



FACULTAT D'INFORMÀTICA DE BARCELONA

GIA UPC
OPTIMITZACIÓ

Disseny d'una xarxa de sensorització

Alumnes :

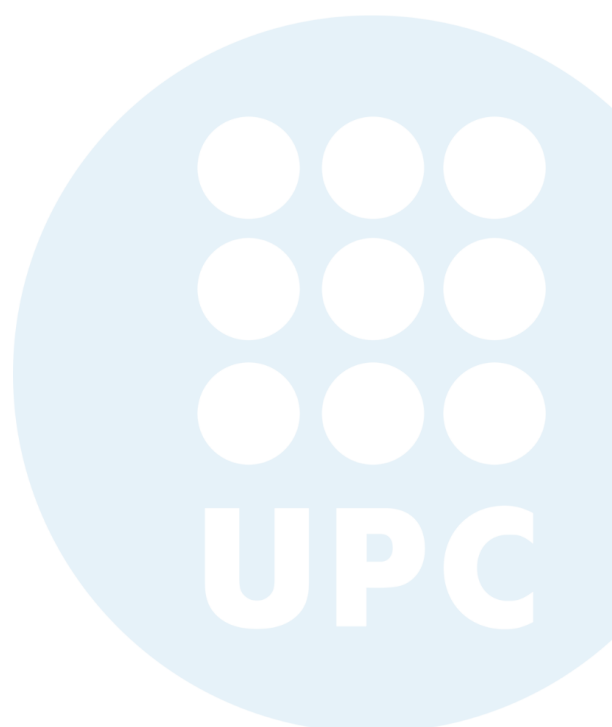
Abril Risso Matas, 45182567

Irene Pumares Benaiges, 54574822

Tutors :

Mari Paz Linares Herreros

April 12, 2024



Contents

1	Introducció	2
2	Implementació del fitxer .dat	3
3	Implementació del fitxer .mod	4
3.1	Conjunts	4
3.2	Paràmetres	4
3.3	Variables	5
3.4	Funció objectiu	5
3.5	Restriccions	5
4	Implementació del fitxer .run	7
5	Solució dels problemes	8
5.1	Apartat A	8
5.2	Apartat B	9
6	Conclusions	10

1 Introducció

Aquesta pràctica té com a objectiu el disseny i implementació d'una xarxa de sensorització per la detecció de vehicles a través de la col·locació de sensors en interseccions seleccionades, específicament en la millora de la gestió del trànsit a l'Eixample de Barcelona, una zona caracteritzada per la seva alta densitat de vehicles.

Per assolir aquest objectiu s'implementarà un model de programació lineal, utilitzant AMPL juntament amb solvers de programació matemàtica com CPLEX. Aquest enfocament ens permetrà resoldre aquest problema de la ubicació òptima dels sensors tenint en compte les restriccions específiques de l'enunciat.

Aquesta pràctica consta de dues parts. La primera part tracta de la configuració inicial de la xarxa de sensors, mentre que la segona introdueix una restricció addicional, l'impediment de la col·locació de dos sensors els quals tenen una distància menor a 300 metres.

2 Implementació del fitxer .dat

Aquest fitxer conté les dades detallades del problema, és a dir, proporciona valors concrets dels conjunts i dels paràmetres del problema que s'utilitzen per a resoldre'l.

Per adaptar les dades d'entrada al nostre model en AMPL, hem implementat un petit programa en Python el qual ens ha permès transformar el format de les dades originals, específicament les llistes d'interseccions que pertanyen a certs camins (INTERSECCIONS_CAMINS) i les parelles d'interseccions veïnes (INTERSECCIONS_VEINES).

Originalment, aquestes dades estaven en format de llistes simples, on cada línia representava una relació entre camins i interseccions, o entre dues interseccions veïnes, amb el format camí intersecció per a INTERSECCIONS_CAMINS i intersecció intersecció per a INTERSECCIONS_VEINES.

Per fer que aquestes dades fossin reconegudes per AMPL com elements units, el programa de python implementat canvia aquestes llistes convertint cada parella de valors en una tupla, és a dir, de la forma: (camí, intersecció) per a INTERSECCIONS_CAMINS i (intersecció1, intersecció2) per a INTERSECCIONS_VEINES.

3 Implementació del fitxer .mod

Aquest fitxer conté la definició del model matemàtic que volem utilitzar. El fitxer descriu els conjunts, paràmetres, variables, restriccions i la funció objectiu del problema d'optimització.

S'han creat dos fitxers d'aquest tipus, un per a cada part, *modelA.mod* i *modelB.mod*, respectivament. El model B és una extensió del model A, ja que conté un conjunt i una restricció més que l'A.

