



**UNIVERSIDAD DE CASTILLA LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

PROYECTO FIN DE CARRERA

Localización de sensores de tráfico

Autor: Alberto Martín Guerrero
Director: Ricardo García Ródenas
Coordinador: Macario Polo Usaola

Septiembre 2006

TRIBUNAL

Presidente:

Vocal:

Secretario:

FECHA DE LA DEFENSA:

CALIFICACIÓN

PRESIDENTE

VOCAL

SECRETARIO

FDO:

FDO:

FDO:

0.1. Agradecimientos

Este proyecto fin de carrera no puede considerarse un proyecto tipo de una escuela de informática. La primera dificultad radica en la materia de estudio: estudios de tráfico. La segunda, la metodología, ya que es diferente a la empleada en las cualquiera de las múltiples prácticas que ha tenido que hacer a lo largo de la carrera. La tercera deriva de los requerimientos matemáticos necesarios, de los que algunos se tratan en los primeros cursos de la carrera como base matemática y otros son totalmente novedosos para mí. Por esto, a la hora de hacer un proyecto de este tipo no son pocas las personas a las que es necesario agradecer su apoyo.

En primer lugar quiero expresar mi agradecimiento a mi familia, de la cual siempre he tenido todo el respaldo que he necesitado y el estímulo necesario para seguir adelante en cada objetivo que me he planteado.

A mis amigos, con los que siempre puedo contar aún cuando el estado de ánimo no está en su mejor momento.

A mis compañeros del laboratorio, donde llevé a cabo la realización del proyecto y que hicieron que todo este tiempo fuera agradable y que resolvieron cualquier duda que tuve.

A la Junta de Comunidades de Castilla La Mancha por la financiación del proyecto de investigación "*Desarrollo, implementación y validación de una metodología para la estimación de matrices de viajes origen-destino a partir de volúmenes de tráfico y matrices desactualizadas. Estudios de casos en ciudades de pequeño y mediano tamaño*", con referencia 2005-COB-701, que me ha permitido disfrutar de un contrato laboral en el que me he podido dedicar a tiempo completo al desarrollo de mi PFC.

A la Universidad de Castilla la Mancha, a través de la Dirección de la Escuela Superior de Informática, por poner a nuestra disposición lo necesario para trabajar en el proyecto.

A Luis Jiménez Linares, director del grupo ORETO en la Escuela Superior de Informática de Ciudad Real, que cedió un espacio en el laboratorio del grupo para desarrollar el proyecto.

A Doroteo Verastegui Rayo, cuya tesis doctoral me orientó en la forma de estructurar la documentación del proyecto.

A Macario Polo Usaola, coordinador del PFC, cuya disposición desinteresada hizo posible este proyecto.

Por último, a Ricardo García Ródenas, director del PFC, por su paciencia en los inicios al

trabajar con alguien para nada acostumbrado a este tipo de proyecto y por todo el tiempo que me ha dedicado para su correcto desarrollo.

0.2. Resumen

Los estudios de movilidad se basan en la existencia de una matriz de viajes origen-destino que sintetiza el patrón de movilidad. Se han desarrollado multitud de metodologías para la estimación de matrices origen-destino. A grosso modo, una taxonomía de estos modelos constaría de tres categorías: i) métodos sintéticos, basados en modelos matemáticos, entrópicos, etc. ii) métodos de estimación basados en encuestas y iii) métodos basados en medidas de flujos en la red (aforos).

En la actualidad la desventaja de los modelos sintéticos es su mala calidad de estimación aunque su obtención es rápida y económica. Los métodos basados en encuesta son los más fiables, pero consumen grandes recursos económicos, humanos y temporales. Los métodos basados en aforos son los que en las últimas dos décadas están teniendo una mayor relevancia, ya que los aforos se pueden recolectar automáticamente, con un bajo coste y las estimaciones obtenidas tienen una calidad aceptable.

En la literatura científica están estudiados considerablemente los modelos de estimación basados en aforos, pero el plan de aforamiento de la red está teniendo una menor atención. Este plan recoge cuántos arcos aforar y cuales. Este PFC se centra en el estudio de ciertos modelos de localización de sensores de tráfico, para determinar el plan de aforamiento.

Los objetivos perseguidos han sido: i) Evaluación computacional de los modelos TCL-P1 y TCL-P2, ii) Fiabilidad de las estimaciones derivadas de los modelos TCL-P1 y TCL-P2 y iii) Algoritmos heurísticos para la resolución del TCL-P1 y TCL-P2. Para abordar el estudio se han resuelto varios problemas. En primer lugar, se ha resuelto el problema de asignación de tráfico (TAP) para las redes disponibles, junto a varias perturbaciones de los datos de demanda de dichas redes. Esta perturbación ha permitido obtener nuevos flujos. Sobre los flujos en equilibrio derivados de la resolución del TAP, se han aplicado los diferentes modelos localización. Con los aforos obtenidos por dichos modelos se han estimado mediante métodos "proporcionales" las matrices origen-destino correspondientes. Para averiguar la fiabilidad de esta la estimación se ha comparado la matriz origen-destino verdadera, que es conocida, con la matriz estimada.

Índice general

0.1. Agradecimientos	3
0.2. Resumen	5
1. Introducción	8
2. Objetivos del proyecto	13
3. Antecedentes	14
3.1. Problema de asignación de tráfico	14
3.1.1. Formulación matemática del modelo de optimización	15
3.2. Problema de estimación de matrices origen-destino (DAP)	18
3.2.1. Modelos de asignación empleados en el DAP	20
3.3. Modelos de localización de sensores (P1 y P2). Otras extensiones	23
3.3.1. Modelo de cubrimiento TCL-P1	24
3.3.2. Modelo de máxima interceptación de flujo neto TCL-P2	25
4. Método de trabajo	27
4.1. Metodología empleada para el objetivo 1	28
4.1.1. Fase de generación de datos. Resolución del problema de asignación de tráfico	30
4.1.2. Evaluación computacional de los problemas de localización (TCL)	46
4.2. Metodología empleada para el objetivo 2	55
4.2.1. Fase de generación de datos	55
4.2.2. Resolución del problema de estimación de matrices O-D	55

<i>ÍNDICE GENERAL</i>	7
5. Resultados	65
5.1. Descripción de los datos	65
5.2. Problema de asignación de tráfico (TAP-DSD)	65
5.3. Objetivo 1: Evaluación computacional de TCL	68
5.4. Objetivo 2: Evaluación de la fiabilidad de las estimaciones de matrices O-D (DAP)	68
6. Conclusiones	74
7. Anexo. Contenido del disco adjunto	78
7.1. Carpeta Redes	78
7.2. Carpeta Localizacion	81
7.3. Carpeta Estimacion	84

Capítulo 1

Introducción

Los sistemas de transporte tienen una gran complejidad en grandes ciudades. Esto motiva la creación de herramientas analíticas para facilitar un conocimiento del sistema y poder hacer predicciones por parte del planificador del sistema de transporte. Dichas herramientas hacen uso de modelos matemáticos. Estos modelos han sido de gran ayuda para identificar soluciones o políticas para solucionar los problemas del sistema y lograr los objetivos propuestos para él, siendo de gran ayuda también para los procesos de planificación y de toma de decisiones. El objetivo de los análisis es obtener una descripción macroscópica del comportamiento de los usuarios de la red de transporte.

La planificación del transporte es un proceso que abarca las siguientes fases:

- **Recopilación de los datos:** Obtener todos los datos relevantes del sistema de transporte a estudio, como pueden ser su infraestructura y su uso, las demandas de transporte en la red, factores de utilización del terreno, estructura de la red de transporte, tipo de empleo y distribución de la renta, etc.
- **Análisis y ajuste de modelos:** Con los datos anteriores se generan una serie de modelos clasificados en:
 - **Modelos de generación/atracción de tráfico:** Justifican los viajes que se generan o que concluyen en una zona de la red en función de sus variables socioeconómicas. Se empieza dividiendo el área de la red por zonas para trabajar en él, con una codificación de la red de transporte y consiguiendo datos para el estudio. Dichos datos

reflejan la actividad económica y demográfica por zonas y se emplean para estimar tanto el número de viajes que se generan, como el que concluyen a esa zona. Por ejemplo, durante las horas que comprenden una mañana laborable, las zonas residenciales generalmente generan viajes y las zonas industriales o comerciales atraen viajes.

Una vez realizada esta fase se obtiene una modelización de la red de transporte de la ciudad representándola como un grafo dirigido. Los arcos del grafo son las vías de la ciudad, las cuales pueden ser de sentido único o de ambos sentidos. Los vértices son intersecciones de estas vías donde los usuarios pueden elegir por cual seguir. La relación arco-vía no es exacta, en algunos casos una vía debe representar usando un conjunto de arcos.

- **Modelos de distribución zonal:** En esta fase se busca conseguir una distribución de los viajes de una zona sobre el espacio de la red. Esta información queda recogida en la llamada matriz de viajes Origen-Destino (O-D), que contiene pares ordenados sobre el conjunto de nodos de la red de transporte y la demanda de viajes (que se considerará fija inicialmente). Estos pares indican la cantidad de viajes entre dos puntos de la red.
- **Modelos de distribución modal:** Cuando las posibilidades de transporte en una red abarcan varios modos, como pueden ser privado, público o desplazarse caminando, es necesario recogerlas en una matriz origen-destino por cada uno de los modos de transporte. A partir de la actual matriz origen-destino se pueden hacer previsiones en cuanto al aumento/disminución de las necesidades de transporte de la red en un futuro
- **Modelos de asignación:** En esta fase se averigua el itinerario de los usuarios de la red de transporte. Para ello la matriz obtenida en la etapa anterior se asigna sobre distintos escenarios para la futura red de transporte y se analiza qué se produce.

Hoy en día estas fases no se aplican de forma secuencial. Hay otros métodos que abarcan varias de estas etapas simultáneamente. Aunque este esquema resulta útil para describir los modelos o para su comparación entre modelos alternativos.

En los estudios de ingeniería de transporte se pone de manifiesto la importancia de la matriz O-D, la cual es la información básica, además de otros parámetros, para conocer el patrón de movilidad.

Estas matrices se han venido estimando de tres maneras:

1. **Estimación directa:** Consiste en la realización de una serie de encuestas a los usuarios de la red. Conlleva un coste importante dependiendo de lo amplia y detallada que se realice. La fiabilidad de los resultados depende de la sinceridad y exactitud de las respuestas de los usuarios. Es significativo el tiempo que se necesita para hacer una estimación de este tipo.
2. **Estimación por modelos de demanda:** Para obtener estimaciones de los flujos O-D se aplican modelos sintéticos de demanda de viajes. Dichos modelos dependen de un conjunto de parámetros que pueden calibrarse específicamente por áreas de estudio o transferidos desde áreas similares empleadas en otros estudios de movilidad.
3. **Estimación mediante volúmenes de tráfico:** Utiliza los sensores de tráfico que se colocan en determinados arcos con el objetivo de actualizar una matriz O-D previa (desactualizada u obtenida por medio de otro procedimiento). En esta alternativa se han centrado gran parte de los esfuerzos en los últimos treinta y cinco años, lo que ha conllevado a la aparición de innumerables trabajos desde el trabajo pionero de [Low1972].

La metodología en la que se basa exige un conjunto de flujos de tráfico o aforos y una matriz O-D desactualizada (obtenida en un estudio anterior u obtenida utilizando cualquiera de los otros dos esquemas de estimación). Los aforos de tráfico es una información que se consigue automáticamente en unos puntos de los arcos de la red de transporte, por lo tanto se encuentran disponibles en áreas urbanas con un coste bajo. En esos puntos se colocan los sensores o cordones de tráfico, que son unos dispositivos que miden el número de vehículos que pasan por un determinado punto en un determinado intervalo de tiempo.

Esta aforación nos indica el flujo que hay en un determinado punto de la red. Al eliminar el sistema de encuestas para estimar la matriz O-D nos deshacemos de los problemas que conlleva, así como de la manipulación de estos datos. Por contra se presentan otros problemas de tipo matemático, como problemas de gran escala computacionalmente más

costosos de resolver por un PC convencional, malas propiedades matemáticas, etc.

Este proyecto fin de carrera se sitúa en esta última alternativa de estimación, la estimación mediante volúmenes de tráfico. En la literatura científica se encuentra poco desarrollado el aspecto de elaborar un plan de aforamiento de la red. Este proyecto intenta ahondar en esta cuestión mostrando la dependencia de los resultados al plan de aforamiento.

En general, la calidad de las matrices estimadas depende en gran medida de la fiabilidad de los datos de entrada (volúmenes de tráfico y matriz de referencia) y éstos del número y localización de los sensores de tráfico. Habitualmente, el impacto de los errores en los datos de entrada ha sido investigado a través de simulaciones numéricas con diferente grado de eficacia del modelo de entrada.

El problema de interés del PFC corresponde con el problema de *localización de sensores de tráfico* de modo que se reduzcan los errores en la estimación de la matriz origen-destino. La mayoría de los modelos de localización han sido desarrollados para determinar la localización óptima de sensores de tráfico para cubrir tanta demanda como sea posible, pero sin una referencia a la matriz O-D dada.

En [Hogson, M.J. 1990] se discutió el problema de la localización de sensores de modo que capturase el mayor flujo en una red de transporte. Asumió que todos los flujos entre cada par O-D escogen el camino mínimo. El estudio en [Berman1992], basado en el trabajo en [Hogson, M.J. 1990], presenta un modelo que determina la localización óptima de un número de sensores de tráfico por medio de la máxima interceptación posible de flujo. Posteriormente en [Berman1995] se realizó un estudio del problema de localización sin el requisito del conocimiento del flujo en los caminos. La información sobre flujo requerida en el modelo es la proporción de flujo desde un nodo dado a sus adyacentes y la proporción de demandas cuyo inicio es cualquier nodo origen de la red.

Los modelos de localización de sensores intentan introducir una forma sistemática para seleccionar un conjunto de arcos cuyo flujo pueda ser observado.

En la literatura científica se han empleado cuatro reglas básicas para la localización de los sensores de tráfico:

- Regla de cubrimiento O-D (TCL-P1): los sensores de tráfico deben ser localizados de tal manera que una cierta proporción de flujo entre cualquier par origen-destino pueda ser

observado.

- Regla del máximo flujo neto: Un número dado de sensores se debe colocar en un subconjunto de arcos que maximice el flujo neto interceptado. Esto es si una determinada fracción de flujo es captada más de una vez ésta es tomada en cuenta una única vez.
- Regla de máxima interceptación de flujo: bajo un cierto número de aforos, los aforos elegidos interceptan tanto flujo como sea posible.
- Regla de independencia lineal de los arcos: los sensores de tráfico deberían localizarse de modo que el flujo recogido en todos ellos sea linealmente independiente.

Estas reglas pretenden que el plan de aforación sea lo más consistente posible y que la información obtenida por medio de dicha aforación sea lo más rica posible.

Para una red de tráfico y una matriz O-D de referencia dadas, los modelos de programación lineal y los procedimientos de solución, están desarrollados para determinar qué arcos aforar bajo la reglas indicadas anteriormente y que satisfacen ciertas restricciones.

Los modelos de localización de sensores desarrollados en la literatura con los que se va a trabajar en este PFC son el TCL-P1 (Traffic count location problem 1) y TCL-P2 (Traffic count location problem 2), ambos vistos en [Yang et al.1998]. El primero, también llamado de cubrimiento, busca el número mínimo de sensores para cubrir toda la demanda de la red. El segundo modelo, también llamado de máxima interceptación de flujo neto, parte del número mínimo de sensores, l , necesarios para cubrir todos los pares de la red. Este modelo parte de que se desean localizar exactamente l sensores de modo que: i) se cubran todos los pares de la red y ii) el flujo neto interceptado sea máximo.

Capítulo 2

Objetivos del proyecto

El objetivo general del proyecto es el desarrollo y evaluación computacional de una metodología para la localización de sensores de tráfico. Este objetivo se compone a su vez de los subobjetivos que se indican a continuación:

- Evaluación computacional de los modelos TCL-P1 y TCL-P2.

Con este objetivo se persigue averiguar si los modelos desarrollados en la literatura son aplicables a problemas reales de grandes dimensiones.

- Fiabilidad de las estimaciones derivadas de los modelos TCL-P1 y TCL-P2.

Con este objetivo se busca obtener alguna medida de la calidad de la estimación de la matriz origen-destino que puede esperarse.

- Desarrollo e implementación de algoritmos heurísticos para el TCL-P1 y TCL-P2.

Este tercer objetivo queda condicionado al resultado del primero. En el caso de que los algoritmos exactos de programación lineal entera no puedan resolver los anteriores modelos en un tiempo razonable, se deberá recurrir a estrategias heurísticas que permitan obtener una "buena" localización en un tiempo de cómputo compatible con los estudios de movilidad.

Capítulo 3

Antecedentes

3.1. Problema de asignación de tráfico

El problema de asignación de tráfico describe el comportamiento de los usuarios de una red de tráfico urbana congestionada. Los modelos de asignación de tráfico es un proceso que asigna una matriz O-D a un conjunto de caminos de la red de tráfico, siguiendo un conjunto de reglas y principios. El resultado de este proceso es un conjunto de flujos en los arcos y caminos de la red.

Los datos básicos para este problema son la nombrada matriz O-D, definidora de la demanda, y un grafo que representa la red de tráfico. Para cada arco existe un función de coste que mide el tiempo en atravesar el arcos en función del flujo de la red.

En la asignación de tráfico se supone que el usuario es *racional*, en el sentido de que elige la ruta que a su juicio es la de mejor coste. Generalmente, este coste está asociado al tiempo que emplea cada usuario en su desplazamiento. Como hay diferentes enfoques respecto a la percepción que tiene el usuario de las condiciones de la red, hay diferentes modelos de asignación adecuados al caso en cuestión.

En este PFC se va a tratar con un modelo *estático* de asignación de tráfico. Este tipo de modelos asumen que existe una situación estacionaria en un determinado intervalos de tiempo.

El otro tipo de modelos de asignación son los *dinámicos*, los cuales recogen variaciones de la demanda.

Los modelos estáticos se pueden clasificar en dos grandes grupos. Estos grupos están condi-

cionados por la forma en la que el usuario percibe la información de la red. En el modelo *determinista* todos los usuarios de la red perciben el coste de la misma forma y saben el coste de todas las rutas (también llamado información perfecta) para tomar sus decisiones. En el modelo *estocástico* los usuarios perciben los costes aleatoriamente, eligen su ruta de acuerdo a su propia percepción. En este PFC se tratan los modelos deterministas con *coste separable*. Esto quiere decir que el coste en un arco sólo es dependiente del volumen del tráfico en dicho arco y no de los demás arcos (caso de costes *asimétricos*).

Estos modelos contienen a su vez dos grandes grupos. Estos grupos son los modelos *sin congestión* y *con congestión*. Los primeros también se denominan modelos de asignación *todo-nada*, porque emplean los caminos mínimos para asignar todo el flujo de un par de demanda. Los segundos emplean modelos de asignación *en equilibrio*. Se basan en que cuando se alcanza el equilibrio en la red, todos los caminos de un par de demanda tienen un coste idéntico.

Esta condición de equilibrio de la red se denomina Primer principio de Wardrop (ver [Wardrop1952]), el cual establece que:

Los costes de viaje de todas las rutas usadas en el equilibrio son iguales y menores que en los que incurriría un único vehículo que utilizara una ruta distinta a la que utiliza en esos momentos.

