



**Universidad de Granada**  
**Departamento de Ciencias de la**  
**Computación e Inteligencia Artificial**

**MODELOS BASADOS EN “SOFT  
COMPUTING” PARA RESOLVER  
PROBLEMAS DE LOCALIZACIÓN**

**David Kutangila Mayoya**



**TESIS DOCTORAL**



**UNIVERSIDAD DE GRANADA**

**E.T.S. DE INGENIERÍA INFORMÁTICA**



**Departamento de Ciencias de la Computación e  
Inteligencia Artificial**

**Modelos basados en “soft computing” para  
resolver problemas de localización**

**TESIS DOCTORAL**

**David Kutangila Mayoya**

**Granada, Enero de 2005**



**Modelos basados en “soft computing” para  
resolver problemas de localización**

**MEMORIA QUE PRESENTA**

**David Kutangila Mayoya**

**PARA OPTAR AL GRADO DE DOCTOR**

**Enero de 2005**

**Director**

**José Luis Verdegay Galdeano**

**DEPARTAMENTO DE CIENCIAS DE LA  
COMPUTACIÓN E INTELIGENCIA ARTIFICIAL**

La memoria titulada **Modelos basados en “Soft Computing” para resolver problemas de localización**, que presenta D. David Kutangila Mayoya para optar al grado de Doctor, ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, bajo la dirección del Dr. D. José Luis Verdegay Galdeano, Catedrático del referido departamento.

Granada, enero de 2005

Editor: Editorial de la Universidad de Granada  
Autor: David Kutangila Mayoya  
D.L.: Gr. 179 - 2005  
ISBN: 84-338-3275-1

*"Tendré un hijo,  
Se llamará Roberto Kutangila Mahidika,  
El añadirá una tesis doctoral al lado esta,  
Le dedico el presente trabajo."*

## *Agradecimientos a*

*Agencia Española de Cooperación Internacional,*

*Todos los miembros del Departamento de Ciencias de la Computación e Inteligencia Artificial,*

*Todos los miembros del grupo de investigación Modelos de Decisión y Optimización (MODO),*

*Prof. José Luís Verdegay Galdeano,*

*Prof. Rev. Ngoy Boliya,*

*Maribel Sancho Royo,*

*Alejandro Sancho Royo,*

*Prof. Vicente Liern,*

*Prof. Maria José Canós Darós,*

*Prof. Carlos Ivorra,*

*Mi mujer Colette Luhilu Kibangu y mis hijas Gracia Kutangila Salangeye,*

*Gloria Kutangila Ikembi, Maribel Kutangila Kibangu,*

*Padres, hermanos, amigos y conocidos,*

*Compañeros en el edificio Mecena (los meceneros),*

*Belén Prados Suárez,*

*Carolina Narváez,*

*Todos los becarios en Ciencias de la Computación e I.A.,*

*Por haber contribuido de cerca o de lejos en la realización de la presente tesis doctoral.*

# ÍNDICE

## **Introducción General, 1**

Planteamiento del problema, 1

Situación actual, 2

Metodología propuesta, 4

## **Capítulo 1: Descripción del Contexto Socio-Económico, 6**

1.1. La red de transporte (las vías), 6

1.2. Circulación, 10

1.3. Transportes colectivos, 13

1.4. Población de Kinshasa en 1999, 19

1.5. Tarifas para el transporte colectivo, 20

## **Capítulo 2: El Problema de la $p$ -Mediana. Modelos exactos y borrosos, 24**

2.1. Introducción, 24

2.2. Una descripción general de los problemas de localización, 24

2.3. El concepto de la  $p$ -mediana, 28

2.4. Formulación matemática del problema de la  $p$ -mediana, 31

2.5. Técnicas de Resolución, 37

2.6. La  $p$ -mediana en incertidumbre, 46

2.6.1. Análisis de sensibilidad y análisis paramétrico, 46

2.6.2. La  $p$ -mediana estocástica, 46

2.6.3. La  $p$ -mediana robusta, 48

2.6.4. La  $p$ -mediana borrosa, 49

2.7. Noción sobre  $p$ -mediana capacitada y  $p$ -mediana no capacitada, 70

2.8. Conclusión, 78

**Capítulo 3: Problemas de Localización de Centro y “Routing”, 79**

- 3.1. Introducción, 79
- 3.2. Problemas de localización con asignación única, 81
- 3.3. Problemas de localización con asignación múltiple, 131
- 3.4. Conclusión, 215

**Capítulo 4: Optimización de la red de transporte de la ciudad de Kinshasa, 217**

- 4.1. Introducción, 217
- 4.2. Diseño del grafo de Kinshasa, 218
- 4.3. Breve presentación del software GAMS, 226
- 4.4. Resultado del cálculo de la p-mediana crisp, 228
- 4.5. Cálculo del grado de cumplimiento de la solución crisp obtenida, 230
- 4.6. Determinación del intervalo del coste óptimo, 234
- 4.7. Centros de transporte, 235
- 4.8. Determinación de líneas de transporte, 239

**Conclusiones y desarrollo de líneas de investigación futuras, 240****Anexos, 243**

- Anexo I. Elementos de conjuntos borrosos, 243
- Anexo II. Conceptos sobre grafos, 255
- Anexo III. Modelos GAMS utilizados, 274

**Bibliografía, 282**



## INTRODUCCION GENERAL

### Planteamiento del problema

La República Democrática del Congo (antiguo Zaire) tiene una superficie total de 2.345.000 km<sup>2</sup>, más de 60 millones de habitantes y se sitúa en el centro de África. Está rodeada por Angola al Suroeste, por Cabinda y la República Popular del Congo al Oeste, por la República Centro Africana y el Sudán al Norte, por Uganda, Rwanda, Burundi, y Tanzania al Este, y por Zambia al Sureste.



El país está dividido en diez provincias (Bandundu, Bas-Congo, Équateur, Kasai-Occidental, Kasai-Oriental, Katanga, Maniema, Nord-Kivu, Orientale, y Sud-Kivu) y un distrito federal (que incluye Kinshasa). Además de Kinshasa, otras áreas urbanas importantes son Boma, Bukavu, Kalemie, Kamina, Kananga, Kisangani, Kolwezi,



Likasi, Lubumbashi, Matadi, Mbandaka, y Mbuji-Mayi. Kinshasa es su capital y la ciudad más grande.

La ciudad de Kinshasa, capital política de la República Democrática del Congo, pasa por una crisis multiforme. Ningún sector es ajeno, aún los que sirven para el desarrollo de los otros, particularmente el transporte público. Nos interesa el sector del transporte público, especialmente el problema de la red de transporte público en Kinshasa, puesto que no se puede hablar de ningún desarrollo excluyendo el del transporte público, por ser este el acompañamiento irremplazable del desarrollo económico.

La vida de una gran aglomeración como Kinshasa requiere todo un conjunto de equipamientos de infraestructura o de servicio y se imagina mal una gran ciudad sin carreteras. Es el caso en la mayor parte de Kinshasa donde la vida diaria no es más fácil que en un pueblo del bosque.

En efecto, bien equipada para hacer vivir a 400.000 habitantes en 1960, Kinshasa, 15 años más tarde, triplicó a su población y construyó nada prácticamente para hacer frente a esta multitud. Además, el mantenimiento de los equipamientos existentes estaba mal garantizado, por lo que su degradación ya no permite el servicio de las zonas que equipan. Por ello, el estudio de la ciudad descubre la yuxtaposición de una ciudad antigua, cómoda y razonada, y nuevas construcciones donde la iniciativa personal no pudo solucionar sino parcialmente los problemas que plantea la vida comunitaria de varios centenares de millares de personas.

No obstante, la red de transporte existente tiene que ser mantenida y optimizada. Esta optimización consiste en establecer en Kinshasa una serie de puntos donde se abrirán algunas facilidades precisamente las estaciones o terminales de autobuses rodeadas por los centros comerciales.

### **Situación actual**

En la actualidad sólo existen centros comerciales y actividades útiles en la zona norte de la ciudad. Dada la gran extensión de Kinshasa esto obliga a millones de personas a efectuar largos y frecuentes desplazamientos desde toda la ciudad hacia la zona norte. Estos desplazamientos no son sencillos de realizar porque:

- 1) La red de transporte es deficitaria: Sobre los 5.109 Km. de rutas, sólo 548 Km. son asfaltados. Así, 4561 Km. de rutas son en tierra, lo que justifica la congestión del tráfico.
- 2) La red de autobuses está en manos privadas, de modo que cada autobús tiene un propietario diferente. Además, la intervención del gobierno es prácticamente nula, y en consecuencia, no existen horarios fijos ni rutas fijas. De hecho, cada conductor decide en cada trayecto hasta dónde llega.
- 3) Una oferta del transporte más baja que su demanda: Los estudios muestran que hay 3.600.000 desplazamientos por día en la ciudad de Kinshasa, pero solamente 196.100 son hechos por los medios de transporte. Así 5,44% de demanda satisfecha y 94,55% de demanda no satisfecha. Los 5,44% de desplazamientos son transportados por 1.142 vehículos. Las investigaciones muestran que en los 196.100 desplazamientos, 3% usan el coche privado, 40% usan el bus o el tren, 20% usan el taxi y 36 % van a pié.
- 4) El crecimiento demográfico: en 1960, Kinshasa contaba 400.000 habitantes, pero en 2001, la población de la ciudad alcanzó más de 4.000.000 de habitantes. Kinshasa será contada como una de las más grandes ciudades del mundo a causa de su población. Según las proyecciones científicas, en 2015 Lagos alcanzará más de 24 millones de habitantes [86], Cairo más de 16,5 millones de habitantes [87] y Kinshasa más de 10 millones de habitantes. Entre las 15 ciudades más grandes que contará el mundo en 2015, en África es necesario contar Cairo, Lagos y Kinshasa. Las otras ciudades serán asiáticas.
- 5) La extensión de la ciudad: el crecimiento demográfico conduce a la extensión de la ciudad (aproximadamente 10.000 km<sup>2</sup>) y así las distancias se largan.

Por consiguiente, a partir de todo esto, se propone como objetivo global de esta memoria, a partir de las metodologías conocidas para el tratamiento de datos imprecisos (básicamente Soft Computing, y por tanto, lógica borrosa)

- Diseñar el establecimiento de una serie de centros comerciales repartidos por toda la ciudad de Kinshasa, que cubran completamente su área metropolitana, y agregar en cada uno de ellos una estación de autobús, de forma que se facilite el transporte urbano a los habitantes de aquella ciudad.

## Metodología propuesta

Deseamos diseñar una nueva red de transporte basada en la solución de los problemas de localización adaptados que nos digan, por ejemplo donde localizar centros comerciales, estaciones de autobús etc. Pero la noción de la red de transporte es en sí mismo un concepto que puede variar dependiendo de la manera en que se considera el transporte [16, 22, 23]. Por lo tanto, los conceptos de nodo de la población, calle/avenida, puntos de travesía relevantes, etc. necesitan ser revisados. Una metodología extremadamente útil para solucionar una variedad de problemas en este contexto es la proporcionada por los conjuntos y sistemas borrosos [51] que nos conducen a tratar problemas borrosos de localización y, correspondientemente, a la definición de red borrosa [70].

Así se propone resolver el problema a partir de lo siguiente:

1. Calcular la mejor localización para los centros comerciales, entendiendo por mejor aquella ubicación que minimice la distancia total que han de recorrer los clientes. Se propone modelizar el problema como una  $p$ -mediana. Para ello, primero hay que diseñar la red. Se ha considerado conveniente repartir la ciudad según las zonas administrativas ya existentes y en cada una de ellas colocar uno o más nodos dependiendo de la población. Estos nodos están situados en aquellas zonas en las que la gente ya acostumbra a coger los autobuses a pesar de que no existe ninguna estación. Las aristas son las calles existentes, y la demanda se ha calculado utilizando el censo del Ministerio de Interior correspondiente al año 1999. Luego fuzzificamos la red sobre el que calculamos la mejor localización borrosa con el nuevo modelo de la  $p$ -mediana borrosa que presentamos en este trabajo.
2. Establecer rutas fijas para los autobuses. Una vez localizados los centros comerciales, se propone agregar en cada uno de ellos una estación de autobuses y establecer rutas fijas.

Por eso en el primer capítulo se describe el contexto socio-económico en el que se va a desarrollar el trabajo. A continuación, en el capítulo segundo, se analizan los modelos de problemas de  $p$ -mediana convencionales (datos perfectamente conocidos) y borrosos (datos conocidos con vaguedad, como será aquí el caso). El tercer capítulo se dedica a la descripción y análisis de los problemas convencionales y borrosos de localización de centro y “routing”, ofreciendo vías operativas de solución. Finalmente, en el último capítulo, los

modelos estudiados se aplican para lograr el objetivo propuesto de diseño de una red de centros y rutas de transporte.

Debido al carácter básico de los conceptos de conjuntos borrosos y grafos, borrosos y no borrosos, que se utilizan a lo largo de toda la memoria, las definiciones y resultados más importantes sobre los mismos se presentan en forma de anexos al final. Así mismo se recoge un anexo descriptivo del software utilizado para la resolución de algunos de los modelos estudiados en la tesis.

La memoria concluye con la exposición del trabajo pendiente para el futuro, y con un apéndice bibliográfico que recopila las referencias utilizadas en este estudio.

## CAPÍTULO I. DESCRIPCIÓN DEL CONTEXTO SOCIO-ECONOMICO

La información presentada bajo este título se extrae del Atlas de Kinshasa [1] y se refiere al período de 1959-1972. Estos datos siguen siendo válidos hasta la fecha dado que no se aumentó nunca la red de transporte, que hoy está caduca y deteriorada. La población que era de 400.000 habitantes hoy ha alcanzado alrededor de 6 millones de habitantes

### I.1. La red de transporte (Las vías)

El plano de la ciudad es una extensa red. Una jerarquía de las vías y barrios aparece inmediatamente: ejes asfaltados cruzan y encuadran la aglomeración donde solo algunos barrios disponen de cierto número de calles asfaltadas.

#### 1. Las vías primarias

Se accede la ciudad por dos accesos principales:

- Al Oeste, la **carretera de Matadi**, enteramente asfaltada, uno de los elementos del "cordón umbilical" que conecta Kinshasa al estuario del Congo. Después de haberse dividido en dos ramas, por una y otra parte del Mampeza, esta carretera se incorpora al borde del río en el fondo de la bahía de Ngaliema, luego se dirige hacia el Este de manera casi rectilínea hasta el Centro Comercial. Este acceso se termina por un bulevar a doble calzada "**el bulevar del 30 de junio**", a partir del paso del río Gombe hasta la Estación Central.
- Se accede a la ciudad por el Este viniendo del interior del país. La carretera en curso de construcción, debería alcanzar las provincias mineras del Este. Por el momento sólo sirve al aeropuerto, la zona industrial de Maluku y una parte de la provincia de Bandundu. La carretera es de doble-calzada. Cruza las extensiones Este de la ciudad luego gira hacia el Noroeste al Intercambiador de Limete y viene finalmente a tropezar con el aeródromo de Ndolo, separado del Centro Comercial por éste y el municipio de Barumbu. A lo largo de su curso, desde el Aeropuerto de N'djili, esta vía lleva el nombre de "**bulevar Lumumba**".

Estos dos accesos se conectan entre Badiadingi y el Intercambiador de Limete por la carretera de **"By-Pass"** que confina el Sur de la aglomeración situada al Oeste del río Ndjili.

Dentro de la aglomeración, **"la avenida 24-Noviembre"** de dirección Norte-sur, pista desde Djelo-Binza hasta el barrio administrativo de Gombe. En su parte Meridional, es amenazada por las erosiones procedente de Lubudi. El **"bulevar Kasa-Vubu"** de dirección Oeste-Este, luego Norte-sur, empieza a Kintambo hasta incorporarse al Centro Comercial. Por último, **"la Avenida de la Universidad"**, de dirección Norte-sur, se alarga del Campús Universitario hasta a la extremidad Septentrional del Bulevar Lumumba.

Estas vías son de calzada simple, de anchura variable, bastante bien establecidas en el Oeste de la ciudad donde encuadran las zonas de extensión. Las vías primarias son insuficientes al Este donde el bulevar Lumumba ni siquiera alcanza directamente el Centro Comercial. De dirección general Norte-sur, estas vías carecen de transversales Oeste-Este, el bulevar Kasa-Vubu sólo desempeñando este papel sobre la parte Occidental de su curso.

## 2. Las vías secundarias

La red de las vías secundarias asfaltadas se desarrolla bien sobre la parte Septentrional y occidental de la ciudad. El Centro Comercial, el barrio administrativo de Gombe, los antiguos barrios (Barumbu, Kinshasa, Lingwala, Kintambo), los nuevos barrios (Kasa-Vubu, Ngiri-Ngiri), los barrios planeados (Bandalungwa, Kalamu, Lemba, Matete), los barrios residenciales de las colinas (Binza, Djelo Binza, Joli-Parc), están divididos por vías asfaltadas de buena calidad que reanudan las orientaciones de las vías primarias y cuyas dimensiones se adaptan al tráfico de los años 60. Se debe tener en cuenta la insuficiencia de los lugares de estacionamiento en el Centro Comercial.

El Sur y el Este de la Ciudad están, por el contrario, casi completamente desprovistos de vías secundarias. Son inexistentes en la parte del municipio de Ngaliema, en los municipios de Selembao, Bumbu, Makala, Ngaba, Kisenso, Masina y muy embrionarias en N'djili y Kimbanseke. Prácticamente, las vías secundarias son las que habían en la ciudad en 1960.

### 3. Las vías terciarias

Es la red de las calles que conducen a las casas. Sólo se ha establecido sobre una pequeña parte de la ciudad: El Centro Comercial, los barrios de construcción europea, los barrios planeados (Bandalungwa, Kalamu, Lemba, Matete). En general, se trata de caminos de tierra en muy mal estado, cenagales en la temporada de lluvia, arenas movedizas en temporada seca. Cuando es posible circular en las zonas planas, se vuelve impracticable para a un vehículo normal en las colinas del Sur o en la valle de Kimanseke y Masina. Además, los ríos, cuyos cursos Sur-Norte trocean la ciudad, se oponen a la circulación transversal, especialmente al Este donde se encuentran los valles. Se aíslan así completamente algunos barrios: el campo Luka (municipio de Ngaliema), y el municipio de Makala, una parte de Kingasani en el municipio de Kimbanseke.

El acceso con automóvil no puede hacerse sino difícilmente a lo largo de itinerarios más o menos fijos, a veces totalmente impracticables al final de la temporada de las lluvias.

Kinshasa dispone pues de las carreteras primarias casi completas y adaptadas al tráfico de los años 60. Las carreteras secundarias son suficientes al Norte y al Oeste de la ciudad, pero son de 1960. Son inexistentes o embrionarias al Sur y al Este de la ciudad, en las zonas de extensión ocupadas desde 1960.

Las carreteras terciarias se han desarrollado bien en el Centro Comercial, los barrios de construcción europea y los barrios planeados construidos entre 1955 y 1960. Es de muy mala calidad en el resto de la Ciudad.





## I.2. Circulación

### I.2.1. El parque automovilístico

En 1972, el parque automovilístico de Kinshasa se presentaba de la siguiente forma:

➤ Motocicletas.....	4.008
➤ Coches ligeros.....	44.260
➤ Jeep.....	1.418
➤ Vehículos de servicio (Camionetas, combis, etc.).....	12.066
➤ Camiones.....	11.885
➤ Autobuses.....	1.111
➤ Tractores.....	520
➤ Remolques.....	687
➤ Otros.....	529
	76.484

Este parque representa alrededor de la mitad de los vehículos en circulación en el conjunto de la República. Esta proporción era solamente de 20% en 1959. Desde entonces, no dejó de crecer en función de la posición cada vez más preponderante de la ciudad en la vida política y económica del país hasta 1972.

La progresión del número de vehículos en circulación es muy rápida de 1970 a 1972, el aumento total del número de vehículos fue de 17% con relación al año anterior, el aumento de los coches siguiendo el mismo ritmo que el de los camiones. Este aumento es incesante por lo que la circulación, hasta entonces bastante fluida, conoce actualmente dificultades en la medida en que la adaptación de carreteras no sigue el mismo ritmo.

La composición del parque evolucionó poco durante los últimos años. Permanece esencialmente compuesto de los coches ligeros. La renovación de este parque se hace al año al ritmo de 8,3% para los coches y al de 20% al año para los camiones.

El ritmo de renovación de los coches ligeros es aún lento y debería acelerarse en los próximos años, lo que implica que numerosos vehículos en circulación que son viejos y que requieren un mantenimiento cuidadoso se renueven.

### **I.2.2. Los recuentos de carreteras.**

El estudio de la circulación se basa en recuentos de carreteras efectuados de diciembre de 1971 a marzo de 1972 por la Misión Francesa de Urbanismo. Se establecieron 34 puntos de recuento sobre los principales ejes de circulación.

La representación cartográfica se hizo según los distintos períodos del día, definidos a partir de las curvas de recuento. Ésta tiene tres picos:

- El primero se sitúa entre las 7 y las 8 y corresponde a los trayectos de ida hacia el trabajo.
- El segundo tiene lugar entre las 11:30 y las 12:30 y corresponde a la vuelta hacia el domicilio en final de mañana. Se señala menos que la anterior ya que no se refiere a los empleados de los servicios públicos que practican el día continuo desde las 7 de la mañana a las 3 de la tarde.
- El tráfico de la tarde culmina entre las 4 y las 6 de la tarde y corresponde al final de las actividades del día.

El tráfico nocturno se reduce sobre la mayoría de los ejes de circulación. Entre estas horas punta, el tráfico es más reducido. Esta distribución muestra bien que el grueso de los desplazamientos en la ciudad se destina a los trayectos de ida y vuelta del domicilio al lugar de trabajo. Los desplazamientos fuera del marco del trabajo son mucho más raros.

### **I.2.3 La repartición del tráfico**

Cualquiera que sea la hora del día, el principal eje de circulación es, al Este de la ciudad, el bulevar Lumumba. Es por esta vía que los habitantes de N'djili, Lemba y Matete tienen acceso al Centro Comercial. En la 16ª calle que cruza al Bulevar Lumumba, cuando se reúnen todos los vehículos que van o que vienen de estas direcciones, pasan cerca de 35.000 vehículos al día. Dos puntas horarias son notables: la mañana desde las 7 a las 8, el tráfico alcanza más de 2.700 vehículos y la tarde desde las 3 a las 4, es de 2.500 vehículos. La Avenida de la Universidad se incorpora al bulevar Lumumba al contacto del centro de la ciudad. Hace transitar más de 20.000 vehículos al día, pero presenta un tráfico muy desequilibrado: 75% de los vehículos contabilizados entran en el centro pero 25%

solamente van hacia la periferia. Eso supone vueltas por otras avenidas y especialmente por el bulevar Kasa-Vubu.

La avenida del 24 de Noviembre es una segunda gran vía de circulación Norte sur. Aquí, el tráfico diario alcanza 23.500 vehículos con dos puntas también de las 7 a las 8 de la mañana y de las 4 a las 5 de la tarde. La primera punta es de 2.300 vehículos, la segunda de 1700 vehículos. Aquí, la punta de la mañana se señala claramente mejor que la de la tarde y en los dos casos, es el tráfico hacia el centro de la ciudad el que es preponderante. La circulación pues se caracteriza menos por las necesidades del trabajo.

La avenida Assossa constituye un eje longitudinal secundario que transita alrededor de 15.000 vehículos al día. Su utilización es diferente de los otros ejes. La punta por hora se sitúa entre las 11 y las 12 con 1.600 vehículos. Conectando Ngiri-Ngiri al centro, a través de los municipios de Kasa-Vubu y Lingwala, la avenida Assossa cruza sectores de actividades artesanales y comerciales y soporta un tráfico en relación con estas actividades.

El tráfico Norte-sur de entrada y salida del Centro Comercial se hace, a la ida por la avenida Bokasa y, a la vuelta, por la avenida Kasa-Vubu hasta el Puente Kasa-Vubu.

El conjunto de estos tráficos representa 40.000 vehículos al día. Se tienen en cuenta dos puntas horarios notables que corresponden al principio y al final del día de trabajo pero las puntas secundarias importantes aparecen en medio de la mañana y en la tarde, mostrando un tráfico a las motivaciones bastante diversificadas.

La circulación transversal se hace, al Norte, por el Bulevar del 30 de Junio y su prolongación la avenida Mundjiba y, al Sur, por el Bulevar Kasa-Vubu.

El bulevar del 30 de Junio ve pasar 28.000 vehículos al día. La punta por hora se sitúa entre las 11 y las 12 y alcanza 30.000 vehículos. Nos encontramos pues aquí también con un complicado problema de tráfico.

El Bulevar Kasa-Vubu hace transitar más de 28.000 vehículos al día; se utiliza intensamente todo el día desde Kintambo hasta Kalamu. En este sector, el tráfico se divide

entre las avenidas Bongolo, Victoire y Sendwe, en dirección del Este de la ciudad o hacia el Centro Comercial.

Concentrando y distribuyendo circulaciones procedentes de toda la ciudad, este barrio que se extiende desde el Puente Kasa-Vubu hasta el cruce de la avenida Bongolo es especialmente activo y animado. El desarrollo de los comercios, la presencia próxima a uno de los más únicos mercados de la ciudad, la concentración de las actividades nocturnas alrededor de Matonge, todo eso refuerza el carácter de este barrio que aparece realmente como el centro de la ciudad.

### **I.3. Transportes colectivos**

#### **I.3.1. La infraestructura**

Los transportes públicos estaban garantizados por dos sociedades principales, la Oficina de Transporte en Común del Zaire (O.T.C.Z) y la Sociedad de Transporte de Kinshasa (S.T.K.). Junto a estas dos sociedades que disponían de un parque de autobuses modernos e instalaciones importantes, siempre han existido pequeñas sociedades o particulares que explotan camiones carrozados especialmente para el transporte de las personas, los "fula-fula". Por fin, muchas empresas de la ciudad garantizan la recogida y el transporte de su personal por sus propios vehículos. Además existía un parque de cerca de 6000 taxis.

El más importante transportador, la O.T.C.Z., fue una oficina nacional sucesora de las compañías concesionarias de transportes urbanos de la colonia y de los primeros años de la independencia. La oficina no tenía el monopolio de los transportes urbanos sino que se beneficiaba de una ayuda importante del Estado. La O.T.C.Z. disponía alrededor de 500 vehículos 40%, de los cuales eran autobuses modernos de gran capacidad (Leyland PSU/5). El resto del material era ya antiguo y de mantenimiento difícil. La falta de recambios se hacía sentir cruelmente, y se inmovilizaba alrededor del 50% del parque continuamente por razones técnicas. La O.T.C.Z. poseía dos depósitos, uno al cruce de las avenidas Kabinda y Assossa, donde se encontraban también los servicios administrativos, otro en Limete cerca del Intercambiador. La O.T.C.Z. empleaba a más de 29.000 asalariados.

La S.T.K. fue una sociedad privada que recibía un apoyo técnico de un grupo alemán. Disponía de un centenar de autobuses modernos (Mercedes 0302). Además, la Sociedad explotaba una flota de taxis. Los garajes se estaban ubicados en Limete sobre la 13ª calle.

Estos dos grupos practicaban métodos de gestión moderna y transportaban alrededor de la mitad de los pasajeros de Kinshasa.

Los fula-fula y los taxi-buses pertenecían a sociedades, o generalmente a particulares. Los propietarios raramente se encargaban de su explotación. Generalmente, éstos alquilaban el vehículo a un conductor por un precio diario (alrededor 45 o 50 Z<sup>1</sup>). La gasolina era a cargo del conductor y el mantenimiento al del propietario.

Tal sistema perjudica mucho a los vehículos ya que el conductor pretende siempre hacer el máximo de los viajes en su día. Se cuenta alrededor de 500 fula-fula en la ciudad.

El sistema de explotación de los coches (taxis) era el mismo que el de los fula-fula con el precio diario pedido por los propietarios, que es de cerca de 10 Z.

En la actualidad, las sociedades de transporte público bien organizadas ya no existen. Sólo los “fula-fula” y los taxis garantizan el transporte de la población. Algunas empresas también garantizan el transporte de ida y vuelta de su personal. Como en los años 59-72, los propietarios de los “fula-fula” o taxis los alquilan a los conductores por un precio diario. Pero, dadas las varias devaluaciones que ha conocido la República Democrática del Congo, dada su economía cada día decreciente que devalúa su moneda, los propietarios de "fula-fula" y taxis tomaron la práctica de fijar el precio diario en moneda fuerte, precisamente el dólar.

### **I.3.2. La red de transporte colectivo**

La red de transporte en común es la establecida por la OTCZ y la STK y los fula-fula siguen las mismas líneas con algunas alternativas.

---

<sup>1</sup> Z (Leer “zaire”) era la moneda nacional cuando el país se denominaba Zaire. Los céntimos de la moneda zaire se denominaban Makuta y se escribían “K”. Un zaire equivalía 100 makuta (1 Z = 100 K).

Esta red es tributaria de las vías asfaltadas y sólo sirve una parte de la ciudad. La mayoría de las líneas se orienta hacia el Centro Comercial: Mercado y Estación Central. Las líneas transversales son raras. Conectan los municipios del Este (Matete, Lemba, Ndjili) al barrio administrativo del Gombe.

La red está constituida esencialmente por tres direcciones yendo hacia el Centro Comercial:

- Al Este, viniendo de Lemba, Matete y los municipios de la orilla derecha de Ndjili y Ngiri-Ngiri.
- En el centro, de Bandalungwa a través de Kasa-Vubu y los antiguos municipios.
- Al Oeste, de Djelo-Binza y Kintambo.

Las líneas del Este y el del Centro se juntan al Puente Kasa-Vubu y transitan a través del municipio de Kinshasa, utilizando la avenida Bokassa a la ida y la avenida Kasa-Vubu a la vuelta. Las líneas del Oeste recorren el Bulevar del 30 de Junio.

Esta red deja de lado los municipios de Bumbu, Makala, Kisenso y la gran parte del de Ngaliema. No se cruzan sino solamente se bordean otros municipios sobre un lado: Ngaba y Masina. Este dispositivo "que bordea" obliga a los viajeros a itinerarios de enfoque a las cuales vienen a añadirse largas esperas a las paradas.

La red distribuye ramificaciones a lo largo de las carreteras asfaltadas fuera de la ciudad, hacia las aglomeraciones periféricas de Kinkole, Nsele, Maluku hacia el Este, Mont-Ngafula hacia el Oeste.

El municipio de Kisenso, situado sobre una colina arenosa, sin carretera construida, posee un sistema particular de transporte utilizando dos pistas a partir de Matete: una sube hasta la cumbre de la colina a la misión católica, otro sigue el valle del Ndjili hasta la estación de tren de Lemba. Estas dos pistas son recorridas por taxis y taxi-buses hasta Matete, de ahí los viajeros pueden embarcarse sobre los autobuses y los fula-fula que sirven el conjunto de la ciudad.

Esta red, tributaria del desarrollo de las vías asfaltadas, es muy insuficiente. Las nuevas líneas permanecen sobre la periferia de la ciudad. La penetración al interior de los

municipios del Sur y del Este de la ciudad no podrá hacerse salvo que se abran algunas vías asfaltadas.

### **I.3.3. El tráfico**

En 1972, las dos sociedades principales OTCZ y STK transportaron respectivamente 126 millones y 29 millones de pasajeros, lo que representa en total 155 millones de pasajeros. Si se cuenta 300 días de tráfico anual, eso representa alrededor de 500.000 pasajeros al día o, si se considera que cada pasajero hace una ida y una vuelta y que la mayoría de los viajes se hacen desde el origen al final de línea, 250.000 personas transportadas cada día. En esta cifra se incluyen los pasajeros de tren transportados por la OTCZ, tres veces al día sobre el trayecto Estación de Lemba, Estación de Central (alrededor de 1600 personas al día).

Queda por considerar el tráfico de los fula-fula. Se puede obtener una cifra mínima basándose en el precio de alquiler. Sabiendo que el precio del billete es 3 K, es necesario pues al menos 1500 pasajeros al día para pagar el importe. Dadas las ganancias realizadas por los conductores, deben transportar alrededor de 2000 pasajeros al día, o sea, si se considera que los viajeros hacen una ida y vuelta al día, 1000 personas al día. Al tomar un porcentaje de inmovilización del 50%, se tiene pues 250 fula-fula en circulación cada día, transportando al menos a 250.000 personas al día.

Según estos cálculos, medio millón de la población de Kinshasa utiliza pues cada día el transporte público y al menos la mitad de ella viaja en los fula-fula.

El parque disponible es muy insuficiente para garantizar este transporte en buenas condiciones. Los vehículos se sobrecargan en las horas de punta y su mantenimiento defectuoso hace que el viaje sea arriesgado.

Una mejora de la situación parece difícil dada la muy fuerte progresión de la demanda si el ritmo de aumento de los vehículos (150 al año) no cambia. Para responder correctamente a la demanda, sería necesario sin duda duplicar muy rápidamente el número de los vehículos en circulación. ¿Pero las carreteras lo soportarían?



La distribución espacial del tráfico se cartografió a partir de los recuentos efectuados en los autobuses de la O.T.C.Z. durante los meses de noviembre de 1972. Las cifras así obtenidas se refieren a los números de personas que suben y descienden del autobús a cada parada. Por diferencia se obtiene el número de personas transportadas entre dos paradas.

Estos recuentos no pudieron realizarse sino sobre una parte de la red. Se refieren al 55% del número de viajeros transportados por la O.T.C.Z.

El plano de flujo que se elaboró a partir de esta información muestra bien el predominio de las líneas del Este de la ciudad. El número de los viajeros transportados entre Ndjili, Masina, Kimbanseke, Lemba, Matete y el Centro Comercial representa un 36% del total de las personas transportadas. Es hacia el Este que se sitúa el centro de gravedad de la población mientras que la principal zona de empleos y comercio permanece cerca del puerto: de dónde este tráfico importante que toma el bulevar Lumumba que aparece bien como la principal vía de circulación de la ciudad.

Se tiene en cuenta, sobre el plano, una diferencia importante entre los movimientos de ida y vuelta, la vuelta siendo más dominante que la ida. Es necesario acordarse que el plano sólo muestra el tráfico de los autobuses. El tráfico de ida tiene lugar sobre todo por la mañana, entre las 6 y las 8. Los autobuses son entonces muy insuficientes y muchos viajeros utilizan los fula-fula. Las vueltas, por el contrario, se extienden sobre la tarde a partir de las 3. Los viajeros prefieren en ese momento utilizar los autobuses, relativamente poco entorpecidos.

Las líneas del centro se refieren a los municipios que tienen las densidades de población más fuertes. Transportan 27% de los viajeros. La convergencia de estas líneas hacia el centro explica la importancia de los flujos registrados sobre el bulevar Kasa-Vubu. Punto de contacto entre las líneas del Centro y del Este, el puente Kasa-Vubu es por otra parte la parada principal de la red después de la del mercado central.

Las líneas del Oeste llegan al Centro Comercial por dos itinerarios: al Norte el bulevar del 30 de junio y sus prolongaciones, al Sur, el bulevar Kasa-Vubu a partir de Kintambo. Estas líneas transportan 21% de los viajeros. Sirven los barrios menos poblados, cuya parte de los habitantes se emplea in situ (municipio industrial de Ngaliema) y dónde reside una población de alto nivel de vida, utilizando para su desplazamiento vehículos personales.

Las principales líneas transversales unen Lemba y Matete al barrio administrativo de la Gombe y Ndjili a la Universidad. Transportan un 16% de los viajeros. Los trayectos transversales se dificultan por falta de carreteras al Sur del bulevar Kasa-Vubu. Por el contrario, sobre el borde de la aglomeración, las líneas utilizando By-Pass poco se frecuentan, sirviendo sólo a una escasa población.

#### **I.3.4 Los Taxis**

Es difícil avanzar las cifras relativas al número de personas transportadas por los taxis. Se intentó una estimación a partir de los ingresos de los conductores de taxi. Se vio más arriba que el precio pagado al propietario del vehículo es de 10Z/día sabiendo que el precio medio de la carrera es de 70K, son necesarios pues 15 carreras para alcanzar esta suma. El beneficio diario de un conductor siendo alrededor de 5Z, serán necesarios 7 a 8 carreras suplementarias para alcanzarlo; lo que hace pues 20 a 25 carreras al día. Sobre los 6000 taxis de la ciudad, se inmovilizan al menos 50% cada día. Sobre la base de 20 carreras al día, se puede considerar que 60.000 a 80.000 viajeros se transportan cada día.

Los taxis se utilizan poco para los transportes relativos al trabajo. Los alquilan sobre todo para transportar enfermos, para ir a los hospitales, para ir en visita a un barrio distante. Según la información proporcionada por los taxistas S.T.K. (los únicos con taxímetros), la longitud media de la carrera sería de 3.200 metros. Esto es una cifra orientativa, la clientela del taxi S.T.K. siendo en gran parte constituida de los extranjeros.