3.1 Conjunts

Els conjunts serveixen per organitzar i definir els elements clau del problema. Cada un representa una col·lecció d'objectes que s'utilitzen en el model.

INTERSECCIONS: Conté els identificadors de totes les interseccions de la xarxa de trànsit de Barcelona. És a dir, cada element representa una intersecció de la ciutat.

CAMINS: Conté els identificadors de tots els camins de la xarxa de trànsit de Barcelona. Cada un representa una ruta possible que podria fer un cotxe d'una intersecció a una altra.

INTERSECCIONS_PROHIBIDES i INTERSECCIONS_FIXES: Són dos subconjunts d'INTERSECCIONS', que contenen els identificadors de les interseccions prohibides i fixes, respectivament. El primer agrupa aquelles on no es pot col·locar un sensor i el segon aquelles on, pel contrari, és obligatori col·locar un sensor.

INTERSECCIONS_CAMINS: Conté parelles d'identificadors de camí i intersecció. Cada parella indica que una intersecció pertany a un camí determinat.

INTERSECCIONS_VEINES: Conté parelles d'identificadors de dues interseccions que són veïnes, és a dir, que es troben a menys de 300 metres l'una de l'altra. Són necessàries per al problema B, per a poder garantir que no es posen sensors en interseccions properes.

3.2 Paràmetres

Els paràmetres són valors constants que es defineixen per descriure les característiques del problema, les condicions del sistema o les dades específiques que s'utilitzen en el model.

f_c: Representa el flux de trànsit associat a cada camí de la xarxa de trànsit de Barcelona. Està indexat pel conjunt 'CAMINS', el que significa que per a cada camí, hi haurà un valor associat al seu flux. Inclou una restricció que indica que no pot ser un valor negatiu, ja que indica la quantitat de vehicles que circulen per un camí.

3.3 Variables

Les variables són els elements que es volen optimitzar o trobar solucions òptimes per al problema en qüestió. En aquest cas, representen les decisions que es volen prendre per a optimitzar la col·locació de sensors de trànsit.

s_i: Representa la presència o absència d'un sensor en cada intersecció. És binària, indicant que només pot prendre valors 0 i 1. Un valor de 1 indica que hi ha un sensor col·locat a la intersecció, mentre que un valor de 0 indica que no hi ha cap sensor.

s_c: Representa, per a cada camí, si aquest té com a mínim dues interseccions amb sensor, és a dir, si està sensoritzat o no. També és binària, per tant, tindrà un valor de 1 si és sensoritzat i prendrà un valor de 0 si no ho és.

3.4 Funció objectiu

La funció objectiu és una expressió matemàtica que indica el que es vol optimitzar en el problema. És a dir, assigna un valor numèric a cada possible solució, de manera que les solucions es poden comparar numèricament per a trobar la millor.

En aquest cas, l'objectiu del problema és maximitzar el fluxe total captat pels sensors. S'ha de tenir en compte que es considera que el fluxe d'un camí es detecta si aquest camí està sensoritzat. Per això, l'expressió de la funció objectiu és la següent: $\sum \{c \text{ in CAMINS}\} f_c[c] * s_c[c]$. El que es calcula és la suma del producte del flux de cada camí ($f_c[c]$) i la variable binària ($s_c[c]$), que indica si el camí està sensoritzat. És a dir, s'està sumant el flux de trànsit de tots els camins que estan sensoritzats.

3.5 Restriccions

Les restriccions són condicions que limiten les possibles solucions del problema. Aquestes poden reflexar condicions físiques, econòmiques, de seguretat o qualsevol tipus de limitació rellevant per el problema. És important definir-les bé per a garantir que la solució sigui factible i compleixi amb els requisits i limitacions del problema.

sensors_maxims: Estableix que la suma de les variables binàries associades a les interseccions (s_i) no pot superar 15. És a dir, com a màxim poden haver-hi 15 sensors en tota la xarxa de trànsit.

camí_sensoritzat: Assegura que cada camí té almenys dues interseccions amb sensor, ja que, com bé indica l'enunciat, perquè un camí estigui sensoritzat cal que dos de les seves interseccions tinguin sensor. Aquesta restricció calcula la suma de les variables (s_i , si la intersecció té sensor o no), per a totes les interseccions que pertanyen a un camí c (cada parella c,i dins del conjunt INTERSECCIONS_CAMINS). Aquesta suma representaria el nombre total de sensors col·locats en les interseccions de cada camí. Aquesta suma hauria de ser menor o igual a dos quan el camí estigui sensoritzat.