Esto quiere decir que ningún usuario de la red puede mejorar su coste de viaje cambiando unilateralmente su ruta. Este principio asume que los usuarios tienen información perfecta y perciben el coste de la misma forma, características que definen los modelos de asignación deterministas. Se denominan "flujos óptimos para el usuario" a aquellos flujos asignados a la red que satisfacen este principio.

El Segundo principio de Wardrop (ver [Wardrop1952]) establece lo siguiente:

Los usuarios eligen la ruta de modo que se minimice el tiempo total de transporte en el red.

De este modo se puede decir que el tiempo de viaje promedio es mínimo. Se denominan "flujos óptimos del sistema" a los flujos asignados a la red que satisfacen este principio.

3.1.1. Formulación matemática del modelo de optimización

Un problema de optimización se caracteriza por tres elementos: *las variables del problema*, que definen el conjunto de decisiones, la *función objetivo*, que evalúa el coste o el beneficio

de la decisión y el *conjunto de soluciones*, que determina las decisiones válidas que pueden llevarse a cabo. Más formalmente, sea $f : S \mapsto \mathfrak{R}$, una función; X , un subconjunto de S que se denomina *conjunto factible de soluciones*; entonces el problema de optimización (en el caso de minimizar) se formula como:

$$\begin{aligned} & \text{Minimizar } f(x) \\ & \text{sujeto a : } x \in X \end{aligned}$$

En este caso la solución es demasiado general, puesto que el conjunto S puede ser cualquier conjunto que satisfaga las restricciones del problema. Por esto, la forma de proceder habitual es adecuarlo a nuestras condiciones particulares. Para ello se le exige a la función f y al conjunto X propiedades suficientemente generales para poder utilizarse en nuestro estudio y suficientemente fuertes para que los resultados sean de interés. Para el conjunto X , se exige que sea cerrado y convexo. Además habrá propiedades aún más fuertes, como la de ser un conjunto definido mediante restricciones lineales. En el caso de la función f hay dos tipos: primero, se exigen condiciones de regularidad local para poder caracterizar los extremos locales del problema y segundo, se exigen propiedades acerca del comportamiento global de la función que permitan garantizar que dichos extremos locales son también extremos globales.

Para el primer grupo, las propiedades más empleadas son la F-diferenciabilidad, la teoría de los *subdiferenciales* (subgradientes) vista en [Rockefella1970] para funciones convexas y la teoría de los *gradientes generalizados* para funciones localmente lipschitzianas vistas en [Clarke1990]. Las propiedades globales más empleadas son la *convexidad* o una generalización de este concepto (bajo la hipótesis de F-diferenciabilidad) denominado *pseudoconvexidad*, visto en [Bazaraa et al.1993]. Un tipo de modelo abordado es el *problema de optimización convexa diferenciable* (CDP). Mas formalmente,

$$\text{Minimizar } f(x), x \in X \tag{3.1}$$

donde la función $f : \mathfrak{R}^n \mapsto \mathfrak{R}$ es continuamente diferenciable, pseudoconvexa (por ejemplo cuando es convexa) y el conjunto X es un subconjunto convexo de \mathfrak{R}^n .

Formulación matemática del problema de asignación de tráfico

Se empleará un modelo matemático CDP para definir el TAP. Para realizar la formulación matemática del problema de asignación de tráfico se comienza con la formulación de las restricciones X . Para describir los requisitos de factibilidad de los flujos se toma v como los flujos en los arcos y K_i como el conjunto de rutas para el par O-D i :

- La primera restricción es que debe satisfacerse la demanda para cada par O-D i :

$$\sum_{k \in K_i} h_k = g_i, \forall i \in I. \quad (3.2)$$

- Los flujos en los caminos, h_k , no pueden ser negativos:

$$h_k \geq 0, \forall k \in K. \quad (3.3)$$

- Los flujos en los arcos y los flujos en las rutas se relacionan de la siguiente manera:

$$\sum_{i \in I} \sum_{k \in K_i} \delta_{ak} h_k = v_a, \forall a \in \mathcal{A}. \quad (3.4)$$

donde $\delta_{ak} = 1$ si la ruta k pasa por el arco a y cero en caso contrario. De esta restricción se deduce que el flujo en un arco a es la suma del flujo de todos los caminos que pasan por dicho arco.

La solución del siguiente problema de optimización caracteriza la situación de equilibrio derivada del primer principio de Wardrop (Problema TAP):

$$\begin{aligned} \text{Minimizar} \quad & Z = \sum_{a \in \mathcal{A}} \int_0^{v_a} c_a(x) dx \\ \text{sujeto a :} \quad & \sum_{k \in K_i} h_k = g_i, \forall i \in I, \\ & \sum_{i \in I} \sum_{k \in K_i} \delta_{ak} h_k = v_a, \forall a \in \mathcal{A}, \\ & h_k \geq 0, \forall k \in K_i, \forall i \in I. \end{aligned}$$

La situación de equilibrio derivada del segundo principio de Wardrop (para costes separables y crecientes) se puede caracterizar como solución del siguiente modelo de optimización (Problema TAP-SE):

$$\begin{aligned}
 \text{Minimizar} \quad & Z = \sum_{a \in \mathcal{A}} c_a(v_a)v_a \\
 \text{sujeto a :} \quad & \sum_{k \in K_i} h_k = g_i, \quad \forall i \in I, \\
 & \sum_{i \in I} \sum_{k \in K_i} \delta_{ak} h_k = v_a, \quad \forall a \in \mathcal{A}, \\
 & h_k \geq 0, \quad \forall k \in K_i, \quad \forall i \in I.
 \end{aligned}$$

En [Bell1983] se formula un modelo que trata de preservar la estructura de la matriz de referencia, añadiendo una nueva restricción y modificando del sistema de la siguiente manera:

$$\text{Minimizar} \quad \left(\frac{1}{2}\right) \sum_{i \in I} \frac{(g_i - \hat{g}_i)^2}{\hat{g}_i} \quad (3.5)$$

$$\text{sujeto a :} \quad \hat{v}_a = \sum_{i \in I} p_i^a g_i ; a \in \hat{A} \quad (3.6)$$

$$g_i \geq 0 \quad (3.7)$$

para todo arco a aforado (con un sensor de tráfico que mide su flujo). Donde p_i^a es la matriz de proporciones que relaciona el número de viajes del par i con el flujo en el arco a , \hat{v}_a es el flujo observado en los arcos aforados, \hat{g}_i es la matriz de referencia desactualizada y conocida; y g_i es la matriz estimada en la actualidad.

3.2. Problema de estimación de matrices origen-destino (DAP)

En esta sección se describen de forma unificada diversos modelos descritos en la literatura para la estimación de matrices O-D. Se ha seguido el enfoque desarrollado en [Verastegui2006].

Ante la necesidad de obtener rápida y económicamente matrices de viajes, diversos autores comenzaron en la década de los setenta con aproximaciones teóricas a la construcción de matrices de viaje a partir de información que no fuera difícil de conseguir, como volúmenes de tráfico en los arcos de la red. Como se dijo en la introducción de este proyecto, en la mayoría de

ciudades los datos se recogían por medio de sensores de tráfico. En [O’Neill1987] se recogen las aproximaciones existentes para sintetizar las matrices de viajes a partir de los sensores de tráfico colocados en los arcos y lo clasifica en dos clases:

- *Técnicas de calibración de parámetros.* Estas técnicas usan el análisis de regresión, lineal o no lineal, para construir *modelos de demanda*, y poder estimar las entradas de la matriz de viajes. En [Willunsen1978], [Nguyen and Dupuis1984] y [O’Neill1987] se revisa la literatura disponible y llegan a la conclusión de estas técnicas requieren datos de zonas para la calibración de los parámetros de los modelos de demanda, y por lo tanto son de limitado uso a nivel práctico, no sólo porque dichos datos son los más difíciles de recoger, sino también porque se quedan anticuado relativamente pronto.
- *Métodos de estimación de matrices basados en conteos.* Estos métodos sólo requieren conteos de tráfico en los arcos de la red y una información a priori sobre la estructura de la matriz de transporte.

Generalmente, estos métodos utilizan información que puede estar disponible en términos de una *matriz de referencia* O-D, \hat{g} , y/o en el número de viajes generados o atraídos en varias zonas. La *matriz de referencia* O-D se puede obtener haciendo una encuesta sobre una muestra o sobre una matriz desactualizada. Esta matriz en ocasiones también se le denomina *target matrix*.

Los distintos métodos desarrollados en la literatura para estimar matrices O-D a partir de volúmenes de tráfico \hat{v} (DAP) se pueden obtener definiendo los elementos siguientes:

- i. Un modelo de asignación.
- ii. Un método de estimación.
- iii. La información disponible.

Una forma unificada para formular el problema es (Problema DAP):

$$\text{Minimizar } F(g) = \eta_1 F_1(g, \hat{g}) + \eta_2 F_2(v, \hat{v})$$

$$\begin{aligned} \text{suje}to\ a : \quad & v \in A(g) \\ & g(\theta) \in G \end{aligned}$$

done A representa algún modelo de asignación, F_1 y F_2 determinan el método de estimación y los parámetros η_1 y η_2 reflejan la confianza relativa en los datos disponibles \hat{v} y \hat{g} . La restricción $g(\theta) \in G$ representa la región factible para el conjunto de matrices O-D dependientes del conjunto de parámetros θ a estimar. En nuestro caso, el conjunto de parámetros a estimar es la propia matriz g , o lo que es lo mismo, $\theta = g$. Las restricciones habituales en estos problemas son la no negatividad de las entradas en la matriz O-D (no tiene sentido un número de viajes negativo) y la información de las zonas disponible.

3.2.1. Modelos de asignación empleados en el DAP

La estimación estática que se trata aquí se refiere a conteos de tráfico para un sólo período determinado, como puede ser la "hora punta". Un detalle clave a la hora hacer esta estimación es la técnica de asignación empleada, que en el modelo genérico DAP viene definido sintéticamente por $A(g)$.

En los arcos de la red de transporte puede existir congestión y esto se puede modelar por funciones de coste. Estas funciones de coste relacionan el volumen de tráfico en el arco y el coste (o el tiempo) para recorrerlo. Los efectos de la congestión es una característica que condiciona los modelos para la asignación de la matriz O-D. Cada modelo asume que la congestión se puede tratar exógenamente (por asignación proporcional) o endógenamente (por asignación en equilibrio). El primero es más adecuado para bajos niveles de congestión o para buenas predicciones de los niveles de congestión. El otro caso es más apropiado en condiciones de congestión significativa. La mayoría de los modelos se basan en costes separables en los arcos, que son los que se tratan en este proyecto, pero también se han manejado casos más generales donde los flujos en arcos vecinos afectan al coste de un arco, como los modelos con desigualdades variacionales.

Asignación proporcional

En este caso asumimos que los volúmenes de tráfico y las proporciones de tráfico p_i^a son independientes el uno del otro. Los volúmenes en los arcos son proporcionales a los flujos O-D (g_i). La proporción de viajeros que escogen una ruta depende de las características de la ruta y del viajero, pero no dependerá de la congestión de la red. Los valores de p_i^a pueden determinarse antes de que la estimación de la matriz O-D se realice. Puede usarse un método de estimación de *todo-nada*, en la que la asignación de tráfico es efectuada por las rutas de mínimo coste para los pares O-D.

P es la matriz de elementos p_i^a , representa la *proporción de viajes* del par i que pasa a través del arco a en tanto por 1.

$$0 \leq p_i^a \leq 1 \quad (3.8)$$

Si se toma un arco dado a , la suma de todos los g_i (Flujos O-D del par i) que atraviesan dicho arco es el volumen o flujo del arco, v_a . En las ecuaciones fundamentales, que relacionan los volúmenes en los arcos y los flujos O-D, se incluye, a veces, un factor R de ocupación de los vehículos (que generalmente tiene un valor 1.25)

$$v_a = \frac{1}{R} \sum_i p_i^a g_i, \quad a \in \mathcal{A} \quad (3.9)$$

donde \mathcal{A} es el conjunto de arcos de la red de transporte. Este grupo de restricciones concretiza la restricción conceptual $v \in A(g)$ del DAP.

Los primeros modelos desarrollados partían de volúmenes de tráfico \hat{v}_a observados en un subconjunto de arcos $\hat{\mathcal{A}}$, unas proporciones p_i^a y la matriz g se determinaba resolviendo el sistema de ecuaciones (3.9) planteado sobre los arcos aforados. Este sistema queda de la siguiente manera:

$$\hat{v}_a = \frac{1}{R} \sum_i p_i^a g_i, \quad a \in \hat{\mathcal{A}} \quad (3.10)$$

En la mayoría de los casos hay muchas matrices O-D que reproducen los conteos de tráfico observados, o lo que es lo mismo, este sistema de ecuaciones es indeterminado y la solución no es única.

El número de pares O-D suele exceder, notablemente, el número de arcos para los que se

han recogido los conteos de tráfico. El problema que se plantea es escoger la *mejor matriz* que reproduce los conteos de tráfico recogidos.

Con el fin de evitar esta sobrespecificación de los parámetros a estimar (las entradas de la matriz O-D) se necesitan más datos, por medio de matrices desactualizadas, y/o realizar determinadas suposiciones sobre el comportamiento de los viajeros para que la solución del problema sea única. En la práctica, lo que se hace es elegir una función F_1 como una función de máxima entropía o de mínima información, añadiendo una matriz de referencia.

La matriz de mínima información estimada se obtiene por medio de la función siguiente:

$$F_1(g) = \sum_{i \in I} g_i \ln \left(\frac{g_i}{\hat{g}_i} \right) \quad (3.11)$$

(puede verse una derivación combinatoria como ejemplo en [Snickars and Wibull1977]).

Todos estos modelos basados en programación matemática intentan determinar la matriz de viajes más idónea consistente con la información contenida en los datos del volumen de tráfico en los arcos, al mismo tiempo, maximizan la entropía o la mínima información con respecto a la matriz de viajes de referencia.

Este tipo de modelos de estimación, pero empleando estimación mínimo cuadrática para F_1 y F_2 , son los empleados en este PFC.

En este PFC se ha empleado un modelo entrópico para estimar la matriz O-D cuando no se dispone de una target matriz. Este modelo adopta la siguiente expresión:

$$\text{Minimizar } \sum_{i \in I} g_i (\log(g_i - 1)) \quad (3.12)$$

$$\text{sujeto a : } \sum_i g_i \delta_i^a - \hat{v}_a = 0 \quad (3.13)$$

$$g_i \geq 0 \quad (3.14)$$

Asignación en equilibrio

Hasta ahora se ha asumido que es posible obtener la matriz la matriz de proporciones P independientemente del proceso de estimación de la matrix de viajes. Donde quiera que la congestión juega un papel importante en la elección de la ruta, esta suposición llega a ser cuestionable como que las proporciones en la elección de la ruta y la matriz de viajes sean indepen-

dientes. Debido a sus ventajas teóricas y prácticas, la asignación en equilibrio es el marco de natural para extender el modelo al caso de redes congestionadas.

Cuando los efectos de la congestión sean importantes, la asignación en equilibrio es un enfoque más realista, porque además, resuelve inconsistencias en los datos de flujos observados. El coste de atravesar un arco depende del volumen de flujo mediante una relación coste-flujo. El valor de los elementos de la matriz P , p_i^a , dependerán del volumen en todos los arcos, no pudiendo determinarse de forma independiente al proceso de estimación de la matriz de viajes. Las ecuaciones (3.9) pasan a ser no lineales en v .

El principal problema al incorporar el equilibrio de Wardrop en la estimación de matrices es que ahora las proporciones de rutas y la matriz de viajes a estimar son independientes. Una manera de resolver este problema es adoptar una aproximación iterativa: asumir una matriz O-D g^l , calcular la matriz de proporciones $P(g^l)$ y estimar g^{l+1} ; repetir el proceso hasta que ambas matrices estén próximas.

En [Cascetta and Nguyen1988] se muestra que la complejidad computacional de un modelo de asignación de la matriz O-D depende, en su mayor parte, de la técnica de asignación utilizada. El uso de una técnica de asignación en equilibrio y un tratamiento explícito de los efectos de la congestión es mucho más gravoso que utilizar la asignación proporcional.

3.3. Modelos de localización de sensores (P1 y P2). Otras extensiones

Hasta ahora se ha estado tratando la importancia de las matrices O-D. En general, la calidad de las matrices estimadas depende en gran medida de los datos de entrada (matriz desactualizada) y número y localización de los cordones de tráfico localizados. Habitualmente, el impacto de los errores en los datos de entrada ha sido investigado a través de simulaciones numéricas, evaluando diferente grado de exactitud en los datos de entrada. Ha habido una atención limitada, al problema de localización de sensores. Basado en el máximo error relativo posible examina la fiabilidad de la matriz O-D estimada con respecto al número y posición de los cordones de tráfico.

La mayoría de los modelos de localización que han sido desarrollados para determinar la

localización óptima de cordones de tráfico para cubrir tanta demanda como sea posible, pero sin una referencia a una matriz O-D dada.

Se trata de seleccionar un conjunto de arcos cuyo flujo observado de la mayor y mejor información posible.

Se han introducido cuatro reglas básicas para la localización de los sensores de tráfico o *Reglas de localización*:

- Regla de cubrimiento O-D (TCL-P1)
- Regla del máximo flujo neto (TCL-P2)
- Regla de máxima interceptación de flujo
- Regla de independencia de los arcos.

En [Yang et al.1998] hay unas conclusiones sobre estas reglas. Básicamente indica que en todo plan de aforación no pueden satisfacer simultáneamente todas estas reglas. En general, la primera regla se considera como una condición necesaria y es la que se aplica en el modelo TCL-P1. Las reglas dos y tres se pueden incorporar a la función objetivo. Este es el caso del modelo TCL-P2, que incorpora la tercera regla, satisfaciendo la primera.

3.3.1. Modelo de cubrimiento TCL-P1

El modelo TCL-P1 formula la regla de cubrimiento O-D. Para ello es necesario introducir la siguiente definición: Se dice que un par está cubierto por un determinado plan de aforamiento si existe un arco aforado que es empleado por algún usuario de dicho par.

Para formular matemáticamente este modelo se define la variables entera $z_a \in \{0, 1\}$, $z_a = 1$ si hay un sensor en el arco a , y 0 en otro caso.

Este modelo determina el número mínimo de sensores necesarios para cubrir todos los pares de la red. A este problema se le llama TCL-P1 (Traffic Count Location Problem 1).

Su función objetivo queda de la siguiente manera:

$$\text{Minimizar } \sum_{a \in A} z_a \quad (3.15)$$

Sus restricciones son las siguientes:

$$\sum_{a \in A} \delta_{i,a} z_a \geq 1, i \in I \quad (3.16)$$

$$z_a \in \{0, 1\}; a \in A \quad (3.17)$$

donde $\delta_{i,a} = 1$ si parte del flujo del par O-D $i \in I$ pasa por el arco $a \in A$, y 0 en otro caso.

La función objetivo determina el número de sensores localizados en la red. El modelo TCL-P1 posee una restricción por cada par O-D $i \in I$. El lado izquierdo de la desigualdad (3.16) proporciona el número de sensores que captan información del par $i \in I$, y se exige que sea mayor o igual que uno.