**I.4. Población de Kinshasa en 1999**

Zona administ.	Nacionales	Extranjeros	Total	Proporción
Bandalungwa	131.527	1.247	132.774	3.06
Barumbu	79.347	9.677	89.024	2.05
Bumbu	222.031	13.313	235.344	5.42
Gombe	17.251	2.425	19.676	0.45
Kalamu	157.166	4.973	162.139	3.73
Kasavubu	75.853	4.417	80.270	1.85
Kinmbanseke	541.214	28.887	570.101	13.13
Kinshasa	81.005	10.602	91.607	2.11
Kintambo	60.963	3.306	64.269	1.48
Kisenso	217.163	12.104	229.267	5.28
Lemba	199.270	1.018	200.288	4.61
Limete	221.830	1.610	223.440	5.15
Lingwala	50.601	3.948	54.549	1.26
Makala	133.186	17.561	150.747	3.47
Maluku	158.976	153	159.129	3.67
Masina	293.451	1.097	294.548	6.78
Matete	168.828	1.218	170.046	3.92
Mont Ngafula	172.308	2.922	175.230	4.04
N'Djili	260.994	23.236	284.230	6.55
Ngaba	104.269	180	104.449	2.41
Ngaliema	443.555	10.148	453.703	10.45
Ngiringiri	90.471	7.627	98.098	2.26
N'Sele	101.284	1.930	103.214	2.38
Selembao	185.657	9.555	195.212	4.5
<b>Total</b>	<b>4.168.200</b>	<b>173.154</b>	<b>4.341.354</b>	<b>100</b>

### **I.5. Tarifas para el transporte colectivo en toda la ciudad de Kinshasa en 1999**

(Anexo al Decreto n° 069 de 12 de abril de 1999 del Gobernador de la Ciudad de Kinshasa, Prof. Théophile Mbemba Fundu)

0. Precio del billete para el taxi, taxi-bús y Kimalu-Malu desde los puntos siguientes: Masina, Kimbanseke, N'djili, Matete, Kisenso, Lemba, Ngaba, Mont-Ngafula, Selembao, Kintambo, Ngaliema, Bandalungwa, Bumbu, Ngiri-Ngiri, Yolo-Ezo hasta Marché Matete, Marché Gambela, Marché Central, Ports Privés, Gombe Royal/Cliniques, Gare Centrale y Centre Ville: 1 Fc (Un Franco congoleño).
1. Precio del billete para el taxi, taxi-bús y Kimalu-Malu circulando entre dos zonas administrativas vecinas: 50 c (Cincuenta céntimos).
2. Precio del billete para el Fula-Fula en toda la ciudad de Kinshasa: 80 c (Ochenta céntimos).
3. Precio del billete para el taxi-bús y bús desde Kingasani/Pascal hasta:
  - Kinkole : 1,20 Fc (1 Franco 20 Céntimos)
  - N'sele : 2,40 Fc
  - Maluku : 3 Fc
  - Menkao : 3 Fc
  - Bitá : 4 Fc
  - Dumi : 5 Fc
  - Mbakana: 7 Fc
4. Precio del billete para el bús y taxi-bús Express : 2 Fc
5. Precio de recorrida en coche desde una zona administrativa a otra:

TRAYECTO	TARIFA
<b>1. <u>Desde Centre Ville hasta:</u></b>	
✓ Victoire	1,20 Fc
✓ Kintambo	1,40 Fc
✓ Bon Marché	1,20 Fc
✓ Kingabwa UCAM	1,20 Fc
✓ Limete	1,60 Fc
✓ Lemba	2,40 Fc
✓ Matete	2,40 Fc
✓ N'djili Sainte Thérèse	2,40 Fc
✓ Kingasani II	2,40 Fc
✓ Masina Pascal	2,40 Fc
✓ Aéroport de N'djili	3,40 Fc
✓ Kimbanseke / Cimétière	3,40 Fc
✓ Masina Petro-Congo	2,00 Fc
✓ Yolo-Ezo	1,80 Fc
✓ Ngiri-Ngiri	1,80 Fc
✓ Bandalungwa	1,80 Fc
✓ Rond-Point Ngaba	2,40 Fc
<b>2. <u>Desde IPN hasta:</u></b>	
✓ Kinzonzi	1,20 Fc
✓ Cité Verte	1,20 Fc
✓ Rond-Point Ngaba	1,40 Fc
✓ Delvaux	1,20 Fc
<b>3. <u>Desde Delvaux hasta:</u></b>	
✓ Barré	1,20 Fc
✓ Kintambo	1,20 Fc
<b>4. <u>Desde Rond-Point Ngaba hasta:</u></b>	
✓ Lemba Terminus	1,20 Fc
✓ Mont-Ngafula Commune	1,20 Fc
✓ Matete	1,20 Fc
✓ Cité Verte	1,20 Fc
✓ Cliniques Universitaires	1,20 Fc
✓ Intendance/Campus	1,20 Fc
✓ Kimwenza Misión	1,40 Fc
<b>5. <u>Desde Lemba hasta:</u></b>	
✓ Matete	1,20 Fc
✓ N'djili Ste Thérèse	1,40 Fc
✓ Limete	1,20 Fc
✓ Kingasani II/Terminus	1,40 Fc

✓ Aéroport	2,40 Fc
✓ Cité Salongo	1,20 Fc
✓ Yolo-Ezo	1,20 Fc
<b>6. <u>Desde N'djili Quartier 1 hasta:</u></b>	
✓ Kimbanseke	1,20 Fc
✓ Kingasani	1,20 Fc
✓ Aéroport	1,40 Fc
✓ Matete Marché	1,20 Fc
✓ Cecomaf	1,20 Fc
<b>7. <u>Desde Kingasani Pascal hasta:</u></b>	
✓ Aéroport de N'djili	1,20 Fc
✓ Mikonga	1,20 Fc
✓ Kinkole	1,60 Fc
✓ N'sele	2,40 Fc
✓ Maluku	3,40 Fc
✓ Route Mokali	1,20 Fc
<b>8. <u>Desde Victoire hasta:</u></b>	
✓ Rond-Point Ngaba	1,60 Fc
✓ Yolo-Ezo	1,20 Fc
✓ Limete	1,20 Fc
✓ Matete	1,40 Fc
✓ Lemba	1,40 Fc
✓ N'djili Sainte-Thérèse	2,40 Fc
✓ Kingasani II	2,40 Fc
✓ Kintambo	1,40 Fc
✓ Aéroport	3,40 Fc
✓ Ngiri-Ngiri	1,20 Fc
✓ Bandalungwa	1,20 Fc
✓ Bon Marché	1,20 Fc
✓ Gombe	1,20 Fc
✓ Lingwala	1,20 Fc
✓ Huileries/Regideso	1,20 Fc
<b>9. <u>Desde Kintambo vers:</u></b>	
✓ Sola	1,20 Fc
✓ IPN	1,40 Fc
✓ Kinsuka Pompage	1,40 Fc
✓ Cimétière	1,20 Fc
✓ Bandalungwa	1,20 Fc
✓ Huileries	1,20 Fc
✓ Barré	1,20 Fc

✓ Gombe Cliniques	1,20 Fc
✓ Rond-Point Mandela	1,20 Fc
<b>10. <u>Desde Pont Matete hasta:</u></b>	
✓ Grande Poste/Ville	1,80 Fc
✓ UCAM	1,20 Fc
✓ AMICO	1,20 Fc
<b>11. <u>Desde Marché Selembao hasta:</u></b>	
✓ Bandal Moulaert	1,20 Fc
✓ IPN	1,20 Fc
<b>12. <u>Desde Matete hasta:</u></b>	
✓ Salongo Sud	1,20 Fc
✓ Limete	1,20 Fc
✓ N'djili Ste-Thérèse	1,40 Fc
✓ Masina/Pascal	1,40 Fc
✓ Kingasani II/Terminus	1,40 Fc
<b>13. <u>Desde Bon Marché hasta:</u></b>	
✓ Limete	1,20 Fc
✓ UCAM	1,20 Fc
✓ Pont-Matete	1,20 Fc
<b>14. <u>Desde Boulevard Lumumba hasta:</u></b>	
✓ Petro-Congo	1,20 Fc

## CAPÍTULO II. EL PROBLEMA DE LA $p$ -MEDIANA. MODELOS EXACTOS Y BORROSOS

### II.1. Introducción

La pertinencia del estudio del problema de la  $p$ -mediana se debe al objetivo que nos fijamos en el presente trabajo, el de localizar de manera óptima los centros comerciales y asignar en cada uno de ellos una estación de autobús, sobre la red de transporte de Kinshasa. Así se realizará la distribución equitativa de los centros comerciales a través de la ciudad y la optimización de la red transporte.

En un problema de localización, en general, se pretende encontrar el emplazamiento (y, tal vez, la asignación) de nuevos centros de servicio mediante técnicas de optimización. Se trabaja en un entorno geográfico (estructura topológica subyacente) donde, si es el caso, se conoce el emplazamiento y características de:

- Los clientes (demanda),
- Los proveedores (oferta),
- Centros de servicio existentes, y
- Otros elementos

Se necesita al menos una función de distancia, un objetivo (o varios) que defina el significado de “atender óptimamente la demanda”.

Uno de los problemas de localización es el de la  $p$ -mediana que utilizaremos en nuestro trabajo. Pero antes de entrar en él, describimos los problemas de localización desde una perspectiva general.

### II.2. Una descripción general de los problemas de localización

En [85] se presenta una descripción de general de los problemas de localización. La mayoría de gente acredita a los griegos Euclides y Pitágoras como los primeros quienes construyeron los modelos geométricos de la distancia. Pero la primera persona quien planteó formalmente un problema de localización y sugirió una solución fue el Emperador Constantino. El solucionaba un problema de localización en una red con solamente las posiciones discretas disponibles para la localización de las legiones romanas. Fermat y Torricelli (ambos matemáticos del siglo décimo sexto) plantearon el problema (con tres



nodos) que llamamos hoy el problema de Weber (the infinite solution space Euclidian distance minisum single site location problem). Este problema también a menudo se llama el problema de Steiner-Weber porque el problema de un sitio único sobre tres nodos coincide con un problema del diseño de red planteado por Steiner. Después Sylvester, matemático del siglo décimo noveno, estudió el problema de la minimización de la máxima distancia euclidiana en el espacio infinito de solución (infinite solution space minimax Euclidian distance problem).

El acontecimiento siguiente en la localización ocurrió en el siglo 20. En el problema original planteado por Steiner, había apenas tres nodos, pero en el problema general de Steiner, los nodos múltiples son conectados en una red de árbol por la arista de menor longitud. La arista de la red no necesita solamente ir de nodo a nodo sino que puede tener juntas intermedias entre los nodos. Estos conectores intermedios se refieren como puntos de Steiner. Weber [103], economista, estudiaba la localización de la industria, y su preocupación ha continuado hasta el presente, estimulada por gente como Hotelling y Hoover. Hotelling creó el famoso "vendedor de helados en una playa", en el cual un nuevo vendedor procura capturar la cuota de mercado máxima del primer vendedor que está situado en el centro de la playa. Éste problema se considera como el precursor de la "localización competitiva".

Más adelante Hoover y Pallander [54] plantearon el problema de Weber de la sola-fábrica con demandas y fuentes de facilidad múltiples, contrariamente al problema original planteado por Weber con tres nodos. Describieron independientemente la creación de líneas con coste de transporte igual para proveer cada punto de la demanda alrededor de cada fuente facilidad. Perseguían así el problema de la multi-fuente, de la multi-demanda de Weber. Pero no fue hasta 1958, con la revolución de la computadora en curso, que Baumol y Wolfe [9] ofrecieron una formulación y un acercamiento de programación matemáticos para el problema de localización de almacenes en una red.

Versan Quon y Charnes [102] ofrecieron una formulación del primer problema de localización rectilínea, un minisum, problema de localización de un único sitio. Estos artículos fueron los primeros que abordaron el problema de localización en el marco de la programación y optimización matemática.

Las demostraciones clásicas de Hakimi sobre la suficiencia nodal para la localización óptima en problemas de minisum en una red siguió rápidamente (problema de la  $p$ -mediana), y con ello llegó el establecimiento y reconocimiento en todo el mundo de las investigaciones en problemas de localización.

Numerosas disciplinas intervienen sobre los problemas de localización: ingenieros eléctricos, científicos regionales, ingenieros mecánicos e industriales, geográficos, ingenieros civiles y ambientales, economistas, informáticos, investigadores operativos, ingenieros químicos, analistas de la política sanitaria, decisores regionales, científicos e ingenieros del transporte, científicos matemáticos, etc.

Aunque el primer problema de localización fue el problema de minisum Euclidiano, los investigadores del campo discreto intentan solucionar problemas de localización en una red donde solamente están elegibles ciertos sitios pre-especificados para tener facilidades. Según ReVelle, estos problemas eran problemas prácticos interesantes y constató que estos programas de cero-uno eran enormemente difíciles de solucionar exactamente en cada caso. Pero, las herramientas para solucionar un problema ayudan a solucionar el siguiente.

La programación entera es la condición indispensable para la solución de los problemas de localización discreta en un espacio de solución. El último paso importante de la programación entera era la invención de branch and bound por Land y Doig (1960). Atacar estos problemas de localización discreta requiere sea una formulación denominada “integer friendly formulation” a la cual aplicar la programación lineal y el branch and bound. Una tal formulación es una formulación para la cual branch and bound no es necesario o en que solamente una cantidad limitada de branch and bound se requiere cuando el procedimiento de suma es necesario.

El progreso en la programación entera es lento y ha habido grandes avances. Las formulaciones “friendly-integer” para el problema de localización de planta, el problema de la  $p$ -mediana, el problema del recubrimiento del conjunto de localización, el problema del recubrimiento máximo y sus variantes, todos han sido creados. Y la relajación Lagrangiana se ha aplicado para solucionar con eficacia muchos de estos problemas también.

El problema de la carga fija, del cual el problema de localización no capacitada de planta es el ejemplo típico, sigue siendo un problema no resuelto en el caso general. No obstante, puede ser considerado solucionado en el caso específico de localización de planta debido a la formulación “integer-friendly” de Balinski y Morris [8][77].

Daskin [27] introdujo una secuencia para contar las variables que eran individualmente iguales a una hasta la suma que contaron. Su modelo maximizó la cobertura prevista de demandas. Hogan y ReVelle [53] también utilizaron la cuenta de variables en el contexto del cálculo de la cobertura de reserva de demandas. Emplearon posteriormente la cuenta de variables estructurando el problema de localización de la disponibilidad máxima y en modelos de la disponibilidad en la protección contra los incendios y el servicio de la ambulancia [86].

Otra idea “integer\_friendly” es la adaptación o la aplicación de la formulación de la trayectoria más corta a los problemas que ocurren en tiempo. ReVelle y Synder [87] aplicaron esta idea a la cosecha del bosque con tiempo, donde la meta es maximizar el valor de la corriente del tiempo de cosechas. Porque el problema es virtualmente igual que la formulación de la trayectoria más corta, las soluciones enteras (integer solutions) casi se aseguran. Hasta ahora, desde que esta formulación fue creada, ReVelle ha encontrado cuatro ajustes industriales, ambientales y geográficos adicionales que pueden aplicarse, cada uno necesitando un modelo integer-friendly para su declaración / solución. Ninguno de éstos se ha implementado todavía.

Lo que parece indudable es que la heurística y la metaheurística son los medios más avanzados para atacar los problemas que no pueden (todavía) incluso ser formulados como programas matemáticos lineales o no lineales. Un ejemplo de tal problema que fue solucionado heurísticamente antes de que un programa lineal entero había sido escrito es el problema de la  $p$ -mediana. Dos heurísticas para la  $p$ -mediana habían sido creadas (por Teitz y Bart [99] y Maranzana [71]) siempre antes de que una formulación de programación matemática hubiera sido ofrecida. Teitz y Bart crearon un procedimiento de la sustitución del vértice; Maranzana desarrolló un procedimiento de la mejora de cluster (cluster-improvement), de la modificación de cluster (cluster modification).

Un nuevo desarrollo que proporciona grandes oportunidades a los científicos de localización es los sistemas de información geográficos, o los GIS, que se han desarrollado

a través de los años 90. Los sistemas de información geográficos utilizan la informática actualizada para crear exhibiciones cautivantes de las soluciones que son resultados de los algoritmos heurísticos y/o exactos desarrollados por los científicos de localización.

### II.3. El concepto de la $p$ -mediana

Las decisiones de localización de la facilidad son inherentemente estratégicas en la naturaleza. Se relacionan con la localización a largo plazo de almacenes, de plantas de producción, de bases del servicio de emergencia, de escuelas, de hospitales etc. y la búsqueda de los vértices de  $p$ -mediana en una red (o grafo) es un problema clásico de localización. Varios autores definieron el problema de la  $p$ -mediana de las siguientes maneras:

- Según Rosing [90], el problema de la  $p$ -mediana consiste en seleccionar  $p$  sitios de facilidades en una red entre los  $n$  vértices o nodos de la demanda de esa red. Escoge esos sitios que reducen la suma ponderada de las distancias de cada nodo de demanda a su sitio de facilidad más cercano a su valor más bajo posible. Argumentan que Hakimi [47][48] demostró que basta con sólo mirar en los nodos de demanda para la selección, mostrando que una solución óptima se contendrá dentro de ese conjunto.
- En [37], Erkut y Bozkaya dice que el modelo de  $p$ -mediana es discutiblemente el modelo más popular en la literatura de localización de la facilidad que trata múltiples facilidades. Dado  $n$  puntos de demanda en algún espacio tal como el plano euclidiano o la red de camino, la meta del modelo es localizar  $p$  facilidades de servicio y asignar los  $n$  puntos de demanda a las facilidades de servicio de manera que se minimice la distancia total de viaje para el servicio.
- Según Bowerman [11], el objetivo del problema de la  $p$ -mediana es escoger  $p$  sitios de un conjunto de sitios candidatos de facilidad de manera que se minimice el coste de atender a un conjunto de localizaciones existentes de demanda de los usuarios.
- Según Hribar [55], el problema de la  $p$ -mediana originalmente propuesto por Hakimi [47][48] es el de localizar  $p$  facilidades para minimizar la suma ponderada

de la distancia de la demanda entre cada nodo de demanda y la más cercana de las facilidades. Hakimi mostró que por lo menos una solución óptima al problema consiste en localizar  $p$  facilidades en los nodos de la red.

- Según [18][19][21][22] el problema de la  $p$ -mediana consiste en localizar  $p$  facilidades para cubrir las demandas dadas de modo que los costes totales de transporte se minimicen. Asumen que estos costes son directamente proporcionales a las distancias que se deben cubrir y a las cantidades de productos que se deben transportar. Se muestra que en una solución óptima, todas las facilidades se localizan en los  $p$  vértices de la red y que la demanda será totalmente cubierta por la facilidad más cercana.
- En [88] Roland presenta el concepto de la  $p$ -mediana de la siguiente manera: dado un conjunto de localizaciones de demanda y los sitios potenciales de facilidad, el objeto del problema de la  $p$ -mediana es identificar las localizaciones de un número predeterminado de las facilidades,  $p$ , de manera que se minimice la distancia total que la demanda debe atravesar para alcanzar la facilidad más cercana. En la práctica, el problema de la  $p$ -mediana ocurre normalmente en una red tal como la red del camino o de la telecomunicación. Hakimi [47][48] demostró que para los casos del problema de la red, una solución óptima existe para la cual las localizaciones de facilidad son nodos de la red.
- En [47] se dice que el problema de la  $p$ -mediana consiste en localizar  $p$  facilidades para minimizar la suma de las distancias de cada usuario a su facilidad más cercana.
- En [70], se presenta el objetivo de la  $p$ -mediana que es el de localizar  $p$  facilidades (o medianas) para minimizar la suma de las distancias de cada vértice de la demanda a su facilidad más cercana.
- En [92], se presenta el problema básico del modelo de la  $p$ -mediana que es el de encontrar  $p$  localizaciones que minimizan la distancia media (o tiempo de viaje) en una red. Sólo los nodos de la red necesitan ser considerados como candidatos de

localización, puesto que hay siempre por lo menos una solución óptima que consiste en localizar las facilidades en los nodos de la red.

■ En [50] se presenta el problema de la  $p$ -mediana de la siguiente manera: considere un conjunto  $L$  de  $m$  facilidades potenciales y un conjunto  $U$  de las localizaciones de  $n$  usuarios dados. El problema de la  $p$ -mediana es localizar simultáneamente las  $p$  facilidades a las localizaciones de  $L$  para minimizar el coste total de transporte satisfaciendo las demandas de los usuarios, cada uno suministrado desde su facilidad más cercana.

■ P. Zhao [106] se presenta el problema de la  $p$ -mediana de la siguiente manera: Considere un problema de la  $p$ -mediana en el plano con la distancia euclidiana. Sea  $P = \{p_1, p_2, \dots, p_n\}$ , un conjunto de  $n$  ( $n$  puede ser más grande) puntos de demanda o localizaciones existentes de facilidad en un área saltada  $A$  con pesas positivas  $\{w_1, w_2, \dots, w_n\}$ . Suponen que  $p$  nuevas facilidades con localizaciones en el conjunto  $X = \{x_1, \dots, x_p\}$  deberán ser localizadas en  $P$ . Denotan como  $f(X)$  el viaje total de cada punto de demanda a su facilidad más cercana, eso es

$$f(X) = \sum_{i=1}^n w_i D(p_i, X),$$

donde  $D(p_i, X) = \min_{x \in X} d(p_i, x)$  y  $d(\cdot)$  es la función de la distancia euclidiana.

El problema de la  $p$ -mediana, entonces, es escoger sitios de facilidad  $X = \{x_1, \dots, x_p\}$  de tal manera que  $f(X)$  se minimice.

■ En [65], se presenta el concepto de la  $p$ -mediana en dos sub problemas de la mediana y del centro de la siguiente manera: Sea  $G = (V, E)$ , un grafo no dirigido, sencillo y conectado con un conjunto de vértices  $V$  y un conjunto de aristas  $E$ . Asociado a cada arista  $e$  en  $E$ , hay un número positivo  $l(e)$  que puede representar la longitud de la arista, la calidad de la calle, el tiempo necesario para cubrirla, etc. Además, asociado a cada vértice  $v$  en  $V$ , hay un peso positivo  $w(v)$  que se interpreta como la importancia de ese vértice. Los dos problemas más representativos en la teoría de localización en redes son los de la mediana y del centro. En ambos, uno debe encontrar el punto de la red  $N$  que minimiza una función de la distancia cargada a los vértices de  $N$ . En el problema de la mediana, la función para ser

minimizada es la suma de las distancias; en el problema del centro, queremos minimizar la distancia máxima.

Concluimos resumiendo todas esas definiciones por la siguiente:

Originalmente propuesto por Hakimi [47][48] quién demostró la posibilidad de localizar óptimamente por lo menos  $p$  facilidades en los nodos de una red, el problema de la  $p$ -mediana consiste en escoger  $p$  nodos de facilidad o localizar  $p$  facilidades en una red finita, conexa, simple, no dirigida, determinista y estática de manera que se minimice, entre cada nodo de la demanda de los usuarios y las  $p$  facilidades más cercanas, la suma ponderada de las distancias, o el coste total de transporte, o el tiempo total de viaje (de recorrido).

#### II.4. Formulación matemática del problema de la $p$ -mediana

Matemáticamente, el problema de la  $p$ -mediana puede formularse de varias maneras. En esta sección, presentamos las formulaciones de algunos autores.

En [50] se consideran un conjunto  $L$  de  $m$  facilidades (o puntos de localización), un conjunto  $U$  de  $n$  usuarios (clientes o puntos de demanda) y una matriz  $n \times m$   $D$  con las distancias (o costes) de viaje para satisfacer la demanda del usuario localizado al punto  $i$  desde la facilidad localizada al punto  $j$ , para todo  $j \in L$  y  $i \in U$ . El objetivo es minimizar la suma de las distancias (o costes de transporte), es decir

$$\min z = \sum_{i \in U} \min_{j \in J} d_{ij}$$

donde  $J \subseteq L$  y  $|J| = p$ .

Al lado de la formulación combinatoria, se presenta una en forma de problema de programación entera:

*p*-mediana – P.Hansen y N. Mladenovic

$$\min z = \sum_i \sum_j d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1 \quad \forall j, \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i, j, \quad (3)$$

$$\sum_j y_j = p \quad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i, j. \quad (5)$$

donde

$y_j$  toma el valor 1 si se localiza una facilidad en  $j$  y 0 si no.

$x_{ij}$  toma el valor 1 si el nodo  $i$  se asigna a la facilidad  $j$  y 0 si no.

La restricción (2) expresa que se debe encontrar (cubrir) la demanda de cada usuario. La restricción (3) impide a cada usuario suministrarse desde un sitio que no tenga facilidad. El número total de facilidades abiertas se fija a  $p$  por la restricción (4).

En [19] se presenta una formulación parecida a la presentada en [50]: Sea  $U = \{1, \dots, m\}$  el índice para el conjunto de  $m$  usuarios  $u_1, u_2, \dots, u_m$ . Además, sean  $w_1, w_2, \dots, w_m$ , las demandas correspondientes de usuario. Sea  $S$ , el conjunto de  $n$  sitios de facilidad.  $d(x, y)$  se define como la distancia entre el sitio de facilidad  $x$  y el usuario  $y$  y  $X(y)$  como el elemento del conjunto  $X \subseteq S$  que es el más cercano de  $y$  de modo que  $X(y) = \arg \min_{x \in X} d(x, y)$ . También  $D(X, y) = d(X(y), y)$  se define como la distancia de  $y$  al elemento más cercano de  $X$ . El conjunto de las soluciones factibles para el problema de la  $p$ -mediana es cualquier subconjunto de  $S$  que tiene exactamente  $p$  elementos. Utilizando estas definiciones, el coste de una solución particular  $X = \{x_1, \dots, x_p\} \subseteq S$  al problema de la  $p$ -mediana se da por

$$f_u(X) = \sum_{i \in U} w_i D(X, u_i) \quad (1)$$

El problema de la  $p$ -mediana consiste en encontrar  $X \subseteq S, |X| = p$  para minimizar (1).



En [44] se presenta la formulación siguiente: Dado el conjunto  $L = \{v_1, v_2, \dots, v_m\}$  de localizaciones potenciales para las facilidades (o puntos de localización), y el conjunto  $U = \{u_1, u_2, \dots, u_n\}$  de los usuarios (o clientes, o puntos de demanda), las entradas de una matriz  $n \times m$   $D = (d_{ij})_{n \times m} = (Dist(u_i, v_j))_{n \times m}$  dan las distancias de viaje (o costes encogidos) para satisfacer la demanda del usuario localizado al punto  $u_i$  desde la facilidad localizada a  $v_j$ , para todo  $v_j \in L$  y  $u_i \in U$ . El objetivo del problema de la  $p$ -mediana es minimizar la suma de estas distancias (o costes de transporte), es decir:

$$\min \sum_{u_i \in U} \min_{v_j \in X} Dist(u_i, v_j),$$

donde

$$X \subseteq L \text{ y } |X| = p.$$

En [91] se presenta la formulación siguiente:

*p*-mediana – D.A. Schilling et al.

$$\min z = \sum_{i \in I} \sum_{j \in J} a_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I, \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i \in I, j \in J, \quad (3)$$

$$\sum_{j \in J} y_j = p \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i \in I, j \in J. \quad (5)$$

donde

$i \in I$  = índice y conjunto de los nodos de demanda,

$j \in J$  = índice y conjunto de los nodos potenciales de facilidad,

$a_i$  = demanda al nodo  $i$ ,

$d_{ij}$  = distancia del nodo  $i$  al nodo  $j$ ,

$p$  = número de facilidades a localizar

$x_{ij} = 1$  si el nodo  $i$  se asigna a la facilidad  $j$  y 0 si no.

$y_j$  toma el valor 1 si una facilidad está abierta en  $j$  y 0 si no.

La función objetivo (1) minimiza la suma ponderada de las distancias asociadas a los nodos de la demanda que se asignan a los sitios de facilidad. El conjunto de restricciones (2) asegura que todos los nodos de demanda se asignan exactamente a una facilidad. El conjunto de restricciones (3) prohíbe asignación a un sitio que no tenga facilidad. El número total de sitios de facilidad abiertos se fija en  $p$  en la restricción (4) y la naturaleza binaria de las decisiones de localización de una facilidad o de asignación a una facilidad se refuerza por el conjunto de restricciones (5).

En [88] se presenta la formulación siguiente: Dado un grafo  $G=(V,E)$  y se desea encontrar un conjunto de nodos,  $S$ , de tamaño  $p$  donde  $S \subset V$  de modo que la suma ponderada de las distancias de los nodos restantes  $\{V-S\}$  al conjunto  $S$  se minimice. Siendo un programa entero binario, el problema de la  $p$ -mediana se puede escribir como:

*p*-mediana – E. Rolland et al.

$$\min \sum_i \sum_j a_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1 \quad \forall i, \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i, j, \quad (3)$$

$$\sum_j y_j = p \quad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i, j. \quad (5)$$

donde

$i \in I =$  índice y conjunto de los nodos de demanda,

$j \in J =$  índice y conjunto de los nodos potenciales de facilidad,

$a_i =$  demanda al nodo  $i$ ,

$d_{ij} =$  distancia del nodo  $i$  al nodo  $j$ ,

$p =$  número de facilidades a localizar

$y_j$  toma el valor 1 si se localiza una facilidad en  $j$  y 0 si no.

$x_{ij}$  toma el valor 1 si el nodo  $i$  se asigna a la facilidad  $j$  y 0 si no.

Explican los conjuntos de restricciones exactamente como D.A. Schilling, K.E. Rosing y C.S. ReVelle [91]

En [37] se presenta la formulación siguiente:

*p*-mediana – E. Erkut et al.

$$\min z = \sum_i \sum_j w_i d_{ij} X_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j X_{ij} = 1 \quad \forall i, \quad (2)$$

$$X_{ij} \leq X_{jj} \quad \forall i, j, \quad (3)$$

$$\sum_j X_{jj} = p \quad (4)$$

$$X_{ij} \in \{0,1\} \quad \forall i, j. \quad (5)$$

donde:

$w_i$  = demanda del nodo  $i$

$d_{ij}$  = distancia del  $i$  nodo al nodo  $j$

$p$  = numero de facilidades a localizar

$X_{ij}$  toma el valor 1 si el punto  $i$  se asigna a la facilidad en  $j$  y 0 si no.

$X_{jj}$  toma el valor 1 si una facilidad está abierta en  $j$  y 0 si no.

En [18] se presenta otra formulación diferente de las vistas anteriormente:

*p*-mediana – M.J. Canos et al

$$\min \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_{ij} = w_j \quad 1 \leq j \leq n, \quad (1)$$

$$x_{ij} \leq y_j \leq w_j y_i \quad 1 \leq i, j \leq n, \quad (2)$$

$$\sum_{i=1}^n y_i = p \quad (3)$$

$$y_i \in \{0,1\} \quad 1 \leq i \leq n,$$

donde  $x_{ij}$  es la demanda del vértice  $v_j$  cubierta por la facilidad al vértice  $v_i$  (si hay una),  $y_i$  es 1 si hay una facilidad al vértice  $v_i$  y 0 si no hay,  $w_i$  es la demanda al vértice  $v_i$ ,  $d_{ij}$  es la distancia de  $v_i$  a  $v_j$ .

Las restricciones (1) aseguran que toda la demanda se cubre en cada nodo  $v_j$ , las restricciones (2) garantizan que solo los vértices con una facilidad suministrarán el producto, y la restricción (3) establece que exactamente  $p$  facilidades se localizarán.

Resumiendo todas las formulaciones presentadas en esta sección, podemos decir en un sentido general que el problema de la  $p$ -mediana se formula de la siguiente manera:

*p*-mediana

$$\min z = \sum_{i \in I} \sum_{j \in J} c_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I, \quad (2)$$

$$\sum_{j \in J} c_i x_{ij} = w_i \quad \forall i \in I, \quad (3)$$

$$x_{ij} \leq y_j \quad \forall i \in I, \quad \forall j \in J, \quad (4)$$

$$\sum_J y_j = p \quad (5)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i \in I, \quad \forall j \in J. \quad (6)$$

donde:

$i \in I$  = índice y conjunto de los nodos de demanda

$j \in J$  = índice y conjunto de los nodos potenciales de facilidad

$c_i$  = demanda realmente cubierta al nodo  $i$

$w_i$  = total de la demanda a cubrir al nodo  $i$

$d_{ij}$  = distancia del nodo  $i$  al nodo  $j$

$p$  = número de facilidades a localizar

$x_{ij}$  toma el valor 1 si el nodo  $i$  se asigna a la facilidad  $j$  y 0 si no.

$y_j$  toma el valor 1 si se abre una facilidad en  $j$  y 0 si no.

La función objetivo (1) minimiza la suma ponderada de las distancias (o coste de transporte) asociadas a los nodos de la demanda que se asignan a los sitios de facilidad. El conjunto de restricciones (2) asegura que todos los nodos de demanda se asignan exactamente a una facilidad. El conjunto de restricciones (3) asegura que toda la demanda se cubre a cada nodo de facilidad. El conjunto de restricciones (4) prohíbe toda asignación a un sitio que no tenga facilidad, es decir, solo los nodos con facilidad suministrarán el producto. La restricción (5) fija el número de las facilidades a  $p$ . El conjunto de restricciones (6) refuerza la naturaleza binaria de las decisiones de localización de una facilidad a un nodo y de asignación de un nodo demanda a una facilidad.

Esta formulación tiene las ventajas de todas las formulaciones precedentes sin tener sus desventajas y nos será útil para el estudio de la  $p$ -mediana borrosa que veremos más abajo.

## II.5. Técnicas de Resolución

En esta sección, resumimos los principales métodos para resolver el problema de la  $p$ -mediana siguiendo la clasificación presentada en [16]: algoritmos de enumeración explícita, métodos heurísticos, técnicas de programación matemática, técnica de teoría de grafos y herramientas visuales:

### II.5.1. Algoritmos de enumeración explícita

El método de enumeración explícita ya fue propuesto por Hakimi en su artículo de 1964 para la 1-mediana [47] y generalizado en 1965 para  $p > 1$  [48]. Hakimi calcula la matriz de distancias más cortas entre todos los pares de vértices y multiplica cada fila por el peso del vértice asociado. A continuación, calcula todos los posibles conjuntos de  $p$ -elementos de  $V$ , siendo  $V$  el conjunto de todos los vértices de la red, y el coste asociado con cada uno de estos subconjuntos. Obviamente, la  $p$ -mediana será aquel subconjunto que tenga el menor coste.

Este algoritmo siempre llega hasta el óptimo, pero para ello debe calcular explícitamente el coste de todas las soluciones posibles. Se hace observar que cuando  $p$  no ha sido fijado de antemano, el número de posibles soluciones es

$$\sum_{j=1}^n \binom{n}{j} = 2^n - 1,$$

es decir, el número de soluciones que el algoritmo debe evaluar crece de forma exponencial con el número de vértices.

Sin embargo, si  $p$  es un valor fijo, el número de posibles soluciones es del orden de  $n^p$ , esto es, es de orden polinomial. Es un buen algoritmo para problemas de tamaño pequeño. Cual es el significado de “pequeño” debe ser decidido para cada problema en concreto. Los algoritmos de enumeración, además de resolver algunas instancias del problema, aportan algunas ideas útiles a la hora de desarrollar otras técnicas eficientes para problemas más grandes o más complicados.

### II.5.2. Métodos heurísticos

El algoritmo de enumeración explícita siempre encuentra el óptimo, pero es a cambio de un alto coste computacional. Por el contrario, un heurístico “es una técnica que busca buenas soluciones (es decir, cercanas al óptimo) a un coste computacional razonable sin ser capaz de garantizar la factibilidad o la optimalidad, o incluso en muchos casos establecer lo cerca que está de la optimalidad una solución factible particular”.

Los heurísticos resultan atractivos debido principalmente a su menor coste computacional, a lo intuitivas que suelen ser sus reglas y a su capacidad de combinación de unos con otros algoritmos exactos o aproximados. Pese a ello, hasta hace tres décadas la mayoría de investigadores todavía eran reacios a utilizarlos para resolver un problema dado a causa de la incertidumbre sobre la optimalidad y la factibilidad de la solución obtenida. Éste no fue, sin embargo, el caso del problema de la  $p$ -mediana. Sus características específicas aseguran la factibilidad y los experimentos computacionales demuestran que las soluciones obtenidas son óptimas.

Por ello, a finales de los 60 ya se habían propuesto los cuatro heurísticos clásicos en los que se han basado los demás:

- Heurístico de construcción [64]: El heurístico de Kuehn y Hamburger empieza suponiendo que ningún centro de servicio está ubicado. Selecciona como primera localización aquel vértice para el cual la función objetivo toma el menor valor. Entonces, entre los  $n-1$  restantes el que hace que la función objetivo disminuya

más, teniendo en cuenta la existencia del primer centro de servicio. Repite el mismo procedimiento hasta que  $p$  vértices han sido elegidos.

