vamos a utilizar LibreOffice Calc, sino GLPK: un solver de tareas de P.L.  
Crearemos un archivo .*mod* con el modelo que GLPK usará, y un script de *Python* que se encargue de generar un archivo de datos *.dat* que GLPK entienda, a partir de un fichero de entrada con una organización específica.

2.2.1. Minimización del impacto de averías

Este problema se puede dividir en dos *subproblemas*: en el primero queremos minimizar el coste total de asignar ciertos buses a ciertos talleres, y en el segundo queremos minimizar el coste total que supone dejar buses sin asignar.

Nosotros, sin embargo, hemos optado por unir esos subproblemas en una única tarea de P.L.,, con una única función objetivo.

Modelado

Para modelar el problema, se definen estos campos:

Variables de decisión

Hemos modelado una variable binaria para las asignaciones de un autobús a una franja.

si el autobús está asignado a la franja ; si no lo está.

Parámetros

* : €/km a asumir si un bus ha sido asignado a un taller. (Constante no negativa)
* : €/pasajero a asumir si un bus no ha sido asignado a un taller. (Constante no negativa)
* : Distancia (en km) entre el bus y el taller.
* : Pasajeros que lleva el bus .

Restricciones

Un autobús solo puede estar asignado a una franja

*(Para cada franja, la suma de los autobuses asignados a ella es, como mucho, 1)*

*(Para cada bus, la suma de las franjas a las que está asignado es, como mucho, 1)*

Función objetivo

Se quiere minimizar el **coste total que suponen los autobuses**, esté este asignado (en cuyo caso se aplicará el coste por asignar), o no (en cuyo caso se aplicará el coste por no haberlo asignado).

Desglosando la función objetivo, podemos ver los dos subproblemas mencionados inicialmente:

* **Autobuses asignados** : Se aplica el coste por kilómetro ()
* **Autobuses sin asignar** : Se aplica el coste por pasajero ()
  + Para conseguir el número de autobuses sin asignar, aprovechamos la variable binaria de asignación y, para cada autobús , utilizamos como “indicador de si ha sido asignado”. Por ello, utilizamos la negación de la variable binaria para solo aplicar el coste por pasajero() a aquellos buses que no han sido asignados.

Análisis de resultados

···