El mínimo número y la localización de los sensores puede obtenerse resolviendo (TCL-P1). Llamemos \hat{l}_0 y $\mathbf{z}^{\hat{l}_0}$ al mínimo número y al vector de localizaciones correspondiente de los sensores de tráfico. Se puede probar fácilmente que $\mathbf{z}^{\hat{l}_0}$ satisface la regla de independencia de los arcos.

La regla de cubrimiento O-D puede ser considerada como una de las condiciones necesarias para una estimación efectiva de la matriz O-D a partir de los sensores de tráfico. Así pues, el número resultante \hat{l}_0 indica el mínimo número de sensores a localizar en la red de tráfico. Sin embargo, la disposición de los sensores de tráfico de acuerdo a $\mathbf{z}^{\hat{l}_0}$ no tiene porqué captar necesariamente la mayor parte de los flujos a través de la red, excepto cuando sólo hay un camino para cada par O-D (situación que se da por ejemplo en la asignación todo-nada).

3.3.2. Modelo de máxima interceptación de flujo neto TCL-P2

De acuerdo a la regla de interceptación de flujo neto máximo, los sensores de tráfico podrían localizarse de tal manera que se observaran la mayor cantidad de flujo neto en la red. Puede darse el caso de que se capte el mismo flujo más de una vez. Por este motivo, el criterio razonable es maximizar el total de los flujos netos observados. Este modelo asume que los flujos en los caminos asociados con la matriz O-D de referencia son conocidos basándose en [Hogson, M.J. 1990].

Su función objetivo queda de la siguiente manera:

$$\text{Maximizar } \sum_{k \in K} h_k y_k \quad (3.18)$$

Sus restricciones son las siguientes:

$$\sum_{a \in A} z_a = \hat{l} \quad (3.19)$$

$$\sum_{a \in A} \delta_{i,a} z_a \geq 1, \quad i \in I \quad (3.20)$$

$$\sum_{a \in A} \delta_{a,k} z_a \geq y_k, \quad k \in K \quad (3.21)$$

$$z_a \in \{0, 1\}; \quad y_k \in \{0, 1\}; \quad a \in A, \quad k \in K \quad (3.22)$$

donde K es el conjunto de caminos con flujo para la matriz O-D de referencia y h_k el flujo del camino $k \in K$. La restricción (3.21) aplica la regla de cubrimiento de la matriz O-D. El parámetro $\delta_{a,k} = 1$ si el camino k emplea el arco a y 0 en otro caso. También asegura que y_k será cero si no hay un sensor localizado en el camino $k \in K$, es decir, y_k indica si hay algún aforo para el camino k . Esto significa que los *efectos de múltiples conteos* desaparecen de la función objetivo ya que los caminos llevan flujos disjuntos y son contados una única vez.

Nótese que el vector de localización obtenido no necesariamente tiene que satisfacer la regla de independencia de los arcos, este hecho depende del número de sensores \hat{l} . En general, \hat{l} será al menos igual o mayor que \hat{l}_0 , así que el problema tiene al menos una solución factible que satisface la regla de cubrimiento O-D. En el caso de que $\hat{l} = \hat{l}_0$, el problema puede explicarse como escoger la mejor solución en términos de la máxima cobertura de flujo entre todas las combinaciones de sensores que satisfacen las reglas de cubrimiento O-D (note que el vector de localización de los sensores $z^{\hat{l}_0}$ que satisface la regla de cubrimiento O-D no es, en general, único).

Capítulo 4

Método de trabajo

En este proyecto se han perseguido dos objetivos fundamentales, por un lado la evaluación computacional de los modelos TCL y por otro la evaluación de su impacto en la fiabilidad de las estimaciones derivadas de dichos modelos TCL. Se han desarrollado dos metodologías de trabajo para cada uno de estos objetivos. El tercer objetivo (Desarrollo e implementación de algoritmos heurísticos para el TCL) que a priori se había propuesto, se ha descartado porque se consiguió resolver exactamente la totalidad de las redes planteadas en un tiempo razonable en el contexto de su aplicación.

Para alcanzar los objetivos se ha planteado una metodología exacta resolución a través del empleo de GAMS¹(www.gams.com) que implementa algoritmos que son el Estado-del-Arte en resolución de problemas de optimización de grandes dimensiones.

La elección de GAMS se debe a sus características:

1. Es un sistema de modelado de alto nivel para programación matemática y optimización. La codificación empleada coincide con el modelado matemático.
2. Tiene disponibles una serie de *Solvers* que implementan diferentes algoritmos del Estado-del-Arte para resolver el problema. Este sistema está maduro en la actualidad y resuelve los problemas en tiempo difícil de alcanzar con otros programas.
3. Es multiplataforma y está disponible en el servicio de supercomputación de la UCLM.

Por todas estas características, GAMS resultó ser la mejor opción. Otra ventaja de utilizar

¹General Algebraic Modeling System

GAMS es que algunos algoritmos empleados en este PFC ya estaban implementados en él. Esto último permitió tener la primera parte del camino andado.

GAMS tiene algunos inconvenientes:

1. No es una herramienta gratuita, sino que los solvers tienen una licencia para cada uno de ellos. En caso de carecer de dicha licencia, los problemas a resolver están limitados a unas pocas variables como *modo de prueba*.
2. Es muy diferente a cualquier lenguaje de programación actual. Esto implica que el aprendizaje de este sistema sea más lento que para cualquier otro lenguaje.
3. La salida de los programas no se visualiza de un modo agradable y comprensible para la mayoría de los usuarios. Otros sistemas disponen de herramientas específicas para la creación de interfaces adecuadas a los resultados.

Para resolver los problemas asignación de tráfico, localización de sensores y estimación de matrices O-D se ha recurrido al PC cuyas características se detallan en el Anexo Recursos hardware. Cuando el tiempo que necesitaba el PC para la resolución de un algoritmo excedía el orden de horas, se optó por resolverlos en el Servicio de Supercomputación de la UCLM. Para poder tener una medida comparativa se resolvieron en su totalidad las redes disponibles. En algunos casos la limitación parecía ser los recursos hardware, porque los requisitos computacionales consumidos en supercomputación indican que el PC debería tener la capacidad de resolverlos igualmente en un tiempo razonable. Según se ha podido observar, los requisitos computacionales a la hora de generar el problema (primera fase de la ejecución de los algoritmos) superan la capacidad del PC en tanto que los hacen inabordables.

4.1. Metodología empleada para el objetivo 1

En la figura 4.1 se muestra un sencillo diagrama para explicar el proceso realizado en la metodología para el objetivo 1.

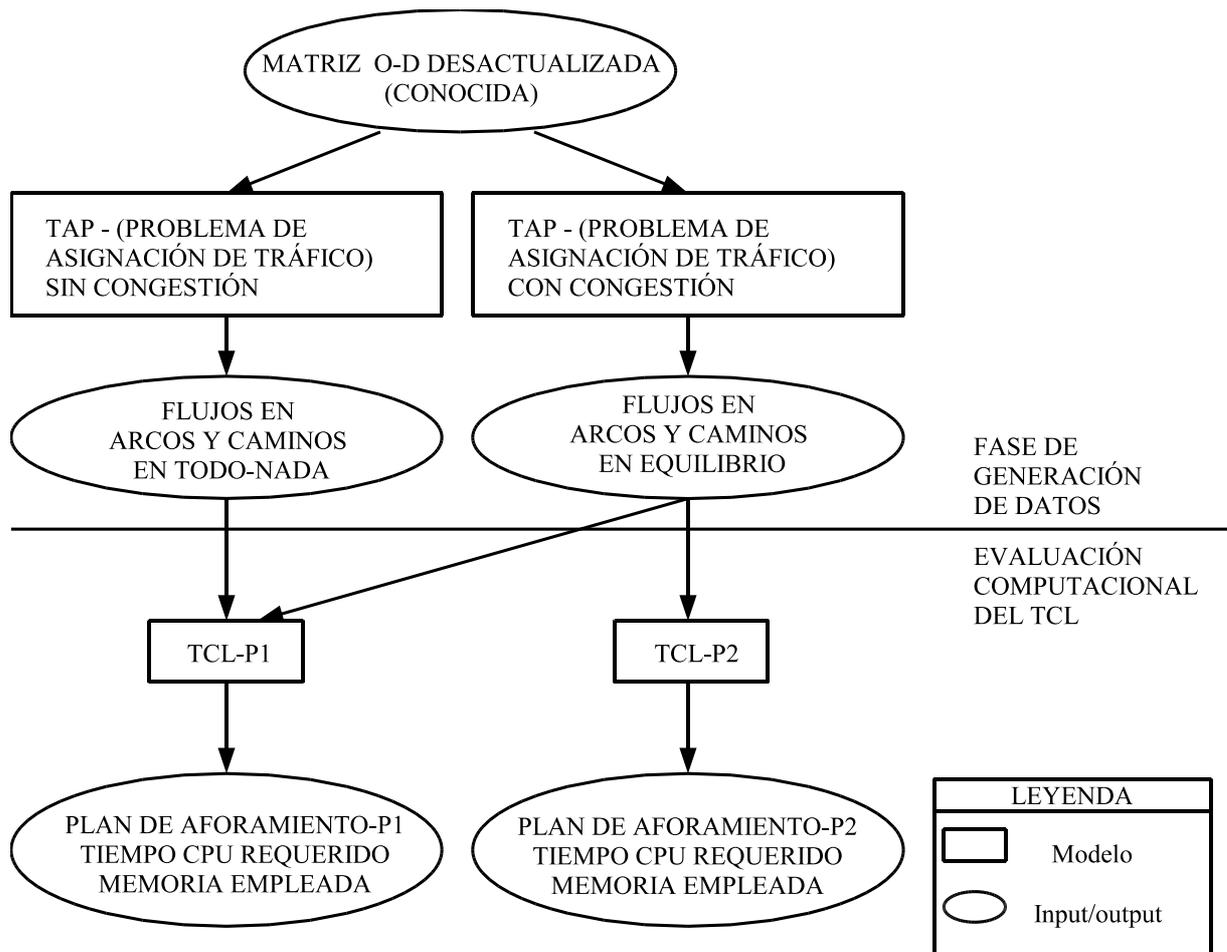


Figura 4.1: Metodología empleada en el objetivo 1

4.1.1. Fase de generación de datos. Resolución del problema de asignación de tráfico

El primer modelo que se desea resolver con el objeto de generar las entradas para los modelos TCL es el problema de asignación de tráfico.

GAMS requiere la formulación explícita de los modelos de optimización. En este caso precisa que se conozcan todos los caminos de la red. Esta tarea resulta inabordable incluso al tratar con redes de dimensiones reducidas porque el número de caminos crece de manera exponencial con el tamaño de éstas. Por esta razón se ha motivado el uso de un algoritmo de descomposición simplicial desagregada (DSD), desarrollado por [Patriksson and Larsson1992]. Este algoritmo genera los caminos según van siendo necesarios. Itera entre una problema de caminos mínimos con el objeto de generar nuevos caminos y el llamado *problema maestro restringido*, que encuentra un equilibrio en el subconjunto de caminos detenidos. La solución a este problema modifica los flujos anteriores, y por tanto cambia el coste en los arcos y vuelve a iterar el proceso.

Este algoritmo fue implementado en GAMS por el profesor Ricardo García Ródenas, director de este PFC, y Doroteo Verastegui Rayo, empleando el algoritmo D'Esopo en la fase de caminos mínimos y los optimizadores de GAMS para resolver el problema maestro restringido.

En GAMS las líneas que comienzan con '*' son comentarios y son ignoradas en la ejecución. El código implementado es el siguiente:

```
*****
* opciones generales y comentarios
*****

$title METODO DE DESCOMPOSICION SIMPLICIAL DESAGREGADA.

$ontext DOROTEO VERASTEGUI RAYO RICARDO GARCIA RODENAS $offtext

$OFFLISTING;

OPTION LIMCOL=0 LIMROW=0 SOLPRINT=OFF PROFILE=1 nlp=minos;

FILE resultados /c:\gams\Redes\ngd\ngd_sc_100.txt/

put resultados;

*****
*DESCRIPCION DE LA RED
*****

$INCLUDE "c:\gams\Redes\ngd\ngd_100.gms";

*****
*
PARAMETROS DEL DSD
*****
```

SET ide(I) conjunto auxiliar de pares fijados a un camino
 act(I) pares no fijados (activos) en cada momento
 KI(K,I) conjunto de caminos para el par I
 NEW(K,O) conjuntos de caminos para el origen o
 RSD(k) puntos extremos en el RMP para el RSD
 MET algoritmos /RSD,DSD,NEW/
 SEL(MET) auxiliar para definicion del metodo
 SUB(MET) metodos seleccionados;
 ALIAS (MET, METMET);

ALIAS (A, AA, AAA); ALIAS (N, NN, NNN); ALIAS (I, II, III);
 ALIAS (K, KK, KKK); ALIAS (o, oo, ooo);

PARAMETER d(K) direccion de busqueda
 haux(k) flujo auxiliar contiene solucion RMP
 haux1(K) flujo auxiliar
 ncam(i) numero de caminos en el par i
 ultcam(I) ultimo camino anyadido al par I
 v0(a) flujo cargado en la red con los pares identificados
 DELTA(A,K) matriz de incidencia arco camino
 Vnew(o,a,k) matriz de puntos extremos en RSD del new algoritmo
 vaux(a) variable auxiliar de flujos en los arcos para calcular el GAP
 Ve(a,k) matriz de puntos extremos en RSD
 CA(a) parametro auxiliara para la linealizacion de costes
 CB(a) parametro auxiliara para la linealizacion de costes
 CAK(K) parametro para calcular el coste del camino k
 CBK(K) parametro para calcular el coste del camino k ;

* Quitado de SCALAR (aparece en redngd.gms)

SCALAR *min_dem minimia demanda necesaria para
 identificar el par /0.0/
 npar numero de pares /0/
 c_1 coste camino 1 (identificaciond de pares) /0/
 c_2 coste camino 2 (identificaciond de pares) /0/
 lam tolerancia en identificacion de pares /1./
 nw numero de pares identificados /0/
 n_cam numero de caminos por par /0/
 k_num camino del par no activo /0/
 der_1 variable implementacion metodo secante /0/
 der_2 variable implementacion metodo secante /0/
 min_h variable implementacion metodo secante /0/
 min_d variable implementacion metodo secante /0/
 min_der variable implementacion metodo secante /0/
 itecount contador de iteraciones metodo secante /0/
 n_sec numero de iteraciones en el metodo de la secante /6/
 hult variable auxiliar de flujo del ultimo camino en el par /0/
 a_aux variable auxiliar /0/
 b_aux variable auxiliar /0/
 suma_a parametro para proyecciones /0/
 suma_b parametro para proyecciones /0/
 cami_neg control para detectar flujo negativo en proyecciones /1/
 U_i coste de equilibrio para el par analizado /0/
 Npro contador de proyecciones en el RMP /0/
 Npro_max maximo numero de proyecciones por RMP /1/
 Master tipo de problema maestro restringido /1/
 Ncol numero de variables del problema maestro /0/
 NCAM numero de caminos empleados en cada momento /0/
 Idemanda numero de demandas para el origen /0/
 demanda demanda analizada /0/
 destino variable nodo auxiliar /0/
 NITER contador de iteraciones /0/
 ITERMAX maximo numero de iteracione principales /0/
 origenaux variable auxiliar para indicar actual origen /0/
 CPURMP tiempo empleado en la resolucion del RMP /0/
 gap medida del error /inf/

```

gapmax          gap maximo admitido                /-1E15/
cpu             tiempo empleado                    /0/
n_iden         iteracion inicial en que se identifica caminos /500/
eli            eliminacion de columnas 1=si 0=no    /1/;

* opciones de problemas maestros para el DSD
*   1= RMP estandar en optimizacion (Patriksson and Larsson)
*   2= Una iteracion de Newton--> funcion objetivo no lineal y costes lineales
*   3= Una iteracion FW en la busqueda de Newton --> programacion lineal
*   4= RMP de optimizacion para desigualdades variacionales
*   5= Reformulacion de 4 con restricciones solo lineales
*   6= Descomposicion para calculo de direccion de newton

** Se declaran las variables de optimizacion.

VARIABLES S_UE      variable del valor de la funcion objetivo
modelo TAP-UE v(a)  flujo en el arco a h(k)      flujo en los
caminos K del par I LAMBDA(K) pesos del RMP en el RSD mu(k,O)
pesos del RMP del new;

POSITIVE VARIABLE h(k), LAMBDA(K), MU(k,O);

EQUATIONS COSTE_UE      funcion objetivo modelo TAP-UE DEMAN(i)
satisfaccion de la demanda del par o-d i DEMAN6(i)
satisfaccion de la demanda del par o-d i activos FLUJODsd(a)
relacion entre flujo camino - flujo arco en el DSD FLUJODsd6(a)
relacion entre flujo camino - flujo arco en el DSD para
descomposicion FLUJOrsd(a)      relacion entre flujo camino - flujo
arco en el RSD FLUJOnew(a)      relacion entre flujo camino - flujo
arco en el new CONVEXO          combinacion convexa CONVEXOnew(o)
combinacion convexa en el nuevo algoritmo;

COSTE_UE..           S_UE=E=
SUM(a, SDATA(a, 't0') * v(a) + SDATA(a, 'd') * v(a) ** SDATA(a, 'm'));

DEMAN(i)..           SUM(k$KI(k,i), h(k)) =E= g(i);
DEMAN6(i)$act(I)..  SUM(k$KI(k,i), h(k)) =E= g(i);

FLUJORSD(a)..       SUM(K$RSD(K), LAMBDA(K) * v(a,k) ) =E= v(a);
FLUJOnew(a)..       SUM(O, SUM(K$NEW(K,O), mu(K,O) * vnew(O,a,k) )
) =E= v(a); CONVEXO.. SUM(K$RSD(K), LAMBDA(K)) =E= 1;
CONVEXOnew(O)..     SUM(K$NEW(k,o), mu(k,O) ) =E= 1; FLUJODSD(a)..
SUM(i, SUM(k$KI(k,i), DELTA(a,k) * h(k))) =E= v(a); FLUJODSD6(a)..
SUM(i$ act(I), SUM(k$KI(k,i) , DELTA(a,k) * h(k))) + v0(a) =E= v(a);

MODEL RMPDSD0/COSTE_UE, DEMAN6, FLUJODSD6/; MODEL
RMPDSD1/COSTE_UE, DEMAN, FLUJODSD/;

MODEL RMPRSD/COSTE_UE, CONVEXO, FLUJORSD/; MODEL RMPnew/COSTE_UE,
CONVEXOnew, FLUJOnew/;

*RMPDSD0.optfile=1; *RMPDSD1.optfile=1; *RMPDSD0.optfile=2;
*RMPDSD1.optfile=2; *RMPRSD.optfile=1;

*****
*                               PARAMETROS DE D'ESOPO
*****

SCALAR Inodo /0/
Iarco/0/
MIN/inf/
principio/1/