- Heurístico de eliminación [41]: Inicialmente todos los vértices tienen un centro de servicio asignado. Elimina aquel que menos aumente la función objetivo y procede del mismo modo hasta que  $n - p$  vértices han sido descartados.
- Heurístico de partición [71]: El heurístico de Maranzana comienza eligiendo  $p$  vértices. Cada punto de demanda es asignado a su centro de servicio más cercano de forma que genera una partición del conjunto de vértices en  $p$  subconjuntos distintos. Para cada uno de ellos calcula la 1-mediana, con lo que obtiene  $p$  nuevas localizaciones. Reasigna de nuevo los puntos de demanda y el proceso se repite hasta que ningún cambio puede hacerse.
- Heurístico de intercambio 1-óptimo [99]: El método de Teitz y Bart empieza dividiendo el conjunto de  $n$  vértices en dos subconjuntos distintos de  $p$  y  $n - p$  elementos respectivamente y asigna un centro de servicio a cada vértice del primer subconjunto. Busca un vértice en cada uno de los subconjuntos de modo que el intercambio de los dos es el que más disminuye la función objetivo. El procedimiento se repite hasta que ningún intercambio mejora la solución que se tiene en ese momento. Un estudio posterior llega a la conclusión de que el aumento de esfuerzo computacional que requiere transformar el heurístico en un método  $k$ -óptimo, con  $k > n$ , no mejora en la misma medida la solución obtenida [33].

Muchos trabajos posteriores están basados en una de estas cuatro técnicas, especialmente en el método de intercambio de Teitz y Bart [99] que, a la vista de los experimentos computacionales [89][99], parece ser el mejor.

Su buen comportamiento se explica porque las soluciones del problema de la  $p$ -mediana han de cumplir tres propiedades que son necesarias, aunque no suficientes, para ser óptimo global: Todos los centros de servicio son medianas locales para los puntos asignados a ellos, todos los puntos demanda se asignan a su centro de servicio más cercano y reemplazarlo por un candidato que no esté en la solución siempre se incurre en un incremento neto del valor de la función objetivo.

Los cuatro heurísticos anteriores tienen en cuenta las dos primeras propiedades pero sólo el de Teitz y Bart tiene en cuenta la tercera.

### II.5.3. Técnicas de programación matemática

El problema de la  $p$ -mediana puede plantearse como el siguiente problema de programación lineal binaria [16]:

$p$ -mediana (P1) – M.J. Canos

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_{ij} = 1 \quad 1 \leq j \leq n \quad (1.1)$$

$$x_{ij} \leq x_{ii} \quad 1 \leq i, j \leq n, \quad i \neq j, \quad (1.2)$$

$$\sum_{i=1}^n x_{ii} = p \quad (1.3)$$

$$x_{ij} \in \{0, 1\} \quad 1 \leq i, j \leq n.$$

donde  $d_{ij}$  es la distancia desde  $v_i$  hasta  $v_j$  y  $x_{ij}$  es 1 si un centro de servicio en  $v_i$  atiende la demanda en  $v_j$  y 0 en caso contrario. Obviamente,  $x_{ii}$  valdrá 1 cuando un centro de servicio se localice en  $v_i$  y 0 en caso contrario.

Las restricciones (1.1) aseguran que se atiende toda la demanda de cada vértice  $v_j$ , las restricciones (1.2) garantizan que sólo los vértices con un centro de servicio servirán el producto y la restricción (1.3) establece que se localizarán exactamente  $p$  centros de servicio.

Las condiciones de integridad de  $x_{ij}$  con  $i \neq j$  pueden relajarse sin que el valor de la función objetivo del problema quede afectado. La propia estructura del problema obliga a asignar cada punto de demanda a su centro de servicio más cercano. Luego  $x_{ij}$  con  $i \neq j$  sólo podría ser no entera si el vértice  $v_j$  tuviese al menos otro centro de servicio, digamos



$v_k$ , tan cercano como el situado en  $v_i$ . En tal caso, la demanda de  $v_j$  podría ser atendida por  $v_i$ , por  $v_k$  o por ambos, en diferentes proporciones. Esto nos daría soluciones alternativas pero no afectaría al valor óptimo de la función objetivo. Basta con reinterpretar  $x_{ij}$  como el tanto por uno de la demanda del vértice  $v_j$  atendida por el centro de servicio del vértice  $v_i$  para que la solución siga siendo válida. En tal caso, quedaría:

*p-mediana (P2) – M.J. Canos*

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_{ij} = 1 \quad 1 \leq j \leq n, \quad (2.1)$$

$$0 \leq x_{ij} \leq y_i \quad 1 \leq i, j \leq n, \quad (2.2)$$

$$\sum_{i=1}^n y_i = p \quad (2.3)$$

$$y_i \in \{0,1\} \quad 1 \leq i \leq n.$$

Nótese que en esta formulación se separan las variables de decisión  $y_i$ , que valen 1 si se localiza un centro de servicio en  $v_i$  y 0 en otro caso, de las variables de asignación  $x_{ij}$  que es el tanto por uno de la demanda del vértice  $v_j$  atendida por  $v_i$ .

Además el problema puede formularse, considerando la demanda total, del siguiente modo:

*p-mediana (P3) – M.J. Canos*

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_{ij} = w_j \quad 1 \leq j \leq n, \quad (3.1)$$

$$0 \leq x_{ij} \leq w_j y_i \quad 1 \leq i, j \leq n, \quad (3.2)$$

$$\sum_{i=1}^n y_i = p \quad (3.3)$$

$$y_i \in \{0,1\} \quad 1 \leq i \leq n.$$

donde  $x_{ij}$  es la demanda del vértice  $v_j$  cubierta por el centro de servicio del vértice  $v_i$  (si es el caso),  $y_i$  es 1 si hay un centro de servicio en el vértice  $v_i$  y 0 en otro caso,  $w_i$  es la demanda del del vértice  $v_i$ ,  $d_{ij}$  es la distancia desde  $v_i$  hasta  $v_j$ .

Los modelos (P2) y (P3) son equivalentes, en el sentido de que un óptimo de (P2) puede transformarse fácilmente en un óptimo de (P3) y viceversa, pero ninguno de ellos ofrece ninguna ventaja computacional respecto al otro.

#### **II.5.4. Técnicas de teoría de grafos**

Los algoritmos basados en la teoría de grafos intentan aprovechar el hecho de que la estructura topológica subyacente del problema es una red. Las dos características del problema más influyentes cuando se diseña un algoritmo son el número de centros de servicio y el tipo de red sobre la que trabajamos, la primera obviamente porque aumentará la complejidad computacional y la segunda debido a la estrecha relación entre las propiedades de la convexidad del problema y los árboles.

Dearing, Francis y Lowe [28], en su trabajo sobre resultados de convexidad en redes aplicados a problemas de localización, aportan varios motivos para que los algoritmos sean mucho más eficientes cuando la red es un árbol, entre ellos destacan: Es posible adoptar y utilizar muchos resultados estándares de la teoría de convexidad para estudiar problemas de localización sobre árboles; la convexidad también sugiere una estructura unificada para estos problemas. Además para cada problema convexo de localización en el plano, hay un problema de localización sobre redes análogo, a menudo no estudiado, que es convexo si y sólo si la red es un árbol y, para algunos problemas de localización sobre redes con un único centro de servicio donde la red no es un árbol, existen problemas equivalentes con un árbol generador. Desafortunadamente, sin embargo, es fácil construir ejemplos de problemas sobre redes con varios centros de servicio para los cuales no hay un problema con un árbol generador.

En 1970, Goldman y Witzgall [46] demostraron el siguiente resultado: Una subred  $S$  contiene al menos una mediana si satisface las siguientes condiciones:

- (i) la suma de los pesos correspondientes a los vértices de  $S$  debe ser al menos tan grande como la mitad de la suma de todos los pesos y
- (ii)  $S$  debe tener puertas, esto es, para cada  $t$  de  $N - S$  debe existir un único  $t'$  de  $S$  (la puerta) de forma que para cada  $s$  de  $S$  se cumple
 
$$d(t, s) = d(t, t') + d(t', s)$$

Una consecuencia inmediata de este teorema de localización es la propiedad de semisuma, que afirma que, dado un árbol, si el peso total de cada subárbol obtenido al eliminar un vértice no es mayor que la mitad del peso total del árbol, dicho vértice es una mediana.

Basándose en estos resultados, Goldman [45] plantea un algoritmo que encuentra una mediana cuando la red es un árbol utilizando las relaciones de incidencia y los pesos de vértices, sin tener que calcular la matriz de distancias. En cada iteración elige un vértice de grado uno del árbol con el que está trabajando en ese momento. Si el peso del vértice seleccionado es al menos tan grande como la mitad de la suma de todos los pesos, este vértice es una mediana. En otro caso, el vértice de grado uno se elimina junto con su arista incidente y su peso se añade al peso del vértice adyacente. El procedimiento se repite con el nuevo árbol. La complejidad del algoritmo es de  $O(n)$  porque, en el peor de los casos, se necesita examinar cada vértice una vez.

Goldman también sugiere como extender su algoritmo a otra clase de redes, aunque sin aportar ninguna evidencia computacional. En el caso de una red cíclica, la generalización es evidente. Otra de sus extensiones ha sido retomada por algunos autores. Goldman utiliza el concepto de puente, es decir, una arista que, al eliminarla, aumenta el número de componentes conexas de la red. Su algoritmo modificado terminaría, o bien en una mediana, o bien cuando ya no existen puentes, con la conclusión de que la mediana está en una de las componentes cíclicas que no se han podido eliminar. En este último caso, y aunque no se puede llegar al óptimo, la red queda reducida de tamaño. Evidentemente, si la red original no contiene ningún puente el tamaño de la red no se puede reducir. El concepto de puente se puede generalizar al concepto de bloque.

Chen y otros [26] consideran una red con ciclos y quienes explican cómo reducirla al grafo asociado por bloques, que siempre es un árbol utilizando una versión modificada del

teorema y del algoritmo de Goldman, construyen un algoritmo de orden lineal para el grafo. Asociado por bloques y encuentran, en la red original, o bien una mediana, o bien un único bloque que contiene todas las medianas.

Lan y Wang [69] abordan un algoritmo de  $O(n)$  para grafos ponderados en los que no se consideran pesos en los vértices o, equivalentemente, todos los vértices son equiponderados. Los autores explican como construir un árbol de búsqueda para el grafo original y que contiene los mismos vértices que éste. El algoritmo recorre todos los vértices del árbol mediante un procedimiento de búsqueda en anchura y siempre encuentra la mediana.

Para el caso de múltiples centros de servicio, Matula y Kolde [72] sugieren un algoritmo dinámico de orden de  $O(n^3 p^2)$  para encontrar la  $p$ -mediana sobre un árbol. Desde entonces, el objetivo ha sido depurar la cota de la complejidad algorítmica. Kariv y Hakimi [58] usan las ideas del algoritmo de Goldman para diseñar un algoritmo de  $O(n^2 p^2)$ . Hsu [56] propone un algoritmo  $O(pn^3)$  en caso de que todos los pesos de los vértices valgan uno. Hassin y Tamir [51] alcanzan la cota de complejidad  $O(pn)$  para el caso particular en que el árbol sea un camino. La mejor cota obtenida hasta el momento es de  $O(pn^2)$  por Tamir [98]. Él propone un algoritmo dinámico que, tras una transformación previa del árbol original en un árbol binario con raíz, recorre este último desde las hojas hasta la raíz, calculando recursivamente dos funciones previamente definidas cuyo valor en la raíz le dará la  $p$ -mediana que busca.

Para el caso de múltiples centros de servicio, sólo se han desarrollado algoritmos, además de en los árboles, en redes cíclicas. En particular Labbé, Peeters y Thisse [67] calculan la  $p$ -mediana sobre una red cíclica en un tiempo  $O(pn^2)$ . Para ello descartan una arista cada vez, utilizan el algoritmo de Hassin y Tamir [51] para redes caminos en el camino resultante y seleccionan la mejor de las soluciones así obtenidas.

### II.5.5. Herramientas visuales interactivas

Los sistemas de ayuda para la toma de decisiones, o SAD, surgidos a principios de los 70, intentan aprovechar las prestaciones de ambos haciendo que las ventajas de uno suplan las carencias del otro y viceversa utilizando tanto modelos de investigación operativa como gráficos, inteligencia artificial y modelización interactiva visual.

En [35] un SAD se define como un sistema interactivo máquina-hombre que ayuda a los decisores, sin reemplazarlos, utiliza datos y modelos, resuelve problemas con diferentes grados de estructuración y se preocupa de facilitar el proceso de toma de decisión antes que de su eficiencia. Aunque los problemas de teoría de grafos se adaptan extraordinariamente bien a las representaciones visuales, existen escasos trabajos en el campo de localización.

Directamente relacionado con la  $p$ -mediana, podemos encontrar en el trabajo de Pirkul, Gupta y Rolland [84] en el que diseñan y proponen el uso del *VisOpt*, una herramienta de optimización interactiva visual diseñada originalmente para la  $p$ -mediana con restricciones de capacidad pero que se puede adaptar fácilmente para una  $p$ -mediana simple. La *interface* del programa reduce al mínimo el uso de números y texto, representando las diferentes demandas por puntos de tamaño proporcional a las unidades demandadas y se utilizan colores para saber que puntos están asignados a cada centro de servicio. Cada vez que el usuario, al que se le permite representar soluciones infactibles que violan las restricciones de capacidad, abre o cierra un centro de servicio, un rápido y sencillo heurístico realiza la asignación de puntos demanda a centros de servicio abiertos. El procedimiento termina cuando el usuario considera que la solución obtenida es satisfactoria. El cálculo también presenta un experimento hecho con estudiantes universitarios familiarizados con técnicas de programación lineal y cuyos resultados se compraran con los resultados obtenidos por un heurístico al uso.

## II.6. La $p$ -mediana en incertidumbre

### II.6.1. Análisis de sensibilidad y análisis paramétrico

El análisis de sensibilidad estudia la curva de intercambio entre las perturbaciones de los datos y el grado de optimalidad de la solución, mientras que el análisis paramétrico caracteriza el camino que seguirán los óptimos cuando los datos del problema varían según un parámetro determinado [16]. Para el problema de la  $p$ -mediana, los estudios de análisis de sensibilidad y paramétricos son escasos y siempre referidos a un único centro de servicio. Esto puede ser debido a que ambos casos no responde a lo que el decisor desea, esto es, a la cuestión de dónde situar un centro de servicio cuando los altos costes de relocalización hagan que la decisión sea irreversible en el horizonte de planificación.

Labbé, Thisse y Wendell [68] estudian la curva de intercambio entre las perturbaciones de los pesos de los vértices y el coste para el caso de la 1-mediana sobre un árbol. Aunque su análisis se puede aplicar a cualquier tipo de perturbación, se enfoca principalmente a perturbaciones multiplicativas donde el factor que multiplica puede interpretarse como un porcentaje de la desviación del valor estimado del peso real del vértice.

Por su parte, Erkut y Tansel [36] presentan un algoritmo en tiempo lineal para determinar la secuencia de medianas sobre un árbol donde las demandas de los vértices son funciones de un parámetro que puede identificarse con el tiempo.

### II.6.2. La $p$ -mediana estocástica

En el problema de la  $p$ -mediana estocástica, los pesos de los vértices, o las longitudes de las aristas, o ambos, son variables aleatorias que siguen una distribución de probabilidad conocida. El objetivo es encontrar  $p$  puntos de la red que minimicen el coste esperado, es decir, encontrar una  $p$ -mediana estocástica [16]. Si sólo la demanda es aleatoria, el problema se reduce a su versión determinista donde los pesos utilizados son media de las variables aleatorias y, en consecuencia, se cumple la propiedad de optimalidad en los vértices.

Frank [42] considera el problema de la mediana cuando los pesos de los vértices son variables aleatorias no negativas independientes y la longitud de las aristas es determinista. En su trabajo posterior [43], Frank plantea la hipótesis de que los pesos de los vértices siguen una distribución normal multivariante. Además ofrece criterios alternativos de optimalidad definiendo la mediana absoluta de probabilidad máxima como un punto en la red donde, dado un número real  $R$ , la probabilidad de que la suma de las distancias ponderadas exceda  $R$  es mínima y la mediana absoluta de varianza mínima como un punto de la red para el cual la variable aleatoria definida como la suma ponderada de las distancias tiene mínima variable.

Wesolowsky [105] supone que los pesos de los puntos demanda, situados en una línea recta, siguen una distribución normal multivariante. Calcula la probabilidad de que el centro de servicio esté óptimamente ubicado en cualquier punto de la ruta y demuestra que sólo los puntos demanda tienen probabilidad distinta de cero.

Cuando la longitud de las aristas es aleatoria, la  $p$ -mediana obtenida sustituyendo las longitudes de las aristas por sus medias no minimiza en general la distancia total esperada, puesto que la métrica viene inducida por el camino más corto y el operador mínimo no es lineal. Sólo el problema de la mediana estocástica sobre un árbol es equivalente al problema determinista cuyos pesos son las medias. En cuanto a la propiedad de optimalidad de los vértices no siempre es precisa aunque sí lo es cuando la distribución de probabilidad se supone discreta, el estado de la red es conocido en cualquier instante y los intervalos de tiempo entre cambios en el estado de la red son mayores que los tiempos de desplazamiento.

Mirchandani [73] plantea un modelo de programación lineal binaria para el problema de la  $p$ -mediana estocástica cuando las distancias son variables aleatorias discretas, añadiendo que si se consideran los pesos de los vértices como variables aleatorias discretas con el mismo número de estados y la misma probabilidad para cada estado que las distancias, la generalización es inmediata.

Mirchandani y Oudjit [74] estudian el problema de la 2-mediana sobre un árbol con demandas conocidas y longitudes de aristas aleatorias con un número finito de estados.

Para este problema proponen un método de enumeración no exhaustivo donde muchas de las combinaciones de dos vértices son rechazadas a priori.

Por su parte, Weaver y Church [104] y estudian computacionalmente los resultados de dos métodos de resolución, el primero basado en el heurístico de Teitz y Bart [99] y el segundo basado en el método del subgradiente para optimizar el dual langrangiano.

Por último, Mirchandini, Oudjit y Wong [75] formulan el problema de la  $p$ -mediana estocástica con pesos y longitudes dadas por variables aleatorias discretas. Para este problema, plantean una aproximación a través de duales, definidos secuencialmente, utilizando como subrutina el algoritmo de Erlenkotter [37].

### II.6.3. La $p$ -mediana robusta

Los modelos de la  $p$ -mediana robusta se han planteado desde dos puntos de vista diferentes [16]: permitiendo desviaciones respecto a la factibilidad y la optimalidad o sólo respecto a la optimalidad. Asumir las hipótesis implícitas en el segundo caso significa, entre otras cosas, que toda la demanda debe ser atendida. Sin embargo, a veces podría interesar dejar una pequeña parte de la demanda insatisfecha a cambio de otras compensaciones como, por ejemplo, una considerable reducción de los costes. Esta idea ha sido considerada en [17], y en [20]. En ambos la incertidumbre de los datos queda reflejada mediante intervalos. Pero, mientras que [17] se realiza la discretización de los escenarios subjetivamente y se plantea un modelo coordinado de programación mixta binaria para una red general que puede resolverse con técnicas estándar, en [20] se aprovecha la estructura de árbol para proponer dos algoritmos que permitan obtener la descripción completa de los escenarios en el caso de la 1-mediana. ), Kouvelis y Yu [63] son los primeros en plantear el problema de la 1-mediana robusta sobre un árbol sin admitir desviaciones respecto a la factibilidad. Estudian diferentes formas de reflejar la incertidumbre de los datos y utilizan como criterio de optimización minimizar la máxima desviación respecto a la optimalidad o *minimax regret*. Cuando los parámetros del problema vienen dados por intervalos, la propiedad de optimalidad en los vértices deja de cumplirse, con lo que hay que distinguir entre el problema restringido a los vértices y el no restringido. Obtienen algoritmos polinomiales para todos sus modelos. Chen y Lin [25], con el mismo criterio de optimización, plantean el problema de la 1-mediana sobre un árbol donde la incertidumbre está especificada por



intervalos. Mejoran el método exhaustivo de Kouvelis y Yu [63] mediante un método de búsqueda descendente restringido a un subconjunto de vértices o una única arista.

Averbakh y Berman [2] desarrollan un algoritmo para el mismo problema, pero sobre una red general, con un orden de complejidad  $O(mn^2 \log n)$  donde  $n$  es el número de vértices de la red y  $m$  el número de aristas.

Por último, Serra y Marianov [92] consideran un número discreto de escenarios y plantean un modelo de programación matemática binaria mixta que resuelven por técnicas estándar. Aplican este modelo a la localización de parques de bomberos en Barcelona.

#### **II.6.4. La $p$ -mediana borrosa**

##### **II.6.4.1. Concepto y formulación matemática del problema borroso de la $p$ -mediana**

En [65] y [66], dijimos que para la formulación de los problemas de localización borrosos sobre un grafo o una red, podemos considerar las cuatro alternativas siguientes para la “fuzzificación” del modelo donde solamente uno de los elementos en el modelo (vértices  $V$ , aristas  $E$ , pesos  $W$  o la longitud  $l$ ) es dado usando técnicas borrosas:

- Problemas de localización con vértices borrosos. Modelan apropiadamente las situaciones donde los vértices de la demanda verifican una cierta condición (por ejemplo, ser un barrio importante) en un grafo dado. Esos problemas sobre un grafo para el cual un conjunto de los vértices de demanda  $V$  es borroso, pertenecen a ese tipo de modelo.
- Problemas de localización con aristas borrosas. Esto es una versión paralela al modelo precedente, cuando las aristas  $E$  están presentes en los modelos con un grado dado y están modeladas por un conjunto borroso de pares de vértices.
- Problemas de localización con pesos borrosos. En muchas situaciones, la información disponible no nos permite asignar valores deterministas a los pesos que representan la importancia de los vértices. En esos casos, en los cuales, por

ejemplo sentencias aparecen como: "el peso del vértice  $v$  es de alrededor  $w$ ", es necesario considerar que los pesos en  $w(v)$  son números borrosos.

- Problemas de localización con longitudes borrosas. Ahora nos referimos al problema de localización sobre un grafo  $G=(V, E)$  donde las longitudes de las aristas no son muy bien conocidas, y están dadas por una función  $l$  de valores borrosos,  $l(e)$ .

Cada uno de estos problemas puede conducirnos a los modelos alternativos que suministrarán nuevas soluciones para los problemas de transporte que queremos considerar.

Así, en las líneas siguientes, fijamos nuestra atención al problema de  $p$ -mediana con pesos borrosos, que es un caso muy importante desde el punto de vista económico, como lo veremos.

Aunque Daniel Serra y Vladimir Marianov [92] presentan un problema de la  $p$ -mediana en una situación de incertidumbre, M.J. Canós, C. Ivorra y Vicente Liern [18][19][22] abordan de manera precisa y profunda el problema borroso de la  $p$ -mediana especialmente en [18]. Parten de su formulación del problema tradicional de la  $p$ -mediana:

*p-mediana (P3) – M.J. Canos*

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_{ij} = w_j \quad 1 \leq j \leq n, \quad (3.1)$$

$$0 \leq x_{ij} \leq w_j y_i \quad 1 \leq i, j \leq n, \quad (3.2)$$

$$\sum_{i=1}^n y_i = p \quad (3.3)$$

$$y_i \in \{0, 1\} \quad 1 \leq i \leq n.$$

donde  $x_{ij}$  es la demanda del vértice  $v_j$  cubierta por el centro de servicio del vértice  $v_i$  (si es el caso),  $y_i$  es 1 si hay un centro de servicio en el vértice  $v_i$  y 0 en otro caso,  $w_i$  es la demanda del del vértice  $v_i$ ,  $d_{ij}$  es la distancia desde  $v_i$  hasta  $v_j$ .

Las restricciones (1) aseguran que toda la demanda se cubre a cada nodo  $v_j$ , las restricciones (2) garantizan que solo los vértices con una facilidad suministrarán el producto, y la restricción (3) establece que exactamente  $p$  facilidades se localizarán.

Este problema tradicional de la  $p$ -mediana, muestra que toda la demanda de cada vértice será totalmente cubierta por la facilidad más cercana, pero no da la libertad al decisor para reducir la demanda que se debe cubrir. La libertad del decisor es fundamental desde el punto de vista económico. M.J. Canós, C.Ivorra y V. Liern [18], proponen una versión borrosa del problema clásico de la  $p$ -mediana. Consideran un conjunto borroso de restricciones para que el decisor sea capaz de tomar en cuenta soluciones que permiten obtener costes significativamente bajos dejando una parte de la demanda sin cubrir. Parten de la definición siguiente:

Sea un objetivo borroso  $\tilde{G}^i$  y una restricción borrosa  $\tilde{C}^i$  en el espacio de alternativas  $X$ . Entonces  $\tilde{G}^i$  y  $\tilde{C}^i$  se combinan para formar una decisión  $\tilde{D}^i$ , que es un conjunto borroso que resulta de la intersección de  $\tilde{G}^i$  y  $\tilde{C}^i$ , es decir  $\tilde{D}^i = \tilde{G}^i \cap \tilde{C}^i$  y correspondientemente

$$\mu_{\tilde{D}^i} = \min\{\mu_{\tilde{G}^i}, \mu_{\tilde{C}^i}\}.$$

Quieren que el grado de pertenencia al conjunto borroso de restricciones  $\tilde{C}^i$  dependa de la demanda actualmente cubierta que cada solución suministra, es decir el grado de factibilidad de cada solución  $(x_{ij}, y_i)$  dependerá de la diferencia entre el total de la demanda que sería cubierta y la realmente cubierta. El grado de factibilidad de cada solución será expresado por la función

$$\sum_{j=1}^n w_j - \sum_{i=1}^n \sum_{j=1}^n x_{ij}$$

y se calcula por la fórmula siguiente:

$$\mu_f(x_{ij}, y_i) = h_f\left(\sum_{j=1}^n w_j - \sum_{i=1}^n \sum_{j=1}^n x_{ij}\right),$$

donde  $h_f$  es una función auxiliar dada por

$$h_f(x) = \begin{cases} 1 & \text{si } x < 0, \\ 1 - x/p_f & \text{si } 0 \leq x \leq p_f \\ 0 & \text{si } x > p_f \end{cases}$$

$p_f$  representa el nivel de tolerancia máxima para que la solución sea considerada factible.

El objetivo borroso  $\tilde{G}^c$  es reducir el coste. En una situación tradicional, con un coste óptimo  $z^*$ , el grado de mejora del objetivo, para cada solución  $(x_{ij}, y_i)$ , es decir el grado de pertenencia a  $\tilde{G}^c$  se define por

$$\mu_g(x_{ij}, y_i) = h_g(z^* - \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij})$$

donde  $h_g$  es otra función auxiliar dada por

$$h_g(x) = \begin{cases} 1 & \text{si } x < 0, \\ x/p_g & \text{si } 0 \leq x \leq p_g \\ 0 & \text{si } x > p_g \end{cases}$$

y  $p_g$  indica hasta cuanto el coste sería reducido para que la mejora sea considerada como completamente satisfactoria.

Las previas consideraciones conducen a este problema borroso de la  $p$ -mediana:

Encontrar  $(x_{ij}, y_i)$

$$\text{s.a.} \quad \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \leq \frac{z^*}{\%} \quad (4)$$

$$\sum_{i=1}^n x_{ij} \leq w_j, \quad 1 \leq j \leq n, \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} \geq \frac{\sum_{i=1}^n w_i}{\%}, \quad (6)$$

$$0 \leq x_{ij} \leq w_j y_i, \quad 1 \leq i, j \leq n, \quad (7)$$

$$\sum_{i=1}^n y_i = p, \quad (8)$$

$$y_i \in \{0, 1\}, \quad 1 \leq i \leq n, \quad (9)$$

La restricción (4) refleja que aunque deseamos minimizar el coste, por debajo de cierta meta  $z^*$ , si la superamos en cierta medida, estamos satisfechos en cierto grado (de cumplimiento de nuestro deseo de optimización). La restricción (5) relaja la restricción (1) del problema tradicional, para permitir soluciones factibles, pero añade la restricción (6) que formaliza la noción de la factibilidad borrosa.

Las restricciones (7) y (8) se interpretan como en el problema tradicional.

Así, pues, el objetivo es encontrar soluciones  $(x_{ij}, y_i)$  que satisfacen todas esas restricciones.

#### II.6.4.2. El grado global de satisfacción de una solución dada.

En [18], los autores denotan por  $\lambda$  el grado de pertenencia al conjunto de decisión  $D_0$ . Lo denominan “grado global de satisfacción” de una solución dada. Se busca encontrar soluciones con el mayor valor de  $\lambda$ . Se pueden encontrar con el problema auxiliar tradicional siguiente:

$$\begin{aligned} & \max \lambda \\ \text{s.a.} \quad & \lambda \leq \mu_g(x_{ij}, y_i), & 1 \leq j \leq n, & 1 \leq i \leq n, & (1) \\ & \lambda \leq \mu_f(x_{ij}, y_i), & 1 \leq j \leq n, & 1 \leq i \leq n, & (2) \\ & 0 \leq x_{ij} \leq w_j y_i, & 1 \leq j \leq n, & 1 \leq i \leq n, & (3) \\ & \sum_{i=1}^n x_{ij} \leq w_j, & 1 \leq j \leq n, & & (4) \\ & \sum_{i \in I} y_i = p, & & & (5) \\ & 0 \leq \lambda \leq 1, & & & (6) \\ & y_i \in \{0, 1\}, & 1 \leq i \leq n. & & (7) \end{aligned}$$

### II.6.4.3. El algoritmo de enumeración para solucionar problemas borrosos de la p-mediana cuando el número de nodos es pequeño

El algoritmo de enumeración se utiliza cuando el número de los vértices no es muy grande. M.J. Canós, C.Ivorra y V. Liern [18], lo presentan detalladamente. Resumimos aquí sus explicaciones:

Se empieza por enumerar los conjuntos posibles  $A_1, \dots, A_k$ , es decir todos los subconjuntos de  $p$  elementos cada uno que cubrirán las demandas de los nodos  $i$ . Los conjuntos  $A_i$  de  $p$  elementos son pues las alternativas posibles de solución. La demanda de cada vértice será completamente cubierta por la facilidad más cercana del conjunto  $A_i$ . Así, para una solución  $A_i$  dada, la distancia que será recorrida para atender al vértice  $v_j$  es dada por

$$D_{ij} = \min_{l \in A_i} d_{lj}$$

Así, el coste total de la cubierta por  $A_i$  será dado por

$$\sum_{j=1}^n w_j D_{ij}.$$

Según Hakimi, el punto mediano se puede calcularse por

$$z^* = \min_{1 \leq i \leq k} \sum_{j=1}^n w_j D_{ij}.$$

Ahora fijamos un punto  $A_i$  y estudiamos la reducción del coste que se puede obtener reduciendo la demanda, en cada vértice  $v_j$ , de la cantidad estipulada  $w_j$  a una nueva cantidad  $x_j$ .

Para cada solución factible determinada por  $x_j$ , podemos calcular su grado de factibilidad, su grado de mejora del objetivo, y su grado global de satisfacción de la siguiente manera:

$$\begin{aligned} \mu_f(x_j) &= h_f \left( \sum_{j=1}^n (w_j - x_j) \right), \\ \mu_g^i(x_j) &= h_g \left( z^* - \sum_{j=1}^n x_j D_{ij} \right), \\ \lambda_i(x_j) &= \min \{ \mu_f(x_j), \mu_g(x_j) \}. \end{aligned}$$

Sea  $\lambda_i^*$ , el grado óptimo de satisfacción que se puede alcanzar con  $A_i$ . Se obtiene solucionando el problema lineal siguiente ( $P_i$ ):

$$\begin{aligned} & (P_i) \\ & \max \lambda_i \\ & \text{s.a.:} \\ & \lambda_i \leq \mu_g^i(x_j), \\ & \lambda_i \leq \mu_f(x_j), \\ & 0 \leq \lambda_i \leq 1 \end{aligned}$$

El mejor grado posible de satisfacción es

$$\lambda^* = \max_{1 \leq i \leq k} \lambda_i^*.$$

Se obtiene en un punto óptimo  $A_i^*$  con las demandas actualmente cubiertas  $x_i^*$ , que constituye la solución borrosa óptima.

Los problemas  $P_i$  se pueden solucionar por métodos usuales, pero M.J. Canós et al proponen el siguiente algoritmo con varias etapas:

(1) Ordenar todos los vértices  $v_{j_1}, \dots, v_{j_n}$  según los valores decrecientes de  $D_{ij}$ .

Observe que si  $n' = n - p$  entonces  $D_{ij} = 0$  para  $k > n'$  (los últimos vértices son los en que se han localizado las facilidades).

(2) Llamar  $x = \sum_{j=1}^n w_j - \sum_{j=1}^n x_j$  la reducción total de la demanda cubierta asociada a

la solución parcialmente factible  $x_j$ . Ahora observamos que para cada unidad reducimos la demanda en cada vértice  $v_{j_k}$ . El coste está reducido en unidades de  $D_{ij_k}$ . Así, entre todas las soluciones  $x_j$ , con un  $x \leq w_{j_1}$  dado, la reducción más grande del coste se obtiene fijando  $x_{j_1} = w_{j_1} - x$  y dejando  $x_{j_k} = w_{j_k}$  para  $k > 1$  (es igual a  $D_{ij_1} x$ ). Si  $w_{j_1} < x \leq w_{j_1} + w_{j_2}$ , entonces reducimos la demanda en el nodo  $v_{j_1}$  a 0 y la demanda en  $v_{j_2}$  a  $w_{j_2} - (x - w_{j_1})$ , y así la mejor reducción del coste vuelve a ser  $D_{ij_1} w_{j_1} + D_{ij_2} (x - w_{j_1})$ , así sucesivamente. En general la

mejor reducción del coste entre todas las soluciones con un valor dado  $x$ , es dada por

$$R(x) = \begin{cases} D_{i,j_1} x & \text{si } x \leq w_{j_1} \\ D_{i,j_1} w_{j_1} + D_{i,j_2} (x - w_{j_1}) & \text{si } w_{j_1} < x \leq w_{j_1} + w_{j_2} \\ \text{M} & \text{M} \\ \sum_{k=1}^{n-1} D_{i,j_k} + D_{i,j_n} \left( x - \sum_{k=1}^{n-1} w_{j_k} \right) & \text{si } \sum_{k=1}^{n-1} w_{j_k} < x \end{cases}$$

La función  $R$  es lineal por trozos, con nodos en los puntos  $a_s = \sum_{k=1}^s w_{j_k}$  para

$s = 1, \dots, n$ . Llamamos  $a_0 = 0$ . Entonces la pendiente de  $R$  en el intervalo  $[a_{s-1}, a_s]$

es  $D_{i,j_s}$ . También  $R(a_s) = \sum_{k=1}^s D_{i,j_k} w_{j_k}$ .

- (3) Considerar la mejor solución parcialmente factible ( $c_i$ ) para un valor dado de  $r$ . Su grado de factibilidad y su grado de mejora del objetivo se pueden calcular como  $\mu_f(x) = h_f(x)$  y

$$\mu_g^i(x) = h_g \left( z^* - \sum_{j=1}^n D_{ij} + R(x) \right).$$

- (4) La función  $\mu_f(x)$  es lineal en el intervalo  $[0, p_f]$ , y estrictamente decreciente de 1 a 0. En el otro lado,  $\mu_g^i(x)$  es lineal por trozos, y es estrictamente creciente desde el punto en que deja el valor 0. Si este punto es menor que  $p_c$ , los dos gráficos tienen un punto único de intersección  $x^*$ , cuya imagen es  $\lambda_i^*$ , el mejor grado de satisfacción asociado al punto  $A_i$ . Pero, debemos saber que el problema  $P_i$  puede tener ninguna solución para algunos valores de  $i$ . Este puede ocurrir cuando el coste para  $A_i$  es muy lejos de  $z^*$ , de tal manera que  $z^*$  no se puede mejorar desde  $A_i$  reduciendo la demanda en  $p_f$  unidades.