Per tant, quan $s_c[c] = 1$ (camí sensoritzat) la suma de les interseccions d'aquest camí haurà de ser menor o igual a 2: $\sum\{(c,i) \text{ in } \text{INTERSECCIONS_CAMINS}\} s_i[i] \geq 2 * s_c[c]$.

interseccions_prohibides: Estableix que les interseccions prohibides (del conjunt 'INTERSECCIONS_PROHIBIDES') no poden tenir sensors. Per tant, totes les variables binàries s_i associades a aquestes interseccions han de ser igual a 0.

interseccions_fixes: Semblant a l'anterior, estableix que les interseccions fixes (del conjunt 'INTERSECCIONS_FIXES') han de tenir sensors. Per tant, totes les variables binàries s_i associades a aquestes interseccions han de ser 1.

distancia_minima_sensor: Aquesta restricció només es troba a l'apartat B. Assegura que les parelles d'interseccions veïnes (definides al conjunt 'INTERSECCIONS_VEINES') no poden tenir sensors simultàneament. Per a garantir-ho, s'estableix que la suma de les variables binàries s_i associades a cada una de les interseccions que formen les parelles ha de ser més petita o igual a 1, ja que mai pot passar que les dos tinguin sensor (en cas que passés, la suma seria $1+1=2$, i no es compliria la restricció).

4 Implementació del fitxer .run

Aquest fitxer conté les instruccions que indiquen a AMPL com ha de resoldre i mostrar el resultat del problema definit als dos fitxers anteriors ('.mod' i '.dat').

En primer lloc, es restableix AMPL a l'estat inicial ('reset'), per a eliminar qualsevol definició o dades anteriors. Aleshores es carrega el fitxer que conté el model ('modelA.mod' o 'modelB.mod'). Tot seguit, carrega també el fitxer de dades ('dadesA.dat' o 'dadesB.dat') que conté les dades del problema. S'especifica el solucionador a utilitzar. En aquest cas, s'ha triat CPLEX per a resoldre el model.

Tot i que en enunciat proposava la utilització d'un altre solver (NEOS) per la segona part de la pràctica, a nosaltres ens ha funcionat correctament el CPLEX. Tot i així si a l'executar-se en el ordinador de l'usuari no funciona correctament, es podria provar amb NEOS (recomanat a l'enunciat).

Un cop tot definit, aleshores ja es pot executar la comanda ('solve') per a la resolució del problema.

Un cop resolt el problema, s'imprimeixen els resultats i les dades que es volen. En aquest cas, apareixen en pantalla els identificadors de les interseccions que han de tenir sensor i finalment el flux total aconseguit amb aquesta distribució de sensors per la ciutat.

En resum, el fitxer '.run' proporciona una seqüència d'instruccions que prepara l'entorn, carrega el model i les dades, tria el solucionador, resol el problema i mostra els resultats. D'aquesta manera s'automatitza el procés d'execució del model.

5 Solució dels problemes

5.1 Apartat A

El fluxe total que s'aconsegueix és de 350.734.

Identificadors de les interseccions amb sensor
5
30
78
20349
41633
41967
41977
44494
44609
44628
45173
45481
45555
45787
49180

Table 1: Identificadors de les interseccions que han de tenir sensor

5.2 Apartat B

El fluxe total que s'aconsegueix és de 350.178.

Identificadors de les interseccions amb sensor
30
78
41633
41653
41970
44494
44522
44609
44628
45173
45481
45555
45787
49180
54839

Table 2: Identificadors de les interseccions que han de tenir sensor

6 Conclusions

Com es pot observar als resultats obtinguts, podem veure una diferència de fluxe del primer apartat respecte el segon degut a la nova restricció afegida a la segona part.

Al primer apartat, s'ha aconseguit un fluxe total de 350.734, utilitzant un total de 15 sensors col·locats en interseccions específicament seleccionades. Aquest resultat demostra la capacitat del model de programació lineal per optimitzar la col·locació dels sensors i maximitzar la detecció de vehicles.

En canvi, a la segona part, on s'ha afegit la restricció de que dos interseccions les quals estan a una distància menor a 300 metres, no poden tenir sensors alhora, per evitar la interferència de les senyals. El fluxe total ha sigut una mica inferior a l'anterior amb un valor de 350.178.

Aquest resultat ens indica que aquesta restricció addicional no ha limitat considerablement l'eficàcia de la xarxa de sensors. A més, indica que al primer apartat el model ja ha tendit a triar interseccions no pròximes per la col·locació dels sensors.