```

```

    final/1/
    origen/0/
    pivot/0/
    L contador de arco /0/
    nodo contador de nodo/0/;

PARAMETER coste(A)      coste del arco estado(N)      estado
del arco (-1 no visitado 0 provisional 1 definitivo) distancia(N)
distancia del origen actual al destino N ultimoarco(N)  contiene
al final de esopo el arco optimo que llega al nodo N ultimonodo(N)
contiene al final de esopo el nodo optimo que llega al nodo N
tabla(N);

*****
*                               INICIALIZACION DEL DSD
*****
SUB('NEW')=no; SUB('RSD')=no; SUB('DSD')=yes;

LOOP (MET$SUB (MET) ,

SEL (METMET)=NO; SEL (MET)= YES;

IF (SEL ('RSD') ,Ve (A,K)=0; put
'*****';
PUT/; PUT '                               RSD  ' ;PUT/; put
'*****';
PUT/; );

IF (SEL ('DSD') ,KI (k,i)=no; put
'*****';
PUT/; PUT '                               DSD  ' ;PUT/; put
'*****';
PUT/; ); IF (SEL ('NEW') ,new (K,O)=no; Vnew (o,A,k)=0; put
'*****';
PUT/; PUT '                               NEW   ' ;PUT/; put
'*****';
PUT/; );

PUT '      ITERACION   pares   caminos  COLUMNAS   CPU
Z          GAP
RELATIVO';PUT/;

Inodo=0; Iarco=0; MIN=inf; principio=1; final= 1; origen=0;
pivot=0; L=0; nodo=0; CPURMP=0; ncam=0; niter=0;

* inicializacion de flujos para la primera generacion de caminos minimos
v.l(a)=0;

WHILE ((NITER LE ITERMAX) and ((gap ge (gapmax)) or (niter le 1))
, *WHILE (NITER LE ITERMAX, NITER=NITER+1; vaux(a)=0;
*****
*PROBLEMA DE GENERACION DE COLUMNAS O DE CAMINOS MINIMOS (USANDO
ALG. D'ESOP)
*****

IF (SEL ('RSD') ,
ncam=ncam+1;
);

origenaux=0; LOOP (N$ ((NDDEM (N) -NDDEM (N-1)) GT 0) ,

```

```

IF (SEL ('new'),
  ncam=ncam+1;
  origenaux=origenaux+1;
);

-----
* INICIALIZACION DEL ALGORITMO DE CAMINOS MINIMOS PARA CADA ORIGEN
-----

* ACTUALIZACION DE COSTES
LOOP (A,
coste (A)=SDATA (a, 't0') +SDATA (a, 'd') *SDATA (a, 'm') * (v.L (a) ** (SDATA (a, 'm') -1.));
);

LOOP (AAA$ (ord (AAA) eq 1), LOOP (NNN$ (ord (NNN) eq 1),
LOOP (III$ (ORD (III) EQ 1), LOOP (KKK$ (ORD (KKK) EQ 1),
LOOP (ooo$ (ord (ooo) eq 1),

-----
* PASO 0: OBTENCION DEL NODO ORIGEN
-----
ORIGEN=ord (N);

-----
* PASO 1: INICIALIZACION DE ETIQUETAS
-----
estado (NN)=-1; distancia (NN)=+INF; ultimoarco (NN)=0;

-----
* PASO 2: INICIALIZACION DEL NODO ORIGEN
-----
estado (NNN+(ORIGEN-1))=0; distancia (NNN+(ORIGEN-1))=0;

-----
* PASO 3: CREACION DE LA TABLA DE DOBLE ENTRADA CON EL NODO ORIGEN
-----
principio=1; final=1; tabla (NNN+(principio-1))=origen;

-----
* PASO 4: BUCLE PRINCIPAL
-----
WHILE ( (final-principio) ge 0,

-----
* PASO 4.1 obtener el primer elemento de la tabla, el llamado pivot
-----
pivot=TABLA (NNN++ (principio-1));

* retirada del primer elemento de la tabla
principio=principio+1;

-----
* PASO 4.2 PARA TODO ARCO
-----
*   CALCULO DEL NUMERO DE ARCOS SALIENTES DE PIVOT
   Iarco= ND (NNN+ (pivot-1)) -ND (NNN+ (pivot-2));

*   CALCULO DEL ARCO ANTERIOR AL PRIMER ARCO DE PIVOT
   L=ND (NNN+ (pivot-2));
   WHILE ( Iarco gt 0,
     Iarco=Iarco-1;
     L=L+1;
     nodo=LA (AAA+ (L-1));
     IF ( (distancia (NNN+ (pivot-1)) + coste (AAA+ (L-1)) lt distancia (NNN+ (nodo-1))),
       distancia (NNN+ (nodo-1))=distancia (NNN+ (pivot-1)) + coste (AAA+ (L-1));

```

```

ultimoarco (NNN+ (nodo-1) )=L;
ultimonodo (NNN+ (nodo-1) )=pivot;
if (estado (NNN+ (nodo-1) ) eq -1,
    estado (NNN+ (nodo-1) )=0;
    final=final+1;
    TABLA (NNN++ (final-1) )=nodo;
    );

if (estado (NNN+ (nodo-1) ) eq 1,
    estado (NNN+ (nodo-1) )=0;
    principio=principio-1;
    TABLA (NNN++ (principio-1) )=nodo;
    );
* fin if
    );
* fin l
    );

estado (NNN+ (pivot-1) )=1; );

*-----
* FINAL DE LA FASE DE GENERACION DE CAMINOS OPTIMOS PARA EL ORIGEN N
*-----

*****
* DEFINICION DE CAMINOS Y ASIGNACION A PARES
*****

* VARIABLES UTILIZADAS
* Idemanda numero de demandas con nodo origen ORIGEN
* demanda demanda actualmente analizada
* origen origen de la demanda actual
* destino variable auxiliar que contiene el nodo actualmente analizado
* ncam numero de caminos o camino analizado

*      CALCULO DEL NUMERO DE DEMANDAS PARA EL NODO N
demanda=NDdem (N-1) ;
Idemanda= NDdem (N) -demanda;

IF (SEL ('new' ),
    new (KKK+ (ncam-1) ,OOO+ (origenaux-1) )=YES;
    );

WHILE ( Idemanda gt 0,
    Idemanda=Idemanda-1;
    demanda=demanda+1;
    IF (SEL ('DSD' ),
        ncam=ncam+1;
        ultcam (III+ (demanda-1) )=ncam;
        KI (KKK+ (ncam-1) , III+ (demanda-1) )=YES;
    );

    destino=LAdem (III+ (demanda-1) );
    WHILE ( (destino ne origen),
        L=ultimoarco (NNN+ (destino-1) );
        IF (SEL ('DSD' ),
            DELTA (AAA+ (L-1) , KKK+ (ncam-1) )=1.;
            );
        IF (SEL ('RSD' ),
            Ve (AAA+ (L-1) , KKK+ (ncam-1) )=Ve (AAA+ (L-1) , KKK+ (ncam-1) )+G (III+ (demanda-1) );
            );
        IF (SEL ('new' ),
            VNEW (OOO+ (ORIGENAU-1) , AAA+ (L-1) , KKK+ (ncam-1) )=
            VNEW (OOO+ (ORIGENAU-1) , AAA+ (L-1) , KKK+ (ncam-1) )+G (III+ (demanda-1) );
            );
        vaux (AAA+ (L-1) )=Vaux (AAA+ (L-1) )+G (III+ (demanda-1) );
        destino=ultimonodo (NNN+ (destino-1) );
    );
);

```

```

    );
* FIN DEL ARTIFICIO AAA, NNN, III Y DE KKK, OOO
);););););

);

*****
* CALCULO DEL GAP
*****
gap=0; LOOP(a, gap=gap+coste(a)*(v.l(a)-vaux(a))););

*****
*PROBLEMA MAESTRO RESTRINGIDO
*****
*-----
*                D   S   D
*-----

IF(SEL('DSD'),

    if(master eq 0,

*inicializacion de la identificacion de pares
    if(niter eq 1,
        npar=card(I);
$INCLUDE      "c:\gams\DSD\ini_par.gms";
    );

    if(niter ge 2,
$INCLUDE      "c:\gams\DSD\ide_par.gms";
    );

    SOLVE RMPDSD0 USING nlp MINIMIZING S_UE;
$INCLUDE      "c:\gams\DSD\eli_par.gms";
    CPURMP=CPURMP+RMPDSD0.resusd;
    );

    if(master eq 1,
        SOLVE RMPDSD1 USING nlp MINIMIZING S_UE;
        loop(i,
            loop(k$ki(k,i),
                if(h.l(k) le eps, ki(k,i)=no;));
            ););
        CPURMP=CPURMP+RMPDSD1.resusd;
        ncol=card(ki);
    );

);

*-----
*                R   S   D
*-----

IF(SEL('RSD'),
    LOOP(KKK$(ORD(KKK) EQ 1),RSD(KKK+(ncam-1))=yes;);
    SOLVE RMPRSD USING nlp MINIMIZING S_UE;
    CPURMP=CPURMP+RMPRSD.resusd;

* eliminacion de columnas
if(eli eq 1,
    loop(k$RSD(k),
        if(lambda.l(k) le eps, rsd(k)=no;);
    );
); ncol=card(rsd); );

*-----
*                N   E   W
*-----

```

```

IF (SEL('NEW'),
SOLVE RMPNEW USING nlp MINIMIZING S_UE;
CPURMP=CPURMP+RMPnew.resusd;

* eliminacion de columnas
if(eli eq 1,
    loop(o,
        loop(k$NEW(k,o),
            if(mu.l(k,o) le eps, new(k,o)=no;);
        ); );
); ncol=card(new); );

*****
* IMPRESION DE RESULTADOS
*****
put$(niter eq 1)niter:10:0,nw:10:0, ncam:10:0,ncol:10:0
CPURMP:12:3, s_ue.l:30:15; put$(niter gt 1) niter:10:0,nw:10:0,
ncam:10:0,ncol:10:0 CPURMP:12:3, s_ue.l:30:15,
gap:25:15, (gap/s_ue.l):25:15; put/; *fin de programa );

*FIN DE CONJUNTO DE METODOS

* contar caminos de la solucion optima
$INCLUDE "c:\gams\DSD\contar.gms";

$INCLUDE "c:\gams\DSD\solucion.gms";

);

```

En el código no se incluyen los datos de cada red, sino que están en un fichero aparte. Para facilitar el uso y comprensión de este código (incorporado en el CD que se adjunta a este PFC) se describen a continuación ciertos parámetros y subíndices empleados en el código.

Cada red está definida por una serie de conjuntos y una serie de matrices, las cuales se detallan a continuación:

- Conjuntos:

- A arcos
- N nodos de la red
- I pares origen-destino
- K cantidad máxima de caminos empleada

- Matrices para describir la topología de la red (la estructura de los datos es la misma que la desarrollada para el algoritmo de caminos mínimos en [Gallo and Pallotino1986]):

- $LA(A)$ lista de nodos finales de cada arco a , ordenada por valores crecientes de los nodos

- $ND(N)$ contiene los punteros que indican el final de la lista de arcos salientes del nodo N .
- Matrices para describir la matriz de viajes origen-destino (emplea la misma estructura que el grafo anterior):
 - $LADDEM(I)$ lista de los nodos destino del par de demanda i
 - $G(I)$ flujos del par de demanda i
 - $NDDEM(N)$ puntero que indica el final de la lista de demandas con nodo origen n en $LADDEM$

Para calcular el coste de atravesar el arco a se emplean las llamadas *funciones BPR*, que tienen la siguiente expresión:

$$\begin{aligned} Coste(a) &= C_a(V_a) = \int_0^{V_a} c_a(x) dx = \int_0^{V_a} \left[c_a^0 + b_a \left(\frac{x}{k_a} \right)^{n_a} \right] dx \\ &= c_a^0 V_a + \frac{b_a}{n_a + 1} \left(\frac{V_a}{k_a} \right)^{n_a + 1} = c_a^0 V_a + d_a V_a^{m_a} \end{aligned}$$

donde $d_a = \frac{b_a}{(n_a + 1)k_a^{n_a + 1}}$, $m_a = n_a + 1$, c_a^0 , es el coste del arco vacío, es decir, el coste de arco sin flujo y k_a es el parámetro relativo a la capacidad del arco a .

La tabla $SDATA$ contiene los datos necesarios para las funciones BPR. Está formada por las columnas t_0 , d_a y m_a que son las que se emplean en estas funciones para cada arco a .

Con todos estos datos el algoritmo DSD tiene todo lo necesario para su ejecución. En el código del DSD hay un parámetro que indica si estamos trabajando con una red *congestionada* o por el contrario *no congestionada*. Si la red se encuentra en el segundo caso, una iteración principal del DSD resuelve el problema TAP. Si se trata de una red congestionada este valor se establece a 5.

Una vez resuelto, en su salida quedan recogidos, como dato más importante, los flujos correspondientes a cada arco, los flujos en los caminos en equilibrio, así como otra serie de información en unas matrices que se detalla a continuación:

- $VOBS(A)$ flujo observado en el arco a .
- $CONGES(A)$ tiempo de congestión en el arco a .

- $ki(k, i)$ matriz de incidencia camino-par. Indica si el camino k satisface al par de demanda i . El número de caminos coincide con el número de demandas cuando se trata de una red *sin congestión*, porque cada demanda utiliza el camino mínimo. En el caso de tratarse de una red *con congestión* el número de caminos para cada demanda es mayor o igual a uno.
- $h(k)$ vector de flujo en cada camino k .
- $\delta_{a,k}$ matriz de incidencia arco-camino. Indica si el arco a está en el camino k .
- $P(i, a)$ matriz de proporciones par-arco. Indica qué proporción de flujo de la demanda i atraviesa el arco a . Cuando trabajamos con una red *sin congestión* estas proporciones son siempre de 1 ó 0, por la asignación *todo-nada*.

Ejemplo: La red de Nguyen-Dupuis

Se utilizará una red de pequeñas dimensiones, más concretamente, la red de Nguyen-Dupuis [Nguyen and Dupuis 1984], cuya representación puede verse en la figura 4.2. El grafo (\mathcal{G}) de esta red está formado por 13 nodos (\mathcal{N}) y 19 arcos (\mathcal{A}). Esta red tiene cuatro demandas ($I = \{i_1, i_2, i_3, i_4\}$) que se especifican de la siguiente manera:

- Demanda $i_1(1 \rightarrow 2)$: 400
- Demanda $i_2(1 \rightarrow 3)$: 800
- Demanda $i_3(4 \rightarrow 2)$: 600
- Demanda $i_4(4 \rightarrow 3)$: 200

Donde los nodos origen de las demandas son el 1 y el 4. Mientras que los nodos destino de las demandas son el 2 y el 3. Es decir, la red cuenta con 4 centroides que son los orígenes/destino de las demandas.

En la tabla 4.1 queda representada matricialmente.

Este formato da nombre a la expresión de matriz de demanda origen-destino. En todo estudio de movilidad la matriz O-D que se describe en la literatura científica es la que se considera como matriz de referencia.

	2	3
1	400	800
4	600	200

Tabla 4.1: Pares de demanda de Nguyen Dupuis

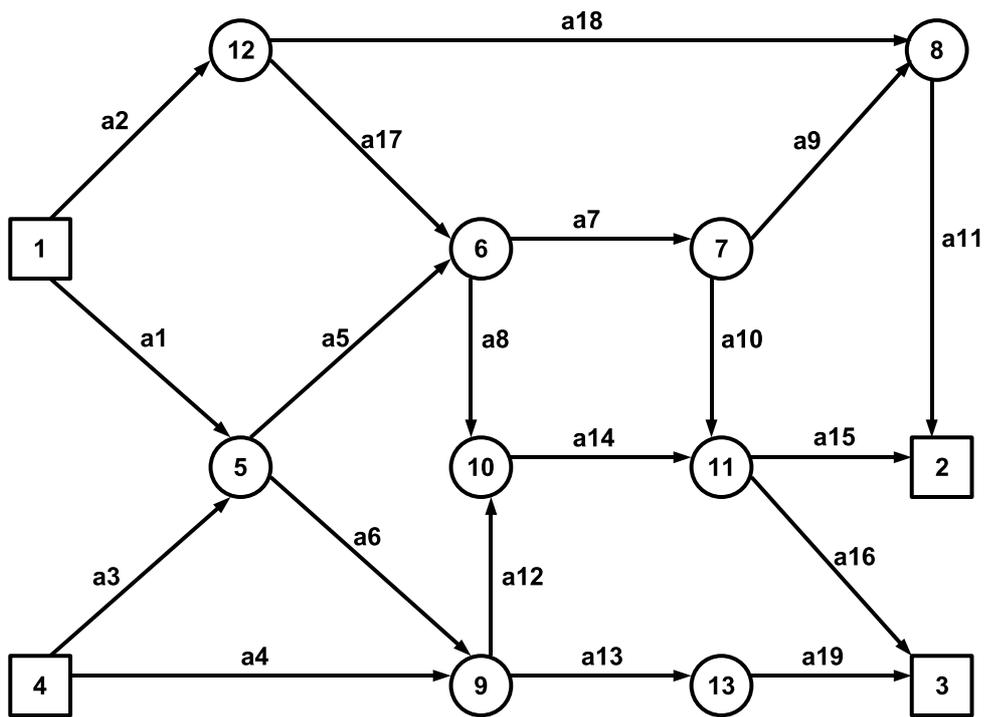


Figura 4.2: Red Nguyen-Dupuis original

Para modelar el efecto de la congestión se asume que el coste de atravesar un arco a depende del flujo que atraviesa dicho arco. Una expresión más usual en los estudios de movilidad son las llamadas funciones BPR, que siguen la siguiente expresión:

$$c_a(v_a) = c_a^0 + b_a \left(\frac{v_a}{k_a} \right)^{n_a}$$

donde c_a^0 es el coste del arco vacío, es decir, el coste de arco si no tiene flujo y k_a es el parámetro relativo a la capacidad del arco a . Cuando se sobrepasa el valor de la capacidad por unidad de tiempo hablamos de congestión en el arco correspondiente.

Empleando las funciones BPR, la función objetivo para el TAP queda de la siguiente manera:

$$\begin{aligned} Z &= \sum_{a \in A} \int_0^{v_a} c_a(x) dx = \sum_{a \in A} \int_0^{v_a} \left[c_a^0 + b_a \left(\frac{x}{k_a} \right)^{n_a} \right] dx \\ &= \sum_{a \in A} c_a^0 v_a + \frac{b_a}{n_a + 1} \left(\frac{v_a}{k_a} \right)^{n_a + 1} = \sum_{a \in A} c_a^0 v_a + d_a v_a^{m_a} \end{aligned}$$

Los parámetros de coste de la red necesarios para la función de coste de los arcos pueden verse en la tabla 4.2. Los valores de los parámetros d_a y m_a se encuentran ya calculados. Los valores en esta tabla permanecen constantes a lo largo del estudio, aunque no así los datos de demanda (matriz O-D).

La solución del TAP representa la situación de asignación en equilibrio.

Con esto queda definida la red de Nguyen-Dupuis. Ahora se pasará a resolver el problema TAP de esta red.

Primer principio de Wardrop. Para este ejemplo, muestra la situación de equilibrio de la tabla 4.3. Puede verse el flujo y el coste en los arcos en la situación de equilibrio. Estos valores corresponden a la asignación de tráfico cuando se considera *congestión*.

Los caminos que absorben la demanda de cada par están compuestos por una serie de arcos que se detallan en la tabla 4.4. En dicha tabla se encuentra el flujo asociado a cada camino y el coste en equilibrio. Puede verse que todos los costes para los caminos de un mismo par son iguales. Puede verse una representación gráfica de los caminos en la figura 4.3.