- (5) A partir de  $x^*$  podemos reconstruir toda la solución del problema  $P_i$ . Sea  $r$  el índice de tal manera que  $a_{r-1} \leq x^* < a_r$ , entonces



$$x_{j_s} = \begin{cases} 0 & \text{para } s < r, \\ w_{j_r} - x^* + a_{r-1} & \text{para } s = r, \\ w_{j_s} & \text{para } r < s \leq n. \end{cases}$$

El algoritmo de enumeración que se presenta más arriba se basa en las siguientes observaciones:

- 1  $r$  es definido como el primer índice de modo que  $\mu_f(a_s) \leq \mu_g^i(a_s)$ . Si existe un valor común para  $\mu_f$  y  $\mu_g^i$ , será estrictamente entre 0 y 1 y así no necesitamos truncar los dos grados. Sólo examinaremos

$$1 - \frac{a_s}{p_f} \leq \frac{z^* - \sum_{j=1}^n D_{ij} w_j + R(a_s)}{p_g}$$

o equivalentemente

$$1 - \frac{a_s}{p_f} \leq \frac{z^* - \mu_s}{p_g},$$

donde

$$\mu_s = \sum_{k=s+1}^n D_{ik} w_{j_k}.$$

- 2 Para encontrar  $r$ , podemos sólo ver los valores de  $s$  desde 1 hasta  $n' = n - p$ , puesto que, obviamente, se echará una solución que reduce las demandas de los vértices de facilidades.
- 3 Una vez que se encuentra  $r$ , el punto  $(r^*, \lambda_i^*)$  se puede obtener como la intersección de dos líneas rectas. Una de ellas es simplemente

$$y = 1 - \frac{r}{p_f}.$$

La otra contiene el punto

$$\left( a_r, \frac{(x^* - \mu_r)}{p_g} \right)$$

y su cota es

$$\frac{D_{j_r}}{p_g}.$$

La geometría elemental nos conduce a la fórmula:

$$\lambda_i^* = \frac{D_{ijk} (p_f - a_r) + z^* - \mu_r}{p_g + p_f D_{ijk}}$$

que es fácilmente manejable.

#### II.6.4.4. Nuestro modelo del problema borroso de la p-mediana

##### II.6.4.4.1. Nuestros conceptos del grafo borroso y de la p-mediana borrosa

Tenemos una red o grafo  $G = (N, A)$ , donde  $N = \{1, \dots, n\}$  es el conjunto de nodos y  $A$  es un conjunto de aristas existentes entre cada par de nodos  $(i, j)$  en el grafo. Cada nodo  $i \in N$  tiene un peso positivo  $w_i$  que representa la población (demanda) en el nodo. Cada arista  $(i, j)$  tiene un peso  $l_{ij}$  que representa la longitud de la arista. A partir de los pesos de aristas existentes, se calcula una matriz  $D$  de distancias  $d_{ij}$  de los caminos más cortos entre los pares de nodos  $(i, j)$ ,  $\forall i, j \in N, i \neq j$ .

En una situación de incertidumbre, los valores  $w_i$  y  $l_{ij}$  son aproximativos e imprecisos. Con tales datos es evidente que localizaciones óptimas obtenidas mediante la p-mediana crisp no serán precisas. El coste óptimo obtenido con la p-mediana crisp tampoco será preciso. Así, el problema básico es determinar el grado de cumplimiento de las localizaciones y coste óptimo cuando los valores  $w_i$  y  $l_{ij}$  son aproximativos e imprecisos.

Así, definimos los subconjuntos borrosos siguientes, para una p-mediana borrosa:

- $\mathcal{W}^c \subset W = \left\{ (w_i | \mu_{\mathcal{W}^c}(w_i)) \right\}$ ,  $\forall i \in N$  es el subconjunto borroso de los pesos  $w_i$  más precisos del conjunto referencial  $W$ .  $\mu_{\mathcal{W}^c}(w_i)$  es el grado de pertenencia del nodo  $i$  al subconjunto borroso  $\mathcal{W}^c$ , en otros términos  $\mu_{\mathcal{W}^c}(w_i)$  es el grado de cumplimiento del peso  $w_i$ .
- $\mathcal{L}^c \subset L = \left\{ (l_{ij} | \mu_{\mathcal{L}^c}(l_{ij})) \right\}$ ,  $\forall i, j \in N$  es el subconjunto borroso de las longitudes  $l_{ij}$  más precisas de las aristas entre los nodos  $i$  y  $j$ .  $\mu_{\mathcal{L}^c}(l_{ij})$  es el grado de pertenencia

de la longitud  $l_{ij}$  al subconjunto borroso  $\mathcal{L}'$ , en otros términos  $\mu_{\mathcal{L}'}(l_{ij})$  es el grado de cumplimiento de la longitud  $l_{ij}$ .

- A partir del subconjunto  $\mathcal{L}' \subseteq L$ , construimos una matriz borrosa  $\mathcal{B}'$  de las distancias más precisas  $d_{ij}$  de los caminos más cortos entre los pares de nodos  $(i, j)$ . Definimos la matriz borrosa por  $\mathcal{B}' = \left\{ \left( d_{ij} \mid \mu_{\mathcal{B}'}(d_{ij}) \right) \right\}, \forall i, j \in N$ , donde  $d_{ij}$  es la distancia del camino más corto entre los nodos  $i$  y  $j$  calculada usando el algoritmo de Floyd-Warshall [26] y  $\mu_{\mathcal{B}'}(d_{ij})$  es el grado de cumplimiento de  $d_{ij}$ .

El objetivo tanto de la p-mediana borrosa como de la p-mediana crisp queda siempre el de hallar localizaciones óptimas de  $p$  facilidades de modo que se minimice la distancia total ponderada o el coste total de recorrida desde cada nodo de demanda al nodo de facilidad más cercano. El aspecto particular de la p-mediana borrosa será de determinar hasta qué punto las localizaciones y el coste óptimos son precisos, es decir el grado de cumplimiento de las localizaciones y coste óptimos.

Entonces la diferencia entre la p-mediana borrosa y la p-mediana crisp es:

- La p-mediana crisp maneja datos precisos (deterministas) con grados de pertenencia 1 y permite hallar localizaciones óptimas precisas para  $p$  facilidades con un coste óptimo preciso con grado de cumplimiento 1. No hay solución imprecisa con el grado de cumplimiento entre 0 y 1.
- La p-mediana borrosa maneja datos aproximativos e imprecisos (borrosos) con grados de pertenencia entre 0 y 1 y permite hallar localizaciones óptimas aproximativas o imprecisas (borrosas) para  $p$  facilidades con un coste óptimo aproximativo, impreciso con grado de cumplimiento incluido en el intervalo  $[0,1]$ .

La p-mediana borrosa tal como acabamos de presentarla es una generalización de la p-mediana crisp y es más aplicable a la realidad que la p-mediana crisp, puesto que la realidad es siempre hecha de incertidumbre.

### II.6.4.4.2. Nuestra formulación del problema borroso de la p-mediana

Con los datos imprecisos definidos aquí arriba, formularemos el problema de la p-mediana de la manera siguiente:

#### *p*-mediana difusa

$$(z^* | \lambda^*) \cong \min \sum_i \sum_j (w_i | \mu_{\mathcal{W}^0}(w_i)) (d_{ij} | \mu_{\mathcal{D}^0}(d_{ij})) x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N, \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j = p, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i, j \in N, \quad (5)$$

donde

$z^*$  = coste total óptimo de transporte,

$\lambda^*$  = grado de cumplimiento del coste óptimo (de la solución óptima)

$\mathcal{W}^0$  = subconjunto borroso de los pesos  $w_i$  más precisos de los nodos  $i$ .

$\mathcal{D}^0$  = matriz borrosa de distancias  $d_{ij}$  más precisas de los caminos más cortos entre los pares de nodos  $(i, j)$ .

$i \in N$  = nodo de demanda,

$j \in N$  = nodo de facilidad,

$w_i$  = población aproximativo o demanda aproximativa del nodo  $i$ ,

$\mu_{\mathcal{W}^0}(w_i)$  = grado de cumplimiento de  $w_i$ .

$d_{ij}$  = distancia aproximativa del camino más corto entre el nodo  $i$  y el nodo  $j$ ,

$\mu_{\mathcal{D}^0}(d_{ij})$  = grado de cumplimiento de  $d_{ij}$ .

$x_{ij} = 1$  si el nodo  $i$  se asigna al nodo  $j$  y 0 si no,

$y_j = 1$  si hay una facilidad en  $j$  y 0 si no,

$p$  = número de facilidades a localizar.

### II.6.4.4.3. Operación sobre los grados de pertenencia

El modelo de la p-mediana borroso que acabamos formular tiene que aplicarse sobre los grafos borrosos. En estos grafos, no disponiendo de las demandas exactas en cada nodo ni de las distancias exactas de las aristas, expresamos estas cantidades o valores acompañados de sus grados de pertenencia. Consecuentemente, la función objetivo del modelo de la p-mediana borrosa que se aplique a un tal grafo no calcula solamente el coste óptimo sino también el grado de cumplimiento de dicho coste.

En la función objetivo del modelo se utiliza el operador de agregación  $\Sigma$  para la suma. Este operador se usará para calcular el coste óptimo. ¿Cuales operaciones debemos efectuar sobre los grados de pertenencia para obtener el grado de cumplimiento de la solución final? Se sabe bien que los elementos de una colección de conjuntos borrosos pueden combinarse para producir un conjunto borroso único mediante operaciones de agregación [83]. Es el caso, por ejemplo, en la intersección e unión de cualquier número de conjuntos borrosos. En general, muchas otras clases de agregación pueden efectuarse sobre los conjuntos borrosos, las operaciones siendo caracterizadas como sigue.

Una agregación es una operación  $n$ -aria  $A:[0,1]^n \rightarrow [0,1]$  que cumple los siguientes requerimientos [83]:

- Condiciones del límite:  $A(0, \dots, 0) = 0$  y  $A(1, \dots, 1) = 1$
- Monotonía:  $A(x_1, \dots, x_n) \geq A(y_1, \dots, y_n)$  si  $x_i \geq y_i, i = 1, \dots, n$ .

Las operaciones de agregación discutidas en [83] son los operadores compensatorias definidas por Zimmermann y Zysno (1980), las sumas simétricas de Dubois y Prade (1980), los operadores de la media discutidos por Dyckhoff y Pedrycz (1984) y el operador de media ponderada ordenada (OWA\*) propuesto por Yager (1988).

Zimmermann y Zysno propusieron una familia de operadores compensatorios en la cual una operación de agregación se presenta como una combinación de puras conexiones lógicas AND y OR facilitando el mecanismo requerido de compensación. La operación definida en Zimmermann y Zysno (1980) se expresa como

---

\* OWA: Ordered Weighted Averaging operator

$$(A \odot B)(x) = [(A \cap B)(x)]^{1-\gamma} [(A \cup B)(x)]^\gamma,$$

donde  $\cap$  y  $\cup$  son operadores de intersección y unión, mientras que  $\gamma$  es un factor de compensación y  $\gamma \in [0,1]$  indica donde el operador actual se localiza entre AND y OR.

Hay también otras maneras de resolver un efecto similar de compensación, como

$$(A \otimes B)(x) = (1-\gamma)[(A \cap B)(x)] + \gamma[(A \cup B)(x)]$$

Las operaciones de intersección y unión pueden modelarse usando los operadores estándares max y min o, en general, las normas triangulares.

Las sumas simétricas facilitan otra opción para agregar los grados de pertenencia. Más formalmente, son funciones de  $n$ -argumentos, denotadas por  $S\_sum$ , tal como además de cumplir las condiciones del límite y la monotonía, son continuos, conmutativas y auto-duales:

$$S\_sum(x_1, \dots, x_n) = 1 - S\_sum(1 - x_1, \dots, 1 - x_n).$$

Dubois y Prade (1980) demostraron que cualquier suma simétrica puede representarse en la forma

$$S\_sum(x_1, \dots, x_n) = \left[ 1 + \frac{\rho(1 - x_1, \dots, 1 - x_n)}{\rho(x_1, \dots, x_n)} \right]^{-1},$$

donde  $\rho$  es cualquier función continua creciente con  $\rho(0, \dots, 0) = 0$ .

Un operador de la media (media generalizada) para  $n$  argumentos es idempotente y conmutativa, además de cumplir la monotonía y las condiciones del límite. Como discutido en Dyckhoff y Pedrycz (1984), la media generalizada toma la forma

$$A(x_1, \dots, x_n) = \sqrt[p]{\frac{1}{n} \sum_{i=1}^n (x_i)^p}, \quad p \in \mathbb{R}, p \neq 0.$$

Es importante notar que la familia de la media generalizada supone algunos casos bien conocidos de operadores de conjuntos borrosos. En particular, tenemos

1. Media Aritmética ( $p = 0$ )  $A(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n (x_i)$
2. Media Geométrica ( $p \rightarrow 0$ )  $A(x_1, \dots, x_n) = (x_1, \dots, x_n)^{\frac{1}{n}}$
3. Media Harmónica ( $p = -1$ )  $A(x_1, \dots, x_n) = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$

$$4. \text{ Minimum } (p \rightarrow -\infty) \quad A(x_1, \dots, x_n) = \min(x_1, \dots, x_n)$$

$$5. \text{ Maximum } (p \rightarrow \infty) \quad A(x_1, \dots, x_n) = \max(x_1, \dots, x_n)$$

El parámetro  $p$  puede referirse como un factor de compensación.

En nuestro modelo del problema de la  $p$ -mediana borrosa, los grados de pertenencia se manipularán usando la operación de media aritmética en vez de la suma y la operación mínimo en vez de la multiplicación. Por eso modelamos el operador de agregación para la media aritmética que utilizaremos en todo el resto de este trabajo como sigue

$$\bar{M}x_i = \frac{1}{n} \sum_{i=1}^n x_i$$

Utilizaremos la forma  $\frac{1}{n} \sum_{i=1}^n x_i$  para calcular los grados de pertenencia en los problemas de

la  $p$ -mediana y la forma  $\bar{M}x_i$  en los problemas de centro.

#### II.6.4.4.4. Solución del modelo

El modelo de la  $p$ -mediana borrosa que acabamos de presentar se resuelve en 2 pasos: primero se calcula normalmente la  $p$ -mediana crisp con los datos aproximativos para hallar las localizaciones y coste óptimos crisp, es decir se utiliza el modelo de arriba sin los grados de pertenencia:

*p*-mediana borrosa: localizaciones y coste óptimos

$$z^* = \min \sum_i \sum_j w_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N, \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j = p, \quad (3)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i, j \in N, \quad (4)$$

Al cabo del paso 1, estaremos en presencia de un grafo en que se conoce el conjunto de  $p$  nodos de facilidad, las asignaciones de los demás  $n-p$  nodos de demanda a los  $p$  nodos de

facilidad. Conociendo que la solución obtenida se basa sobre datos aproximativos de cuyos grados de pertenencia  $\mu_{w_i}(w_i)$  y  $\mu_{d_{ij}}(d_{ij})$  disponemos, los vamos a utilizar para calcular el grado de cumplimiento de la solución obtenida en el paso 1 usando la fórmula siguiente :

*p*-mediana borrosa: cálculo del grado de cumplimiento

$$\lambda^* = \frac{1}{n-p} \sum_{i=1}^{n-p} \left( \frac{1}{p} \sum_{j=1}^p \min(\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij})) x_{ij} \right) \quad (1)$$

$$\text{s.a.} \quad \sum_{j=1}^p x_{ij} = 1 \quad \forall i = 1, \dots, n-p \quad (2)$$

$$\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij}) \in [0, 1] \quad \forall i = 1, \dots, n-p \quad \forall j = 1, \dots, p \quad (3)$$

La sentencia (1) calcula el grado de cumplimiento  $\lambda^*$  de la solución óptima  $Z^*$  obtenida en el paso 1 siempre que cada nodo de demanda se asigne a un solo nodo de facilidad (2) y que los pesos de los nodos y de las aristas sean grados de pertenencia (3). Si  $w_i$  y  $d_{ij}$  se conocen con certeza absoluta, es decir si  $\mu_{w_i}(w_i) = 1, \forall i \in N$  y  $\mu_{d_{ij}}(d_{ij}) = 1, \forall d_{ij} \in D$ , entonces  $\lambda^* = 1$ , no hace falta ir al paso 2, pues se trata de una *p*-mediana crisp. Pero si  $w_i$  y  $d_{ij}$  no se conocen con certeza absoluta, es decir si  $\mu_{w_i}(w_i) \in [0, 1], \forall i \in N$  y  $\mu_{d_{ij}}(d_{ij}) \in [0, 1], \forall d_{ij} \in D$ , entonces, por los pasos 2 y 3, tenemos  $\lambda^* \in [0, 1]$ ; se trata de una *p*-mediana borrosa. Así pues, la *p*-mediana borrosa es una generalización de la *p*-mediana crisp.

#### II.6.4.4.5. Algoritmo y programa de resolución

Intentar ahora escribir un algoritmo para el problema de la *p*-mediana es querer reinventar la rueda porque varios algoritmos proliferan en la literatura tanto para el problema de la *p*-mediana como para los de centro y routing. El aspecto particular del problema de la *p*-mediana borrosa como de los de centro y routing que examinamos a continuación es el cálculo del grado de cumplimiento. Así, proponemos ahora el algoritmo del cálculo del grado de cumplimiento de una solución óptima de la *p*-mediana.



✓ **Algoritmo : Grado de cumplimiento p-Mediana**

Datos de entrada:

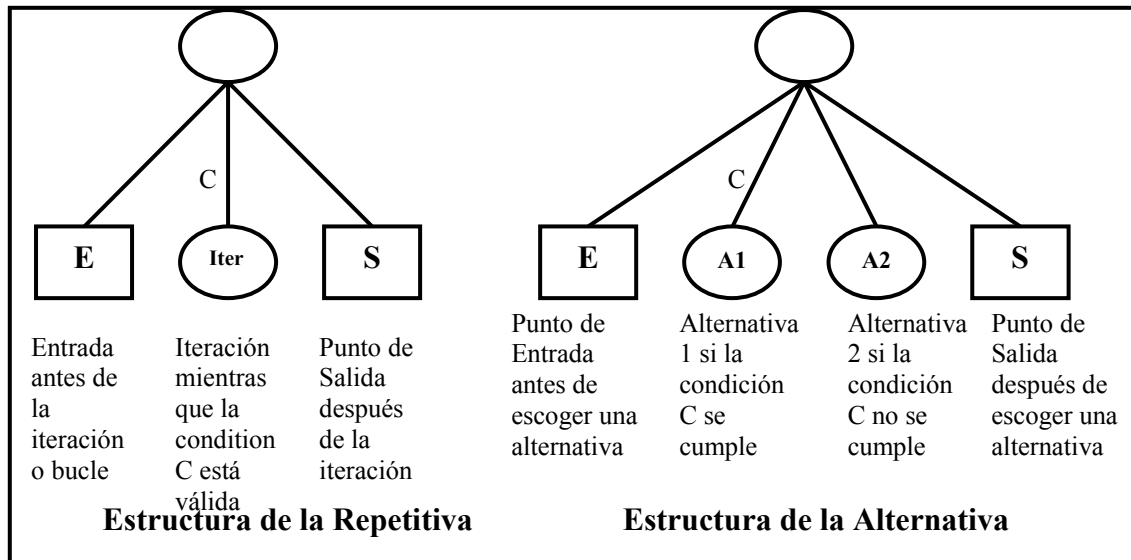
- El vector  $W_{(n-p)}$  de los grados de pertenencia  $\mu_{\beta_0}(w_i)$
- La matriz  $D_{(n-p) \times p}$  de los grados de pertenencia  $\mu_{\beta_0}(d_{ij})$
- La matriz de las asignaciones  $X_{(n-p) \times p} = [x_{ij}]_{(n-p) \times p}$ . Esta matriz debe construirse tomando en cuenta la restricción  $\sum_j x_{ij} = 1, \forall i$ .

VARIABLES UTILIZADAS EN EL ALGORITMO:

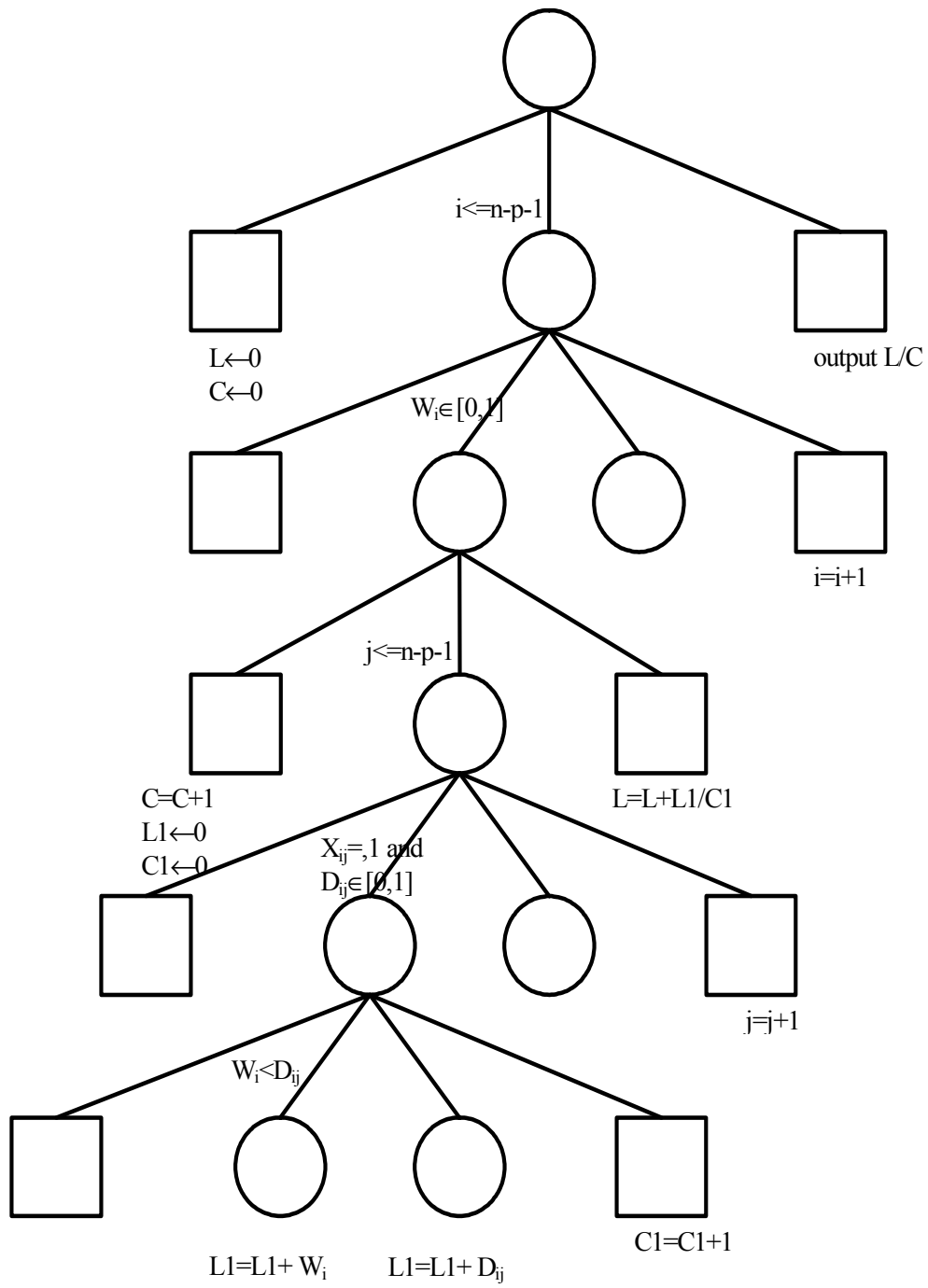
- L1 y L almacenan el resultado calculado del bucle más interior al bucle más exterior, respectivamente
- C1 y C son contadores que sirven como denominadores en los cálculos del bucle más interior al bucle más exterior respectivamente.
- i y j son índices de acceso a los datos de entrada.
- n almacena el número de nodos de la red (grafo).
- p almacena el número de nodos con facilidad.

A lo largo de este trabajo construiremos algoritmos usando los árboles programáticos seguidos de los pseudocódigos correspondientes. Los árboles programáticos son una herramienta eficaz de construcción de algoritmos estructurados permitiéndonos analizar un problema según la metodología denominada "Top-Down Programming". El problema se analiza desde arriba es decir desde su estructura general bajando hasta hallar sus estructuras particulares o elementales.

Un árbol programático se construye mediante dos estructuras básicas: La repetitiva (iteración) y la alternativa.



El árbol programático del cálculo del grado de cumplimiento para la p-mediana se presentará como sigue:



✓ **Pseudocódigo: Grado de cumplimiento p-Mediana**

```

L←0
C←0
for i=0 to n-p-1
    if  $W_i \in [0, 1]$ 
        C1←0
        L1←0
        C=C+1
        for j=0 to p-1
            if  $X_{ij}=1$  and  $D_{ij} \in [0, 1]$ 
                if  $W_i < D_{ij}$ 
                    L1=L1+Wi
                else
                    L1=L1+Dij
            C1=C1+1
        j=j+1
        L=L+L1/C1
    i=i+1
output L/C

```

Tiempo computacional:  $O(n^2)$

**II.6.4.4.6. Implementación del algoritmo en C++**

```

//=====
//Fuzzy p-Median: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Wednesday, November 24th, 2004
//Department of Computer Sciences and Artificial
//Intelligence
//University of Granada (Spain)
//=====

```

```

#include <iostream.h>
#include <stdlib.h>
main(){
    double L=0,L1;
    int C=0,C1;
    cout << "Fuzzy P-Median\n";
    cout << "\nNumber of nodes: ";
    cin >> n;
    cout << "\nNumber of facility nodes: ";
    cin >> p;

    //Declaring input data
    double W[n-p], D[n-p][p];
    bool X[n-p][p];

    //Entering input data
        // Will be written later
    //Processing data
    for (int i=0; i<=n-p-1;){
        if (W[i]>=0 && W[i]<=1){
            C1=0;
            L1=0;
            ++C;
            for (int j=0; j<=p-1;){
                if ((X[i][j])&&
                    D[i][j]>=0 && D[i][j]<=1){
                    if (W[i]<D[i][j]){
                        L1+=W[i];
                    }
                    else
                        L1+=D[i][j];
                }
                ++C1;
            }
            ++j;

```

```

    }
    L += L1/C1;
  }
  ++i;
}
L /= C;
Cout << "\n\nThe degree of certainty is " << L;
return 0;
}

```

## II.7. Noción sobre $p$ -mediana capacitada y $p$ -mediana no capacitada

### II.7.1. Concepto y formulación matemática del problema de la $p$ -mediana capacitada

El problema de la  $p$ -mediana capacitada (CPMP) busca solucionar la localización óptima de las  $p$  facilidades, considerando las distancias y capacidades para el servicio que cada facilidad ofrecerá. El servicio total pedido por los vértices identificados por los clusters de la  $p$ -mediana no puede exceder la capacidad de su facilidad (véanse Luiz A.N. Lorena et al [70]). Su formulación matemática se presenta de la siguiente manera:

*p*-mediana capacitada

$$\min \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}, \quad (1)$$

$$\text{s.a.} \quad \sum_{j \in N} x_{ij} = 1, \quad i \in N, \quad (2)$$

$$\sum_{j \in N} x_{jj} = p, \quad (3)$$

$$\sum_{i \in N} q_i x_{ij} \leq Q x_{jj}, \quad j \in N, \quad (4)$$

$$x_{ij} \in \{0, 1\}, \quad i \in N, \quad j \in N, \quad (5)$$

donde

$N = \{1, \dots, n\}$  es el conjunto de índices de entidades a asignar y también de las facilidades posibles, donde  $p$  facilidades estarán localizadas;  $q_i$  es la demanda de cada entidad y  $Q$  la

capacidad de cada facilidad posible;  $[d_{ij}]_{n \times m}$  es una distancia de la matriz;  $[x_{ij}]_{n \times m}$  es la asignación de la matriz, con  $x_{ij} = 1$  si la entidad  $i$  se asigna a la facilidad  $j$ , y  $x_{ij} = 0$  en el caso contrario;  $y_j = 1$  si la facilidad  $j$  está seleccionada (abierta) y  $y_j = 0$ , en el caso contrario.

Las restricciones (2) y (3) imponen que cada entidad se asigna a una sola facilidad y que el número de facilidades a localizar es  $p$ . La restricción (4) impone que una capacidad total de la facilidad debe ser respetada, y la restricción (5) resfuera el carácter binario de las decisiones de asignación y de localización a un nodo.

Observamos que esta formulación de Luiz A.N. Lorena et al [70] presenta la dificultad para calcular el coste de transporte, puesto que la primera sentencia de su formulación no incluye la variable  $q_i$ , o la demanda del usuario. En efecto, el coste de transporte es proporcional no sólo a la distancia de recorrido sino también a la cantidad del producto transportada (cantidad de demanda cubierta). Además la capacidad  $Q$  debe tener un índice que indica a qué facilidad se refiere. En efecto hay  $p$  facilidades con capacidades  $Q_j$ ,  $j = 1, \dots, p$ .

Así, presentamos la formulación siguiente para el problema de la  $p$ -mediana capacitada:

### *$p$ -mediana capacitada*

$$\min \sum_{i \in N} \sum_{j \in N} w_i d_{ij} x_{ij}, \quad (1)$$

$$\text{s.a.} \quad \sum_{j \in N} x_{ij} = 1, \quad i \in N, \quad (2)$$

$$\sum_{j \in N} y_j = p, \quad (3)$$

$$\sum_{i \in N} w_i x_{ij} \leq q_j y_j, \quad j \in N, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i, j \in N, \quad (5)$$

Se puede notar que la diferencia entre el problema de la  $p$ -mediana clásico tal como lo presentamos en las secciones precedentes y el problema de la  $p$ -mediana capacitada reside solamente en que el problema de la  $p$ -mediana capacitada fija la cantidad del servicio o de

producto que la facilidad  $j$  suministra. Lo que es muy importante desde el punto de vista económico. El problema de la  $p$ -mediana capacitada permite al decisor localizar sus facilidades cualquiera sea la capacidad de estas facilidades. Cubrirá sólo, toda o parte de la demanda que corresponda a la capacidad de sus facilidades. Mientras que en el problema de la  $p$ -mediana clásico el decisor tendrá que cubrir toda la demanda de los usuarios. Se nota también que en el problema de la  $p$ -mediana capacitado existe un conjunto de  $p$  capacidades de las  $p$  facilidades.

Formularemos el problema de la  $p$ -mediana capacitada borrosa como sigue:

*p-mediana capacitada borrosa*

$$(z^* | \lambda^*) \cong \min \sum_i \sum_j (w_i | \mu_{\beta_0}(w_i)) (d_{ij} | \mu_{\beta_0}(d_{ij})) x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j = p, \quad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i, j \in N, \quad (5)$$

$$\sum_{i \in N} w_i x_{ij} \leq q_j y_j, \quad \forall j \in N \quad (6)$$

donde

$q_j =$  capacidad de la facilidad en el nodo  $j$ .

Nuestro modelo de la  $p$ -mediana capacitada borrosa nos permite también afirmar que la  $p$ -mediana capacitada borrosa es una generalización de la  $p$ -mediana capacitada crisp.

En efecto, el modelo se resuelve en dos tiempos: primero se calcula normalmente la  $p$ -mediana capacitada crisp con los datos aproximativos para hallar las localizaciones y el coste óptimos imprecisos, es decir se utiliza el modelo de arriba sin los grados de pertenencia:



*p*-mediana capacitada borrosa: localizaciones y coste óptimos

$$z^* = \min \sum_i \sum_j w_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N, \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j = p, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i, j \in N, \quad (5)$$

$$\sum_{i \in N} w_i x_{ij} \leq q_j y_j, \quad j \in N, \quad (6)$$

Al cabo del paso 1, estaremos en presencia de un grafo en que se conoce el conjunto de  $p$  nodos de facilidad, las asignaciones de los demás  $n-p$  nodos de demanda a los  $p$  nodos de facilidad. Conociendo que la solución obtenida se basa sobre datos aproximativos cuyos grados de pertenencia  $\mu_{w_i}(w_i)$  y  $\mu_{d_{ij}}(d_{ij})$  disponemos, los vamos a utilizar para calcular el grado de cumplimiento de la solución obtenida en el paso 1 usando el modelo siguiente:

*p*-mediana capacitada borrosa: cálculo del grado de cumplimiento

$$\lambda^* = \frac{1}{n-p} \sum_{i=1}^{n-p} \left( \frac{1}{p} \sum_{j=1}^p \min(\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij})) x_{ij} \right) \quad (1)$$

$$\text{s.a.} \quad \sum_{j=1}^p x_{ij} = 1 \quad \forall i = 1, \dots, n-p \quad (2)$$

$$\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij}) \in [0, 1] \quad \forall i = 1, \dots, n-p \quad \forall j = 1, \dots, p \quad (3)$$

La función objetivo (1) de este submodelo calcula el grado de cumplimiento  $\lambda^*$  de la solución óptima  $Z^*$  obtenida en el paso 1 siempre que cada nodo de demanda se asigne a un solo nodo de facilidad (2) y que los pesos de los nodos y de las aristas sean grados de pertenencia (3).

Si  $w_i$  y  $d_{ij}$  se conocen con certeza absoluta, es decir si  $\mu_{w_i}(w_i) = 1, \forall i \in N$  y  $\mu_{d_{ij}}(d_{ij}) = 1, \forall d_{ij} \in D$ , entonces  $\lambda^* = 1$  y no hace falta ir al paso 2, pues se trata de la  $p$ -

mediana capacitada crisp. Pero si  $w_i$  y  $d_{ij}$  no se conocen con certeza absoluta, es decir si  $\mu_{p^*}(w_i) \in [0,1]$ ,  $\forall i \in N$  y  $\mu_{p^*}(d_{ij}) \in [0,1]$ ,  $\forall d_{ij} \in D$ , entonces, por el paso 2, tenemos  $\lambda^* \in [0,1]$ ; se trata de la p-mediana capacitada borrosa. Así pues, la p-mediana capacitada borrosa generaliza la p-mediana capacitada crisp.

Se utiliza el mismo algoritmo y el mismo programa presentados más arriba para solucionar este modelo.

### **II.7.2. Concepto y formulación matemática del problema de la p-mediana no capacitada**

I.R. de Farias Jr [40], explica el problema de la p-mediana no capacitada de la siguiente manera:

Sea  $M = \{1, \dots, m\}$  un conjunto de almacenes, y  $N = \{1, \dots, n\}$  un conjunto de los clientes. El problema de la p-mediana no capacitada es escoger no más de p almacenes para atender a los clientes a un coste mínimo, donde la capacidad de cada almacén individual es por lo menos grande como la suma de las demandas de todos los clientes.

Sea  $x_{ij}$  la fracción de la demanda del cliente j cubierta desde el almacén i,  $y_i = 1$  o 0 dependiendo de si el almacén i está abierto o no, respectivamente,  $c_{ij} \geq 0$  es el coste para enviar una unidad del producto desde el almacén i hasta el cliente j, y  $d_j > 0$  la demanda del cliente j,  $i \in M, j \in N$ , el problema de la p-mediana no capacitada puede formularse de la siguiente manera:

*p*-mediana no capacitada – I.R. de Farias Jr

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} d_j x_{ij},$$

$$\text{s.a.} \quad \sum_{i \in M} x_{ij} = 1, \quad j \in N, \quad (1)$$

$$x_{ij} \leq y_i, \quad i \in M, \quad j \in N, \quad (2)$$

$$\sum_{i \in M} y_i \leq p, \quad (3)$$

$$x_{ij} \geq 0, \quad i \in M, \quad j \in N, \quad (4)$$

$$y_i \in \{0,1\}, \quad i \in M. \quad (5)$$

La variable  $c_{ij}$  que representa el coste de transporte de una unidad del producto desde el nodo  $i$  (sitio de facilidad) al nodo  $j$  (nodo demanda), es el equivalente de la variable  $d_{ij}$  que representa la distancia de recorrido en el problema de la  $p$ -mediana capacitada y en el problema clásico de  $p$ -mediana. Pues el coste de transporte y la distancia de recorrido entre dos puntos son conceptos equivalentes. La variable  $d_j$  es el equivalente de la variable  $c_j$ , representando la demanda al nodo del usuario. La variable  $x_{ij}$  juega el mismo papel como en los otros problemas de la  $p$ -mediana.

Ahora intentemos escribir de nuevo esta misma formulación pero utilizando las mismas variables como en los otros problemas de la  $p$ -mediana y considerando que el índice  $i \in I$  representa el conjunto de los nodos de demanda (usuarios) y el índice  $j \in I$  los nodos de facilidad. La misma formulación puede escribirse de la siguiente manera:

*p*-mediana no capacitada

$$\min \sum_{i \in N} \sum_{j \in N} w_i d_{ij} x_{ij},$$

$$\text{s.a.} \quad \sum_{j \in N} x_{ij} = 1, \quad i \in N, \quad (1)$$

$$x_{ij} \leq y_j, \quad i \in N, \quad j \in N, \quad (2)$$

$$\sum_{j \in N} y_j \leq p, \quad (3)$$

$$x_{ij} \geq 0, \quad i \in N, \quad j \in N, \quad (4)$$

$$x_{ij}, y_j \in \{0,1\}, \quad i \in N, \quad j \in N, \quad (5)$$

Ahora, se nota que la diferencia entre el problema clásico de la  $p$ -mediana y el problema de la  $p$ -mediana no capacitada reside en la restricción que fija el número de facilidades a localizar. En el problema clásico de la  $p$ -mediana, ese número tiene que ser  $p$  mientras que el problema de la  $p$ -mediana no capacitada admite que ese número sea inferior a  $p$  en la única condición que toda la demanda de los usuarios se cubra en un coste mínimo de transporte. Según la formulación del modelo borroso de la  $p$ -mediana que hemos presentado más arriba, el de la  $p$ -mediana no capacitada borrosa se presentará de la manera siguiente:

*p-mediana no capacitada borrosa*

$$(z^* | \lambda^*) \equiv \min \sum_i \sum_j (w_i | \mu_{\beta^0}(i)) (d_{ij} | \mu_{\beta^0}(d_{ij})) x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N, \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j \leq p, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i, j \in N, \quad (5)$$

Aquí también afirmamos que la  $p$ -mediana borrosa no capacitada es una generalización de la  $p$ -mediana crisp no capacitada.

