

# Optimizació GIA UPC

## Pràctica 2: Disseny de la red de sensorització per a l'Eixample de Barcelona



Figura 1 Ubicació de los 14 sensores seleccionados en el proyecto in4Mo (2013).

### Objetivo

Los encargados de gestionar el tráfico en Barcelona deben colocar en l'Eixample sensores de detección de vehículos. Los sensores de detección de que disponen se pueden ubicar únicamente en las intersecciones, y se considera que un sensor colocado en una intersección determinada será capaz de captar todos los vehículos que pasen por ella.

Para una cierta hora del día, se dispone de los caminos utilizados en la red y de cuantos vehículos utilizan cada uno de ellos (flujo). Un camino partirá de una intersección e irá recorriendo calles e intersecciones hasta llegar a su intersección destino. Se considerará que el camino está sensorizado si al menos dos de sus intersecciones lo están. Además, se considera que el flujo de un camino se ha detectado si dicho camino está sensorizado.

Además:

- 1) Debido a problemas técnicos, no todas las intersecciones pueden contar con un sensor, es decir, se tiene un conjunto de intersecciones "prohibidas" en las que no se puede instalar sensor.
- 2) Algunas de las intersecciones deben contar con un sensor sí o sí, con lo que serán consideradas intersecciones "fijas" que seguro deben aparecer en la solución.

El objetivo de los gestores es ubicar como máximo 15 sensores de vehículos, de forma que el flujo captado por los mismos durante ese período de tiempo, sea máximo y se cumplan todas las restricciones del problema.



- (a) Formula un **modelo de programación lineal entera** que solucione el problema de los gestores. Resuélvelo utilizando AMPL + un solver de programación matemática.
- (b) En una segunda fase del proyecto, se descubre que dichos sensores cuentan con una limitación física que consiste en que la detección falla si dos sensores se encuentran demasiado cerca, a menos de 300 metros, puesto que su señal de captación se solapa. Esto habrá que evitarlo, con lo que en la solución no deben aparecer sensores que estén a menos de dicha distancia. Para cada intersección se ha definido un subconjunto de intersecciones vecinas donde quedan incluidas las intersecciones que se encuentran en la misma calle a distancia menor o igual a 300 metros de la intersección considerada. Para construir dichos subconjuntos se han considerado las distancias entre intersecciones tomando el punto medio de las coordenadas de todos los nodos que forman dicha intersección.

Formula un **modelo de programación lineal entera** que solucione el problema de los gestores teniendo en cuenta esta restricción técnica. Resuélvelo utilizando AMPL + un solver de programación matemática.

### Datos

El problema se debe resolver para el conjunto de datos que aparece en el archivo *OPT23-24\_Datos práctica 2.txt* que acompaña este enunciado. Los datos proporcionados incluyen los caminos, las intersecciones, las intersecciones fijas, las intersecciones prohibidas, qué intersecciones tiene cada camino, el flujo de cada camino y dada una intersección, todas sus intersecciones vecinas (los que están a menos de 300m).

Por lo tanto, en encontraréis:

- *paths* es el conjunto de todos los identificadores de los caminos de la red
- *intersections* es el conjunto de identificadores de todas las intersecciones de la red
- *fixed* es el conjunto de los identificadores de las intersecciones que tienen que tener detector sí o sí en la solución.
- *prohibited* es el conjunto de los identificadores de las intersecciones que no pueden tener detector en la solución.
- *path\_intersections* es el conjunto que especifica qué intersecciones están en cada camino utilizando sus correspondientes identificadores
- *path\_flow* es el flujo que tiene cada camino
- *intersection\_neighborhood* es el conjunto que especifica para cada intersección quienes son sus intersecciones vecinas (es decir, las que están a menos de 300 m en este caso), utilizando sus correspondientes identificadores. [solo para el apartado (b)]

### Solvers

En el apartado (a), se puede utilizar cualquiera de los solvers incluidos en vuestra versión de estudiantes de AMPL, siempre que resuelva problemas de programación lineal entera (CPLEX, Gurobi, etc.).

En el apartado (b), superaréis el número de restricciones que permite vuestra licencia de AMPL (500 variables, 500 restricciones). En este caso, utilizad NEOS (<https://neos-server.org/neos/solvers/index.html>) para resolverlo. [Nota: el fichero .run no debe cargar ni el fichero .mod, ni el fichero .dat, simplemente los *displays* de las variables que se quieran mostrar después de la instrucción *solve*.]

### Entrega de la práctica

- El ejercicio se debe realizar en grupos de dos alumnos.
- Solo uno de los dos miembros del grupo debe colgar la práctica en Atenea, que debe ser un .zip o un .rar que contenga:
  - Ficheros .mod, .dat y .run en AMPL
  - Un fichero .pdf que contenga:
    - Nombre, apellidos y DNIs de los miembros del grupo.
    - Formulación matemática de los modelos de programación lineal entero propuestos para resolver ambos apartados.
    - Modelos y datos en AMPL (copiar el código en el entregable) de ambos apartados.
    - La descripción resumida de la implementación realizada, incluyendo los solvers que se utilicen para su resolución.
    - La solución que aporta la mejor localización de los sensores y el flujo total captado en cada uno de los apartados.