Otro dato de gran interés conseguido al resolver la asignación de tráfico es la matriz $P_{i,a}$, que relaciona la proporción de demanda del par i que atraviesa cada arco a . En la tabla 4.5 se

Arco	c_a^0	d_a	m_a
a1	7.0	0.00625	2.0
a2	9.0	0.00500	2.0
a3	9.0	0.00500	2.0
a4	12.0	0.00250	2.0
a5	3.0	0.00375	2.0
a6	9.0	0.00375	2.0
a7	5.0	0.00625	2.0
a8	13.0	0.00250	2.0
a9	5.0	0.00625	2.0
a10	9.0	0.00625	2.0
a11	9.0	0.00625	2.0
a12	10.0	0.00250	2.0
a13	9.0	0.00250	2.0
a14	6.0	0.00125	2.0
a15	9.0	0.00250	2.0
a16	8.0	0.00500	2.0
a17	7.0	0.00125	2.0
a18	14.0	0.00250	2.0
a19	11.0	0.00500	2.0

Tabla 4.2: Parámetros de la función de coste de los arcos

Arco	Flujo	Coste
a1	675.144	15.439
a2	524.856	14.249
a3	102.571	10.026
a4	697.429	15.487
a5	416.187	6.121
a6	361.528	11.711
a7	356.416	9.455
a8	184.626	13.923
a9	102.571	6.282
a10	253.845	12.173
a11	502.571	15.282
a12	497.429	12.487
a13	561.528	11.808
a14	682.056	7.705
a15	497.429	11.487
a16	438.472	12.385
a17	124.856	7.312
a18	400.000	18.000
a19	561.528	16.615

Tabla 4.3: Flujos y costes de los arcos en el equilibrio

Demanda	Camino	Flujo	Coste
$i_1(1 \rightarrow 2)$	a2 - a11 - a18	400.00	47.53
$i_2(1 \rightarrow 3)$	a1 - a5 - a7 - a10 - a16	253.84	55.57
	a2 - a8 - a14 - a16 - a17	124.85	55.57
	a1 - a6 - a13 - a19	361.52	55.57
	a1 - a5 - a8 - a14 - a16	59.77	55.57
$i_3(4 \rightarrow 2)$	a4 - a12 - a14 - a15	497.42	47.16
	a3 - a5 - a7 - a9 - a11	102.57	47.16
$i_4(4 \rightarrow 3)$	a4 - a13 - a19	200.00	43.91

Tabla 4.4: Caminos usados en equilibrio

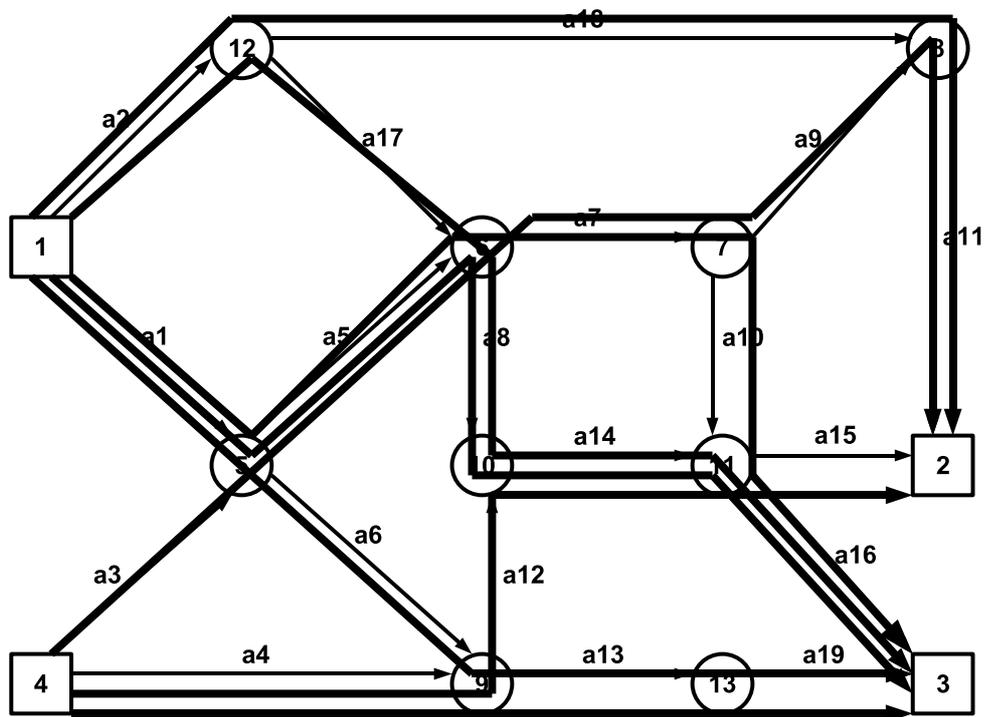


Figura 4.3: Red Nguyen-Dupuis con los caminos sobreimpresionados

Par	Arco	Proporción
i1	a2	1.0000000000000000
i1	a11	1.0000000000000000
i1	a18	1.0000000000000000
i2	a1	0.843929963420921
i2	a2	0.156070036579080
i2	a5	0.392019539592957
i2	a6	0.451910423827964
i2	a7	0.317306621173164
i2	a8	0.230782954998873
i2	a10	0.317306621173164
i2	a13	0.451910423827964
i2	a14	0.230782954998873
i2	a16	0.548089576172037
i2	a17	0.156070036579080
i2	a19	0.451910423827964
i3	a3	0.170951312796997
i3	a4	0.829048687203003
i3	a5	0.170951312796997
i3	a7	0.170951312796997
i3	a9	0.170951312796997
i3	a11	0.170951312796997
i3	a12	0.829048687203003
i3	a14	0.829048687203003
i3	a15	0.829048687203003
i4	a4	1.0000000000000000
i4	a13	1.0000000000000000
i4	a19	1.0000000000000000

Tabla 4.5: Matriz de proporciones $P(i, a)$

muestra su contenido.

Del mismo modo se resuelve una perturbación de la matriz de demandas al 115 % para servir de base en la estimación. La forma de perturbar se explicó en detalle en este mismo capítulo. Tras resolver esta red perturbada tenemos que los caminos elegidos para cubrir la demanda son los mismos que en la matriz original. Los arcos que forman cada camino son idénticos. La diferencia radica tanto en el flujo como en el coste, puesto que estos dependen del número de viajes en la red. En la tabla 4.6 se encuentran los flujos y los coste para la red perturbada y en la tabla 4.7 aparece la matriz de proporciones $P(i, a)$

Demanda	Camino	Flujo	Coste
$i_1(1 \rightarrow 2)$	a2 - a11 - a18	459.99	50.06
$i_2(1 \rightarrow 3)$	a1 - a5 - a7 - a10 - a16	272.77	58.05
	a2 - a8 - a14 - a16 - a17	171.07	58.05
	a1 - a6 - a13 - a19	410.65	58.05
	a1 - a5 - a8 - a14 - a16	65.49	58.05
$i_3(4 \rightarrow 2)$	a4 - a12 - a14 - a15	578.00	48.85
	a3 - a5 - a7 - a9 - a11	111.99	48.85
$i_4(4 \rightarrow 3)$	a4 - a13 - a19	230.00	45.64

Tabla 4.6: Caminos usados en equilibrio para la red perturbada

Par	Arco	Proporción	$\delta_{i,a}$
i1	a2	1.0000000000000000	1
i1	a11	1.0000000000000000	1
i1	a18	1.0000000000000000	1
i2	a1	0.814044600629254	1
i2	a2	0.185955399370746	1
i2	a5	0.367680992799100	1
i2	a6	0.446363607830148	1
i2	a7	0.296492991155365	1
i2	a8	0.257143401014488	1
i2	a10	0.296492991155365	1
i2	a13	0.446363607830148	1
i2	a14	0.257143401014488	1
i2	a16	0.553636392169852	1
i2	a17	0.185955399370746	1
i2	a19	0.446363607830148	1
i3	a3	0.162314074437436	1
i3	a4	0.837685925562565	1
i3	a5	0.162314074437436	1
i3	a7	0.162314074437436	1
i3	a9	0.162314074437436	1
i3	a11	0.162314074437436	1
i3	a12	0.837685925562565	1
i3	a14	0.837685925562565	1
i3	a15	0.837685925562565	1
i4	a4	1.0000000000000000	1
i4	a13	1.0000000000000000	1
i4	a19	1.0000000000000000	1

Tabla 4.7: Matriz de proporciones $P_{i,a}$ de la red perturbada y su correspondiente matriz $\delta_{i,a}$

4.1.2. Evaluación computacional de los problemas de localización (TCL)

Como se ilustra en la figura 4.1, con la fase anterior se generan los datos de entrada para los modelos TCL. Como puede observarse, se requiere resolver varias veces los modelos TCL con diferentes entradas. Se ha optado por separar los modelos de los ficheros de datos y esto hace que sea más fácil tratarlos. En el problema anterior, si se trataba de una red *congestionada* o no, se indicaba en el algoritmo. En el caso que nos ocupa se da un caso u otro según el fichero de entrada (salida del TAP) que tomemos, es decir, si se toma la entrada para el problema de asignación *sin congestión* generaremos un fichero con los datos de este tipo.

La ejecución del DSD permite la generación de los datos necesarios para los modelos TCL-P1 y TCL-P2. Para estos modelos se han realizado fundamentalmente dos tareas:

- Depuración de datos

A continuación se detallan el origen de los datos y el tratamiento dado en caso de que sea necesario:

- $G(i)$ Demandas: se toma directamente del fichero de la red.
- $F(k)$ Flujos: se toman del fichero de salida del TAP-DSD todos aquellos flujos cuyo valor sea mayor que 0.
- $\delta_{i,a}$ Incidencia par-arco: se calcula con la matriz de proporciones $P(i, a)$ del fichero de salida del TAP-DSD.
- $D(i)$ Destino del par i : se toma directamente del fichero de la red ($LADEM(i)$).
- $\delta_{a,k}$ Incidencia arco-camino: se toma del fichero de salida del TAP-DSD.
- $C(i)$ Subconjunto de pares: se utiliza para limitar el número de demandas de la red a resolver.

La salida del algoritmo TCL es un fichero de texto con el resultado de los dos problemas, Modelo de cubrimiento TCL-P1 y Modelo de máxima interceptación de flujo TCL-P2. Puede verse más adelante en la implementación en GAMS, que ambos modelos se abordan en el mismo programa. En primer lugar indica el número de arcos aforables en la red. A continuación aparecen los resultados del primer problema, que se componen del número de arcos aforados y de una lista indicando cada uno de ellos. Los resultados del

segundo problema dan más información. Antes de la lista de los arcos aforados aparece una serie de datos:

- El número de aforos o arcos aforados, igual que en el problema anterior.
 - El flujo bruto aforado, suma del flujo de todos los caminos que atraviesan un arco aforado y que se calcula según: $\sum_k h_k \sum_a \delta_{a,k} z_a$ donde h_k es el flujo en el camino k , $\delta_{a,k}$ es la matriz de incidencia arco-camino, que indica que el arco a forma parte del camino k y z_a indica la presencia de aforo en el arco a .
 - El flujo neto aforado, que es la función objetivo del problema, el flujo capturado en los arcos aforados.
 - la demanda interceptada, que se calcula de acuerdo a $\sum_i k_i D(i)$, donde k_i toma el valor 1 si $\sum_a \delta_{i,a} z_a \geq 1$ y 0 en otro caso. $D(i)$ es el destino del par i .
 - Los caminos interceptadas, caminos que atraviesan un arco aforado que se calculan según $\sum_k y_k$.
- Programa en GAMS para la resolución de los modelos TCL-P1 y TCL-P2

```

$title MODELO DE CUBRICION

OPTION
LIMCOL=0
LIMROW=0
SOLPRINT=OFF
PROFILE=1
iterlim=5000000
reslim=1000000;
*mip=cplex;
*stepsum=1;

* Redes sin congestión
*$INCLUDE      "C:\gams\Localizacion\ngd\Datos_ngd_sc.gms";
* Redes con congestión
*$INCLUDE      "C:\gams\Localizacion\ngd\Datos_ngd_c.gms";

FILE

* Resultados SIN congestion
* resultados /C:\gams\Localizacion\resultados\ngd_sc.txt/;
* Resultados CON congestion
* resultados /C:\gams\Localizacion\resultados\ngd_c.txt/;

put resultados;

VARIABLES *x(i)      pares cubiertos

```

```

*Npar(i)  numero de veces que
*se puede interceptar un par con el numero de aforos dados
*Nreal(i) numero de veces que se intercepta el par r(i)

aux      pares cubiertos para calcularlos en la rutina

y(k)     existencia de aforo en la ruta

z(a)     existencia de aforo en el arco a

P_1      numero minimo de aforos

P_2      Flujo interceptado

BINARY VARIABLE z,y,x; Integer variable Npar,Nreal;

Scalar L numero de aforos;

Scalar W numero de pares interceptados;

Scalar Fa flujo aforado;

Scalar Di Demanda interceptada;

EQUATIONS

COST_p1   Funcion objetivo modelo de cubrimiento

COST_p2   Funcion objetivo modelo de captacion de flujo neto
maximo

cubr(i)   Restriccion de cubrimiento de pares

Aa        Restriccion en el numero de arcos aforados

Ar(k)     Restriccion de existencia de aforos en las rutas ;

*cubr(i)$ (c(i)).. SUM(a,DELTA_ia(i,a)*z(a)) =g= c(i);

Aa.. SUM(a,z(a))=e= L;

Ar(k).. SUM(a,Delta_ak(a,k)*z(a)) =g= y(k);

COST_p1..          P_1 =e= SUM(a,z(a));

COST_p2..          P_2 =e= SUM(k,F(k)*y(k));

MODEL tcl_p1 /cost_p1,cubr/;

MODEL tcl_p2 /cost_p2,aa,ar,cubr/;

```

```

put 'Numero de arcos: ' card(a)/; put /;

*****

*MODELO P1 DE CUBRIMIENTO

*****

SOLVE tcl_p1 using mip minimizing P_1;

put 'MODELO DE CUBRIMIENTO P1'; put/; put/; put 'numero minimo de
aforos ' ;put P_1.l:30:0; Put/;put/;put/;

*****

*RUTINA PARA ENUMERAR LOS ARCOS AFORADOS

*****

put 'arcos a aforar' put/; loop(a, if(z.l(a) gt 0, put a.tl; put
/;); );

put
'*****' put/;
put
'*****' put/;
put/;put/;

*****

*MODELO P2 DE INTERCEPTACION DE MAXIMO FLUJO NETO

*****

put 'MODELO DE INTERCEPTACION DE MAXIMO FLUJO NETO P2';
put/;put/;

*ENCABEZADOS DE LAS COLUMNAS

put ' aforos'; put ' flujo bruto aforado'; put ' flujo neto
aforado';
put ' Demanda interceptada'; put ' pares interceptados';
put ' rutas interceptadas'; put /; put/;

SOLVE tcl_p2 using mip maximicing P_2;

*CALCULO DEL FLUJO BRUTO AFORADO

```

```

Fa=SUM(k,F(k)*SUM(a,Delta_ak(a,k)*z.l(a)));

*CALCULO DE LA DEMANDA INTERCEPTADA

loop (i, if (SUM(a,DELTA_ia(i,a)*z.l(a)) ge 1, r.l(i) = 1;
           else r.l(i) = 0;)););

Di=SUM(i,r.l(i)*D(i));

*CALCULO DEL NUMERO DE PARES QUE SE AFORAN

w=SUM(i,r.l(i));

*CALCULO DEL NUMERO DE RUTAS QUE SE INTERCEPTAN MAS DE UNA VEZ

Rutas= sum(k,y.l(k));

*IMPRIMIMOS LOS DATOS CALCULADOS

put L:5:0; PUT Fa:25:0; put P_2.1:25:0; put Di:20:0; put W:20:0;
put rutas:20:0 put/;

*****

*RUTINA PARA ENUMERAR LOS ARCOS AFORADOS

*****

put 'arcos a aforar' put/; loop(a, if(z.l(a) gt 0, put a.tl; put
/;));)

```

En el código hay dos puntos a destacar:

- Hay una línea que limita el tratamiento de los pares a aquellos válidos. Estos pares se identificaron en la matriz $c(i)$ como se indicó anteriormente en este apartado.
- El modelo TCL-P2 descrito en la literatura científica tenía un bucle originalmente. Como se explicó en el capítulo de antecedentes, este modelo parte de un número de sensores y

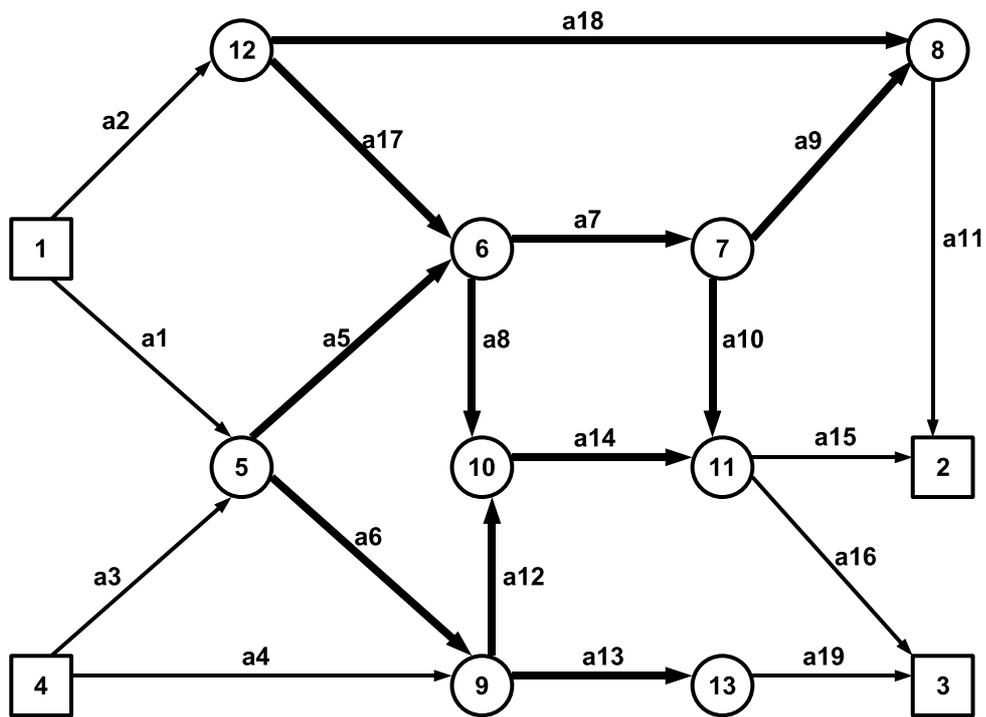


Figura 4.4: Arcos no conectores de la red Nguyen-Dupuis en trazo grueso

resuelve el modelo. Este bucle hace que el modelo haga más iteraciones aumentando el número de sensores hasta aforar la red completamente. Como en este caso el número de sensores viene dado del modelo TCL-P1 se ignoró este bucle.

Ejemplo

Lo primero que hay que hacer antes de comenzar con este problema es hacer una selección de los arcos *no conectores* (arcos reales), es decir tenemos que limitar el conjunto de arcos a un conjunto que sea aforable (arcos observables en la realidad).