El modelo se resuelve en dos tiempos. Primero se calcula la  $p$ -mediana no capacitada crisp, es decir se utiliza el modelo de arriba sin los grados de pertenencia:

*p-mediana no capacitada borrosa: localizaciones y coste óptimos*

$$z^* = \min \sum_i \sum_j w_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_j x_{ij} = 1, \quad \forall i \in N, \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i, j \in N, \quad (3)$$

$$\sum_j y_j \leq p, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i, j \in N, \quad (5)$$

Al cabo del paso 1, estaremos en presencia de un grafo en que se conoce el conjunto de  $p$  nodos de facilidad, las asignaciones de los demás  $n-p$  nodos de demanda a los  $p$  nodos de facilidad. Conociendo que la solución obtenida se basa sobre datos aproximativos cuyos grados de pertenencia  $\mu_{w_i}(w_i)$  y  $\mu_{d_{ij}}(d_{ij})$  disponemos, los vamos a utilizar para calcular el grado de cumplimiento de la solución obtenida en el paso 1 usando el modelo siguiente :

*p*-mediana no capacitada borrosa: cálculo del grado de cumplimiento

$$\lambda^* = \frac{1}{n-m} \sum_{i=1}^{n-m} \left( \frac{1}{m} \sum_{j=1}^m \min(\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij})) x_{ij} \right), \quad (1)$$

$$\text{s.a.} \quad \sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, n-m \quad m \leq p \quad (2)$$

$$\mu_{w_i}(w_i), \mu_{d_{ij}}(d_{ij}) \in [0,1] \quad \forall i = 1, \dots, n-m, \forall j = 1, \dots, m \quad m \leq p \quad (3)$$

La función objetivo (1) de este submodelo calcula el grado de cumplimiento  $\lambda^*$  de la solución óptima  $Z^*$  obtenida en el paso 1 siempre que cada nodo de demanda se asigne a un solo nodo de facilidad (2) y que los pesos de los nodos y de las aristas sean grados de pertenencia incluidos en el intervalo  $[0,1]$  (3).

Si  $w_i$  y  $d_{ij}$  se conocen con certeza absoluta, es decir si  $\mu_{w_i}(w_i) = 1, \forall i \in N$  y  $\mu_{d_{ij}}(d_{ij}) = 1, \forall d_{ij} \in D$ , entonces  $\lambda^* = 1$  y no hace falta ir al paso 2, pues se trata de una *p*-mediana no capacitada crisp. Pero si  $w_i$  y  $d_{ij}$  no se conocen con certeza absoluta, es decir si  $\mu_{w_i}(w_i) \in [0,1], \forall i \in N$  y  $\mu_{d_{ij}}(d_{ij}) \in [0,1], \forall d_{ij} \in D$ , entonces, por el paso 2, tenemos  $\lambda^* \in [0,1]$ ; se trata de una *p*-mediana no capacitada borrosa. Así pues, la *p*-mediana no capacitada borrosa generaliza la *p*-mediana no capacitada crisp.

Utilizamos el mismo algoritmo y el mismo programa presentados más arriba para solucionar este modelo

## II.8. Conclusión

En esta parte hemos visto detalladamente el problema de la  $p$ -mediana de que nos serviremos para optimizar la red de transporte de la ciudad de Kinshasa. Hemos visto que el modelo de la  $p$ -mediana tradicional se utiliza sobre un grafo determinista para localizar  $p$  centros de facilidad de manera que se minimice la suma ponderada de las distancias de recorrida entre los nodos de demanda y los centros de facilidad más cercanos. El modelo de la  $p$ -mediana tradicional se formula de muchas maneras, pero se interpretan siempre de la misma manera puesto que se trata de minimizar el coste de transporte.

A partir de la tesis doctoral de M.J. Canós [16], hemos descubierto las técnicas de resolución de un problema de la  $p$ -mediana clásico o tradicional y el problema de la  $p$ -mediana en incertidumbre pero su formulación de la  $p$ -mediana borrosa en [18] y [19] se aplica a los grafos deterministas aunque económicamente interesante.

Hemos formulado un modelo del problema borroso de la  $p$ -mediana para los grafos borrosos y hemos que hemos extendido a los problemas de la  $p$ -mediana capacitada y no capacitada. El aspecto particular que diferencia la  $p$ -mediana borrosa de la crisp es que la primera calcula el grado de cumplimiento de la solución obtenida por la segunda. Así para calcular el grado ese grado de cumplimiento, hemos presentado un modelo correspondiente seguido de algoritmo de solución que hemos implementado en el lenguaje de programación C++.

Después de localizar  $p$  facilidades, el trabajo siguiente consiste en determinar un o dos centros de transporte y rutas fijos para los autobuses. Una de las alternativas para establecer esas rutas es que entre todos los  $p$  centros de facilidad, uno de ellos será considerado como central de modo que todos los autobuses tendrán como destino este nodo. Esto implica que una persona deberá hacer como máximo un trasbordo, sea cual sea el destino elegido. Así, en la forma secuencial se trataría de elegir el punto de la  $p$ -mediana que sea el mejor en algún sentido, por ejemplo, el que esté a menor distancia de todos los demás. O bien se trataría de un problema de localización como el de la  $p$ -mediana central alguno similar. Lo que nos permite abordar los problemas de centros.

## CAPÍTULO III. POBLEMAS DE LOCALIZACIÓN DE CENTRO Y “ROUTING”

### III.1. Introducción

El problema básico del diseño del sistema radial de centros (SRC) se describe como sigue: dada una red de nodos donde cada par de nodos tiene un flujo dado, determinar cuales nodos establecer como centros de modo que cada flujo pase primero a través de uno o dos centros antes de que alcance su destino [34]. Los problemas de diseño de un sistema radial de centros (SRC) se refieren también como problemas de localización de centro, y son problemas crisp. Después de haber recorrido toda la literatura sobre los problemas de localización de centros, nos hemos dado cuenta de que hasta la fecha, no existen modelos borrosos de localización de centros sobre una red borrosa. Supongamos que tenemos una red de nodos  $N = \{1, L, n\}$  donde cada par de nodos  $(i, j)$  tiene un flujo aproximativo dado por  $W_{ij}$  y un coste aproximativo por unidad de flujo dado por  $C_{ij}$ . Estos dos datos siendo imprecisos, se desea determinar cuales nodos establecer como centros de modo que cada flujo pase primero a través de uno o dos centros antes de que alcance su destino. Así, en este trabajo estaremos simultaneamente estudiando los modelos crisp existentes y desarrollando modelos borrosos equivalentes. Los problemas de localización de centro tienen muchas aplicaciones. Ocurren generalmente en casos donde el tráfico (como correos, paquetes de telecomunicación, o pasajeros aéreos) se debe transportar de su punto de origen a su destino, pero donde es indeseable, impracticable o extremadamente costoso dedicar conexiones de transporte exclusivas entre cada par origen-destino (O-D) [31][32].

Para ilustrar la situación [32], considérese un servicio de correo en el que se tiene un conjunto de distritos postales que intercambian correos (es decir, el tráfico). Cada distrito postal puede representarse por un nodo. Dos distritos postales pueden conectarse por una conexión de transporte, representada como una arista dirigida. Si hay una arista conectando los distritos  $i$  y  $j$ , entonces esta puede llevar correos directamente desde  $i$  hasta  $j$ . El conjunto entero de nodos y aristas se conoce como una red. A menudo, en los problemas de localización de centro hay una corriente de tráfico entre cada par ordenado de nodos. En otras palabras, existe una corriente de tráfico que debe fluir de cada nodo a cada otro nodo. Típicamente es muy costoso dedicar un medio de transporte a cada corriente de tráfico. Por eso se utilizan los sistemas de

centro. Los centros que se escogen entre los nodos (distritos postales) de la red obran como centros de consolidación o centros de clasificación para el tráfico (correos) y se asume generalmente que están completamente interconectados. Facilitan el flujo de tráfico e incorporan un descuento del volumen en el flujo inter-centro. El flujo directo entre nodos no centros no está permitido. Por el contrario, el flujo originando en (o destinado a) un nodo no centro debe coleccionarse en (o distribuirse por) un nodo centro. Se estipula que cada envío de tráfico se debe mandar por uno o (por lo mucho) dos centros.

Hay un coste asociado para enviar el flujo entre dos nodos cualesquiera, que se basa en los factores tales como el tiempo o la distancia. En algunos casos hay también un coste fijo para establecer un centro [32].

Hay muchas variantes de los problemas de localización de centro. Por ejemplo, se puede especificar que  $p$  de los nodos son centros. Estos se refieren como problemas de localización de  $p$ -centro. Los problemas de localización de centro pueden incluir una sola o asignación múltiple de los nodos no centros a los centros. Si un nodo está solamente asignado, entonces todo tráfico desde ese nodo viaja directamente hasta o desde solamente un centro; si está múltiplemente asignado, entonces su tráfico puede dividirse y puede viajar directamente hasta o desde más de un centro. Además, los problemas de localización de centro pueden ser capacitados o libres de cualquier capacidad. Esto significa que la cantidad del flujo que puede estar redirigida o clasificada por un centro estará restringida a algún volumen máximo, conocido como su capacidad y denotado por  $\Gamma_k$  para un centro  $k$  [32] y puede haber un coste fijo asociado al establecimiento de un nodo dado como centro [31]. En la tabla siguiente resumimos las variantes de los problemas de localización de centro y routing que vamos a analizar en este capítulo:



*Variantes de los problemas de centro*

<i>Problemas con asignación única</i>	<i>Problemas con asignación múltiple</i>
USApHMP <sup>1</sup> Problema de p-mediana central no capacitada con asignación única	UMApHMP <sup>2</sup> Problema de p-mediana central no capacitada con asignación múltiple
USApHLP <sup>3</sup> Problema de localización de p-centro no capacitado con asignación única	UMApHLP <sup>4</sup> Problema de localización de p-centro no capacitado con asignación múltiple
Phap <sup>5</sup> Problema de asignación de p-centro	
	UMAHLP <sup>6</sup> Problema de localización de centro no capacitado con asignación múltiple
	CMAHLP <sup>7</sup> Problema de localización de centro capacitado con asignación múltiple

### III.2. Problemas de localización con asignación única

Los problemas de localización de centro con asignación única se describen como sigue [31]: Tenemos la localización de un conjunto de nodos  $N = \{1, L, n\}$ , el volumen del flujo ( $W_{ij}$ ) que se debe transportar entre cada par de nodos O-D ( $i, j$ ) y el coste por unidad del flujo ( $C_{ij}$ ) entre cada par O-D de ( $i, j$ ). Se asume que  $C_{ij} = C_{ji}$ . En el conjunto  $N$  de nodos, se seleccionan  $p$  nodos como centros (especialmente  $p = 2$  o  $3$ ). Los centros son los puntos de consolidación y redirección para el flujo y están completamente interconectados. Todo flujo transcurre vía los nodos centros y cada nodo no centro debe asignarse a un único nodo centro. La localización de los centros y la asignación de los nodos a los centros se escogen de modo que se minimice el coste total del sistema. Muchas veces, este problema se refiere como el problema de p-mediana centro no capacitada con asignación única. Aquí  $p$  representa el número de centros en el problema. También se considera el caso donde los nodos centros se conocen a priori; se refiere a este problema como el problema de asignación de p-centro. Para ambos problemas, tales redes tienen generalmente la ventaja de pocas conexiones de transporte y a menudo atraen un descuento sobre el flujo inter-centro. Se debe notar que todo

<sup>1</sup> USApHMP: Uncapacitated Single Allocation p-Hub Median Problem

<sup>2</sup> UMApHMP: Uncapacitated Multiple Allocation p-Hub Median Problem

<sup>3</sup> USApHLP: Uncapacitated Single Allocation p-Hub Location Problem

<sup>4</sup> UMApHLP: Uncapacitated Multiple Allocation p-Hub Location Problem

<sup>5</sup> pHAP: p-Hub Allocation Problem

<sup>6</sup> UMAHLP: Uncapacitated Multiple Allocation Hub Location Problem

<sup>7</sup> CMAHLP: Capacitated Multiple Allocation Hub Location Problem

flujo que debe transportarse entre nodos tiene tres componentes separados: la colección (nodo origen – centro), la transferencia (inter-centro) y la distribución (centro – nodo destino). Cada componente tiene un coeficiente del coste por unidad de  $\chi, \alpha$  y  $\delta$  respectivamente. Por eso, el coste de enviar  $W_{ij}$  unidades de flujo de  $i$  a  $j$  vía los centros  $k$  y  $l$  sería la suma de  $\chi W_{ij} C_{ik}$  (el coste de colección de  $W_{ij}$  unidades de flujo por el centro  $k$ ), de  $\alpha W_{ij} C_{kl}$  (el coste de transferir  $W_{ij}$  unidades de flujo del centro  $k$  al centro  $l$ ) y  $\delta W_{ij} C_{lj}$  (el coste de distribuir  $W_{ij}$  unidades del flujo del centro  $l$  a  $j$ ). A menudo, debido a las eficiencias de transferencia inter-centro,  $\alpha < 1$  es el factor de descuento, mientras que  $\chi, \delta = 1$ .

### III.2.1. Problemas USA $p$ HMP y pHAP

#### III.2.1.1. Aspecto crisp de los problemas USA $p$ HMP y pHAP

El USA $p$ HMP fue estudiado por Aykin[3][4][5][6][7], Campbell [13][14][15], Ernst y Krishnamoorthy [38], Helm y Venkataramanan [52], Klineciewicz [61][62], O’Kelly [78][79][80][81][82], D.Skorin-Kapov y J.Skorin-Kapov [93], D.Skorin-Kapov et al. [94] Smith et al [95] y Sohn y Park [96]. El USA $p$ HMP fue formulado por la primera vez por O’Kelly (1987) como un problema entero cuadrático, la función objetivo para esta formulación era no convexa. Campbell [14] reformuló el problema como un programa mixto lineal entero con  $O(n^4)$  variables. Ernst y Krishnamoorthy [38] presentaron un programa mixto lineal entero utilizando  $O(n^3)$  variables para resolver el USA $p$ HMP.

La gran parte del trabajo reciente en la localización se ha centrado en la solución de los problemas más grandes. Utilizando técnicas y formulaciones desarrolladas en [31], se puede ahora empezar a solucionar problemas de tamaño práctico para el USA $p$ HMP y el pHAP. Se presenta una formulación que resuelva el USA2HMP y USA2HMP utilizando  $O(n^2)$  variables. Se presenta también una segunda formulación que soluciona el 2HAP y 3HAP usando  $O(n)$  variables. Finalmente se presenta una formulación para el USA $p$ HMP que es válida para  $p$  centros, que utiliza solamente  $O(n^2)$  variables y restricciones. Se revisa las formulaciones matemáticas existentes para el USA $p$ HMP y pHAP.

Para el USA $\rho$ HMP, se define los parámetros de las formulaciones consideradas como sigue [31]:

$C_{ij}$  el coste por unidad del flujo de  $i$  a  $j$ ,

$W_{ij}$  el montante del flujo de  $i$  a  $j$ ,

$$V_{ij} = \begin{cases} W_{ij} + W_{ji} & \forall i > j \\ 0 & \text{en el caso diferente} \end{cases}$$

$O_i = \sum_{j \in N} W_{ij}$  el flujo total originando en el nodo  $i$ ,

$D_i = \sum_{j \in N} W_{ji}$  el flujo total destinado para el nodo  $i$ ,

$\chi$ ,  $\alpha$  y  $\delta$  coeficientes respectivos de la colección, transferencia y distribución,

$n$  el número de nodos,

$p$  el número de nodos que se deben escoger como centros,

$h$  el conjunto de nodos centros conocidos.

Las variables utilizadas en las formulaciones siguientes son:

$X_{ijkl}$  la fracción del flujo total del nodo  $i$  a  $j$  rutada vía los centros  $k$  y  $l$ ,

$Y_{kl}^i$  el flujo del nodo  $i$  vía los centros  $k$  y  $l$

$E_{kl} + E_{lk}$  el flujo entre centros  $k$  y  $l$ .  $E_{kk}$  se define siempre a cero,

$Q_{ik}$  el coste de transferir entre centros el flujo originando en  $i$ , si  $i$  se asigna a  $k$ ,

$t_{ik}$  el flujo del nodo  $i$  que se transfiere entre centros si el nodo  $i$  se asigna al centro  $k$ ,

$$H_{ik} = \begin{cases} 1 & \text{si el nodo } i \text{ se asigna al centro } k \\ 0 & \text{si no} \end{cases}$$

$$B_{ij} = \begin{cases} 1 & \text{si los nodos } i \text{ y } j \text{ se asignan al mismo centro} \\ 0 & \text{si no} \end{cases}$$

La primera formulación de USA $\rho$ HMP fue presentada por O'Kelly [80] como un programa entero cuadrático. La función objetivo de su formulación era no-convexa y consecuentemente difícil para resolver. Con pocos cambios de notación, la formulación de O'Kelly se expresa como sigue [31]:

*Formulación USApHMP – Q*

$$\min \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \sum_{j \in N} W_{ij} (\chi C_{ik} H_{ik} + \alpha C_{kl} H_{ik} H_{jl} + \delta C_{lj} H_{jl}) \quad (1)$$

$$\text{sujeto a } \sum_{k \in N} H_{kk} = p, \quad (2)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (4)$$

$$H_{ik} \in \{0,1\} \quad \forall i, k \in N. \quad (5)$$

La ecuación (2) especifica el número de centros y (3) es la restricción de asignación única. La ecuación (4) asegura que un nodo puede asignarse solamente a un centro. Esta formulación tiene  $O(n^2)$  variables, a pesar de que este problema es difícil resolverse debido a la no-convexidad de su función objetivo. Después Campbell [14] desarrolló en 1994 una formulación MILP (Programa Mixto Lineal Entero) con  $O(n^4)$  para evitar la no-convexidad de la función objetivo. Pero Skorin-Kapov y sus compañeros [94] modificaron la formulación MILP de Campbell. La formulación modificada se expresa como sigue [31]:

*Formulación USApHMP – N<sup>4</sup>*

$$\min \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \sum_{j \in N} W_{ij} C_{ijkl} X_{ijkl} \quad (6)$$

$$\text{s.a. } \sum_{k \in N} H_{kk} = p, \quad (7)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (8)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (9)$$

$$\sum_{l \in N} X_{ijkl} = H_{ik} \quad \forall i, j, k \in N, \quad (10)$$

$$\sum_{k \in N} X_{ijkl} = H_{jl} \quad \forall i, j, l \in N, \quad (11)$$

$$H_{ik} \in \{0,1\} \quad X_{ijkl} \geq 0 \quad \forall i, j, k, l \in N \quad (12)$$

Las ecuaciones (10) y (11) aseguran que el volumen correcto del flujo se mueve entre nodos y que el flujo se consolida en los centros.

La *Formulación USApHMP*– $N^4$  tiene  $(n^4 + n^2)$  variables de las cuales  $n^2$  son binarias y  $(2n^3 + n^2 + n + 1)$  son restricciones lineales. Esta formulación es significativamente más fácil resolverse que su equivalente *USApHMP*– $Q$ . La limitación es que aún para  $n$  relativamente pequeño, el problema vuelve a ser intratable debido al gran número de variables y restricciones. Por eso, Ernst y Krishnamoorthy [38] desarrollaron una formulación con muy pocas variables y restricciones en el esfuerzo de solucionar los problemas de tamaño práctico. Con pocos cambios de notación, así se presenta dicha formulación [31]:

*Formulación USApHMP*– $N^3$

$$\min \sum_{i \in N} \sum_{k \in N} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \alpha C_{kl} Y_{kl}^i \quad (13)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p, \quad (14)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (15)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (16)$$

$$\sum_{l \in N} Y_{kl}^i - \sum_{l \in N} Y_{lk}^i = O_i H_{ik} - \sum_{j \in N} W_{ij} H_{jk} \quad \forall i, k \in N, \quad (17)$$

$$H_{ik} \in \{0,1\} \quad Y_{kl}^i \geq 0 \quad \forall i, k, l \in N \quad (18)$$

La ecuación (17) describe un problema de flujo multiproducto, donde cada producto representa el flujo de tráfico originando en un nodo particular.

La *Formulación USApHMP*– $N^3$  tiene  $(n^3 + n^2)$  variables de las cuales  $n^2$  son binarias y  $(2n^2 + n + 1)$  son restricciones lineales. Así el *USApHMP*– $N^3$  es  $O(n)$  más pequeño que el *USApHMP*– $N^4$  en ambos números de variables y de restricciones. Aún así, problemas más grandes de alrededor  $n=50$  vuelven a ser también difíciles resolverse debido a las limitaciones de la memoria. Los problemas más grandes con asignación única se habían solucionado con el enfoque basado en el camino más corto (Shortest Path Based Approach – SPBA) descrito por Ernst y Krishnamoorthy [39].

En cuanto al problema de asignación de  $p$ -centro ( $p$ -Hub Allocation Problem –  $p$ HAP), Sohn y Park [97] formularon el 2HAP como [31]:

*Formulación 2HAP –  $N^2$*

$$\min \sum_{i=3}^n \sum_{k=1}^2 C_{ik} H_{ik} (\chi O_i + \delta D_i) + C_{12} \sum_{i=2}^n \sum_{j \in N, j < i} V_{ij} B_{ij} \quad (19)$$

$$\text{s.a.} \quad \sum_{k=1}^2 H_{ik} = 1 \quad \forall i \in N, \quad (20)$$

$$H_{11} = 1 \quad H_{22} = 1 \quad (21)$$

$$H_{i1} - H_{j1} - B_{ij} \leq 0 \quad \forall i = 2, \dots, n, \quad \forall i > j, \quad (22)$$

$$-H_{i1} + H_{j1} - B_{ij} \leq 0 \quad \forall i = 2, \dots, n, \quad \forall i > j, \quad (23)$$

$$B_{ij} \geq 0 \quad \forall j \in N, \quad \forall i > j, \quad (24)$$

$$0 \leq H_{ik} \leq 1 \quad i \in N, \quad k \in \{1, 2\}. \quad (25)$$

La ecuación (20) es la restricción de asignación única, (21) define los nodos centros (están definidos a 1 o 2 sin pérdida de la generalidad) y (22) y (23) definen un politopo con puntos enteros extremos. Esta formulación necesita solamente  $H_{i1} \forall i > 2$ , puesto que  $H_{i2} = 1 - H_{i1} \forall i > 2$ . Esta formulación tiene  $O(n^2/2)$  variables y  $O(n^2)$  restricciones. Una formulación similar para el 3HAP se puede encontrar en Sohn y Park [97]. Se refiere a esta formulación encontrada en Sohn y Park [97] como  $3HAP - N^2$ .

Con el propósito de solucionar problemas aún más grandes, se formula el USA2HMP y USA3HMP utilizando solamente  $O(n^2)$  variables. Es la primera formulación lineal que utiliza  $O(n^2)$  para resolver un  $USApHMP$  [31]:

*Formulación USA2,3HMP –  $N^2$*

$$\min \sum_{i \in N} \sum_{k \in N} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{k \in N} \sum_{l \in N} \alpha C_{kl} E_{kl} \quad (26)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p \quad p = 2 \text{ o } 3 \quad (27)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (28)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in \{N : i \neq k\}, \quad (29)$$

$$t_{ik} \geq \sum_{j \in \{N : j < i\}} V_{ij} H_{ik} - \sum_{j \in \{N : j < i\}} (V_{ij} H_{jk}) \quad \forall i \in \{N : i \neq k\}, \quad \forall k \in N, \quad (30)$$

$$\sum_{l \in N} E_{kl} = \sum_{i \in \{N : i > j\}} t_{ik} \quad \forall k \in N, \quad (31)$$

$$\sum_{k \in N} E_{kl} \leq \sum_{j \in N} \left[ \left( \sum_{i \in \{N : i > j\}} V_{ij} \right) H_{jl} \right] \quad \forall l \in N, \quad (32)$$

$$\sum_{l \in N} E_{kl} - \sum_{l \in N} E_{lk} = \sum_{i \in N} \left[ \sum_{j \in \{N : j < i\}} V_{il} - \sum_{j \in \{N : j < i\}} V_{ji} \right] H_{ik} \quad \forall k \in N, \quad (33)$$

$$H_{ik} \in \{0, 1\}, \quad E_{kl}, t_{ik} \geq 0 \quad \forall i, k, l \in N. \quad (34)$$

Las tres primeras restricciones (27)-(29) describen la estructura de la red en el problema. Más específicamente, la ecuación (27) especifica que para esta formulación debe haber dos o tres centros, (28) es la restricción de asignación única y (29) asegura que los nodos están solamente asignados a los centros. Estas restricciones, juntas con el primer término de la función objetivo producen el coste correcto asociado a la colección y distribución del flujo en la red. La restricción siguiente (30) describe el volumen mínimo del flujo que un nodo dado debe transferir entre centros. Por eso (31) determina el volumen del flujo que está transferido del centro  $k$  a otros centros, puesto que (32) y (33) ambos restringen al flujo transitar entre nodos centros. Las restricciones (31) y (33) determinan también el volumen total del flujo destinado a cualquier centro vía una transferencia  $(\sum_l E_{lk})$ . El volumen total del flujo destinado a cualquier centro vía una transferencia, junto con el volumen del flujo transferido desde cualquier centro, lleva un coste único asociado a la transferencia del flujo en un sistema de dos o tres centros (en un espacio euclidío). Esto ya no es verdad en una red de cuatro o más centros, así este modelo está válido solamente para una red de dos o tres centros. La *Formulación USA2,3HMP* -  $N^2$  tiene  $(3n^2 - 2n)$  variables y  $(2n^2 + 2n + 1)$  restricciones. Así *USA2,3HMP* -  $N^2$  es  $O(n)$  más pequeño que *USApHMP* -  $N^3$  en el número de variables y tiene aproximadamente el mismo número de restricciones. Hay que notar que se pudiera formular *USA2HMP* en  $(3n^2 + n)/2 - 1$  variables, no obstante en la práctica una tal formulación exige más tiempo computacional para resolver un problema que la formulación que se acaba de presentar. La formulación *USA2,3HMP* -  $N^2$  desarrollada en [31] es más efectiva que *USApHMP* -  $N^3$  cuando se resuelvan problemas utilizando un código comercial

MILP. Además, para los problemas más grandes, resolver el programa mixto lineal entero de [31] con un código comercial es más efectivo que el acercamiento basado en el camino más corto (SPBA) desarrollado por Ernst y Krishnamoorthy [39]. Luego, se presenta el primer programa mixto lineal entero para el *USApHMP* que requiere solamente  $O(n^2)$  variables y restricciones pero válido para todo  $p$ . Esta formulación incorpora las ideas desarrolladas por Kaufman y Broeckx [60] para el problema de asignación cuadrática:

*Formulación USApHMP –  $N^2$*

$$\min \sum_{i \in N} \sum_{k \in N} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{i \in N} \sum_{k \in N} Q_{ik} \quad (35)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p, \quad (36)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (37)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (38)$$

$$Q_{ik} + \alpha \sum_{j \in N} \sum_{l \in N} V_{ij} C_{kl} (1 - H_{ik}) \geq \alpha \sum_{j \in N} \sum_{l \in N} V_{ij} C_{kl} H_{jl} \quad \forall i, k \in N, \quad (39)$$

$$H_{ik} \in \{0, 1\}, \quad Q_{ik} \geq 0, \quad \forall i, k \in N. \quad (40)$$

La ecuación (36) especifica el número de los centros mientras que (37) es la restricción de asignación única. La ecuación (38) asegura que un nodo puede asignarse a un centro. Finalmente (39) afirma que  $Q_{ik}$  debe ser al menos el coste de transferir el flujo de un nodo  $i$  vía el centro  $k$  (a todos los otros centros), si y solo si el nodo  $i$  se asigna al centro  $k$  (y cero si no, puesto que si  $i$  no se asigna a  $k$  entonces  $Q_{ik} + M \geq \alpha \sum_{j \in N} \sum_{l \in N} V_{ij} C_{kl} H_{jl}$  y  $M$  es grande,  $Q_{ik} = 0$ ). No obstante puesto que esto es una minimización del problema,  $Q_{ik}$  es simplemente el coste de transferir el flujo del nodo  $i$  vía el centro  $k$ . Una pequeña variación de la formulación *USApHMP –  $N^2$*  produce una formulación válida para cualquiera de los problemas siguientes:

- Problema de localización de centro capacitado con asignación única,
- Problema de  $p$ -mediana centro capacitada o no capacitada con asignación única.

La formulación *USApHMP –  $N^2$*  tiene  $2n^2$  variables de las cuales  $n^2$  son binarias y  $(2n^2 - n + 1)$  restricciones lineales. Se hubiera podido desarrollar algunas variantes de esta



formulación. En la práctica se notó que en aún el mejor de los casos el tiempo computacional requerido para resolver cualquier variante del  $USApHMP-N^2$  era más grande que el requerido para resolver la de  $USApHMP-N^3$  o de  $USA2,3HMP-N^2$ . La razón principal es que  $USApHMP-N^2$  tiene  $O(n^4)$  elementos no-cero mientras que las otras formulaciones mencionadas requieren solamente  $O(n^3)$  elementos no-cero. Transformando la formulación  $USA2,3HMP-N^2$  en un programa mixto lineal entero para resolver el  $2HAP$  y el  $3HAP$ , se llega a una formulación siguiente [31]:

*Formulación pHAP-N*

$$\min \sum_{i \in N} \sum_{k \in h} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{k \in h} \sum_{l \in h} \alpha C_{kl} E_{kl} \quad (41)$$

$$\text{s.a.} \quad H_{kk} = 1 \text{ o } 3 \quad \forall k \in h, \quad |h| = 2, \quad (42)$$

$$\sum_{k \in h} H_{ik} = 1 \quad \forall i \in N, \quad (43)$$

$$H_{ik} \leq H_{kk} \quad \forall i \in \{N : i \neq k\}, \quad k \in h, \quad (44)$$

$$t_{ik} \geq \sum_{j \in \{N : j < i\}} V_{ij} H_{ik} - \sum_{j \in \{N : j < i\}} (V_{ij} H_{jk}) \quad \forall i \in \{N : i \neq 1\}, \quad \forall k \in h, \quad (45)$$

$$\sum_{l \in h} E_{kl} = \sum_{i \in \{N : i \neq 1\}} t_{ik} \quad \forall k \in h, \quad (46)$$

$$\sum_{k \in h} E_{kl} \leq \sum_{j \in N} \left[ \left( \sum_{i \in \{N : i > j\}} V_{ij} \right) H_{jl} \right] \quad \forall l \in h, \quad (47)$$

$$\sum_{k \in h} E_{kl} - \sum_{l \in h} E_{lk} = \sum_{i \in N} \left[ \sum_{j \in \{N : j < i\}} V_{ij} - \sum_{j \in \{N : j < i\}} V_{ji} \right] H_{ik} \quad \forall k \in h, \quad (48)$$

$$H_{ik} \in \{0, 1\}, \quad E_{kl}, t_{ik} \geq 0, \quad \forall i \in N, \quad \forall k, l \in h. \quad (49)$$

Esta formulación tiene  $(2pn + p^2 - 2p)$  variables donde  $p$  es el número de localizaciones forzadas de centros. Para el  $2HAP$  y  $3HAP$  el número de localizaciones es 2 y 3, respectivamente. La formulación  $pHAP-N$  es más efectiva que las formulaciones de Sohn y Park para solucionar el  $2HAP$  y el  $3HAP$ .

### III.2.1.2. Aspecto borroso de los problemas USApHMP y pHAP

#### III.2.1.2.1. Problema USApHMP borroso

##### ✓ Formulación del modelo

Para formular el USApHMP borroso, partiremos de la formulación crisp con muy pocas variables y restricciones desarrollada por Ernst y Krishnamoorthy [38] en el esfuerzo de solucionar los problemas de tamaño práctico:

*Formulación USApHMP –  $N^3$*

$$\min \sum_{i \in N} \sum_{k \in N} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \alpha C_{kl} Y_{kl}^i \quad (1)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p, \quad (2)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (4)$$

$$\sum_{l \in N} Y_{kl}^i - \sum_{l \in N} Y_{lk}^i = O_i H_{ik} - \sum_{j \in N} W_{ij} H_{jk} \quad \forall i, k \in N, \quad (5)$$

$$H_{ik} \in \{0,1\} \quad Y_{kl}^i \geq 0 \quad \forall i, k, l \in N \quad (6)$$

En la incertidumbre se supone que los datos siguientes son aproximativos, vagos o imprecisos:

- $C_{ik}$  el coste por unidad para coleccionar el flujo del nodo  $i$  al centro  $k$ , el mismo variable se utiliza para el coste de la distribución.
- $C_{kl}$  el coste por unidad para transferir el flujo del centro  $k$  a  $l$ ,
- $W_{ij}$  el montante del flujo de  $i$  a  $j$ ,
- $O_i = \sum_{j \in N} W_{ij}$  el flujo total originando enl nodo  $i$ ,
- $D_i = \sum_{j \in N} W_{ji}$  el flujo total destinado para el nodo  $i$ ,
- $Y_{kl}^i$  el flujo del nodo  $i$  vía los centros  $k$  y  $l$

Estos datos serán elementos de los subconjuntos borrosos siguientes:

- $\mathcal{C}_{ij}^{\alpha} = \left\{ \left( C_{ij} \mid \mu_{\mathcal{C}_{ij}^{\alpha}}(C_{ij}) \right) \right\}, \forall i, j \in N$ , subconjunto borroso de costes unitarios  $C_{ij}$  más ciertos del flujo entre los nodos  $i$  y nodo  $j$ , donde  $\mu_{\mathcal{C}_{ij}^{\alpha}}(C_{ij})$  es el grado de pertenencia del coste  $C_{ij}$  al subconjunto borroso  $\mathcal{C}_{ij}^{\alpha}$ , o mejor dicho, su grado de cumplimiento.

- $\mathcal{W}^{\circ} = \left\{ \left( W_{ij} \mid \mu_{\mathcal{W}^{\circ}}(W_{ij}) \right) \right\}, \forall i, j \in N$ , subconjunto borroso de flujos  $W_{ij}$  más ciertos entre los nodos  $i$  y  $j$ , donde  $\mu_{\mathcal{W}^{\circ}}(W_{ij})$  es el grado de pertenencia del flujo  $W_{ij}$  al subconjunto borroso  $\mathcal{W}^{\circ}$ , o mejor dicho, su grado de cumplimiento.
- $\mathcal{O}^{\circ} = \left\{ \left( O_i \mid \mu_{\mathcal{O}^{\circ}}(O_i) \right) \right\}, \forall i \in N$ , subconjunto borroso de flujos  $O_i = \sum_{j \in N} W_{ij}$  más ciertos para todo  $i$ , donde  $\mu_{\mathcal{O}^{\circ}}(O_i) = \mathbf{M} \mu_{\mathcal{W}^{\circ}}(W_{ij})$  es el grado de pertenencia del flujo  $O_i$  al subconjunto borroso  $\mathcal{O}^{\circ}$ , o mejor dicho, su grado de cumplimiento. Recordamos que utilizamos  $\mathbf{M}$  como operador de agregación para la media aritmética.
- $\mathcal{D}^{\circ} = \left\{ \left( D_i \mid \mu_{\mathcal{D}^{\circ}}(D_i) \right) \right\}, \forall i \in N$ , subconjunto borroso de flujos  $D_i = \sum_{j \in N} W_{ji}$  más ciertos para todo  $i$ , donde  $\mu_{\mathcal{D}^{\circ}}(D_i) = \mathbf{M} \mu_{\mathcal{W}^{\circ}}(W_{ji})$  es el grado de pertenencia del flujo  $D_i$  al subconjunto borroso  $\mathcal{D}^{\circ}$ , o mejor dicho, su grado de cumplimiento.
- $\mathcal{Y}^{\circ} = \left\{ \left( Y_{kl}^i \mid \mu_{\mathcal{Y}^{\circ}}(Y_{kl}^i) \right) \right\}, \forall i, k, l \in N$ , subconjunto borroso de flujos  $Y_{kl}^i$  más ciertos para todo  $i, k$  y  $l$ , donde  $\mu_{\mathcal{Y}^{\circ}}(Y_{kl}^i)$  es el grado de pertenencia del flujo  $Y_{kl}^i$  al subconjunto borroso  $\mathcal{Y}^{\circ}$ , o mejor dicho, su grado de cumplimiento.

Dados los subconjuntos borrosos definidos aquí arriba, desarrollamos el modelo USApHMP Borroso siguiente:

#### Formulación USApHMP Borroso

$$(Z^* \mid \lambda^*) \cong \min \sum_{i \in N} \sum_{k \in N} (C_{ik} \mid \mu_{\mathcal{C}^{\circ}}(C_{ik})) H_{ik} \left[ \chi(O_i \mid \mu_{\mathcal{O}^{\circ}}(O_i)) + \delta(D_i \mid \mu_{\mathcal{D}^{\circ}}(D_i)) \right] +$$

$$\sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \alpha(C_{kl} \mid \mu_{\mathcal{C}^{\circ}}(C_{kl})) (Y_{kl}^i \mid \mu_{\mathcal{Y}^{\circ}}(Y_{kl}^i)) \quad (1)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p, \quad (2)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (4)$$

$$\sum_{l \in N} (Y_{kl}^i \mid \mu_{\mathcal{Y}^{\circ}}(Y_{kl}^i)) - \sum_{l \in N} (Y_{lk}^i \mid \mu_{\mathcal{Y}^{\circ}}(Y_{lk}^i)) \cong$$

$$(O_i \mid \mu_{\mathcal{O}^{\circ}}(O_i)) H_{ik} - \sum_{j \in N} (W_{ij} \mid \mu_{\mathcal{W}^{\circ}}(W_{ij})) H_{jk} \quad \forall i, k \in N, \quad (5)$$

$$H_{ik} \in \{0,1\} \quad Y_{kl}^i \geq 0 \quad \forall i, k, l \in N \quad (6)$$

El modelo se resuelve en dos pasos. En el primer paso, se calcula el USApHMP crisp utilizando el modelo de arriba sin los grados de certeza:

$$Z^* = \min \sum_{i \in N} \sum_{k \in N} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \alpha C_{kl} Y_{kl}^i \quad (1)$$

$$\text{s.a.} \quad \sum_{k \in N} H_{kk} = p, \quad (2)$$

$$\sum_{k \in N} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i, k \in N, \quad (4)$$

$$\sum_{l \in N} Y_{kl}^i - \sum_{j \in N} Y_{lk}^i = O_i H_{ik} - \sum_{j \in N} W_{ij} H_{jk} \quad \forall i, k \in N, \quad (5)$$

$$H_{ik} \in \{0,1\} \quad Y_{kl}^i \geq 0 \quad \forall i, k, l \in N \quad (6)$$

Al final del primer paso, nos encontraremos con un grafo en que se conoce el conjunto de  $p$  centros, las asignaciones de los demas  $n-p$  nodos no centros a los  $p$  nodos centros. Sabiendo que la solución del paso 1 se basa sobre datos aproximativos cuyos grados de certeza  $\mu_{\mathcal{C}^o}(C_{ik}), \mu_{\mathcal{C}^o}(C_{kl}), \mu_{\mathcal{O}^o}(O_i), \mu_{\mathcal{D}^o}(D_i)$  y  $\mu_{\mathcal{Y}^o}(Y_{kl}^i)$  disponemos, los utilizaremos para calcular el grado de cumplimiento de dicha solución usando la fórmula y restricciones siguientes:

*Grado de cumplimiento USApHMP*

$$\lambda^* = \frac{\prod_{i=1}^{n-p} \prod_{k=1}^p \left( \mu_{\mathcal{C}^o}(C_{ik}) H_{ik} \wedge \frac{1}{2} (\mu_{\mathcal{O}^o}(O_i) + \mu_{\mathcal{D}^o}(D_i)) \right) + \prod_{i=1}^{n-p} \prod_{k=1}^p \prod_{l=1}^p \left( \mu_{\mathcal{C}^o}(C_{kl}) \wedge \mu_{\mathcal{Y}^o}(Y_{kl}^i) \right)}{2} \quad (1)$$

$$\text{s.a.} \quad \sum_{k=1}^p H_{ik} = 1 \quad \forall i \in N \quad (2)$$

$$\mu_{\mathcal{C}^o}(C_{ik}), \mu_{\mathcal{O}^o}(O_i), \mu_{\mathcal{D}^o}(D_i), \mu_{\mathcal{C}^o}(C_{kl}), \mu_{\mathcal{Y}^o}(Y_{kl}^i) \in [0,1] \quad \forall i, k, l \in N \quad (3)$$

Obviamente, este modelo del USApHMP borroso generaliza el USApHMP crisp. En efecto si los grados de certeza  $\mu_{\mathcal{C}^o}(C_{ik}), \mu_{\mathcal{C}^o}(C_{kl}), \mu_{\mathcal{O}^o}(O_i), \mu_{\mathcal{D}^o}(D_i), \mu_{\mathcal{Y}^o}(Y_{kl}^i) \in \{0,1\} \quad \forall i, k, l \in N$ , entonces  $\lambda^* = 1$ , así pues, se trata de un USApHMP crisp. Pero si los grados de certeza

$\mu_{\vartheta^0}(C_{ik}), \mu_{\vartheta^0}(C_{kl}), \mu_{\vartheta^0}(O_i), \mu_{\vartheta^0}(D_i), \mu_{\vartheta^0}(Y_{kl}^i) \in [0,1], \forall i, k, l \in N$ , entonces  $\lambda^* \in [0,1]$ . En este caso se trata de un USApHMP disufo.

✓ **Algoritmo de cálculo del grado de cumplimiento USApHMP**

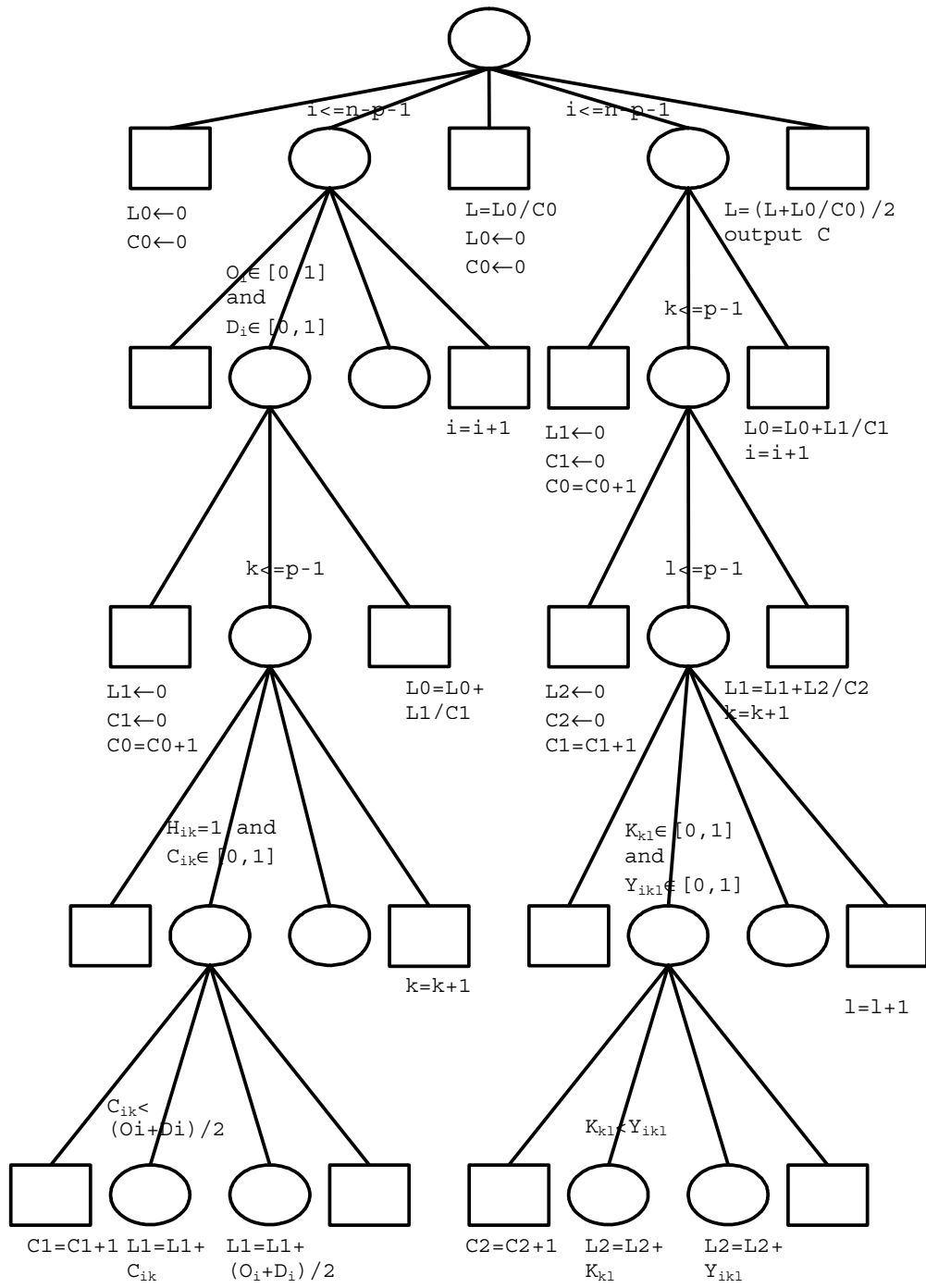
En la entrada, tenemos los datos siguientes:

- El vector  $D_{(n-p)} = [\mu_{\vartheta^0}(D_i)]_{(n-p)}$
- El vector  $O_{(n-p)} = [\mu_{\vartheta^0}(O_i)]_{(n-p)}$
- La matriz de las asignaciones  $H_{(n-p) \times p} = [H_{ik}]_{(n-p) \times p}$
- La matriz  $C_{(n-p) \times p} = [\mu_{\vartheta^0}(C_{ik})]_{(n-p) \times p}$
- La matriz  $K_{p \times p} = [\mu_{\vartheta^0}(C_{kl})]_{p \times p}$
- La matriz  $Y_{(n-p) \times p \times p} = [\mu_{\vartheta^0}(Y_{kl}^i)]_{(n-p) \times p \times p}$

Además de los datos de entrada, utilizamos las variables siguientes:

- $L$  Grado de cumplimiento de la solución
- $L2, L1$  y  $L0$  resultados intermedios calculados en los diferentes bucles del algoritmo, de los bucles más interiores a los bucle más exteriores, respectivamente
- $C2, C1$  y  $C0$  contadores que sirven como denominadores en los cálculos en los diferentes bucles del algoritmo, desde los bucles más interiores a los bucles más exteriores, respectivamente
- $i, j, k$  y  $l$  índices que permiten el acceso a los datos de entrada
- $n$  número total los nodos de la red
- $p$  número de nodos centros

Presentamos gráficamente la estructura de las operaciones para calcular el grado de cumplimiento de la solución obtenida usando el USApHMP:



Arbol programático de cálculo del Grado de certeza USApHMP

✓ Pseudocódigo : Grado de cumplimiento USApHMP

```

L0 ← 0
C0 ← 0
for i = 0 to n - p - 1
    if  $O_i \in [0,1]$  and  $D_i \in [0,1]$ 
        C1 ← 0
        L1 ← 0
        C0 = C0 + 1
        for k = 0 to p - 1
            if  $H_{ik} = 1$  and  $C_{ik} \in [0,1]$ 
                C1 = C1 + 1
                if  $C_{ik} < \frac{1}{2}(O_i + D_i)$ 
                    L1 = L1 +  $C_{ik}$ 
                else
                    L1 = L1 +  $\frac{1}{2}(O_i + D_i)$ 
            k = k + 1
        L0 = L0 +  $\frac{L1}{C1}$ 
    i = i + 1
L =  $\frac{L0}{C0}$ 
L0 ← 0
C0 ← 0
for i = 0 to n - p - 1
    C1 ← 0
    L0 ← 0
    C0 = C0 + 1
    for k = 0 to p - 1
        C2 ← 0
        L2 ← 0
        C1 = C1 + 1
        for l = 0 to p - 1
            if  $K_{kl} \in [0,1]$  and  $Y_{ikl} \in [0,1]$ 
                C2 = C2 + 1
                if  $K_{kl} < Y_{ikl}$ 
                    L2 = L2 +  $C_{kl}$ 
                else
                    L2 = L2 +  $Y_{ikl}$ 
            l = l + 1

```

$$L1 = L1 + \frac{L2}{C2}$$

$$k = k + 1$$

$$L0 = L0 + \frac{L1}{C1}$$

$$i = i + 1$$

$$L = \frac{1}{2} \left( L + \frac{L0}{C0} \right)$$

output  $\lambda$

Tiempo computacional =  $np(p+1)$

### ✓ Implementación del algoritmo en C++

```
//*****//
//Fuzzy USApHMP Degree of certainty
//Author: David Kutangila Mayoya
//Date: Sunday, November 21st, 2004
//Department of Computer Sciences and
//Artificial Intelligence
//University of Granada (Spain)
//*****//
#include <iostream.h>
main()
{
    int n,p,C0=0,C1,C2;
    double L0=0,L1,L2;
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of Hubs: ";
    cin >> p;

    //Declaring input data

    bool H[n-p][p];
    double D[n-p],O[n-p],C[n-p][p],Y[n-p][p][p];
```



```

//Entering input data

//will be written later

//Processing input data

for (int i=0; i<=n-p-1;){
    if (O[i]>=0&&O[i]<=1&&D[i]>=0&&D[i]<=1){
        C1=0;
        L1=0;
        ++C0;
        for (int k=0; k<=p-1;){
            if (H[i][k]&&C[i][k]>=0&&C[i][k]<=1){
                ++C1;
                if (C[i][k]<(O[i]+D[i])/2)
                    L1 += C[i][k];
                else
                    L1 += (O[i]+D[i])/2;
            }
            ++k;
        }
        L0 += L1/C1;
    }
    ++i;
}

double L = L0/C0;
L0=0;
C0=0;
for (int i=0; i<=n-p-1;){
    C1=0;
    L1=0;
    ++C0;

```

```

for (int k=0; k<=p-1;){
    C2=0;
    L2=0;
    ++C1;
    for (int l=0; l<=p-1;){
        if (K[k][l]>=0 && K[k][l]<=1 &&
            Y[i][k][l]>=0 && Y[i][k][l]<=1){
            ++C2;
            if (K[k][l]< Y[i][k][l])
                L2 += K[k][l];
            else
                L2 += Y[i][k][l];
        }
        ++l;
    }
    L1 += L2/C2;
    ++k;
}
L0 += L1/C1;
++i;
}
L = (L + L0/C0)/2;
cout << "\nDegree of certainty USApHMP: " << L;
return 0;
}

```

### III.2.1.2.2. Problema pHAP borroso

#### ✓ Formulación del modelo

Para el pHAP borroso, partiremos de la formulación crisp siguiente desarrollada en [31]:

*Formulación pHAP – N*

$$\min \sum_{i \in N} \sum_{k \in h} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{k \in h} \sum_{l \in h} \alpha C_{kl} E_{kl} \quad (1)$$

$$\text{s.a.} \quad H_{kk} = 1 \text{ o } 3 \quad \forall k \in h, \quad |h| = 2, \quad (2)$$

$$\sum_{k \in h} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i \in \{N : i \neq k\}, \quad k \in h, \quad (4)$$

$$t_{ik} \geq \sum_{j \in \{N : j < i\}} V_{ij} H_{ik} - \sum_{j \in \{N : j < i\}} (V_{ij} H_{jk}) \quad \forall i \in \{N : i \neq 1\}, \quad \forall k \in h, \quad (5)$$

$$\sum_{l \in h} E_{kl} = \sum_{i \in \{N : i \neq 1\}} t_{ik} \quad \forall k \in h, \quad (6)$$

$$\sum_{k \in h} E_{kl} \leq \sum_{j \in N} \left[ \left( \sum_{i \in \{N : i > j\}} V_{ij} \right) H_{jl} \right] \quad \forall l \in h, \quad (7)$$

$$\sum_{k \in h} E_{kl} - \sum_{l \in h} E_{lk} = \sum_{i \in N} \left[ \sum_{j \in \{N : j < i\}} V_{ij} - \sum_{j \in \{N : j < i\}} V_{ji} \right] H_{ik} \quad \forall k \in h, \quad (8)$$

$$H_{ik} \in \{0, 1\}, \quad E_{kl}, t_{ik} \geq 0, \quad \forall i \in N, \quad \forall k, l \in h. \quad (9)$$

Suponemos la incertidumbre en los mismos datos que en el USApHMP borroso, salvo que en vez de la variable  $Y_{kl}^i$  utilizamos  $E_{kl}$  que es el flujo del nodo entre los centros  $k$  y  $l$ . Siendo impreciso  $E_{kl}$  pertenecerá el subconjunto borroso  $\tilde{E}_{kl} = \{(E_{kl} | \mu_{\tilde{E}_{kl}}(E_{kl}))\}$ ,  $\forall k, l \in N$ , de los flujos  $E_{kl}$  más ciertos entre los centros  $k$  y  $l$ , donde  $\mu_{\tilde{E}_{kl}}(E_{kl})$  es el grado de pertenencia del flujo  $E_{kl}$  al subconjunto borroso  $\tilde{E}_{kl}$ , o mejor dicho, su grado de cumplimiento.

Así, desarrollamos el modelo pHAP Borroso siguiente:

*pHAP Borroso*

$$(Z^* | \lambda^*) \cong \min \sum_{i \in N} \sum_{k \in h} (C_{ik} | \mu_{\phi^0}(C_{ik})) H_{ik} \left[ \chi(O_i | \mu_{\phi^0}(O_i)) + \delta(D_i | \mu_{\beta^0}(D_i)) \right] +$$

$$\sum_{k \in h} \sum_{l \in h} \alpha(C_{kl} | \mu_{\phi^0}(C_{kl})) (E_{kl} | \mu_{\beta^0}(E_{kl})) \quad (1)$$

$$\text{s.a.} \quad H_{kk} = 1 \quad \forall k \in h, \quad |h| = 2 \text{ o } 3 \quad (2)$$

$$\sum_{k \in h} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i \in \{N : i \neq k\}, \quad k \in h, \quad (4)$$

$$t_{ik} \stackrel{\phi^0}{\geq} \sum_{j \in \{N : j < i\}} V_{ij} H_{ik} - \sum_{j \in \{N : j < i\}} (V_{ij} H_{jk}) \quad \forall i \in \{N : i \neq 1\}, \quad \forall k \in h, \quad (5)$$

$$\sum_{l \in h} E_{kl} \cong \sum_{i \in \{N : i \neq 1\}} t_{ik} \quad \forall k \in h, \quad (6)$$

$$\sum_{k \in h} E_{kl} \stackrel{\phi^0}{\geq} \sum_{j \in N} \left[ \left( \sum_{i \in \{N : i > j\}} V_{ij} \right) H_{jl} \right] \quad \forall l \in h, \quad (7)$$

$$\sum_{k \in h} E_{kl} - \sum_{l \in h} E_{lk} \cong \sum_{i \in N} \left[ \sum_{j \in \{N : j < i\}} V_{ij} - \sum_{j \in \{N : j < i\}} V_{ji} \right] H_{ik} \quad \forall k \in h, \quad (8)$$

$$H_{ik} \in \{0, 1\}, \quad E_{kl}, t_{ik} \geq 0, \quad \forall i \in N, \quad \forall k, l \in h. \quad (9)$$

El modelo se resuelve en dos pasos. En el primer paso, se calcula  $Z^*$  usando el modelo de arriba sin los grados de certeza:

$$Z^* = \min \sum_{i \in N} \sum_{k \in h} C_{ik} H_{ik} (\chi O_i + \delta D_i) + \sum_{k \in h} \sum_{l \in h} \alpha C_{kl} E_{kl} \quad (1)$$

$$\text{s.a.} \quad H_{kk} = 1 \quad \forall k \in h, \quad |h| = 2 \text{ o } 3 \quad (2)$$

$$\sum_{k \in h} H_{ik} = 1 \quad \forall i \in N, \quad (3)$$

$$H_{ik} \leq H_{kk} \quad \forall i \in \{N : i \neq k\}, \quad k \in h, \quad (4)$$

$$t_{ik} \geq \sum_{j \in \{N : j < i\}} V_{ij} H_{ik} - \sum_{j \in \{N : j < i\}} (V_{ij} H_{jk}) \quad \forall i \in \{N : i \neq 1\}, \quad \forall k \in h, \quad (5)$$

$$\sum_{l \in h} E_{kl} = \sum_{i \in \{N : i \neq 1\}} t_{ik} \quad \forall k \in h, \quad (6)$$

$$\sum_{k \in h} E_{kl} \leq \sum_{j \in N} \left[ \left( \sum_{i \in \{N: i > j\}} V_{ij} \right) H_{jl} \right] \quad \forall l \in h, \quad (7)$$

$$\sum_{k \in h} E_{kl} - \sum_{l \in h} E_{lk} = \sum_{i \in N} \left[ \sum_{j \in \{N: j < i\}} V_{ij} - \sum_{j \in \{N: j < i\}} V_{ji} \right] H_{ik} \quad \forall k \in h, \quad (8)$$

$$H_{ik} \in \{0,1\}, \quad E_{kl}, t_{ik} \geq 0, \quad \forall i \in N, \quad \forall k, l \in h. \quad (9)$$

Al cabo del primer paso, nos encontraremos con un grafo en que se conoce el conjunto de  $p$  centros, las asignaciones de los demas  $n-p$  nodos no centros a los  $p$  nodos centros. Sabiendo que la solución del paso 1 se basa sobre datos aproximativos cuyos grados de certeza  $\mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{O}^0}(O_i), \mu_{\mathcal{D}^0}(D_i)$  y  $\mu_E(E_{kl})$  disponemos, los utilizaremos para calcular el grado de cumplimiento de dicha solución usando la fórmula y restricciones siguientes:

*Grado de cumplimiento pHAP*

$$\lambda^* = \frac{\prod_{i=1}^{n-p} \prod_{k=1}^p \left( \mu_{\mathcal{C}^0}(C_{ik}) H_{ik} \wedge \frac{1}{2} (\mu_{\mathcal{O}^0}(O_i) + \mu_{\mathcal{D}^0}(D_i)) \right) + \prod_{k=1}^p \prod_{l=1}^p (\mu_{\mathcal{C}^0}(C_{kl}) \wedge \mu_{\mathcal{E}^0}(E_{kl}))}{2} \quad (1)$$

$$\text{s.a.} \quad \sum_{k=1}^p H_{ik} = 1 \quad \forall i \in N \quad (2)$$

$$\mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{O}^0}(O_i), \mu_{\mathcal{D}^0}(D_i), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{E}^0}(E_{kl}) \in [0,1], \quad \forall i \in N, \quad \forall k, l \in h. \quad (3)$$

Obviamente, este modelo del pHAP borroso generaliza el pHAP crisp. En efecto si los grados de certeza  $\mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{O}^0}(O_i), \mu_{\mathcal{D}^0}(D_i), \mu_{\mathcal{E}^0}(E_{kl}) \in \{0,1\} \quad \forall i, k, l \in N$ , entonces  $\lambda^* = 1$ , así pues, se trata de un pHAP crisp. Pero si los grados de certeza  $\mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{O}^0}(O_i), \mu_{\mathcal{D}^0}(D_i), \mu_{\mathcal{E}^0}(E_{kl}) \in [0,1], \quad \forall i, k, l \in N$ , entonces  $\lambda^* \in [0,1]$ . En este caso se trata de un pHAP borroso.

✓ **Algoritmo de cálculo del grado de cumplimiento pHAP**

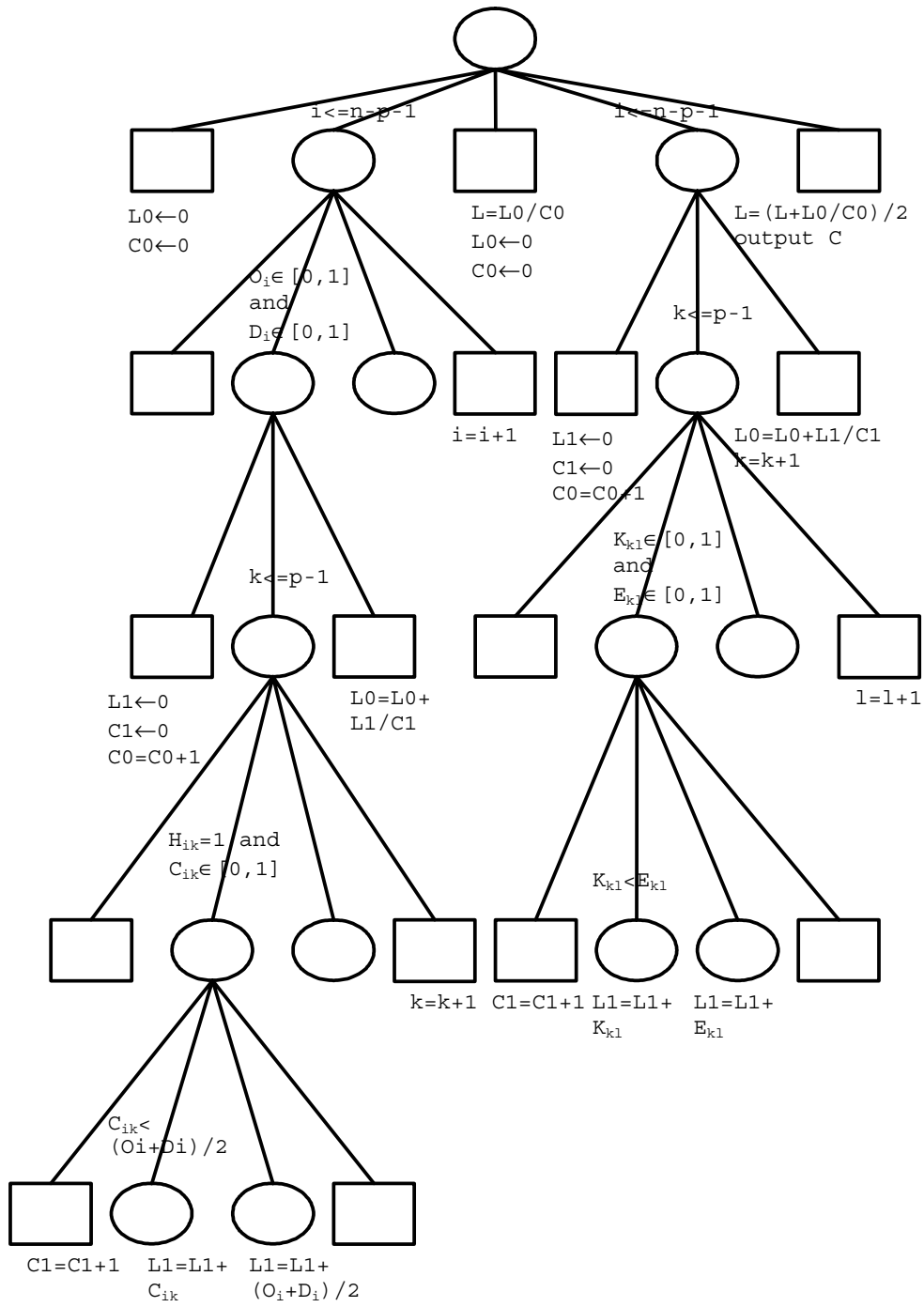
En la entrada, tenemos los datos siguientes:

- El vector  $D_{(n-p)} = [\mu_{\beta^0}(D_i)]_{(n-p)}$
- El vector  $O_{(n-p)} = [\mu_{\theta^0}(O_i)]_{(n-p)}$
- La matriz de las asignaciones  $H_{(n-p) \times p} = [H_{ik}]_{(n-p) \times p}$
- La matriz  $C_{(n-p) \times p} = [\mu_{\theta^0}(C_{ik})]_{(n-p) \times p}$
- La matriz  $K_{p \times p} = [\mu_{\theta^0}(C_{kl})]_{p \times p}$
- La matriz  $E_{p \times p} = [\mu_{\theta^0}(E_{kl})]_{p \times p}$

Además de los datos de entrada, utilizamos las variables siguientes:

- $L$  Grado de cumplimiento de la solución
- $L1$  y  $L0$  resultados intermedios calculados en los diferentes bucles del algoritmo, de los bucles más interiores a los bucle más exteriores, respectivamente
- $C1$  y  $C0$  contadores que sirven como denominadores en los cálculos en los diferentes bucles del algoritmo, desde los bucles más interiores a los bucles más exteriores, respectivamente
- $i, j, k$  y  $l$  índices que permiten el acceso a los datos de entrada
- $n$  número total los nodos de la red
- $p$  número de nodos centros

Construimos el algoritmo mediante el árbol programático siguiente:



Arbol programático de cálculo del Grado de certeza pHAP

✓ **Pseudocódigo: grado de cumplimiento pHAP**

$L0 \leftarrow 0$

$C0 \leftarrow 0$

for  $i = 0$  to  $n - p - 1$

if  $O_i \in [0,1]$  and  $D_i \in [0,1]$

$C1 \leftarrow 0$

$L1 \leftarrow 0$

$C0 = C0 + 1$

for  $k = 0$  to  $p - 1$

if  $H_{ik} = 1$  and  $C_{kl} \in [0,1]$

$C1 = C1 + 1$

if  $C_{ik} < \frac{1}{2}(O_i + D_i)$

$L1 = L1 + C_{ik}$

else

$L1 = L1 + \frac{1}{2}(O_i + D_i)$

$k = k + 1$

$L0 = L0 + \frac{L1}{C1}$

$i = i + 1$

$L \leftarrow \frac{L0}{C0}$

$L0 \leftarrow 0$

$C0 \leftarrow 0$

for  $k = 0$  to  $p - 1$

$C1 \leftarrow 0$

$L1 \leftarrow 0$

$C0 = C0 + 1$

for  $l = 0$  to  $p - 1$

if  $K_{kl} \in [0,1]$  and  $E_{kl} \in [0,1]$



```

    C1 = C1 + 1
    if  $K_{kl} < E_{kl}$ 
        L1 = L1 +  $K_{kl}$ 
    else
        L1 = L1 +  $E_{kl}$ 

    l = l + 1
    L0 = L0 +  $\frac{L1}{C1}$ 
    k = k + 1
L =  $\frac{1}{2} \left( L + \frac{L0}{C0} \right)$ 
output  $\lambda^*$ 

```

Tiempo Computacional:  $2np$

### ✓ Implementación del algoritmo en C++

```

//*****//
//Fuzzy pHAP: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Sunday, November 21st, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada (Spain)
//*****//
#include <iostream.h>
main()
{
    int n,p,C0=0,C1;
    double L0=0,L1
    cout << "p-Hub Allocation: Degree of certainty\n";
    cout << "\nNumber of nodes: ";
    cin >> n;

```

```

cout << "\nNumber of Hubs: ";
cin >> p;

//Declaring input data
int H[n-p][p];
double D[n-p], O[n-p], C[n-p][p],K[p][p],E[p][p];
//Entering input data
    // Will be written later
//Processing input data
for (int i=0; i <= n-p-1;){
    if (O[i]>=0 && O[i]<=1 && D[i]>=0 && D[i]<=1){
        C1=0;
        L1=0;
        ++C0;
        for (int k=0; k<=p-1;){
            if (H[i][k]&&C[i][k]>=0&&C[i][k]<=1){
                ++C1;
                if (C[i][k]<(O[i]+D[i])/2)
                    L1 += C[i][k];
                else
                    L1 += (O[i]+D[i])/2;
            }
            ++k;
        }
        L0 += L1/C1;
    }
    ++i;
}
double L = L0/C0;
L0=0;
C0=0;
for (int k=0; k<=p-1;){
    C1=0;

```

```

L1=0;
++C0;
for (int l=0; l<=p-1;){
    if (K[k][l]>=0 && K[k][l]<=1 &&
        E[k][l]>=0 && E[k][l]<=1){
        ++C1;
        if (K[k][l]<E[k][l])
            L1 += K[k][l];
        else
            L1 += E[k][l];
    }
    ++l;
}
L0 += L1/C1;
++k;
}
L = (L + L0/C0)/2;
cout << "\nDegree of certainty pHAP: " << L;
return 0;
}

```

### III.2.2. Problema USApHLP

#### III.2.2.1. Aspecto crisp del problema USApHLP

En el problema de localización de  $p$ -centro con asignación única, se revisa la formulación de Skorin-Kapov [94]. Cada nodo no centro tiene que conectarse a exactamente uno de los  $p$  centros. Usando la simetría del coste de flujo por unidad (el coste del flujo del nodo  $i$  al nodo  $j$  es el mismo que el coste del flujo del nodo  $j$  al nodo  $i$ ), se puede reducir el número de variables y restricciones de la formulación por más de la mitad. [96] proporciona una formulación mezclada entera para el modelo con localizaciones fijas de centros, donde se consideran los costes fijos para abrir los enlaces. En los datos de Skorin-Kapov [94], el coste del flujo por unidad es simétrico, es decir,  $c_{ij} = c_{ji}$ . Cuando el coste del flujo por unidad se proporcione a la distancia, entonces  $C_{ijkn} = C_{jimk}$  y se puede reducir el número de variables y restricciones en la formulación de Skorin-Kapov [94] por más de la mitad de la manera siguiente. Sea  $v_{ij}$  (para  $i < j$ ) el flujo total bi-direccional entre los nodos  $i$  y  $j$ , así  $v_{ij} = W_{ij} + W_{ji}$  si  $i < j$ . Entonces, la formulación puede reducirse a la siguiente [96]:

( $p$ -SA)

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^n \sum_{m=1}^n v_{ij} C_{ijkn} x_{ijkn}$$

$$\text{s.a.} \quad \sum_{k=1}^n z_{kk} = p, \quad (1)$$

$$\sum_{k=1}^n z_{ik} = 1, \quad i = 1, \dots, n, \quad (2)$$

$$z_{ik} \leq z_{kk}, \quad i, k = 1, \dots, n, \quad i \neq k, \quad (3)$$

$$\sum_{m=1}^n x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k = 1, \dots, n \quad (4)$$

$$\sum_{k=1}^n x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m = 1, \dots, n, \quad (5)$$

$$z_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, n,$$

$$x_{ijkm} \geq 0, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k, m = 1, \dots, n.$$

donde la variable  $z_{ik} = 1$  si el nodo  $i$  se asigna al centro  $k$ , y 0 en el caso contrario. Si un nodo  $k$  se escoge como centro, se denota por  $z_{kk} = 1$ . Las restricciones (4) y (5) refuerzan el requerimiento de una asignación única. Esta formulación ( $p$ -SA) tiene  $n^2 0/1$  variables,  $n^3 (n-1)/2$  variables continuas, y  $1+n^3$  restricciones.

A pesar de que se ha reducido el número de variables y restricciones, el tamaño de la formulación es todavía muy largo para resolver grandes problemas aún si el valor de  $p$  es pequeño. Para una formulación de este problema, se considera una modificación del modelo ( $p$ -SA) como sigue [96]. Sea  $H$  el conjunto de centros fijos con  $|H| = p$  y  $f_{ik}$  el coste fijo para abrir una conexión entre el nodo  $i$  y el centro  $k$ .

( $p$ -FH)

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k \in H} \sum_{m \in H} v_{ij} C_{ijkm} x_{ijkm} + \sum_{i=1, i \notin H}^n \sum_{k \in H} f_{ik} z_{ik}$$

$$\text{s.a.} \quad z_{kk} = 1 \quad k \in H, \quad (1)$$

$$\sum_{k \in H} z_{ik} = 1, \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{m \in H} x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k \in H, \quad (3)$$

$$\sum_{k \in H} x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m \in H, \quad (4)$$

$$z_{ik} \in \{0, 1\}, \quad i = 1, \dots, n, \quad k \in H$$

$$x_{ijkm} \geq 0, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k, m \in H.$$

La formulación ( $p$ -FH) tiene  $pn 0/1$  variables,  $p^2 n(n-1)/2$  variables continuas, y  $p + (1-p)n + pn^2$  restricciones.

Con el propósito de tomar en cuenta los efectos de la congestión, en [34] los autores consideran el problema de localización de  $p$ -centro no capacitado con asignación única, en el que se refieren como *USApHLP*, que se formula de la siguiente manera:

[USApHLP]:

$$\min \sum_i \sum_{j>i} \sum_k \sum_m F_{ijkm} X_{ijkm} \quad (1)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = 1 \quad \forall i, \quad (2)$$

$$Z_{ik} \leq Z_{kk} \quad \forall i, k, \quad (3)$$

$$\sum_k Z_{kk} = p, \quad (4)$$

$$\sum_m X_{ijkm} = Z_{ik} \quad \forall k, i, j > i, \quad (5)$$

$$\sum_k X_{ijkm} = Z_{jm} \quad \forall m, i, j > i, \quad (6)$$

$$X_{ijkm}, Z_{ik} \in \{0,1\} \quad \forall i, j > i, k, m, \quad (7)$$

donde  $k$  y  $m$  son índices para las localizaciones de los centros potenciales,  $i$  y  $j$  son índices para los nodos de origen y destino,  $p$  es el número de los centros a localizar, y  $F_{ijkm}$  es el coste total de enviar el flujo  $(i, j)$  a través del camino  $(i, k, m, j)$ . Este coste se da por  $W_{ij}(C_{ik} + \alpha C_{km} + C_{mj})$  donde  $C_{ij}$  es el coste unitario de viaje del nodo  $i$  al nodo  $j$ ,  $W_{ij}$  es el flujo de  $i$  a  $j$  y  $\alpha$  es el descuento por unidad entre los centros. Las variables de decisión  $Z_{ik}$ ,  $Z_{kk}$  y  $X_{ijkm}$  son variables de decisión binarias que toman el valor 1 si el nodo  $i$  se asigna al centro  $k$ , el nodo  $k$  está seleccionado como un centro, y el flujo  $(i, j)$  está enviado de  $i$  a  $k$  a  $m$  a  $j$ , respectivamente.