El primer paso para esto es determinar los centroides de la red. En este caso ya se tiene el dato de que los nodos 1,2,3 y 4 son centroides. Una vez determinados se marcan todos los arcos cuyo origen o destino es un centroide para dejar el resto como el subconjunto de arcos no conectores. En la figura 4.4 aparecen en trazo grueso aquellos arcos aforables.

El modelo TCL puede localizar hasta 11 sensores de tráfico. Estos sensores de tráfico forman el conjunto Z . Al descartar los arcos conectores el conjunto Z está restringido, y lo forman las variables $z_5, z_6, z_7, z_8, z_9, z_{10}, z_{12}, z_{13}, z_{14}, z_{17}, z_{18}$. Como se explicó anteriormente, se parte de

una matriz O-D de referencia, que se ha supuesto $1,15g$, siendo g la matriz verdadera. A partir de la matriz $P_{i,a}$ se obtiene la correspondiente matriz de incidencia par-arco, $\delta_{i,a}$ en la tabla 4.7, y con ésta el modelo TCL-P1 se plantea de la siguiente manera:

$$\text{Minimizar : } z_5 + z_6 + z_7 + z_8 + z_9 + z_{10} + z_{12} + z_{13} + z_{14} + z_{17} + z_{18}$$

sujeto a :

$$z_{18} \geq 1$$

$$z_5 + z_6 + z_7 + z_8 + z_{10} + z_{12} + z_{13} + z_{14} + z_{17} \geq 1$$

$$z_5 + z_7 + z_9 + z_{14} \geq 1$$

$$z_{13} \geq 1$$

$$z_5, z_6, z_7, z_8, z_9, z_{10}, z_{12}, z_{13}, z_{14}, z_{17}, z_{18} \in \{0, 1\}$$

Si ejecutamos el algoritmo de Localización TCL-P1 el resultado aparece en la figura 4.5, donde los sensores de tráfico se localizan en los arcos de trazo grueso.

Un hecho a destacar es que este problema no tiene solución única, por ejemplo z_{11}, z_{13}, z_{18} o z_7, z_{13}, z_{18} . satisfacen igualmente las restricciones del problema. Con el modelo TCL-P2 el objetivo es encontrar entre las soluciones óptimas de TCL-P1 aquella que tenga el máximo flujo neto de entre todas ellas.

Cuando se resuelve Modelo de Localización TCL-P2 el resultado es el que aparece en la figura 4.6. Al igual que en la figura anterior están marcados en trazo grueso los arcos con sensores de tráfico.

Para formular el modelo TCL-P2 se introdujeron tantas nuevas variables como caminos hay en el equilibrio (ver tabla 4.6). Por lo tanto se introduce el conjunto de variables Y , que indican la existencia de aforo del camino k . Este conjunto está formado por: $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8$.

La función objetivo queda de la siguiente manera:

$$\text{Maximizar } 459,99 * y_1 + 272,77 * y_2 + 171,07 * y_3 + 410,65 * y_4 \\ + 65,49 * y_5 + 578,00 * y_6 + 11,99 * y_7 + 230,00 * y_8$$

Sujeto a restricciones del modelo TCL – P1

Restricciones adicionales :

$$\begin{aligned}
z_5 + z_6 + z_7 + z_8 + z_9 + z_{10} + z_{12} + z_{13} + z_{14} + z_{17} + z_{18} &= 3 \\
z_{18} &\geq y_1 \\
z_5 + z_7 + z_{10} &\geq y_2 \\
z_8 + z_{14} + z_{17} &\geq y_3 \\
z_6 + z_{13} &\geq y_4 \\
z_5 + z_8 + z_{14} &\geq y_5 \\
z_{12} + z_{14} &\geq y_6 \\
z_5 + z_7 + z_9 &\geq y_7 \\
z_{13} &\geq y_8 \\
y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8 &\in \{0, 1\}
\end{aligned}$$

La localización de estos sensores en la figura 4.5 responde a las restricciones del modelo y a la estructura de la red. Como puede verse en la tabla 4.4 hay dos pares con un único camino, lo que explica en concreto los sensores en los arcos a_{13} y a_{18} . El otro sensor se coloca en a_5 puesto que dicho arco consigue cubrir las otras dos demandas. Para tener mejor una idea del motivo por el que se aforan estos arcos es conveniente mirar la figura 4.4 y el planteamiento anterior. En esta figura se observa que estos arcos son atravesados por caminos pertenecientes a diferentes demandas, lo que implica que con un sensor se capta más de una demanda. Los arcos a_{13} y a_{18} son atravesados por un camino único, por lo que no queda más remedio que aforarlos para cubrir la demanda que satisfacen.

La localización de los sensores en la figura 4.6 es diferente al modelo anterior. Esto es, puesto que en el modelo anterior sólo se buscaba cubrir todas las demandas y ahora el objetivo es capturar el máximo flujo posible. Los arcos a_{13} y a_{18} tienen que aforarse de nuevo obligatoriamente para poder capturar el flujo de esas dos demandas satisfechas por un único camino. El otro arco aforado, a_{14} , es el que nos permite capturar el máximo flujo de las otras dos demandas. También en este caso es interesante mirar la figura 4.3 para darnos cuenta de que dicho arco es atravesado por varios caminos y la tabla 4.4 junto con la tabla 4.5 para ver su flujo.

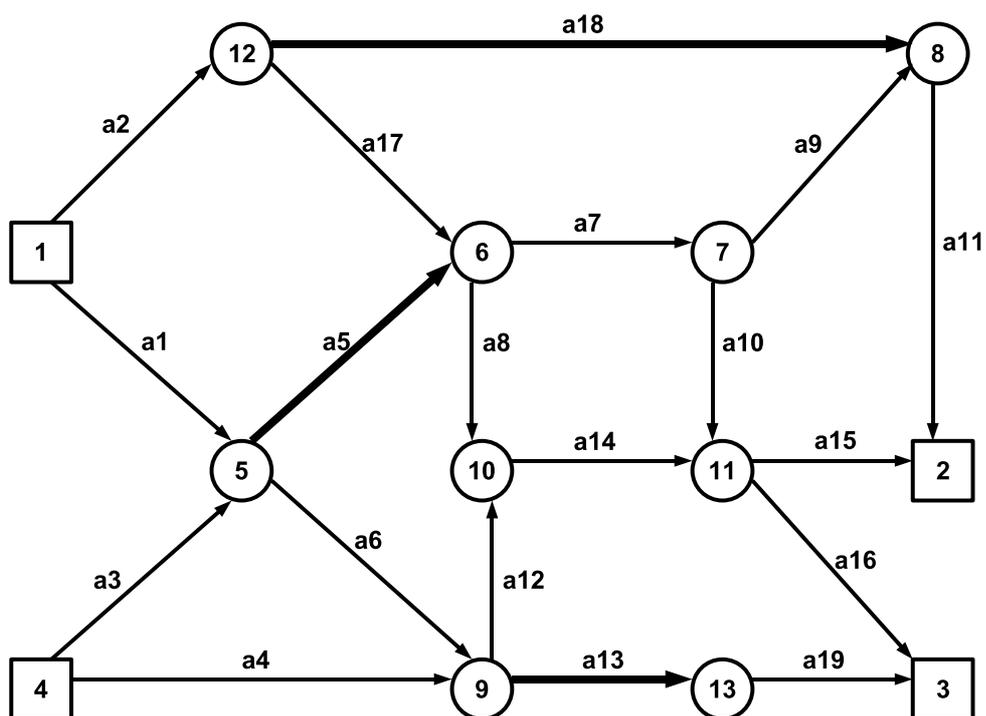


Figura 4.5: Localización de sensores por el Modelo TCL-P1 de la red Nguyen-Dupuis

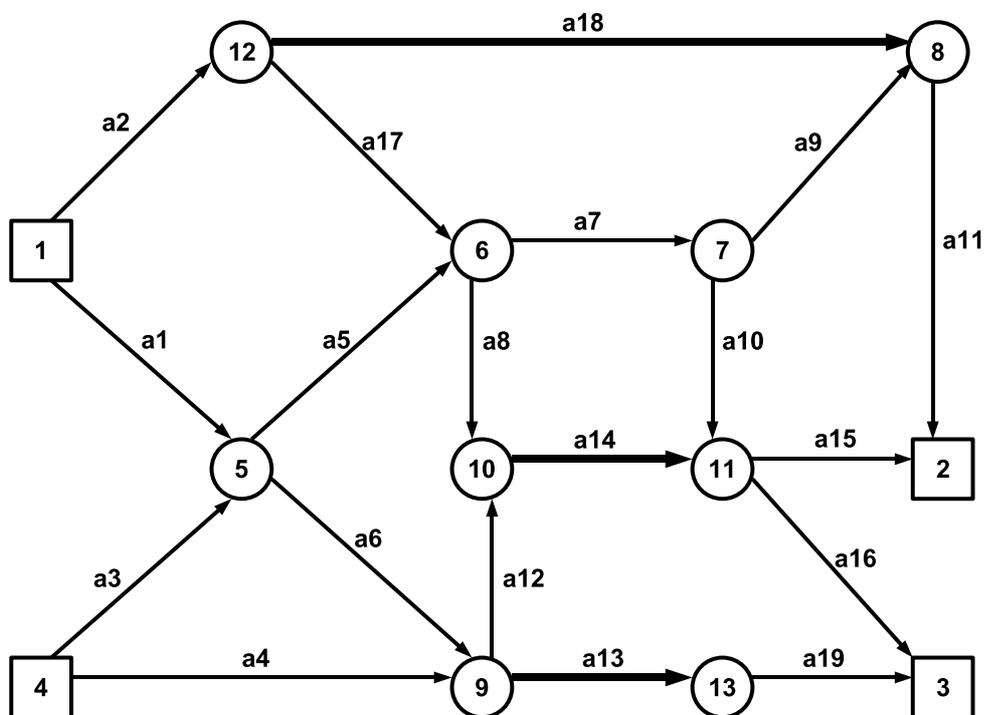


Figura 4.6: Localización de sensores por el Modelo TCL-P2 de la red Nguyen-Dupuis

4.2. Metodología empleada para el objetivo 2

4.2.1. Fase de generación de datos

Para lograr el segundo objetivo se realizó un estudio de simulación mostrado en la figura 4.7. Con este estudio se pretende valorar la calidad de las matrices estimadas en función de la calidad de la localización de los sensores de tráfico.

La filosofía de este estudio es hacer una simulación de una situación práctica, en la que los aforos contienen información de la verdadera (actual) matriz O-D. Se dispone de esta información junto a una matriz de referencia.

La matriz verdadera se obtuvo perturbando la target matrix de las siguientes formas:

- Se multiplicó por el factor "0,8" para quedar como el 80 % de la matriz verdadera.
- Se multiplicó por el factor "1,3" para quedar como el 130 % de la matriz verdadera.
- Se supuso que la target matrix era una estimación insesgada de la verdadera matriz O-D y se asumió que el error de la estimación del par i era una normal de media 0 y de desviación típica ρg_i donde ρ se escogió el 0,03 para todas las redes salvo para Barcelona, caso en el que escogió el valor de 1,3 que resultaban más adecuados.

4.2.2. Resolución del problema de estimación de matrices O-D

En el capítulo de Antecedentes se introdujeron diferentes modelos empleados en la estimación de matrices O-D. En este PFC se ha optado por *modelos proporcionales (sin congestión)*. Más concretamente, se ha empleado un estimador mínimo cuadrático para la situación en la que se dispone de una matriz de referencia ([Ortuzar and Willumsen 1994]). El otro modelo empleado es el *modelo entrópico* dado en la misma referencia que el anterior, en el caso de prescindir de matriz de referencia y disponer únicamente de aforos. Estos modelos se han mostrado en el capítulo de Antecedentes.

Ambos modelos se han codificado en GAMS. El código del primero es el siguiente:

```
* MODELO 1 DE ESTIMACION DE MATRICES MEJORADO
* (Improved matrix estimation model)

OPTION LIMCOL=0 LIMROW=0 SOLPRINT=OFF nlp=minos; *PROFILE=1;
```

```

* Contiene los datos perturbados.
* Flujo observado y matriz de proporciones P(i,a)
* Ademas tiene la salida correspondiente red_sc/c_80/110/130/rd20/rd30.txt

*****
* NGD - ngd
*****
* Redes sin congestion
$include "ngd/datos_ngd_sc_80.gms";

*$include "ngd/datos_ngd_sc_130.gms";

*$include "ngd/datos_ngd_sc_rd30.gms";

*Redes congestionadas

*$include "ngd/datos_ngd_c_80.gms";

*$include "ngd/datos_ngd_c_130.gms";

*$include "ngd/datos_ngd_c_rd30.gms";

VARIABLES

est1          Error en la estimacion del modelo 1

positive variable G_est(i) Matriz a estimar;

SCALAR

diferencia          Diferencia para la matriz estimada /0/;

EQUATIONS

fobj_modelo1        Funcion objetivo del modelo de estimacion 1

rest1(a)            Restriccion1: flujo observado - viajes = 0

mayor(i)            Todo viaje mayor que 0;

rest1(a)$afor(a)..  Vobs(a) =e= sum( i , G_est(i)*P(i,a) );
mayor(i)..          G_est(i) =g= 0.00000001;

fobj_modelo1..      est1 =e=
                    0.5*sum(i, (G_est(i)-G(i)) * (G_est(i)-G(i))/G(i));

MODEL estimacion1 /fobj_modelo1,rest1,mayor/;
estimacion1.workspace=1000;
estimacion1.optfile=1;
estimacion1.reslim=100000.0;
estimacion1.iterlim=100000;

*Solucion inicial
g_est.l(i)=g(i);

```

```

SOLVE estimacion1 using nlp minimizing est1;

* Imprimir resultados
put 'Modelo 1: Modelo de estimacion de matrices mejorado' // /;

put 'Funcion objetivo =' est1.l:30:10/ /;

* Matriz estimada
put 'G_est(i) Matriz estimada'//;
loop (i,
  put i.tl G_est.l(i):30:15 /;
); put ' '// /;

* Diferencia de la estimacion
put 'Diferencia estimada-referencia'//;
loop (i,
  put i.tl (G_est.l(i)-G_p(i)):30:15 /;
); put ' '// /;

loop (i, diferencia = diferencia + power(G_est.l(i)-G_p(i),2) );
diferencia = sqrt(diferencia)/card(i); put 'Diferencia: '
diferencia:20:10 /;

```

Para el segundo modelo se tiene el código:

```

* MODELO 2 DE ASIGNACION NO PROPORCIONAL
* (Non-proportional assignment)

OPTION LIMCOL=0 LIMROW=0 SOLPRINT=OFF nlp=minos; *PROFILE=1;

* Contiene los datos perturbados.
* Flujo observado y matriz de proporciones P(i,a)
* Ademas tiene la salida correspondiente red_sc/c_80/110/130/rd20/rd30.txt

*****
* NGD - ngd
*****
* Redes sin congestion
$include "ngd/datos_ngd_sc_80.gms";

*$include "ngd/datos_ngd_sc_130.gms";

*$include "ngd/datos_ngd_sc_rd30.gms";

*Redes congestionadas

*$include "ngd/datos_ngd_c_80.gms";

*$include "ngd/datos_ngd_c_130.gms";

*$include "ngd/datos_ngd_c_rd30.gms";

VARIABLES

```

```

est2          Error en la estimacion del modelo 2

positive variable G_est(i) Matriz a estimar;

SCALAR

diferencia    Diferencia para la matriz estimada /0/;

EQUATIONS

fobj_modelo2   Funcion objetivo del modelo de estimacion 2

rest1(a)       Restriccion1: flujo observado - viajes = 0

mayor(i)       Todo viaje mayor que 0;

rest1(a)$afor(a)..  Vobs(a) =e= sum( i , G_est(i)*P(i,a) );
mayor(i)..       G_est(i) =g= 0.00000001;

fobj_modelo2..  est2 =e= sum( i ,G_est(i)*(log(G_est(i))- 1.0 ) );

MODEL estimacion2 /fobj_modelo2,rest1,mayor/;
estimacion2.workspace=1000; estimacion2.optfile=1;
estimacion2.reslim=100000.0; estimacion2.iterlim=100000;

*Solucion inicial g_est.l(i)=g(i);

SOLVE  estimacion2 using nlp minimizing est2;

put / 'Modelo 2: Modelo de asignacion no proporcional' / /; put
'Funcion objetivo =' est2.l:20:15 / /;

* Matriz estimada
put 'G_est(i) Matriz estimada' / /; loop (i$(G(i) gt 0.0),
  put i.tl G_est.l(i):30:15 /;
); put ' / / / /;

* Diferencia de la estimacion
put 'Diferencia estimada-referencia' / /; loop (i$(G(i) gt 0),
  put i.tl (G_est.l(i)-G_p(i)):30:15 /;
); put ' / / /;

diferencia = 0;

loop (i, diferencia = diferencia + power(G_est.l(i)-G_p(i),2) );

diferencia = sqrt(diferencia)/card(i);

put 'Diferencia: ' diferencia:20:15 /;

```

En el código implementado de ambos modelos hay varios puntos a destacar:

- Las primeras líneas contienen las opciones habituales de GAMS.

- Las líneas siguientes se emplean para indicar el fichero de entrada al modelo.
- Aunque parezca redundante forzar que $G_{est}(i)$ sea positiva y a su vez tenga una restricción $G_{est}(i) > 0,00000001$, se hace de esa manera para evitar problemas como divisiones por cero y logaritmos de números negativos.
- La solución inicial es una manera de llegar más rápidamente a la solución.

Para evaluar los modelos de localización TCL-P1 y TCL-P2 se añadió una regla de control para proporcionar aforos. Esta regla consistió en tomar aforos de forma aleatoria. El número de aforos que se tomó fue el indicado en el modelo TCL-P1, al igual que hace el modelo TCL-P2.

Esta regla sirve de control y permite comprobar si el hecho de elegir los aforos con un criterio aumenta la información obtenida en la muestra.

La comparación de los resultados se hizo perturbando las matrices O-D redes de tráfico, resolviéndolas y luego tomando los aforos en los arcos indicados por los modelos TCL y por la otra regla. Estas perturbaciones permiten trabajar con la misma red bajo condiciones de demanda diferentes. Como la matriz de la que provienen los aforos es conocida (matriz original), se puede comparar con la matriz obtenida en el proceso de estimación.

Puede verse que el estudio requiere resolver tres problemas de optimización. En la parte superior izquierda se encuentra el problema de asignación de tráfico (TAP), a su derecha se encuentra el de localización (TCL) y bajo ambos se encuentra el problema de estimación (DAP). En el diagrama quedan indicados tanto los modelos como la entrada y salida de datos de ellos. Puede verse que la target matrix O-D es dato de entrada tanto del problema de localización como de estimación.

En el fichero de entrada se recogen los datos necesarios para la estimación. Uno de ellos es un conjunto de aforos para una localización de sensores. Estos se toman de la salida del TCL o de otra técnica de aforación (esto se verá en detalle en el apartado siguiente). El contenido del fichero de entrada es el siguiente:

- $G(i)$ de la red original tal y como se describe para el problema de la asignación de tráfico (TAP-DSD).
- $P(i, a)$ de la salida del problema de asignación de tráfico (TAP-DSD).