Las restricciones (2) aseguran que cada nodo está asignado a exactamente un centro. Las restricciones (3) garantizan que un nodo no se asignará a un nodo en que no se ha abierto un centro. La restricción (4) requiere que se abran  $p$  centros. Las restricciones (5) y (6) aseguran que para cualquier flujo  $(i, j)$  que utiliza la conexión intercentro  $(k, m)$ , los nodos  $i$  y  $j$  tienen que estar asignados a los centros  $k$  y  $m$ , respectivamente. La función objetivo (1) minimiza el coste total de enviar flujos a través de uno o dos centros. Esta función de coste no explica una congestión posible que resulta de la consolidación de largos montantes de flujo en los centros seleccionados. Este ocurrirá típicamente cuando los costes intercentros reducidos inducirán al

modelo concentrar largos montantes de flujo en ciertos centros, causando que estos estén sobre-utilizados.

Según [34], la congestión ocurre en la mayoría de los aeropuertos (centros), donde numerosos vuelos convergen a las horas de punta. Cuando muchos flujos se dirigen a un centro, la congestión aumenta y causa un aumento dramático en los costes. Esta relación puede explicarse por una función de coste de la forma:

$$f(u) = au^b,$$

donde  $u$  es el flujo en un centro,  $a$  y  $b$  son constantes positivas con  $b \geq 1$ .

Sirviéndose de la notación en *USApHLP*, el flujo a través del centro  $k$  es  $\sum_i \sum_{j>i} \sum_m W_{ij} X_{ijkm}$ . Por eso, el coste de la congestión en el centro  $k$  es

$$a \left( \sum_i \sum_{j>i} \sum_m W_{ij} X_{ijkm} \right)^b = a \left( \sum_i \sum_{j>i} W_{ij} Z_{ik} \right)^b,$$

donde la última igualdad se obtiene usando la restricción (5)

$$\sum_i \sum_{j>i} \sum_m W_{ij} X_{ijkm} = \sum_i \sum_{j>i} W_{ij} \sum_m X_{ijkm} = \sum_i \sum_{j>i} W_{ij} Z_{ik} \quad \forall k.$$

Incorporando la función de coste de la congestión en la función objetivo de *USApHLP*, [34] obtiene el problema de localización de centro no capacitado con asignación única con congestión cuya formulación sigue:

[*USApHLPC*]:

$$\min \sum_i \sum_{j>i} \sum_k \sum_m F_{ijkm} X_{ijkm} + \sum_k a \left( \sum_i \sum_{j>i} \sum_m W_{ij} X_{ijkm} \right)^b$$

$$\text{s.a.} \quad \sum_k Z_{ik} = 1 \quad \forall i, \quad (2)$$

$$Z_{ik} \leq Z_{kk} \quad \forall i, k, \quad (3)$$

$$\sum_k Z_{kk} = p, \quad (4)$$

$$\sum_m X_{ijkm} = Z_{ik} \quad \forall k, i, j > i, \quad (5)$$

$$\sum_k X_{ijkm} = Z_{jm} \quad \forall m, i, j > i, \quad (6)$$

$$X_{ijkm}, Z_{ik} \in \{0,1\} \quad \forall i, j > i, k, m, \quad (7)$$

En [57] B.Y. Kara y B.Ç. Tansel examinan los aspectos computacionales del problema de p-centro con asignación única sobre los bases de un modelo básico y un nuevo modelo. Presentan una formulación combinatoria del problema de p-centro para el caso de asignación única y demuestran su aspecto "NP-Hard". A pesar de que los problemas de localización de centro se definen de costumbre basándose sobre un grafo completo cuyos costes de aristas satisfacen la desigualdad triangular, se apartan de esta tradición y definen el problema sobre una red de transporte física que se supone conectada pero no necesariamente completa.

### III.2.2.2. Aspecto borroso del problema USApHLP

Desarrollaremos dos modelos para el problema de localización de p-centro no capacitado con asignación única difuso a los que nos referiremos como USApHLP borroso y USApHLPC borroso, partiendo de las formulaciones crisp desarrolladas en [96] y [34].

#### III.2.2.2.1. El USApHLP Borroso

##### ✓ Formulación del modelo

Para el USApHLP borroso, partimos de la siguiente formulación [96]:

$$(p\text{-FH})$$

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k \in H} \sum_{m \in H} v_{ij} C_{ijkm} x_{ijkm} + \sum_{i=1, i \notin H}^n \sum_{k \in H} f_{ik} z_{ik}$$

$$\text{s.a.} \quad z_{kk} = 1 \quad k \in H, \quad (1)$$

$$\sum_{k \in H} z_{ik} = 1, \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{m \in H} x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k \in H, \quad (3)$$



$$\sum_{k \in H} x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m \in H, \quad (4)$$

$$z_{ik} \in \{0, 1\}, \quad i = 1, \dots, n, \quad k \in H \quad (5)$$

$$x_{ijkm} \geq 0, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k, m \in H. \quad (6)$$

En esta formulación, suponemos el incertudumbre sobre los siguientes datos:

- $v_{ij} = W_{ij} + W_{ji}$  (para  $i < j$ ), el flujo total bi-direccional entre los nodos  $i$  y  $j$ ,
- $C_{ijkm}$ , el coste del flujo por unidad del nodo  $i$  a  $j$  vía los centros  $k$  y  $m$ . Cuando el coste del flujo por unidad se proporciona a la distancia, entonces  $C_{ijkm} = C_{jimk}$ .
- $f_{ik}$  el coste fijo para abrir una conexión entre el nodo  $i$  y el centro  $k$ .

Asumamos que estos datos pertenecen a los subconjuntos borrosos siguientes:

- $\mathcal{V}^{\circ} = \left\{ \left( v_{ij} \mid \mu_{\mathcal{V}^{\circ}}(v_{ij}) \right) \right\}, \forall i, j \in N$ , subconjunto borroso de los flujos  $v_{ij}$  ciertos, donde 
$$\mu_{\mathcal{V}^{\circ}}(v_{ij}) = \frac{\mu_{W^{\circ}}(W_{ij}) + \mu_{W^{\circ}}(W_{ji})}{2}, \forall i, j \in N$$
, es el grado de cumplimiento de  $v_{ij}$ .
- $\mathcal{C}^{\circ} = \left\{ \left( C_{ijkm} \mid \mu_{\mathcal{C}^{\circ}}(C_{ijkm}) \right) \right\}, \forall i, j \in N, \forall k, m \in H$ , subconjunto borroso de los costes  $C_{ijkm}$ , donde  $\mu_{\mathcal{C}^{\circ}}(C_{ijkm})$  es el grado de cumplimiento del coste  $C_{ijkm}$ ,
- $\mathcal{F}^{\circ} = \left\{ \left( f_{ik} \mid \mu_{\mathcal{F}^{\circ}}(f_{ik}) \right) \right\}, \forall i \in N, \forall k \in H$ , subconjunto borroso de los costes fijos  $f_{ik}$  ciertos, donde  $\mu_{\mathcal{F}^{\circ}}(f_{ik})$  es el grado de cumplimiento del coste fijo  $f_{ik}$ .

Estos subconjuntos borrosos nos permiten formular el problema borroso de localización de p-centro no capacitado con asignación única como sigue:

*USApHLP borroso:*

$$(Z^* | \lambda^*) \cong \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k \in H} \sum_{m \in H} (v_{ij} | \mu_{\beta_0}(v_{ij})) (C_{ijkm} | \mu_{\beta_0}(C_{ijkm})) x_{ijkm} + \sum_{i=1, i \notin H}^n \sum_{k \in H} (f_{ik} | \mu_{\beta_0}(f_{ik})) z_{ik} \quad (1)$$

$$\text{s.a.} \quad z_{kk} = 1 \quad k \in H, \quad (2)$$

$$\sum_{k \in H} z_{ik} = 1, \quad i = 1, \dots, n, \quad (3)$$

$$\sum_{m \in H} x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k \in H, \quad (4)$$

$$\sum_{k \in H} x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m \in H, \quad (5)$$

$$z_{ik} \in \{0, 1\}, \quad i = 1, \dots, n, \quad k \in H \quad (6)$$

$$x_{ijkm} \geq 0, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k, m \in H. \quad (7)$$

El modelo se resuelve en dos pasos. Primero se calcula  $Z^*$  usando el modelo de arriba sin los grados de certeza, es decir el modelo USApHLP crisp:

*USApHLP crisp:*

$$Z^* = \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k \in H} \sum_{m \in H} v_{ij} C_{ijkm} x_{ijkm} + \sum_{i=1, i \notin H}^n \sum_{k \in H} f_{ik} z_{ik} \quad (1)$$

$$\text{s.a.} \quad z_{kk} = 1 \quad k \in H, \quad (2)$$

$$\sum_{k \in H} z_{ik} = 1, \quad i = 1, \dots, n, \quad (3)$$

$$\sum_{m \in H} x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k \in H, \quad (4)$$

$$\sum_{k \in H} x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m \in H, \quad (5)$$

$$z_{ik} \in \{0, 1\}, \quad i = 1, \dots, n, \quad k \in H \quad (6)$$

$$x_{ijkm} \geq 0, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k, m \in H. \quad (7)$$

Segundo se calcula el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el paso 1 usando la fórmula y restricciones siguientes:

*Grado de cumplimiento USApHLP:*

$$\lambda^* = \frac{\prod_{i=1}^{n-p} \prod_{j=1}^{n-p} \prod_{k=1}^p \prod_{m=1}^p \left[ \left( \mu_{\mathcal{V}_0}(v_{ij}) \wedge \mu_{\mathcal{C}_0}(C_{ijkm}) \right) x_{ijkm} \right] + \prod_{i=1}^{n-p} \prod_{k=1}^p \left( \mu_{\mathcal{F}_0}(f_{ik}) \right) z_{ik}}{2} \quad (1)$$

$$\text{s.a.} \quad \sum_{k \in H} z_{ik} = 1, \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{m \in H} x_{ijkm} = z_{ik}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad k \in H, \quad (3)$$

$$\sum_{k \in H} x_{ijkm} = z_{jm}, \quad i = 1, \dots, n-1, \quad j = i+1, \dots, n, \quad m \in H, \quad (4)$$

$$\mu_{\mathcal{V}_0}(v_{ij}), \mu_{\mathcal{C}_0}(C_{ijkm}), \mu_{\mathcal{F}_0}(f_{ik}) \in [0, 1],$$

$$i = 1, \dots, n-p, \quad j = 1, \dots, n-p \quad k, m \in H. \quad (5)$$

Obviamente, este modelo USApHLP borroso generaliza el USApHLP crisp. En efecto si los grados de certeza  $\mu_{\mathcal{C}_0}(C_{ijkm}), \mu_{\mathcal{V}_0}(V_{ij}), \mu_{\mathcal{F}_0}(f_{ik}) \in \{0, 1\} \quad \forall i, j \in N$ , y  $\forall k, m \in H$ , entonces  $\lambda^* = 1$ , así pues, se trata de un USApHLP crisp. Pero si los grados de certeza  $\mu_{\mathcal{C}_0}(C_{ijkm}), \mu_{\mathcal{V}_0}(V_{ij}), \mu_{\mathcal{F}_0}(f_{ik}) \in [0, 1], \quad \forall i, j \in N$ , y  $\forall k, m \in H$ , entonces  $\lambda^* \in [0, 1]$ . En este caso se trata de un USApHLP borroso.

### ✓ Algoritmo de cálculo del grado de cumplimiento USApHLP

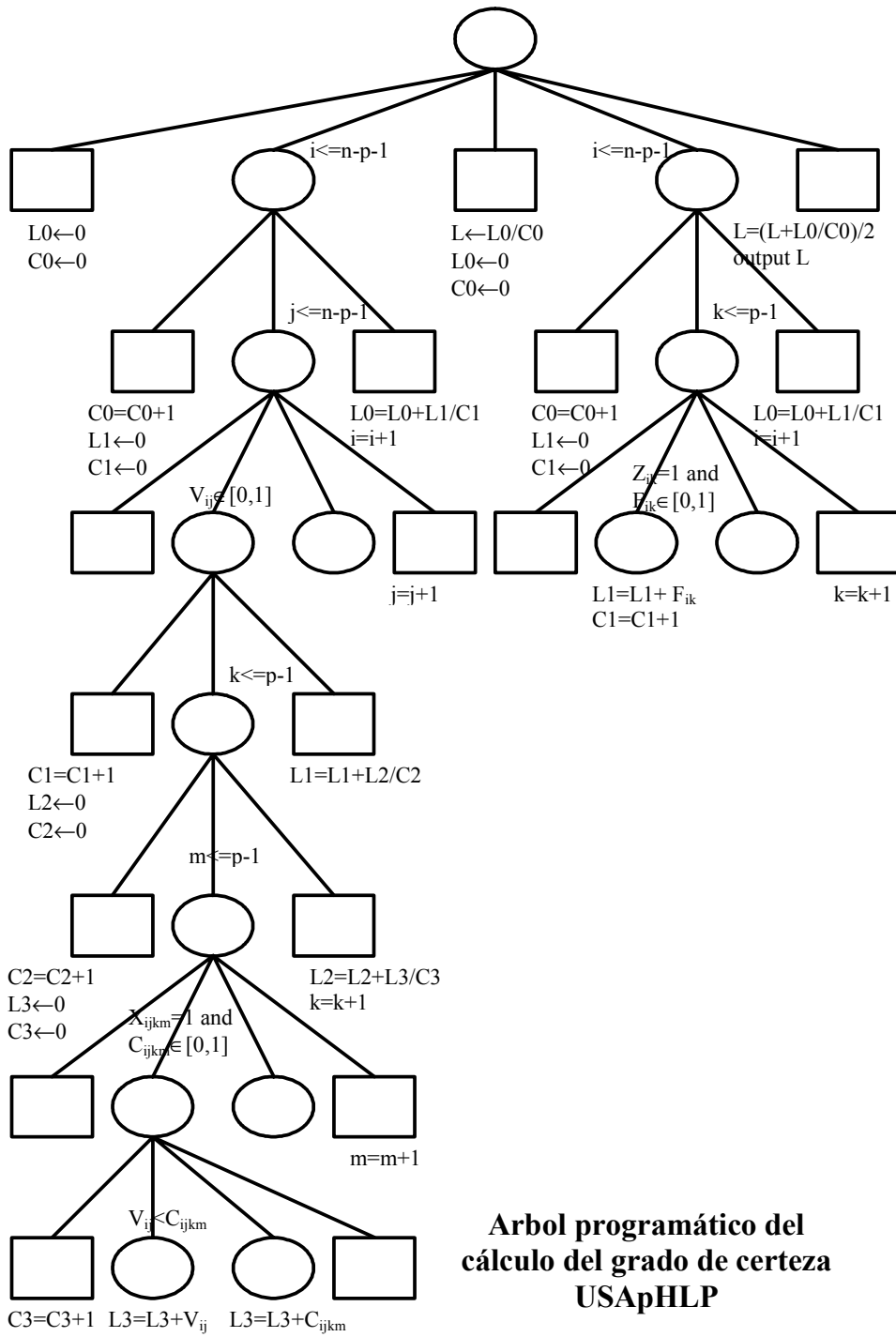
En la entrada, tenemos los datos siguientes:

- La matriz  $X_{(n-p) \times (n-p) \times p \times p} = [x_{ijklm}]_{(n-p) \times (n-p) \times p \times p}$
- La matriz  $Z_{(n-p) \times p} = [z_{ik}]_{(n-p) \times p}$
- La matriz  $C_{(n-p) \times (n-p) \times p \times p} = [C_{ijklm}]_{(n-p) \times (n-p) \times p \times p}$
- La matriz  $V_{(n-p) \times (n-p)} = [\mu_{\rho}(V_{ij})]_{(n-p) \times (n-p)}$
- La matriz  $F_{(n-p) \times p} = [\mu_{\rho}(f_{ik})]_{(n-p) \times p}$

Además de los datos de entrada, utilizamos las siguientes variables:

- $L$  Grado de cumplimiento de la solución
- $L3, L2, L1$  y  $L0$  resultados intermedios calculados en los diferentes bucles del algoritmo, de los bucles más interiores a los más exteriores, respectivamente.
- $C3, C2, C1$  y  $C0$  contadores que sirven como denominadores en los cálculos en los diferentes bucles del algoritmo, de los bucles más interiores a los bucles más exteriores, respectivamente
- $i, j, k, m$ , índices que permiten el acceso a las diferentes matrices en la memoria
- $n$  número total de los nodos de la red
- $p$  número de nodos centros

Construimos el algoritmo a partir del árbol siguiente:



Tiempo computacional:  $np(np+1)$  o  $O(n^2 p^2)$

✓ **Peudocódigo: Grado de cumplimiento USApHLP**

```

L0 ← 0
C0 ← 0
for i = 0 to n - p - 1
    C0 = C0 + 1
    L1 ← 0
    C1 ← 0
    for j = 0 to n - p - 1
        if  $V_{ij} \in [0,1]$ 
            C1 = C1 + 1
            L2 ← 0
            C2 ← 0
            for k = 0 to p - 1
                C2 = C2 + 1
                L3 ← 0
                C3 ← 0
                for m = 0 to p - 1
                    if  $X_{ijkm} = 1$  and  $C_{ijkm} \in [0,1]$ 
                        C3 = C3 + 1
                        if  $V_{ij} < C_{ijkm}$ 
                            L3 = L3 +  $V_{ij}$ 
                        else
                            L3 = L3 +  $C_{ijkm}$ 
                    m = m + 1
                L2 = L2 +  $\frac{L3}{C3}$ 
            k = k + 1
            L1 = L1 +  $\frac{L2}{C2}$ 
        j = j + 1
    L0 = L0 +  $\frac{L1}{C1}$ 
i = i + 1

```

```

L ←  $\frac{L0}{C0}$ 
L0 ← 0
C0 ← 0
for i = 0 to n - p - 1
    C0 = C0 + 1
    L1 ← 0
    C1 ← 0
    for k = 0 to p - 1
        if  $Z_{ik} = 1$  and  $F_{ik} \in [0, 1]$ 
            L1 = L1 +  $F_{ik}$ 
            C1 = C1 + 1
        k = k + 1
    L0 = L0 +  $\frac{L1}{C1}$ 
    i = i + 1
L =  $\frac{1}{2} \left( L + \frac{L0}{C0} \right)$ 
output L

```

✓ **Implementación del algoritmo en C++**

```

//*****//
//Fuzzy USApHLP: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Sunday, November 21st, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada (Spain)
//*****//
#include <iostream.h>
main()
{
    int n,p,C0=0,C1,C2,C3;
    double L0=0,L1,L2,L3;
    cout << "Number of nodes: ";

```

```

cin >> n;
cout << "\nNumber of Hubs: ";
cin >> p;
//Declaring input data
bool X[n-p][n-p][p][p], Z[n-p][p];
double C[n-p][n-p][p][p], V[n-p][n-p], F[n-p][p];
//Entering input data
    // will be written later
//Processing input data
for (int i=0; i<=n-p-1;){
    ++C0;
    L1=0;
    C1=0;
    for (int j=0; j<=n-p-1;){
        if (V[i][j]>=0 && V[i][j]<=1){
            ++C1;
            L2=0;
            C2=0;
            for (int k=0; k<=p-1;){
                ++C2;
                L3=0;
                C3=0;
                for (int m=0; m<=p-1;){
                    if (X[i][j][k][m] &&
                        C[i][j][k][m]>=0 &&
                        C[i][j][k][m]<=1){
                        ++C3;
                        if (V[i][j]<C[i][j][k][m])
                            L3+=V[i][j];
                        else
                            L3+=C[i][j][k][m];
                    }
                }
            }
        }
    }
}

```



```

        ++m;
    }
    L2 += L3/C3;
    ++k;
}
L1 += L2/C2;
}
++j;
}
L0 += L1/C1;
++i;
}
double L = L0/C0;
L0=0;
C0=0;
for (int i=0; i<=n-p-1;){
    ++C0;
    L1=0;
    C1=0;
    for (int k=0; k<=p-1;){
        if (Z[i][k] && F[i][k]>=0 && F[i][k]<=1){
            L1 += F[i][k];
            ++C1;
        }
        ++k;
    }
    L0 += L1/C1;
    ++i;
}
L = (L + L0/C0)/2;
cout << "\nDegree of certainty (USApHLP): " << L;
return 0;
}

```

### III.2.2.2.2. El USApHLPC Borroso

#### ✓ Formulación del modelo

Para el USApHLPC borroso, partiremos de la siguiente formulación [34]:

[USApHLPC]:

$$\min \sum_i \sum_{j>i} \sum_k \sum_m F_{ijkm} X_{ijkm} + \sum_k a \left( \sum_i \sum_{j>i} \sum_m W_{ij} X_{ijkm} \right)^b$$

s.a.  $\sum_k Z_{ik} = 1 \quad \forall i, \quad (2)$

$Z_{ik} \leq Z_{kk} \quad \forall i, k, \quad (3)$

$\sum_k Z_{kk} = p, \quad (4)$

$\sum_m X_{ijkm} = Z_{ik} \quad \forall k, i, j > i, \quad (5)$

$\sum_k X_{ijkm} = Z_{jm} \quad \forall m, i, j > i, \quad (6)$

$X_{ijkm}, Z_{ik} \in \{0,1\} \quad \forall i, j > i, k, m, \quad (7)$

En la función objetivo de esta formulación, supondremos la incertidumbre en los siguientes datos:

- $C_{ij}$ , el coste unitario de viaje del nodo  $i$  al nodo  $j$ . Así suponemos que  $\exists C \supset \mathcal{C}^b = \left\{ \left( C_{ij} \mid \mu_{\mathcal{C}^b}(C_{ij}) \right) \right\}, \forall i, j$ , el subconjunto borroso de los costes  $C_{ij}$  ciertos, donde  $\mu_{\mathcal{C}^b}(C_{ij})$  es el grado de cumplimiento del coste  $C_{ij}$ , o su grado de pertenencia al subconjunto borroso  $\mathcal{C}^b$ . Este coste no aparece en la formulación del modelo pero permite la construcción del coste  $F_{ijkm}$ .
- $W_{ij}$ , el flujo del nodo  $i$  al nodo  $j$ . Así suponemos que  $\exists W \supset \mathcal{W}^b = \left\{ \left( W_{ij} \mid \mu_{\mathcal{W}^b}(W_{ij}) \right) \right\}, \forall i, j$ , el subconjunto borroso de los flujos  $W_{ij}$  ciertos, donde  $\mu_{\mathcal{W}^b}(W_{ij})$  es el grado de cumplimiento del flujo  $W_{ij}$ , o su grado de pertenencia al subconjunto borroso  $\mathcal{W}^b$ .

- $F_{ijkm}$ , el coste total de enviar el flujo  $(i, j)$  a través del camino  $(i, k, m, j)$ . Este coste se define por  $F_{ijkm} = W_{ij} (C_{ik} + \alpha C_{km} + C_{mj})$ . Así podemos suponer que  $\exists F \supset \mathcal{F} = \left\{ \left( F_{ijkm} \mid \mu_{\mathcal{F}}(F_{ijkm}) \right) \right\}, \forall i, j, k, m$ , el subconjunto borroso de los costes  $F_{ijkm}$  más ciertos, donde  $\mu_{\mathcal{F}}(F_{ijkm}) = \mu_{\mathcal{W}}(W_{ij}) \wedge \left( \frac{\mu_{\mathcal{C}}(C_{ik}) + \mu_{\mathcal{C}}(C_{km}) + \mu_{\mathcal{C}}(C_{mj})}{3} \right)$  es el grado de cumplimiento del coste  $F_{ijkm}$ , o su grado de pertenencia al subconjunto borroso  $\mathcal{F}$ .

Pues, con estos datos vagos, presentamos el modelo USApHLPC siguiente:

$$(Z^* \mid \lambda^*) \equiv \min \sum_i \sum_{j>i} \sum_k \sum_m \left( F_{ijkm} \mid \mu_{\mathcal{F}}(F_{ijkm}) \right) X_{ijkm} + \sum_k a \left( \sum_i \sum_{j>i} \sum_m \left( W_{ij} \mid \mu_{\mathcal{W}}(W_{ij}) \right) X_{ijkm} \right)^b$$

$$\text{s.a.} \quad \sum_k Z_{ik} = 1 \quad \forall i, \quad (2)$$

$$Z_{ik} \leq Z_{kk} \quad \forall i, k, \quad (3)$$

$$\sum_k Z_{kk} = p, \quad (4)$$

$$\sum_m X_{ijkm} = Z_{ik} \quad \forall k, i, j > i, \quad (5)$$

$$\sum_k X_{ijkm} = Z_{jm} \quad \forall m, i, j > i, \quad (6)$$

$$X_{ijkm}, Z_{ik} \in \{0, 1\} \quad \forall i, j > i, k, m, \quad (7)$$

El modelo se resuelve en 2 pasos. Al paso 1 se calcula el coste óptimo crisp  $Z^*$  usando el modelo sin tomar en cuenta los grados de certeza de la función objetivo, es decir simplemente el USApHLP crisp. Luego, al paso 2 se calcula el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el paso 1, usando los grados de certeza de la función objetivo de la manera siguiente:

Grado de cumplimiento USApHLPC:

$$\lambda^* = \frac{\prod_{i,j,k,m} \mu_{\beta_0}(F_{ijkm}) \cdot X_{ijkm} + \prod_k \left( \prod_{i,j,m} \mu_{\beta_0}(W_{ij}) \cdot X_{ijkm} \right)}{2} \quad (1)$$

$$\text{s.a.} \quad \sum_m X_{ijkm} = 1 \quad \forall k, i, j > i, \quad (2)$$

$$\sum_k X_{ijkm} = 1 \quad \forall m, i, j > i, \quad (3)$$

$$\mu_{\beta_0}(F_{ijkm}), \mu_{\beta_0}(W_{ij}) \in [0,1] \quad \forall i, j > i, k, m, \quad (4)$$

Puesto que  $\lambda^* \in [0,1]$ , podemos afirmar que el USApHLPC borroso generaliza el crisp.

### ✓ Algoritmo de cálculo del grado de cumplimiento USApHLPC

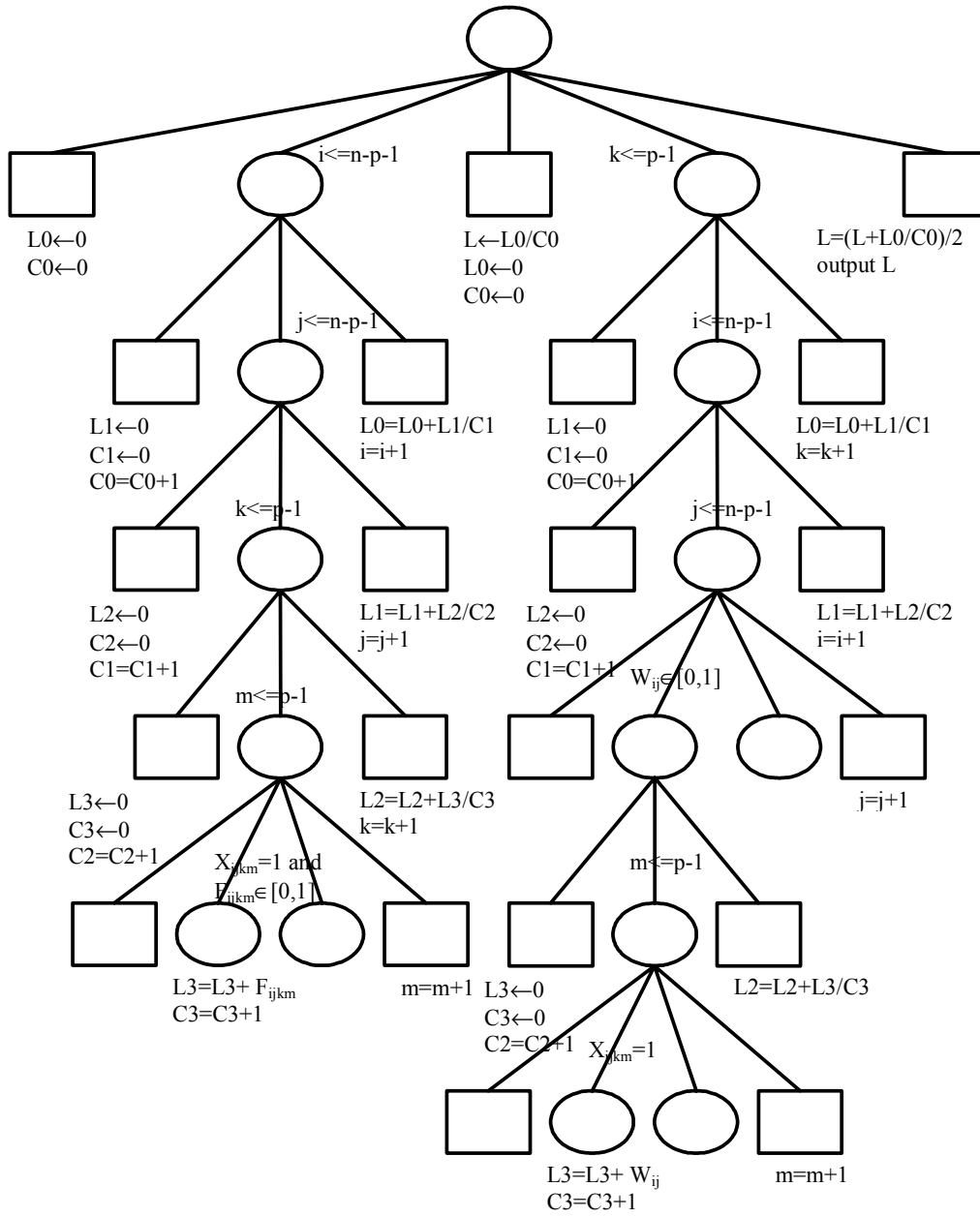
En la entrada, tenemos los datos siguientes:

- La matriz  $X_{(n-p) \times (n-p) \times p \times p} = [X_{ijkm}]_{(n-p) \times (n-p) \times p \times p}$
- La matriz  $F_{(n-p) \times (n-p) \times p \times p} = [\mu_{\beta_0}(F_{ijkm})]_{(n-p) \times (n-p) \times p \times p}$
- La matriz  $W_{(n-p) \times (n-p)} = [\mu_{\beta_0}(W_{ij})]_{(n-p) \times (n-p)}$

En el algoritmo, utilizamos las variables siguientes:

- L Grado de cumplimiento de la solución
- L3, L2, L1 y L0 resultados intermedios calculados en los diferentes bucles del algoritmo, de los bucles más interiores a los bucles más exteriores, respectivamente.
- C3, C2, C1 y C0 contadores que sirven como denominadores en los cálculos en los diferentes bucles del algoritmo, de los bucles más interiores a los bucles más exteriores, respectivamente.
- $i, j, k, m$ , índices que permiten el acceso a las diferentes matrices en la memoria
- $n$  número total los nodos de la red
- $p$  número de nodos centros

Contruiremos nuestro algoritmo a partir del árbol programático siguiente:



**Arbol programático del  
cálculo del grado de certeza  
USApHLPC**

✓ Pseudocódigo: Grado de cumplimiento USApHLPC

```

L0 ← 0
C0 ← 0
for i = 0 to n - p - 1
  L1 ← 0
  C1 ← 0
  C0 = C0 + 1
  for j = 0 to n - p - 1
    L2 ← 0
    C2 ← 0
    C1 = C1 + 1
    for k = 0 to p - 1
      L3 ← 0
      C3 ← 0
      C2 = C2 + 1
      for m = 0 to p - 1
        if  $X_{ijkm} = 1$  and  $F_{ijkm} \in [0, 1]$ 
          L3 = L3 +  $F_{ijkm}$ 
          C3 = C3 + 1
          m = m + 1
        L2 = L2 +  $\frac{L3}{C3}$ 
      k = k + 1
    L1 = L1 +  $\frac{L2}{C2}$ 
  j = j + 1
  L0 = L0 +  $\frac{L1}{C1}$ 
  i = i + 1
L ←  $\frac{L0}{C0}$ 
L0 ← 0
C0 ← 0
for k = 0 to p - 1

```

```

L1 ← 0
C1 ← 0
C0 = C0 + 1
for i = 0 to n - p - 1
  L2 ← 0
  C2 ← 0
  C1 = C1 + 1
  for j = 0 to n - p - 1
    if  $W_{ij} \in [0,1]$ 
      L3 ← 0
      C3 ← 0
      C2 = C2 + 1
      for m = 0 to p - 1
        if  $X_{ikm} = 1$ 
          L3 = L3 +  $W_{ij}$ 
          C3 = C3 + 1
        m = m + 1
      L2 = L2 +  $\frac{L3}{C3}$ 
    j = j + 1
  L1 = L1 +  $\frac{L2}{C2}$ 
  i = i + 1
  L0 = L0 +  $\frac{L1}{C1}$ 
  k = k + 1
L =  $\frac{1}{2} \left( L + \frac{L0}{C0} \right)$ 
output L

```

Tiempo computacional:  $2n^2p^2$  o  $O(n^2p^2)$

✓ **Implementación en C++ : Grado de cumplimiento USApHLPC**

```

//*****//
//Fuzzy USApHLPC: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Sunday, November 21st, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada (Spain)
//*****//
#include <iostream.h>
main()
{
    int n,p,C0=0,C1,C2,C3;
    double L0=0,L1,L2,L3;
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of Hubs: ";
    cin >> p;
    //Declaring input data
    int X[n-p][n-p][p][p];
    double W[n-p][n-p], F[n-p][n-p][p][p];
    //Entering input data
        //will be written later
    //Processing input data
    for (int i=0;i<=n-p-1;){
        L1=0;
        C1=0;
        ++C0;
        for (int j=0; j<=n-p-1;){
            L2=0;
            C2=0;
            ++C1;

```



```

for (int k=0; k<=p-1;){
    L3=0;
    C3=0;
    ++C2;
    for (int m=0; m<=p-1){
        if (X[i][j][k][m] &&
            F[i][j][k][m]>=0 &&
            F[i][j][k][m]<=1){
            L3 += F[i][j][k][m];
            ++C3;
        }
        ++m;
    }
    L2 += L3/C3;
    ++k;
}
L1 += L2/C2;
++j;
}
L0 += L1/C1;
++i;
}
double L = L0/C0;
L0=0;
C0=0;
for (k=0; k<=p-1;){
    L1=0;
    C1=0;
    ++C0;
    for (i=0; i<=n-p-1;){
        L2=0;
        C2=0;
        ++C1;

```

```

for (j=0; j<=n-p-1;){
    if (W[i][j]>=0 and W[i][j]<=1 ){
        L3=0;
        C3=0;
        ++C2;
        for (m=0; m<=p-1;){
            if (X[i][j][k][m]){
                L3 += W[i][j];
                ++C3;
            }
            ++m;
        }
        L2 += L3/C3;
    }
    ++j;
}
L1 += L2/C2;
++i;
}
L0 += L1/C1;
++k;
}
L=(L + L0/C0)/2;
cout << "\nDegree of certainty = " << L << endl;
return 0;
}

```

### III.3. Problemas de localización con asignación múltiple

Los problemas de localización de centro con asignación múltiple se describen de la manera siguiente: tenemos una red de nodos y aristas dirigidas en la cual los nodos comunican entre ellos. Hay un coste por unidad de flujo sobre cada una de las aristas de la red. Todos los flujos deben consolidarse en los centros y cada nodo se asigna a uno o muchos centros. El flujo entre cualquier par de nodos necesita ser enviado vía uno o dos nodos centros. Por ejemplo, el flujo entre dos nodos no centros debe seguir un camino nodo – centro – nodo o nodo – centro – centro – nodo. La localización de los centros se escoge de tal forma que se minimice el coste total del sistema. Para algunas variantes se consideran también los costes fijos para establecer los centros y las capacidades sobre el montante del flujo que puede coleccionar un centro.

Se presentan tres variantes de los problemas de localización de centro con asignación múltiple:

- El problema de  $p$ -mediana central no capacitada con asignación múltiple (UMApHMP: Uncapacitated Multiple Allocation  $p$ -Hub Median Problem);
- El problema de localización de centro capacitado con asignación múltiple (CMAHLP: Capacitated Multiple Allocation Hub Location Problem);
- El problema de localización de centro no capacitado con asignación múltiple (UMAHLPP: Uncapacitated Multiple Allocation Hub Location Problem).