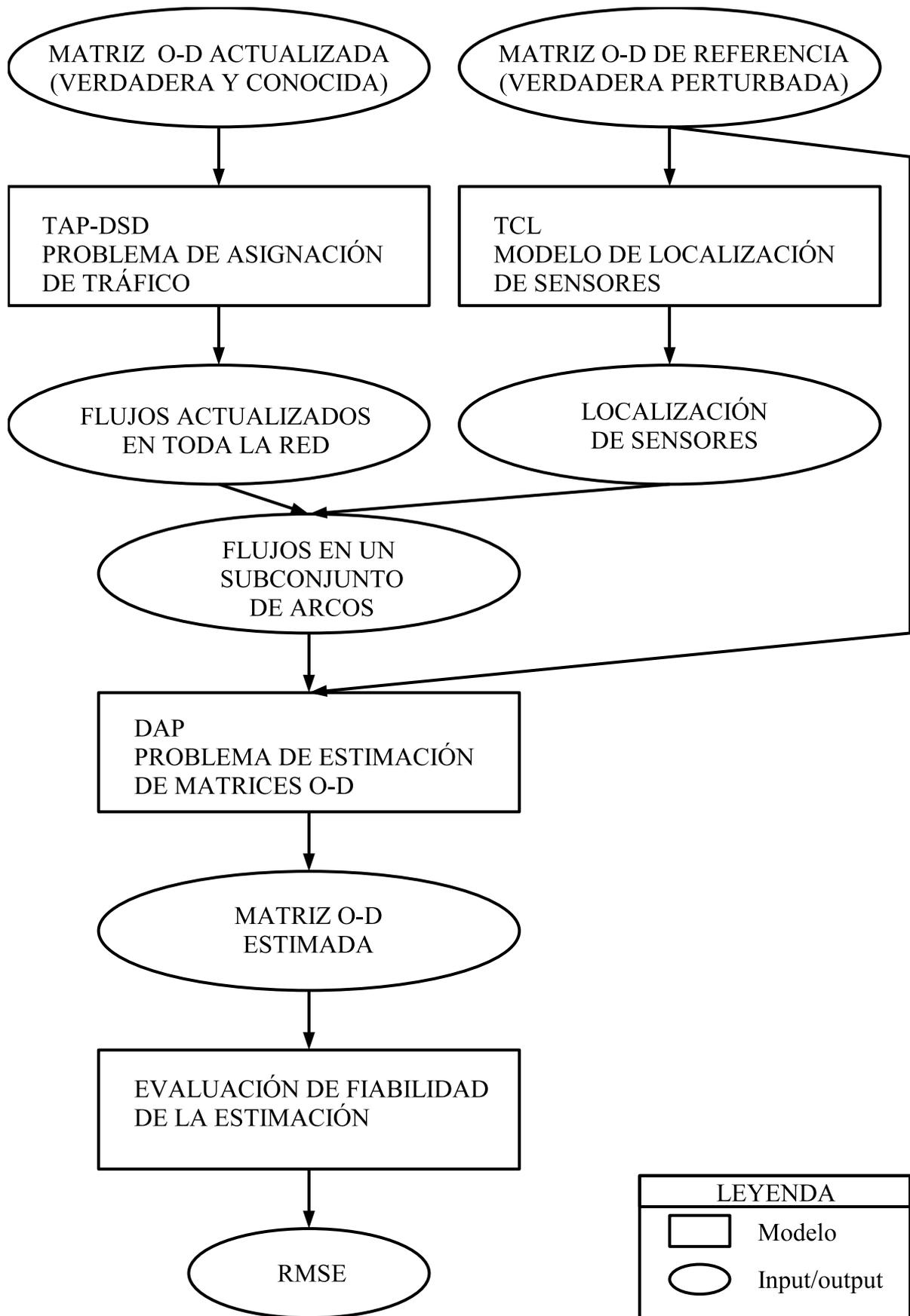


Figura 4.7: Diagrama del estudio

- $G_p(i)$ matriz de demandas perturbada (verdadera), se calcula multiplicando $G(i)$ por el factor adecuado.
- $\delta_{i,a}$ matriz de incidencia par-arco, calculada a partir de la matriz $P(i, a)$ como se ha hecho en otros algoritmos.

La salida del algoritmo nos da los resultados para los modelos de estimación empleados. El contenido del fichero de salida es el siguiente:

- El valor de la función objetivo.
- La matriz estimada $\hat{G}(i)$.
- La matriz perturbada $G_p(i)$, dato de entrada.
- La diferencia entre ellas al compararlas de siguiendo el *Root mean square error* (RMSE): $\frac{\sqrt{\sum_i (\hat{G}(i) - G_p(i))^2}}{\text{card}(i)}$, donde $\text{card}(i)$ es el número de elementos de I , es decir, el número de demandas.

Las tablas con los resultados de los diferentes problemas se encuentran en el capítulo siguiente.

Ejemplo

En este ejemplo se trata de estimar la matriz O-D según los aforos conseguidos de los modelos de localización, de tomar aforos aleatoriamente o tomar los de mayor flujo de la red. La fiabilidad de la matriz obtenida nos dará idea de la calidad del método de localización empleado.

Como se dijo al principio del ejemplo, la matriz que queremos estimar es la misma perturbada con una demanda a un 115 % de demanda respecto de la original. Tenemos los datos de esta perturbación una vez asignada y una serie de arcos aforados.

Modelo de estimación proporcional mínimo cuadrático con matriz de referencia

Se comienza planteando el modelo de estimación mínimo cuadrático con matriz de referencia. La matriz de verdadera es la matriz O-D perturbada. Lo que se pretende es estimarla a partir de aforos de la red y de la matriz O-D de referencia (matriz O-D original). Dado que esta

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	460.00	0.00
i2	920.00	943.77	23.77
i3	690.00	636.14	53.85
i4	230.00	219.38	10.61
RMSE			14.95

Tabla 4.8: Estimación con matriz de referencia con localización TCL-P1

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	460.00	0.00
i2	920.00	904.60	15.39
i3	690.00	694.72	4.72
i4	230.00	236.87	6.87
RMSE			4.37

Tabla 4.9: Estimación con matriz de referencia con localización TCL-P2

matriz verdadera y la estimada son conocidas, pueden compararse para evaluar la calidad de la estimación.

A lo largo de las tablas siguientes se van a mostrar las matrices que se han obtenido por medio de modelos de estimación para varios planes de aforamiento. La medida que aparece en la última fila de cada tabla es el Error Cuadrático Medio, calculado como se indicó anteriormente.

En el caso de localización aleatoria no se han indicado las diez matrices que estima el método en realidad, sino que se han mostrado sólo cinco de ellas. Debajo de cada matriz aparece su media y en la parte inferior izquierda aparece la media aritmética de los resultados de cada matriz estimada.

En esta red el mejor resultado lo consigue el modelo de localización TCL-P2. El método de localización consigue mejores resultados que el TCL-P1 porque la cantidad de flujo capturada

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	459.08	0.91
i2	920.00	901.54	18.45
i3	690.00	695.66	5.66
i4	230.00	238.23	8.23
RMSE			5.25

Tabla 4.10: Estimación con matriz de referencia con localización por máximo flujo

Par	Matriz verdadera	Estimada 1	Estimada 2	Estimada 3	Estimada 4	Estimada 5
i1	460.00	400.00	460.00	400.00	400.00	476.99
i2	920.00	920.00	920.00	904.60	800.00	828.63
i3	690.00	690.00	600.00	694.72	599.99	600.00
i4	230.00	200.00	200.00	236.87	199.99	200.00
Media estimaciones: 26.11		16.77	23.76	15.70	41.09	33.23

Tabla 4.11: Estimación con matriz de referencia con localización aleatoria

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	460.00	0.00
i2	920.00	447.93	472.06
i3	690.00	2.32	687.67
i4	230.00	192.71	37.28
RMSE			208.73

Tabla 4.12: Modelo entrópico con localización TCL-P1

es mayor. Podría decirse que se trata de una aproximación al modelo TCL-P2 más relajado. El método de localización aleatorio no obtiene buenos valores en ninguna de sus estimaciones, si unimos los malos resultados con el tiempo empleado, unas diez veces mayor que el resto, hace que sea el método menos indicado.

Modelo de estimación de máxima entropía (ME)

En el modelo anterior nos servíamos de una red de referencia para poder estimar una nueva matriz. En este modelo, la matriz de referencia desaparece, con lo que en la función objetivo del modelo lo que se pretende es maximizar la entropía.

A lo largo de las siguientes tablas hay que tener en cuenta las aclaraciones del punto anterior.

En esta red tan pequeña los resultados de estimación sin matriz de referencia obtienen unos

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	460.00	0.00
i2	920.00	637.06	282.93
i3	690.00	177.50	512.49
i4	230.00	3.58	226.41
RMSE			156.91

Tabla 4.13: Modelo entrópico con localización TCL-P2

Par	Matriz verdadera	Matriz estimada	Diferencia
i1	460.00	187.76	272.23
i2	920.00	430.34	489.65
i3	690.00	384.22	305.77
i4	230.00	210.30	19.69
RMSE			159.63

Tabla 4.14: Modelo entrópico con localización por máximo flujo

Par	Matriz Referencia	Estimada 1	Estimada 2	Estimada 3	Estimada 4	Estimada 5
i1	460.00	1.00	1.00	315.53	394.50	1.00
i2	920.00	272.77	637.06	315.53	236.57	192.38
i3	690.00	111.99	177.50	1.00	1.00	192.38
i4	230.00	1.00	3.58	1.00	1.00	1.00
Media estimaciones: 238.05		252.00	194.44	238.97	249.84	255.00

Tabla 4.15: Modelo entrópico con localización aleatoria

resultados mucho peores que en el caso anterior. No se puede decir que el modelo localización TCL-P2 y que el método de localización por máximo flujo tengan buenos resultados, pero son más recomendables que los restantes. En el capítulo de resultados puede verse como para redes de gran tamaño la estimación obtiene mayor fiabilidad.

Capítulo 5

Resultados

5.1. Descripción de los datos

En las pruebas realizadas, se ha trabajado con las redes de tráfico descritas en la tabla 5.1.

El origen de las redes es diferente. La red ficticia de Nguyen-Dupuis puede consultarse en [Nguyen and Dupuis1984]. La red de Ciudad Real la proporcionó el Departamento de Ingeniería de transporte de la UCLM. Las redes de Winnipeg y Barcelona se encuentran disponibles en la web de Bar-Gera (<http://www.bgu.ac.il/bargera/tntp/>).

Las pruebas se han realizado sobre un PC y sobre los ordenadores del servicio de supercomputación (SSC), cuyas especificaciones aparecen en la tabla 5.2. Los tiempos están con una precisión de milésima.

5.2. Problema de asignación de tráfico (TAP-DSD)

En la tabla 5.3 se encuentran recogidos los resultados del algoritmo TAP-DSD para las diferentes perturbaciones de la matriz O-D consideradas. Las redes de Winnipeg y Barcelona resul-

Redes	Nº demandas	Nº arcos	Nº nodos	Nº centroides
Nguyen-Dupuis	4	19	13	4
Ciudad Real	456	546	230	25
Winnipeg	4345	2836	1052	140
Barcelona	7922	2522	1020	108

Tabla 5.1: Redes de tráfico

	Procesadores	Memoria RAM
PC	1xPentium 4 3.0 GHz	1 GB
SSC (Cantor)	8xAlpha EV67 667 MHz	10 GB
SSC (Mulhacen)	8xAlpha EV67 667 MHz	4 GB
SSC (Aneto)	2xAlpha EV65 600 MHz	1 GB

Tabla 5.2: Recursos hardware

taron inabordables para el PC, estas redes se resolvieron en el Servicio de Supercomputación (SSC) de la UCLM. En dicho sistema se garantiza una máquina para el proceso que se está ejecutando, pero no se puede asegurar que se emplee el mismo tiempo al ejecutar el mismo proceso varias veces. Esto ocurre de esta manera porque el servicio tiene varias máquinas con diferentes recursos y la configuración de E/S no asegura un disco para cada proceso, sino que asignada el proceso a la máquina que con menos carga procesa en ese momento. De esta forma, se ve claramente en la red de Barcelona que algunas perturbaciones emplean el doble de tiempo para su resolución. En el caso de la red de Winnipeg no es tan evidente, la mayoría de los tiempos se mueven en las mismas cifras salvo tres casos en los que notamos que prácticamente emplean la mitad de tiempo. Esto se debe a que la máquina menos potente estaba con menos carga de trabajo en ese momento y el sistema nos la asignó para tener el resultado lo antes posible, es decir, aseguró el mejor tiempo de respuesta del sistema, pero no el tiempo de ejecución del proceso. Esto no fue un problema, ya que el objetivo era evaluar computacionalmente los modelos, y hasta en la máquina menos potente el objetivo se alcanzó. Las máquinas empleadas en la tabla 5.3 fueron Aneto y Mulhacen. Si se mira junto a la tabla 5.2 puede verse que memoria RAM es un recurso determinante para el tiempo empleado.

La mayoría del tiempo empleado en resolver el TAP es el empleado en la generación del problema. Cuando el problema tiene unas dimensiones importantes, el tiempo empleado en la generación del problema supera con creces al empleado en resolver el problema. Esta es la parte de ejecución crítica para el PC, ya que carece de los recursos necesarios para abordarlo. Merece la pena poner especial atención en las estructuras de datos del problema, de manera que la generación de los datos sea lo más eficiente posible.

Ngd	100 %	130 %	80 %	Aleatorio
Sin congestión	0,171	0,124	0,109	0,140
Con congestión	0,496	0,309	0,496	0,372
GAP	0,001E-9	0,282E-9	0,001E-9	28.109E-9
GAP rel.	3,728E-10	0,000	2.009E-10	5,903E-10
Ciudad Real	100 %	130 %	80 %	Aleatorio
Sin congestión	2,421	2,530	2,436	2,531
Con congestión	25,059	25,746	24,606	30,092
GAP	41,441	807,446	0,576	99,205
GAP rel.	0,221E-3	2,993E-3	0,004E-3	0,528E-3
Winnipeg	100 %	130 %	80 %	Aleatorio
Sin congestión	1.544,391	3.021,583	2.994,633	5.664,750
Con congestión	22.882,095	18.782,383	10.040,250	16.447,950
GAP	2.935,693	1,108,447	487,021	2.589,141
GAP rel.	3,860E-3	10,621E-3	0,825E-3	3,413E-3
Barcelona	100 %	130 %	80 %	Aleatorio
Sin congestión	18.638,200	19.145,933	8.714,567	18.872,717
Con congestión	53.690,750	24.887,900	22.804,083	49.661,983
GAP	509,176	3.203,969	269,845	988,222
GAP rel.	0,429E-3	2,033E-3	0,288E-3	0,805E-3

Tabla 5.3: Tiempos de CPU

5.3. Objetivo 1: Evaluación computacional de TCL

La primera observación del estudio de las tablas 5.19 y 5.21 es que los resultados son idénticos cuando se trata de redes sin congestión en los problemas TCL-P1 y TCL-P2. Nótese que el porcentaje de arcos aforados en ambos modelos coincide. Esto se debe que el número de sensores óptimos que obtiene TCL-P1 se impone como restricción a TCL-P2. Cuando se trata el caso de redes sin congestión ambos modelo coinciden. Esto se explica porque existe un único camino con flujo para cada par. El objetivo del modelo TCL-P2 es capturar el máximo flujo neto de cada par. Dado que el número de sensores cubre completamente el conjunto de pares y que todo el flujo de un par circula por un único camino, el resultado es que cuando se afora un arco se intercepta todo el flujo del par.

Otro aspecto importante de los resultados es que el PC no dispone de recursos para las redes de tamaño considerable (ver tabla 8.6, error *INSUFFICIENT SPACE TO UPDATE U-FACTOR*). En todos los casos el problema aparece al generar el problema, al igual que ocurría en el problema anterior. De igual forma, el servicio de supercomputación resolvió las redes necesarias para el estudio (ver tabla 8.11), con un rendimiento muy superior al del PC (ver tabla 8.16).

Por último, se puede observar que el porcentaje de arcos aforados es pequeño en la mayoría de los casos. Esto tiene gran interés puesto que implica utilizar un número reducido de sensores de tráfico con un consiguiente coste menor. La excepción es la red Nguyen-Dupuis, que es muy pequeña y no corresponde a modelos de ciudad reales.

5.4. Objetivo 2: Evaluación de la fiabilidad de las estimaciones de matrices O-D (DAP)

Como primera observación, hay que decir que para leer las tablas hay ciertas consideraciones. La primera que el índice empleado en la comparación (RMSE) depende de la escala en la que han sido medidas las matrices O-D. Escala para ilustrar esta afirmación si se reemplazan

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,124	2,0
Ciudad Real	7,54 %	0,233	2,0
Winnipeg	7,64 %	99,000	11,0
Barcelona	INSUFFICIENT SPACE TO UPDATE U-FACTOR		

Tabla 5.4: Resultados Modelo TCL-P1 sin congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,140	2,0
Ciudad Real	7,76 %	0,217	2,0
Winnipeg	INSUFFICIENT SPACE TO UPDATE U-FACTOR		
Barcelona	INSUFFICIENT SPACE TO UPDATE U-FACTOR		

Tabla 5.5: Resultados Modelo TCL-P1 con congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,186	2,0
Ciudad Real	7,54 %	115,000	3,0
Winnipeg	7,64 %	1953,000	19,5
Barcelona	INSUFFICIENT SPACE TO UPDATE U-FACTOR		

Tabla 5.6: Resultados Modelo TCL-P2 sin congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,249	2,0
Ciudad Real	7,76 %	8,000	3,0
Winnipeg	INSUFFICIENT SPACE TO UPDATE U-FACTOR		
Barcelona	INSUFFICIENT SPACE TO UPDATE U-FACTOR		

Tabla 5.7: Resultados Modelo TCL-P2 con congestión

Tabla 5.8: Ejecución en PC

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,001	1,8
Ciudad Real	7,54 %	0,050	2,9
Winnipeg	7,64 %	0,883	27,5
Barcelona	9,97 %	1,117	33,3

Tabla 5.9: Resultados Modelo TCL-P1 sin congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,001	1,8
Ciudad Real	7,76 %	0,083	2,9
Winnipeg	8,03 %	0,883	28,0
Barcelona	10,03 %	0,033	25,6

Tabla 5.10: Resultados Modelo TCL-P1 con congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	18,18 %	0,017	1,7
Ciudad Real	7,54 %	1,083	3,9
Winnipeg	7,64 %	100,600	44,0
Barcelona	9,97 %	118,817	51,4

Tabla 5.11: Resultados Modelo TCL-P2 sin congestión

	PC	SSC	Memoria (MB)
Ngd	18,18 %	0,001	1,7
Ciudad Real	7,76 %	0,467	3,9
Winnipeg	8,03 %	104,217	45,6
Barcelona	10,03 %	123,870	39,3

Tabla 5.12: Resultados Modelo TCL-P2 con congestión

Tabla 5.13: Ejecución en SSC

	PC	SSC	Razón (PC/SSC)
Ngd	0,124	0,001	124,000
Ciudad Real	0,233	0,050	4,660
Winnipeg	99,000	0,883	112,118
Barcelona	INSUFFICIENT...	1,117	n.d.

Tabla 5.14: Tiempos Modelo TCL-P1 sin congestión

	PC	SSC	Razón (PC/SSC)
Ngd	0,140	0,001	140,000
Ciudad Real	0,217	0,083	2,614
Winnipeg	INSUFFICIENT ...	0,883	n.d.
Barcelona	INSUFFICIENT ...	0,033	n.d.

Tabla 5.15: Tiempos Modelo TCL-P1 con congestión

	PC	SSC	Razón (PC/SSC)
Ngd	0,186	0,017	10,941
Ciudad Real	115,000	1,083	106,187
Winnipeg	1953,000	100,600	19,414
Barcelona	INSUFFICIENT ...	118,817	n.d.

Tabla 5.16: Tiempos Modelo TCL-P2 sin congestión

	Arcos aforados (%)	Tiempo de cpu (s)	Memoria (MB)
Ngd	0,249	0,001	249,000
Ciudad Real	8,000	0,467	17,131
Winnipeg	INSUFFICIENT ...	104,217	n.d.
Barcelona	INSUFFICIENT ...	123,870	n.d.

Tabla 5.17: Tiempos Modelo TCL-P2 con congestión

Tabla 5.18: Comparativa de ejecución PC-Supercomputación

	Perturbación	TCL-P1 y P2	Aleatorios
Ngd	80 %	0,0000000001	45,7626299815
	130 %	0,0000000002	68,6255429661
	Aleatoria	64,5481333946	124,3705969218
	Media	15,9161193659	59,7661314688
Ciudad Real	80 %	0,1290	0,4871
	130 %	0,1935	0,7306
	Aleatoria	0,1559	0,1664
	Media	0,1594	0,4613
Winnipeg	80 %	0,00002025	0,0001497
	130 %	0,00049985	0,0002414
	Aleatoria	0,00005767	0,0001852
	Media	0,0001925	0,0001921
Barcelona	80 %	0,00002734	0,0002216
	130 %	0,00030277	0,0003608
	Aleatoria	0,00138113	0,0014012
	Media	0,00057041	0,0006612

Tabla 5.19: Modelo con matriz de referencia para redes sin congestión

	Pert.	TCL-P1	TCL-P2	Aleatorios
Ciudad Real	80 %	1,9408	1,9511	0,4877
	130 %	3,1436377436	3,1159109491	0,8328557076
	Aleat.	2,4582699977	2,4416964283	0,3234330324
	Media			
Winnipeg	80 %	0,0003302578	0,0002623539	0,0000678929
	130 %	0,0003986254	0,0004171096	0,0001018095
	Aleat.	0,0003155496	0,0003254663	0,0000657723
	Media			
Barcelona	80 %	0,0005657639	0,0005705313	0,0001198758
	130 %	0,0010143590	0,0010802981	0,0001798704
	Aleat.	0,0017280406	0,0007828897	0,0013811539
	Media			

Tabla 5.20: Modelo con matriz de referencia para redes con congestión

las matrices g y \hat{g} mediante un cambio de escala por αg y $\alpha \hat{g}$, respectivamente, se obtiene:

$$RMSE = \frac{\sqrt{\sum_i (\alpha G_i - \alpha \hat{G}_i)^2}}{|I|} = \alpha \frac{\sqrt{\sum_i (G_i - \hat{G}_i)^2}}{|I|} \quad (5.1)$$

Esto quiere decir que la comparación de los resultados debe hacerse únicamente para cada red y sus perturbaciones, pero no pueden compararse entre sí.

	Perturbación	TCL-P1 y P2	Aleatorios
Ngd	80 %	56,5685425438	186,3644592017
	130 %	91,9238831456	303,0707659539
	Aleatoria	42,9050840927	142,3752072172
	Media	72,4166005398	230,4653997553
Ciudad Real	80 %	1,6939	1,9121
	130 %	2,9556	3,1302
	Aleatoria	2,2010	2,4093
	Media	2,2835	2,483866667
Winnipeg	80 %	0,00102683	0,005107
	130 %	0,00159164	0,005089
	Aleatoria	0,00126788	0,005100
	Media	0,00129545	0,005098667
Barcelona	80 %	0,001192	0,002820
	130 %	0,001636	0,002835
	Aleatoria	0,001978	0,003140
	Media	0,001602	0,001602

Tabla 5.21: Modelo de máxima entropía para redes sin congestión

Capítulo 6

Conclusiones

En este proyecto se ha pretendido desarrollar y evaluar computacionalmente una metodología para la localización de los sensores de tráfico. Para llegar a los objetivos indicados en este proyecto el trabajo no se ha limitado únicamente a este problema, sino que se ha aplicado a los métodos de estimación para estudiar cómo varían en función de los aforos obtenidos.

En el primer objetivo del proyecto, *Evaluación computacional de los modelos TCL*, se han comprobado los recursos y los resultados de los modelos TCL-P1 y P2. Si bien, en un primer momento pareció que podrían ser inabordables (por este motivo se planteó el tercer objetivo), sólo las redes más grandes lo fueron en alguno de sus casos. Sin embargo, gracias al Servicio de Supercomputación de la UCLM, todos los modelos se resolvieron en un tiempo razonable.

Para el segundo objetivo, *Fiabilidad de las estimaciones derivadas de los modelos TCL*, se combinaron los resultados de la localización de sensores con los dos métodos de estimación utilizados. Previamente se habían resuelto las redes disponibles, junto con sus perturbaciones, para poder comparar las matrices estimadas y para poder tener los flujos en un subconjunto de arcos, que son los que se aforaban.

A grandes rasgos, los resultados fueron los esperados. En la mayoría de los casos la mejor estimación se obtuvo al tomar los aforos indicados por los modelos de localización.

Al fijarse en la aforación aleatoria para la red de Ciudad Real en el caso del modelo de estimación con matriz de referencia, puede observarse que los resultados coinciden. Esto se explica porque los arcos aforados en esta red son relativamente pocos. Cuando se aforan aleatoriamente pocos arcos lo que ocurre es que los aforos en ambos casos han correspondido a arcos que pertenecen a caminos que satisfacen la demanda completa. Puede decirse que en ambos casos

se ha estimado la matriz sin congestión, puesto que si todas las demandas de la red se cubren con un único camino cada una, aparentemente la red no está congestionada, porque estos caminos son *caminos mínimos*.

Para el modelo de estimación de estimación sin matriz de referencia sólo tenemos resultados de redes sin congestión. Para obtener los resultados de redes con congestión se necesitaría una matriz de referencia de la que se pudiera extraer la matriz de proporciones necesaria.

La evaluación computacional de la estimación no fue necesaria, puesto que en ningún caso el problema fue más pesado, computacionalmente hablando, que los abordados hasta ahora.

El tercer objetivo, *Algoritmos heurísticos para el TCL*, se descartó, puesto que los algoritmos del TCL eran perfectamente abordables para los casos más complicados gracias al Servicio de Supercomputación. Esto, unido a que los algoritmos heurísticos no garantizan una solución óptima, hizo definitiva la decisión de descartar estos algoritmos.

Bibliografía

- [Bazaraa et al.1993] Bazaraa, M. S., H. D. Sherali, and C. M. Shetty : 1993, 'Nonlinear Programming: Theory and Algorithms (second ed.). New York, NY: John Wiley and Sons.
- [Berman1992] Berman, O., Larson, R., and Fouska , N.: 1992, 'Optimal location of discretionary service facilities'. In: *Transportation Science*, **26**, 201-211.
- [Berman1995] Berman, O., Krass, D. and Xu, C.W.: 1995, 'Locating discretionary service facilities based on probabilistic costumers flows'. In: *Transportation Science*, **29**, 276-290.
- [Bell1983] Bell, M.G.H.: 1983, 'The estimation of an origin-destination matrix from traffic counts.' In: *Transportation Research* **17(2)**, 198-217.
- [Cascetta and Nguyen1988] Cascetta, E. and S. Nguyen : 1988, 'An unified framework for estimating or updatin origin/destinatio matrices from traffic counts.' In: *Transportation Research B* **22**, 437-455.
- [Clarke1990] Clarke, F. H.: 1990, 'Optimization and Nonsmooth Analysis'. Philadelphia: SIAM
- [Gallo and Pallotino1986] Gallo, G. and Pallotino, S.: 1986, 'Shortest path methods: a unifying approach' In: *Mathematical Programming Study* **26**, 38-64
- [García2001] R. García : 2001, 'Metodología para el diseño de redes de transporte y para la elaboración de algoritmos en programación matemática convexa diferenciable'. Escuela Técnica Superior de Ingenieros Aeronáuticos, Universidad Politécnica de Madrid.
- [Hogson, M.J. 1990] Hogson, M.J.: 1990, 'A flow-capturing location-allocation model'. In: *Geographical Analysis*, **32**, 271-279.

- [Low1972] Low, D.E.: 1972, 'A new approach to transportation systems modelling.' In: *Traffic Quarterly*, **26(3)**, 391-404
- [Nguyen and Dupuis1984] Nguyen S., and Dupuis, C.: 1984, 'An Efficient Method for Computing Traffic Equilibria in Networks with Asymmetric Transportation Cost'. In: *Transportation Science*, **18**, 185-202.
- [O'Neill1987] O'Neill, W. A.: 1987, 'Origin-destination trip table estimation using traffic counts.' Ph. D. Dissertation, University of New York at Buffalo, NY.
- [Ortuzar and Willumsen1994] Ortúzar, J. de D. and Willumsen, L.G.: 1994, 'Modelling Transport'. *Model estimation for traffic counts*, **12.4**, 359-378.
- [Patriksson and Larsson1992] Patriksson, M. and Larsson, M.: 1992, 'Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem'. In *Transportation Science*
- [Rockefella1970] Rockafellar, R. T.: 1970, 'Convex Analysis'. Princeton, NJ: Princeton University Press
- [Snickars and Wibull1977] Snickars, F. and J. Wibull : 1977, ' A minimum information principle, theory and practice'. In: *Regional Science and Urban Economics*, **7**.
- [Verastegui2006] Verastegui, Doroteo.: 2006, 'Estimación de matrices origen-destino y calibración de parámetros en el problema de asignación de tráfico en redes congestionadas'. Escuela técnica superior de ingenieros industriales de Ciudad Real, Universidad de Castilla la Mancha.
- [Wardrop1952] Wardrop, J. G.: 1952, 'Some Theoretical Aspects of Road Traffic Research'. In: *Proceedings of the Institute of Civil Engineers Part II*. pp. 325-378.
- [Willunsen1978] Willumsen, L.: 1987, 'Estimation of an O-D matrix from traffic counts: a review.' In: *Working paper 99*. Institute for Transport Studies, University of Leeds.
- [Yang et al.1998] Hai Yang, Jing Zhou : 1998, 'Optimal traffic counting locations for origin-destination matrix estimation'. In: *Transport Research B*, **32 (2)**, 109-126.

Capítulo 7

Anexo. Contenido del disco adjunto

En el disco adjunto se encuentran todos los ficheros empleados en la realización de este PFC. El contenido está separado en tres carpetas:

- **Redes:** Esta carpeta contiene una subcarpeta por cada una de las redes empleadas en este PFC.
- **Localizacion:** Contenedora una serie de subcarpetas. Cada una de ellas contiene todos los ficheros necesarios para el problema de localización. Hay una de ellas, llamada *resultados*, con la salida de los modelos TCL para cada una de las redes.
- **Estimacion:** Contiene la implementación de los modelos de estimación, así como subcarpetas contenedoras de todos los ficheros necesarios de cada red. Los resultados de cada uno de los modelos se recogen en una subcarpeta llamada *resultados*.

7.1. Carpeta Redes

Dentro de cada una de las subcarpetas pueden encontrarse una serie de ficheros que obedecen la siguiente nomenclatura:

Nombre de la red_congestión_demanda.gms

donde *Nombre de la red* corresponde a una de las cuatro empleadas en el PFC: Ngd (Nguyen Dupuis), ciudad real3 (Ciudad Real), winnipeg y barcelona. *congestión* puede ser «sc» para el caso sin congestión y «c» para el otro caso. Por último, *demanda* indica el valor de la demanda

de la red: 100 corresponde a la red original, el resto de valores (80, 130, $rd30$) corresponden a las perturbaciones indicadas en el capítulo 4.

Por ejemplo, el fichero *ngd_sc_115.gms* corresponde a la red de Nguyen Dupuis sin congestión con la demanda perturbada al 115 % de su valor original.

Estos ficheros contienen la implementación del problema de asignación y los datos de la red a tratar.

La salida del modelo de asignación se almacena en unos ficheros de texto que se rigen por la misma nomenclatura comentada anteriormente.

Por ejemplo, la salida del fichero anterior se encuentra en el fichero *ngd_sc_115.txt*.

Dentro de cada carpeta hay un fichero llamado *normal.gms*, cuya única función es generar las matrices de demanda aleatoria con los factores indicados, de manera que puedan incluirse en el fichero adecuado.

Produce dos ficheros de salida: *normal.txt*, que contiene la matriz O-D original y las perturbadas, y *salida_rdom.txt* que contiene las matrices perturbadas con el formato que emplea GAMS. Con este último fichero se puede tomar la matriz e introducirla directamente en los ficheros de perturbaciones aleatorias.

El código de este programa auxiliar es:

```
$title Calculo de valores aleatorios a traves de normal de la
matriz O-D
```

```
$include "C:\gams\Redes\ngd\ngd_100.gms";
```

```
* Trabajamos con la matriz G(i);
```

```
FILE
```

```
resultados      /c:\gams\Redes\ngd\normal.txt/
```

```
salida /c:\gams\Redes\ngd\salida_rdom.txt/
```

```
put resultados;
```

VARIABLE

aleatorios(i) Valores aleatorios de G(i);

put "Matriz original"/;

loop (i,

 put i.tl G(i)/;

); put /;

* Matriz perturbada con factor 0.02

put resultados;

put "Matriz perturbada con factor: 0.02"/;

loop (i,

 aleatorios.l(i) = normal(G(i),sqr(G(i)*0.02));

 put i.tl aleatorios.l(i):12:3 /;

); put /;

put salida; put "G(i) Matriz perturbada con factor: 0.02"/;

put "/";

loop (i, put i.tl aleatorios.l(i):12:3 /);

put "/;"/;

put /;

* Matriz perturbada con factor 0.003

```

put resultados;

put "Matriz perturbada con factor: 0.03"/;
loop (i,
    aleatorios.l(i) = normal( G(i),sqr( G(i)*0.03 ) );
    put i.tl aleatorios.l(i):12:3 /;
);
put /;

put salida;
put "G(i) Matriz perturbada con factor: 0.03"/;

put "/";
loop (i, put i.tl aleatorios.l(i):12:3 /);
put "/;"/;

put /;

```

7.2. Carpeta Localizacion

En el fichero *TCLP2.gms* se encuentran implementados los modelos de localización TCL-P1 y TCL-P2. En las primeras líneas de este fichero hay que quitar el "*" para indicar el fichero de la red que se quiere resolver por los citados modelos.

Cada uno de estos ficheros se encuentra en una subcarpeta junto a los demás ficheros necesarios de la misma red.

Los ficheros de la red pueden ser *sin congestión* o *con congestión*. En el primer caso el nombre del fichero será del tipo *datos_red_sc.gms* y *datos_red_c.gms* para el segundo caso.

Estos ficheros se han generado por un sencillo programa. Este programa facilita la tarea de recopilar los datos necesarios de varios ficheros. Todos los ficheros necesarios para los modelos de localización están en la misma carpeta. Los ficheros son la salida de los modelos de

asignación para la matriz conocida y verdadera y el fichero que contiene la descripción de la red.

El nombre del fichero que tiene implementado el programa sigue la siguiente nomenclatura:

procesar_datos_red_congestion.gms

donde *red* y *congestion* toman los valores que se indican en el apartado anterior.

Por ejemplo, *procesar_datos_ngd_c.gms* genera el fichero *datos_ngd_c.gms* que contiene los datos de la red de Nguyen Dupuis para el modelo de localización con congestión.

Los resultados de los modelos para congestión y sin congestión se almacenan en la subcarpeta *resultados*.

Por ejemplo, la salida del modelo anterior es *ngd_c.txt*.

El código de este programa es el siguiente:

```
*Informacion de la red

$include "C:\gams\Localizacion\ngd\ngd_100.gms";

*Salida del DSD

*Sin congestion

$include "C:\gams\Localizacion\ngd\ngd_sc_100.txt";

*Con congestion

*$include "C:\gams\Localizacion\ngd\ngd_c_100.txt";

file datos /C:\gams\Localizacion\ngd\datos_ngd_sc.gms/;

put datos;

* Incluye los conjuntos de la red.
```

```

put '*Incluye los conjuntos de la red'//;

put '$include "C:\gams\Localizacion\ngd\sets_ngd.txt";'//;

*Para el Servicio de Supercomputacion

*put '$include "ngd/sets_ngd.txt";'//;

*put '$include "C:\gams\Localizacion\ngd\ngd_100.gms";'//;

*put '$include "C:\gams\Localizacion\ngd\ngd_sc_100.txt";'//;

put 'c(i) constante;'//;

put //'Parameter'//;

*matriz G(i) put //'G(i) matriz de demanda origen - destino'//;
loop (i,
  put i.tl G(i):20:30/;
); put '//';

put 'F(k) vector de flujos'//;
loop (k$(h(k) gt 0.0),
  put k.tl h(k):20:30/;
); put '//';

*P_2
put 'D(i) Destino del par i'//;
loop (i,

```

```

    put i.tl ladem(i):10:0/;
); put '//';

```

```
*P_2
```

```

put 'delta_ak(a,k)'/';
loop( (a,k),
    if( delta(a,k) > eps,
        put a.tl '.'k.tl 1/;
    );
); put '/;'/';

```

```
put 'c(i)=1;'/';
```

7.3. Carpeta Estimacion

En esta carpeta se encuentran los ficheros que implementan los modelos de estimación. Estos ficheros son:

- modelo1.gms contiene la implementación del modelo de estimación de mínimos cuadrados para los planes de aforación proporcionados por los modelos de localización TCL-P1 y TCL-P2.
- modelo1aleatorio.gms contiene la implementación del modelo de estimación de mínimos cuadrados para el plan de aforación aleatorio.
- modelo2.gms contiene la implementación del modelo de estimación de máxima entropía para los planes de aforación proporcionados por los aforos de los modelos de localización TCL-P1 y TCL-P2 (en este caso los aforos que nos proporcionan los modelos de localización coinciden, porque este modelo sólo estima redes sin congestión).
- modelo2aleatorio.gms contiene la implementación del modelo de estimación de máxima entropía para el plan de aforación aleatorio.

El resultado de los modelos de estimación se recoge en la carpeta *resultados*. Dentro de esta carpeta se encuentran dos subcarpetas: *modelo1*, que contiene los resultados del modelo de estimación de mínimos cuadrados y *modelo2*, que contiene los resultados del modelo de máxima entropía.

La nomenclatura de los ficheros recogidos en estas carpetas es:

estimacion_red_congestión_demanda.txt

donde *estimacion* puede ser:

- *estimacion* cuando el plan de aforación lo proporcionan los aforos del modelo TCL-P1.
- *estimacion2* cuando el plan de aforación lo proporcionan los aforos del modelo TCL-P2.
- *estimacion_aleatorios* cuando el plan de aforación lo proporcionan los aforos aleatorios.

congestión y *demanda* toman los valores de apartados anteriores.