Los problemas CMAHLP, UMAHLPP, UMAApHMP se definen más precisamente de la manera siguiente: tenemos un grafo completamente dirigido  $G = (N, A)$  donde  $N = \{1, \dots, n\}$  es el conjunto de nodos y  $A = N \times N$  es el conjunto de aristas dirigidas. Una solución factible a estos problemas es un conjunto de centros  $H \subseteq N$  y una asignación de nodos no centros a los centros. Todos los pares de nodos comunican entre ellos vía los centros. El flujo del nodo fuente (nodo origen)  $i \in N$  al nodo destino  $j \in N$  se refiere como flujo  $(i, j)$ . En estos problemas, ningún flujo directo está permitido entre dos nodos no centros.  $x_{ijkl}$  es el volumen del flujo  $(i, j)$  enviado vía los centros  $k$  y  $l$ . Existe la desigualdad triangular entre los nodos, así el flujo viaja a través de dos centros como máximo.  $x_{ijkk}$  denota el caso donde se emplea

exactamente un único centro intermedio. Se exige que  $x \geq 0$  y  $\sum_{k \in H} \sum_{l \in H} x_{ijkl} = W_{ij}$ , donde  $W_{ij}$  es el volumen dado del flujo que se debe enviar de  $i$  a  $j$  para cada  $i, j \in N$ . Se refiere a  $x$  como un “routing” factible compatible con  $H$  y una solución al problema es  $(H, x)$ .  $C_{ij}$  es el coste de envío de una unidad del flujo directamente de  $i$  a  $j$  para cada  $i, j \in N$ .  $C_{ij} = C_{ji}$ .

Cada flujo tiene tres componentes separados: *Colección* (del nodo origen  $i$  al centro  $k$ ), *transferencia* (centro  $k$  al centro  $l$ ) y *distribución* (del centro  $l$  al nodo  $j$ ). Cada uno de estos componentes tiene un coeficiente del coste que es  $\chi$ ,  $\alpha$  y  $\delta$  respectivamente. Por ejemplo, el coste asociado a una unidad de flujo que se colecciona desde el nodo  $i$  por el centro  $k$  es  $\chi C_{ik}$  para cada  $i, k \in N$ . Debido a las eficiencias de transferencia inter-centro, se asume que  $\chi, \delta > \alpha$ .

$C(H, x) = F(H) + C_H(x)$  denota el coste de una solución  $(H, x)$ , donde  $F(H) = \sum_{k \in H} F_k$  y  $F_k$  denota el coste fijo de establecer un centro al nodo  $k$ ,  $C_H(x) = \sum_{(i,j) \in A} \sum_{k \in H} \sum_{l \in H} c_{ijkl} x_{ijkl}$  y  $c_{ijkl} = \chi C_{ik} + \alpha C_{kl} + \delta C_{lj} \quad \forall i, j, k, l \in N$ . En el UMA $p$ HMP  $F(H)$  es siempre cero puesto que no hay normalmente ningún coste fijo para localizar un centro.

La diferencia entre CMAHLP y UMAHLP es que CMAHLP tiene una restricción de capacidad de  $\Gamma_k$  sobre el montante del flujo que debe entrar (y/o salir) en un centro  $k$  dado. Si Campbell [14] considera las restricciones de capacidad a un centro sobre ambas la colección y la transferencia, N. Boland et al.[10] consideran las restricciones de capacidad solamente sobre la colección porque su trabajo está motivado por una aplicación postal.

La diferencia entre UMA $p$ HMP y UMAHLP es que en UMA $p$ HMP el número de centros está especificado ( $|H| = p$ ), mientras que UMAHLP determina implícitamente el número de centros que se exigen vía el modelo de decisión.

### III.3.1. Formulaciones de los problemas CMAHLP, UMAHLP y UMA $\rho$ HMP

#### III.3.1.1. Modelos CMAHLP, UMAHLP y UMA $\rho$ HMP crisp

Se presenta formulaciones MILP para cada uno de los problemas UMA $\rho$ HMP, UMAHLP y CMAHLP [10]. Todas las formulaciones utilizan  $O(n^3)$  variables y restricciones y se basan sobre la primera formulación de UMA $\rho$ HMP dada por [38]. La formulación UMA $\rho$ HMP de [38] utiliza las variables:

$Z_{ik}$  = el montante del flujo originando en el nodo  $i$  y coleccionado por el centro  $k$ .

$Y_{kl}^i$  = el montante del flujo originando en el nodo  $i$ , coleccionado por el centro  $k$  y distribuido por el centro  $l$ .

$X_{lj}^i$  = el montante del flujo originando en el nodo  $i$  destinado para el nodo  $j$  y distribuido por el centro  $l$ .

$$H_k = \begin{cases} 1 & \text{si el nodo } k \text{ es un centro} \\ 0 & \text{si no.} \end{cases}$$

La formulación de UMA $\rho$ HMP de [38], con cambios de notación menores se da por [10]:

UMA $\rho$ HMP

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] \quad (1)$$

$$\text{s.a.} \quad \sum_k H_k = p \quad (2)$$

$$\sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (6)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall l, j, \quad (7)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l. \quad (8)$$

La restricción (2) especifica el número de centros mientras que las restricciones (3) - (5) representan las ecuaciones de divergencia para el problema de flujo sobre una red para cada producto o artículo  $i$ . La restricción (6) asegura que ningún flujo se colecciona por un nodo no centro y la restricción (7) asegura que ningún flujo se distribuye por un nodo no centro.

La formulación  $UMApHMP$  tiene  $(2n^3 + n^2 + n)$  variables de las cuales  $n$  son binarias y  $(4n^2 + n + 1)$  son restricciones lineales.

El  $CMAHLP$  fue formulado por primera vez por J.F. Campbell (1994), utilizando  $O(n^4)$  variables. Después, en [32] presentó una formulación que utiliza solamente  $O(n^3)$  variables, parecida a la de  $UMApHMP$ :

$CMAHLP$

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (9)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (6)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall l, j, \quad (7)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l, \quad (8)$$

$$\sum_i Z_{ik} \leq \Gamma_k H_k \quad \forall k. \quad (10)$$

El UMAHLP se formula como sigue [10]:

*UMAHLP*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (9)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (6)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall l, j, \quad (7)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l. \quad (8)$$

Esta formulación tiene  $(2n^3 + n^2 + n)$  variables de las cuales  $n$  son binarias y  $(4n^2 + n)$  son restricciones lineales.

En [10] se presentan algunas características de la solución óptima en los problemas CMAHLP, UMAHLP y UMA $\rho$ HMP que son:

**Propiedad 1:** En cualquier solución  $(H, x)$  a un UMAHLP o UMA $\rho$ HMP, para cada centro  $h \in H$ ,  $x_{h_{jkl}} = 0$  para cada  $k, l \in H$  con  $k \neq h$  y todo  $j \in N$ .

Eso quiere decir que en los problemas de localización de centro en los cuales no hay restricciones de capacidad sobre la colección, cada centro colecciona su propio flujo que está fuera de su límite, en cada solución óptima. Como consecuencia, el montante del flujo originando en el centro  $h$  coleccionado por otro centro  $k$ ,  $k \neq h$ , debe ser cero en cada solución óptima.

En efecto, el coste de enviar un flujo vía el camino  $h \rightarrow k \rightarrow l \rightarrow j$  (coeficiente  $x_{h_{jkl}}$ ) es  $\chi C_{hk} + \alpha C_{kl} + \delta C_{lj}$ . Por tanto el coste de enviar un flujo vía el camino  $h \rightarrow l \rightarrow j$ , con  $h$

como el centro de colección (coeficiente  $x_{ijhl}$ ), es  $\alpha C_{hl} + \delta C_{lj}$ ; por la desigualdad triangular ( $C_{hl} \leq C_{hk} + C_{kl}$ ) y puesto que  $\alpha < \chi$ , esto es el camino más económico para el flujo  $(i, j)$

**Propiedad 2:** En cualquier solución  $(H, x)$  a un UMAHLP, UMA $\rho$ HMP o CMAHLP, para cada centro  $h \in H$ ,  $x_{jhlk} = 0$  para cada  $l, k \in H$  con  $k \neq h$  y todo  $j \in N$ .

Eso quiere decir que en los problemas de localización de centros en los cuales no hay restricciones de capacidad sobre la distribución, cada centro distribuirá su propio flujo que está en su límite en cada solución óptima. Como consecuencia, el montante del flujo destinado para un centro  $h$  distribuido por otro centro  $k$ ,  $k \neq h$ , debe ser cero en cualquier solución óptima. Puesto que CMAHLP no tiene restricción de capacidad sobre la distribución, esta propiedad se aplica tan a él como a UMA $\rho$ HMP y UMAHLP.

No obstante, en los problemas con colección capacitada no podemos asumir que los centros con capacidad suficiente coleccionarán sus propios flujos. Consideremos un CMAHLP con  $N = \{1, 2, 3\}$ ,  $F = (100, 0, 0)$ ,  $\Gamma = (10, 1, 1)$ ,  $W_{12} = 1$ ,  $W_{23} = 1$  y  $W_{ij} = 0$  para todo  $(i, j) \in A \setminus \{(1, 2), (2, 3)\}$ ,  $C_{12} = C_{21} = C_{23} = C_{32} = 1$ ,  $C_{13} = C_{31} = 2$ ,  $\delta = \chi = 2$ , y  $\alpha = 1$ . El nodo 1 no es obviamente un centro en la solución óptima debido a su alto coste fijo, con tal que una cierta solución factible exista sin el centro 1. Puesto que los nodos 1 y 3 no tienen ningún coste fijo y una capacidad combinada de 2, deben ambos ser elegidos como centros si el nodo 1 no es un centro. Ahora el flujo a partir de 2 puede coleccionarse por el nodo 2 o el nodo 3. El flujo desde 1 a 2 también tiene dos opciones:  $1 \rightarrow 2$  o  $1 \rightarrow 3 \rightarrow 2$ . Esto conduce a dos soluciones posibles:

1. El flujo a partir del nodo 1 se colecciona en 2 y el flujo a partir del nodo 2 se colecciona en 3. Esto conduce a un coste de  $\chi(C_{12} + C_{23}) = 4$ .
2. El flujo a partir del 1 se colecciona en 3 y se transfiere a 2, mientras que el flujo a partir del 2 se colecciona en 2 y se transfiere a 3. Este da un coste de  $\chi C_{13} + \alpha C_{32} + \chi C_{22} + \alpha C_{23} = 6$ .



Por lo tanto la primera solución, dondel flujo del centro 2 es coleccionado por el centro 3, es óptima y el nodo 1 no es un nodo centro posible. No es necesariamente el caso que un centro coleccionará su propio flujo de salida si la colección está capacitada.

Por tanto, si se consideran problemas con colección capacitada, no se puede necesariamente asumir que los centros con suficiente capacidad de colección coleccionarán su propio flujo.

**Propiedad 3:** En cualquier solución óptima  $(H, x)$  a un UMAHLP o UMA $p$ HMP, para todo  $i, k, l \in N$ , con  $\chi C_{ik} + \alpha C_{kl} > \chi C_{il}$ ,  $x_{ijkl} = 0$  para cada  $j \in N$ .

Eso quiere decir que para los problemas de localización de centro con colección no capacitada, en cada solución óptima, ningún flujo se enviará desde un nodo  $i$  a un centro  $l$  vía un centro  $k$  si es más económico enviar un flujo directamente a  $l$ .

En efecto, sea  $(H, x)$  una solución óptima a un UMAHLP o UMA $p$ HMP, y sea  $i, k, l \in N$  tal que  $\chi C_{ik} + \alpha C_{kl} > \chi C_{il}$ . Supongamos que para cada  $j \in N$  tenemos  $x_{ijkl} = \omega > 0$ . (En este caso,  $k, l \in H$ .) Podemos construir una solución factible  $(H, y)$  que tenga un coste estrictamente menor que el de  $(H, x)$  y así llegar a una contradicción. Nuestro “routing” factible  $y$  se obtiene re-enviando  $\omega$  unidades del flujo  $(i, j)$  coleccionado por  $k$  y distribuido por  $l$  en el flujo  $x$ , de forma que estén coleccionadas y distribuidas por  $l$ . Precisamente se define

$$y_{fghm} = \begin{cases} x_{ijll} + \omega, & \text{si } (f, g, h, m) = (i, j, l, l), \\ x_{ijkl} - \omega, & \text{si } (f, g, h, m) = (i, j, k, l), \\ x_{fghm}, & \text{por lo demás,} \end{cases}$$

para todo  $f, g, h, m \in N$ . Claramente  $y$  es un “routing” factible compatible con  $H : y \geq 0$ , los cambios relacionados al “routing”  $x$  implican solamente el flujo  $(i, j)$  y la colección aumentada al centro  $l$ , el montante neto del flujo  $(i, j)$  enviado no cambia, y no hay capacidades sobre la colección. Ahora el coste de  $(H, y)$  se da por

$$\begin{aligned} & C(H, y) \\ &= F(H) + \sum_{(f,g) \in A} \sum_{h \in H} \sum_{m \in H} y_{fghm} \\ &= F(H) + \sum_{(f,g) \in A} \sum_{h \in H} \sum_{m \in H} x_{fghm} + c_{ijll} \omega - c_{ijkl} \omega \end{aligned}$$

$$= C(H, x) + \omega(\chi C_{il} + \delta C_{lj} - \chi C_{ik} - \alpha C_{kl} - \delta C_{lj})$$

$$= C(H, x) + \omega(\chi C_{il} - \chi C_{ik} - \alpha C_{kl}) < C(H, x)$$

puesto que  $\omega > 0$  y  $\chi C_{ik} + \alpha C_{kl} > \chi C_{il}$ . Esto contradice la optimalidad de  $(H, x)$ .

**Propiedad 4:** En cualquier solución óptima  $(H, x)$  a un UMAHLP, UMA $p$ HMP o CMAHLP, para todo  $j, k, l \in N$  con  $\alpha C_{kl} + \delta C_{lj} > \delta C_{kj}$ ,  $x_{ijkl} = 0$  para todo  $i \in N$ .

Eso quiere decir que para los problemas de localización de centro con distribución no capacitada, se observa que en cada solución óptima, ningún flujo estará enviado de un centro  $k$  a un nodo  $j$  vía un centro  $l$  si es más económico enviar el flujo directamente desde  $k$ .

**Propiedad 5:** En cualquier solución óptima  $(H, x)$  a un UMAHLP o UMA $p$ HMP, para todo  $i, l \in N$  y  $k \in H$  con  $\chi C_{ik} + \alpha C_{kl} < \chi C_{il}$ ,  $x_{ijlh} = 0$  para todo  $j, h \in N$ .

Es decir, en toda solución óptima a un problema de localización de centro con colección no capacitada, ningún flujo estará coleccionado por un centro  $l$  si es más económico coleccionarlo al centro  $k$  luego transferirlo a  $l$ .

En efecto, sea  $(H, x)$  una solución óptima a un UMAHLP o UMA $p$ HMP, y sea  $i, l \in N$  y  $k \in H$  sea tal que  $\chi C_{ik} + \alpha C_{kl} < \chi C_{il}$ . Supongamos que para algunos  $j, h \in N$  tenemos  $x_{ijlh} = \omega > 0$ . (En este caso,  $l, h \in H$ ). Podemos construir una solución factible  $(H, y)$  que tenga un coste estrictamente menor que el de  $(H, x)$  y así llegar a una contradicción. Nuestro “routing” factible  $y$  se obtiene re-enviando  $\omega$  unidades del flujo  $(i, j)$  coleccionado por  $l$  y distribuido por  $h$  en el flujo  $x$ , de forma que esté coleccionado por  $k$ . Precisamente,

$$y_{efgm} = \begin{cases} x_{ijkh} + \omega, & \text{si } (e, f, g, m) = (i, j, k, h) \\ x_{ijlh} - \omega, & \text{si } (e, f, g, m) = (i, j, l, h) \\ x_{efgm}, & \text{por lo demás} \end{cases}$$

para todo  $e, f, g, m \in N$ . Claramente  $y$  es un “routing” factible compatible con  $H: y \geq 0$ , los cambios relacionados al “routing”  $x$  implican solamente el flujo  $(i, j)$  y la colección aumentada al  $k$ , que es un centro por las condiciones de la propiedad, el montante neto del flujo  $(i, j)$

enviado no ha cambiado, y no hay capacidades sobre la colección. Así, está claro que  $(H, y)$  es una solución factible. Ahora, el coste de  $(H, y)$  es:

$$\begin{aligned}
& C(H, y) \\
&= F(H) + \sum_{(e,f) \in A} \sum_{g \in H} \sum_{m \in H} y_{efgm} \\
&= F(H) + \sum_{(e,f) \in A} \sum_{g \in H} \sum_{m \in H} x_{efgm} + c_{ijkh} \omega - c_{ijlh} \omega \\
&= C(H, x) + \omega (\chi C_{ik} + \alpha C_{il} - \chi C_{il} - \alpha C_{lh} - \delta C_{lj}) \leq C(H, x) + \omega (\chi C_{ik} + \alpha (C_{kl} + C_{lh}) - \chi C_{il} - \alpha C_{lh})
\end{aligned}$$

por la V-desigualdad  $< C(H, x)$

puesto que  $\omega > 0$  y  $\chi C_{ik} + \alpha C_{kl} < \chi C_{il}$ . Esto contradice la optimalidad de  $C(H, x)$ .

**Propiedad 6:** En cualquier solución óptima  $(H, x)$  a un UMAHLP, UMA $\rho$ HMP o CMAHLP, para todo  $j, k \in N$  y  $l \in H$  con  $\alpha C_{kl} + \delta C_{lj} < \delta C_{kj}$ ,  $x_{ijhk} = 0$  para todo  $i, h \in N$ .

Aunque las propiedades 3 y 5 son simétricas, hay una sutil diferencia entre ellas; en la propiedad 3, la condición del coste puede aplicarse a tres nodos cualesquiera y el resultado será válido, pero en la propiedad 5 uno de los nodos tiene que ser un centro. Es la misma diferencia también entre las propiedades 4 y 6.

También en [10] se considera el flujo  $(i, i)$ , y se deduce algunas propiedades que los “routings” óptimos deben satisfacer con respecto a este tipo de flujo. En primer lugar, se observa que si  $C_{ik} \leq C_{ih}$  para algunos  $i, k, h \in N$ ,  $h \neq k$ , entonces como corolario a la propiedad 3, (siempre que no haya capacidades en colección),  $x_{iik} = 0$ , y como corolario a la propiedad 4 (siempre que no haya capacidades en distribución),  $x_{iikh} = 0$ , para cualquier routing óptimo  $x$ . Así, en efecto, en las soluciones óptimas para los problemas no capacitados, el flujo  $(i, i)$  nunca se transfiere entre centros, y un centro colecciona el flujo  $(i, i)$  si y solamente si el mismo centro lo distribuye también.

**Propiedad 7:** En cualquier solución óptima  $(H, x)$  a un UMA $\rho$ HMP o un UMAHLP, para todo  $i, k, h \in N$ ,  $x_{iikh} = 0$  si  $k \neq h$  o si  $k \notin \arg \min_{l \in H} C_{il}$ .

Es decir, en una solución óptima a un problema no capacitado, todo flujo  $(i, i)$  se coleccionará y se distribuirá por un centro que es el más cercano a él. En efecto, sea  $(H, x)$  una solución óptima a un UMAPHMP o un UMAHLP y sea  $i \in N$ . Consideremos  $k, h \in N$  con  $k \neq h$ . Ahora, sea  $C_{ik} \leq C_{ih}$  o  $C_{ih} \leq C_{ik}$ . En primer caso, tenemos por la propiedad 4, y puesto que  $C_{kh} > 0$ , que  $x_{ikh} = 0$ . En el segundo, obtenemos el mismo resultado por la propiedad 3 y también puesto que  $C_{kh} > 0$ . Ahora, consideremos  $k \notin \arg \min_{l \in H} C_{il}$  y supongamos  $x_{ikk} = \omega > 0$ . En este caso  $k$  tiene que ser un centro, es decir  $k \in H$ . Podemos construir una solución factible  $(H, y)$  que tenga un coste estrictamente menor que el de  $(H, x)$  y así llegar a una contradicción. Nuestro routing factible  $y$  se obtiene re-rutando las  $\omega$  unidades del flujo  $(i, i)$  coleccionadas y distribuidas por  $k$  en el flujo  $x$ , de tal forma que estén coleccionadas y distribuidas por  $j \in \arg \min_{l \in H} C_{il}$ . Por las definiciones de  $j$  y  $k$ ,  $C_{ij} < C_{ik}$ . Se define

$$y_{efgm} = \begin{cases} x_{ijj} + \omega, & \text{si } (e, f, g, m) = (i, i, j, j) \\ x_{ikk} - \omega, & \text{si } (e, f, g, m) = (i, i, k, k) \\ x_{efgm}, & \text{para los demás} \end{cases}$$

para todo  $e, f, g, m \in N$ .  $y$  es un routing factible compatible con  $H : y \geq 0$ , los cambios relacionados al routing  $x$  implican solamente el flujo  $(i, i)$  y la colección y distribución aumentadas a  $j$ , que es un centro por definición, el montante neto del flujo  $(i, i)$  enviado no ha cambiado, y no hay capacidades sobre la colección o la distribución. Así, está claro que  $(H, y)$  es una solución factible. Ahora, el coste de  $(H, y)$  se da por

$$\begin{aligned} C(H, y) &= F(H) + \sum_{(e,f) \in A} \sum_{g \in H} \sum_{m \in H} y_{efgm} \\ &= F(H) + \sum_{(e,f) \in A} \sum_{g \in H} \sum_{m \in H} x_{efgm} + c_{ijj} \omega - c_{ikk} \omega \\ &= C(H, x) + \omega (\chi C_{ij} + \delta C_{ji} - \chi C_{ik} - \delta C_{ki}) \\ &= C(H, x) + \omega (\chi + \delta) (C_{ij} - C_{ik}) \end{aligned}$$

por la simetría de  $C < C(H, x)$

puesto que  $\omega > 0$ ,  $\chi + \delta > 0$  y  $C_{ij} < C_{ik}$ . Esto contradice la optimalidad de  $(H, x)$ .

Para los problemas de  $p$ -centro, [10] deduce implícitamente alguna información útil sobre  $\arg \min_{l \in H} C_{il}$  para cada  $i \in N$ . Sea  $\pi^i$  una permutación de los nodos de modo que  $C_{i\pi_1^i} \geq C_{i\pi_2^i} \geq \dots \geq C_{i\pi_n^i}$ . En cualquier conjunto de centros, para un problema de  $p$ -centro, habrá un centro más cerca a  $i$  que los  $p-1$  nodos más lejos de  $i$ . Precisamente, debe existir  $j \in H$  con  $C_{ij} \leq C_{i\pi_p^i}$ . Así,  $\min_{l \in H} C_{il} \leq C_{i\pi_p^i}$ , y se deduce que para todo  $k \in N$  con  $C_{ik} > C_{i\pi_p^i}$ ,  $k \notin \arg \min_{l \in H} C_{il}$ ,  $x_{iikk} = 0$  para cada  $x$ , un routing óptimo para un problema de  $p$ -centro no capacitado, por la propiedad 7. Así se ha demostrado la propiedad 8, donde  $\Delta_i = C_{i\pi_p^i}$  para cada  $i \in N$ .

**Propiedad 8:** En cada solución óptima  $(H, x)$  a un UMAPHMP,  $x_{iikk} = 0$  para todo  $i \in N$  y todo  $k \in N$  con  $C_{ik} > \Delta_i$ .

Este resultado no es tan fuerte como debería ser en caso de que haya varios nodos lejos de  $i$  de distancia exactamente  $\Delta_i$  para algún  $i \in N$  con  $W_{ii} > 0$ . Claro que pudiéramos tener  $x_{iikk} > 0$  para cualquier  $k \in \arg \min_{l \in H} C_{il}$ , y cualquier  $x$  un routing óptimo, pero debe existir un routing  $\hat{x}$  que tenga un coste idéntico al de  $x$ , y por eso óptimo, con  $\hat{x}_{iikk} = 0$  para todo  $k \neq \hat{k}$  para un único  $\hat{k} \in \arg \min_{l \in H} C_{il}$ . En efecto, si el conjunto óptimo de centro  $H$  fuera conocido, se pudiera escoger un  $k^i \in \arg \min_{l \in H} C_{il}$  para cada  $i \in N$  y requerir que todo flujo  $(i, i)$  esté coleccionado y distribuido por  $k^i$  para todo  $i \in N$  sin eliminar todos los routings óptimos, es decir, debe existir al menos un routing óptimo satisfaciendo este requisito. [10] utiliza estas ideas para formular una propiedad que es ligeramente más fuerte que la propiedad 8, pero que debe eliminar algunos routings óptimos.

**Propiedad 9:** Dado un UMAPHMP, para cada  $i \in N$  escoger un conjunto  $F_i \subseteq N$  con  $|F_i| = p-1$  y  $C_{ik} \geq C_{il}$  para todo  $k \in F_i$  y todo  $l \in N \setminus F_i$ . Entonces existe una solución óptima  $(H, x)$  al UMAPHMP con  $x_{iikk} = 0$  para todo  $i \in N$  y todo  $k \in F_i$ .

Combinando esta propiedad con la propiedad 7, vemos que para todo  $i \in N$  y todo  $k \in F_i$ , donde los conjuntos  $F_i$  son tal como definidos en la propiedad 9,  $x_{iik} = 0$  y  $x_{ikh} = 0$  para todo  $h \in N$  en algún routing óptimo  $x$  para UMAPHMP.

N.Boland et al. [10] utilizan algunas de las propiedades de las características de las soluciones óptimas en CMAHLP, UMAHLP y UMAPHMP para desarrollar técnicas de preprocesamiento y apretamiento de las restricciones para las formulaciones presentadas. Las propiedades presentadas en las características de las soluciones óptimas en CMAHLP, UMAHLP y UMAPHMP caracterizan las soluciones óptimas en vez de las soluciones factibles, y así las restricciones que se desarrollan no serán necesariamente válidas, pero se satisfarán por todas, o al menos algunas soluciones óptimas. Las propiedades todas requieren sea una colección no capacitada, una distribución no capacitada, o ambas, así solamente pocas propiedades pueden aplicarse al CMAHLP. No obstante, si se considera agregar algunas variables en la formulación de CMAHLP, vemos aparecer una estructura variable del flujo del límite superior, y por lo tanto podemos determinar restricciones del recubrimiento del flujo que son las restricciones válidas que se pueden utilizar para apretar la formulación. También observamos que las variables (complementadas) de la localización del centro deben satisfacer una restricción de la mochila, y por lo tanto las desigualdades del recubrimiento de la mochila pueden ser aplicadas; En hecho, ambas observaciones son consecuencia de la sola observación que las variables  $(Z, H)$  satisfacen las restricciones de la localización de la capacidad, y así que el trabajo de K.Aadril (1998) sobre las desigualdades válidas para CFL puede ser aplicado.

Primero, con el propósito de consolidar las restricciones, considerando la propiedad 1, que estipula que para los problemas con colección no capacitada, el centro  $k \in H$  no coleccionará un flujo originando en cualquier otro centro  $h \in H$ , tenemos para todo  $h \in N$ :

$$\sum_{k \neq h} Z_{hk} \leq O_h(1 - H_h) \quad (11)$$

Aquí  $O_i = \sum_j W_{ij}$  es el montante total del flujo originando en el nodo  $i$ . Esta restricción puede escribirse usando ecuaciones (3) y (6) como

$$Z_{hh} = O_h H_h \quad (12)$$

para todo  $h \in N$ . Se observa que añadir (12) en la formulación introduce pocos elementos no cero que añadir (11). En la práctica, añadir (12) es computacionalmente más efectiva que (11).

Considerando la propiedad 2 que estipula que para todos los problemas con distribución no capacitada, el flujo destinado al centro  $h \in H$  no se distribuirá por cualquier otro centro  $k \in H$ . Así, la ecuación siguiente está válida para todo  $i, k, h \in N$  con  $k \neq h$ :

$$X_{kh}^i \leq W_{ih} (1 - H_h) \quad (13)$$

La restricción (13) puede escribirse usando ecuaciones (4) y (7):

$$X_{hh}^i = W_{ih} H_h \quad (14)$$

para todo  $i, h \in N$ . Se observa que (14) es más computacionalmente efectiva que (13).

En [10] se utiliza las propiedades 3, 5 y 8 para desarrollar técnicas de preprocesamiento para eliminar algunas de las variables de las formulaciones.

Cuando  $\chi C_{ik} + \alpha C_{kl} > \chi C_{il}$  para todo  $i, k, l \in N$  y que la colección no está capacitada, entonces por la propiedad 3, ningún flujo originando en  $i$  estará coleccionado por  $k$  y distribuido por  $l$ , así se puede poner  $Y_{kl}^i = 0$ .

Desafortunadamente la propiedad 4 no se puede utilizar en la misma manera: estipula que si la distribución no está capacitada y que  $\alpha C_{kl} + \delta C_{lj} > \delta C_{kj}$  para algunos  $j, k, l \in N$ , entonces ningún flujo destinado para  $j$  estará coleccionado por  $k$  y distribuido por  $l$ . Puesto las formulaciones no están simétricas, no hay una variable análoga a la variable  $y$  que se aplique al flujo de destino en el lugar del flujo de origen.

La propiedad 5 estipula que para los problemas con colección no capacitada, si  $\chi C_{ik} + \alpha C_{kl} < \chi C_{il}$  para algunos  $i, l \in N$  y algún  $k \in H$ , entonces ningún flujo originando en  $i$  estará coleccionado por  $l$ . Esto se puede utilizar para generar la restricción:

$$Z_{il} \leq O_i (1 - H_k) \quad (15)$$

para todo  $i, k, l \in N$ . Esto es justamente una amplia clase de restricciones, así no se puede intentar añadirlas en la formulación. No obstante es posible utilizar estas restricciones para

fijar las variables  $Z_{il}$  en el árbol de Branch and Bound. Desafortunadamente, en este caso, no hay ninguna análoga para las variables  $Y$ .

La propiedad 6 estipula que para los problemas con distribución no capacitada, si  $\alpha C_{kl} + \delta C_{lj} < \delta C_{kj}$  para algunos  $j, k \in N$  y algún  $l \in N$ , entonces ningún flujo destinado para  $j$  se distribuirá por el centro  $k$ . Esto se puede utilizar para generar la restricción:

$$\sum_i X_{kj}^i \leq D(1 - H_l) \quad \forall j, k, l \in N \quad (16)$$

Puesto que no se sabe  $H$  a priori, no hay manera directa de tener una ventaja de la propiedad 7. No obstante, si se mezcla con la propiedad 9, se observa que preprocesando una UMAPHMP se puede escoger los conjuntos  $F_i$  para cada  $i \in N$  según la propiedad 9, y se puede asegurar que ningún flujo  $(i, i)$  estará distribuido o coleccionado por nodos en  $F_i$ . No obstante, tenemos solamente variables que anotan la distribución de flujo  $(i, i)$ : las variables  $X_{ki}^i$ . Así se fija  $X_{ki}^i = 0$  para todo  $k \in F_i$  para todo  $i \in N$ , es decir, se elimina en el preprocesamiento todas las variables  $X_{ki}^i$  con  $k \in F_i$ . Además, una vez que se fija un centro  $l$  por el proceso de Branching, es posible fijar  $X_{ki}^i$  para todo  $k$  más lejos de  $i$  que  $l$ .

Las variables (complementarias) de localización de centro deben satisfacer una restricción de la mochila, y así las desigualdades de la recubrimiento de la mochila se pueden aplicar. Además emerge una estructura del flujo superior de una variable, y así se puede aplicar las desigualdades del recubrimiento del flujo. Claramente, tiene que cumplirse:

$$\sum_{k \in N} \Gamma_k H_k \geq \Omega \quad (17)$$

para que  $H$  induzca un conjunto de centros factibles, donde  $\Omega = \sum_{i \in N} \sum_{j \in N} W_{ij}$  es el flujo total. A partir de esta restricción, las restricciones del recubrimiento de la mochila se pueden deducir. Cualquier conjunto  $J \subset N$  con  $\sum_{k \in N \setminus J} \Gamma_k < \Omega$  se denomina un recubrimiento, y para cualquier recubrimiento  $J$ , la restricción del recubrimiento de la mochila

$$\sum_{k \in J} H_k \geq 1$$

es válida.



Ahora, para todo  $k \in N$ , considerando las variables agregadas  $\sum_{i \in N} Z_{ik}$  juntas con las variables binaras de localización de centro  $H_k$  de la formulación de CMAHLP en [10], se observa que la estructura del flujo superior aparece. Tenemos

$$\sum_{i \in N} Z_{ik} \leq \Gamma_k H_k$$

para todo  $k \in N$  (esto es precisamente (10)) y por (3)

$$\begin{aligned} \sum_{k \in N} \left( \sum_{i \in N} Z_{ik} \right) &= \sum_{i \in N} \left( \sum_{k \in N} Z_{ik} \right) \\ &= \sum_{i \in N} \left( \sum_{j \in N} W_{ij} \right) = \Omega \end{aligned}$$

$D \subseteq N$  se denomina un recubrimiento si  $\sum_{k \in D} \Gamma_k > \Omega$ , y recubrimiento mínimo si  $D \setminus \{k\}$  no es un recubrimiento para todo  $k \in D$ . Se utiliza  $\lambda_D$  para denotar el exceso de capacidad de un recubrimiento  $D$ , es decir  $\lambda_D = \sum_{k \in D} \Gamma_k - \Omega$ . Ahora por resultados estándares, tenemos

$$\sum_{k \in D} \sum_{i \in N} Z_{ik} + \sum_{k \in D} (\Gamma_k - \lambda_D)(1 - H_k) \leq \Omega$$

que es válido para cualquier recubrimiento mínima  $D$ . Esto es equivalente a

$$\sum_{k \in D} \sum_{i \in N} Z_{ik} - \sum_{k \in D} (\Gamma_k - \lambda_D) H_k \leq (|D| - 1) \lambda_D \quad (18)$$

El propósito de estas restricciones es cortar los puntos fraccionarios de la forma  $H_k = 1$  para cada  $k \in D \setminus \{l\}$ ,  $H_l = (\Gamma_l - \lambda_D) / \Gamma_l$ , para algún  $l \in D$ . Se observa que para  $D$  un recubrimiento mínima es  $0 < H_l < 1$ . Se observa también que la capacidad total de esta solución fraccionaria es

$$\sum_{k \in D \setminus \{l\}} \Gamma_k H_k + \Gamma_l H_l = \sum_{k \in D \setminus \{l\}} \Gamma_k + (\Gamma_l - \lambda_D) = \Omega,$$

mientras que el coste fijo es

$$\sum_{k \in D \setminus \{l\}} F_k + F_l \left( \frac{\Gamma_l - \lambda_D}{\Gamma_l} \right) < \sum_{k \in D} F_k$$

Así [10] agrega  $n$  nuevas variables,  $\{Z_k : k \in N\}$  y  $2n$  nuevas restricciones:

$$Z_k = \sum_{i \in N} Z_{ik}, \quad \forall k \in N \quad (19)$$

y

$$Z_k \leq \Gamma_k H_k, \quad \forall k \in N. \quad (20)$$

Claramente no se necesita la restricción (10) en la formulación, puesto que en efecto se han agregado solamente  $n$  nuevas restricciones. Se podría también añadir

$$\sum_{k \in N} Z_k = \Omega \quad (21)$$

Se reformula las tres variantes de los problemas de localización de centro: UMApHLP, UMAHLP, CMAHLP. Se da una indicación del tamaño del problema proporcionando el número de variables y restricciones en cada nueva formulación antes del preprocesamiento. Cuando en la reformulación de cualquier problema se ponen números en el final del nombre del problema es para indicar cual de las cinco propiedades se han aplicado al problema.

*Problema UMApHMP-1,2,3,5*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] \quad (1)$$

$$\text{s.a.} \quad \sum_k H_k = p \quad (2)$$

$$\sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (8)$$

$$Z_{hh} = O_h H_h \quad (12)$$

$$X_{hh}^i = W_{ih} H_h \quad (14)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, i \neq k \quad (22)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (23)$$

La ecuación (12) y la desigualdad (22) ambas proporcionan una restricción más fuerte que la desigualdad original (6). De la misma manera (14) y (23) son mejores que (7). Esta formulación tiene  $(2n^3 + n^2 + n)$  variables de las cuales  $n$  son binarias y  $(5n^2 + 2n + 1)$  son restricciones.

*Problema UMAHLP-1,2,3*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (9)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l, \quad (8)$$

$$Z_{hh} = O_h H_h \quad (12)$$

$$X_{hh}^i = W_{ih} H_h \quad (14)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, i \neq k, \quad (22)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j \quad (23)$$

Esta formulación tiene el mismo número de variables como el UMAHLP-N y  $(5n^2 + 2n)$  restricciones.

*Problema CMAHLP-2*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (9)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (6)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l, \quad (8)$$

$$X_{hh}^i = W_{ih} H_h \quad (14)$$

$$Z_k = \sum_{i \in N} Z_{ik} \quad \forall k \in N, \quad (19)$$

$$Z_k \leq \Gamma_k H_k \quad \forall k \in N, \quad (20)$$

$$\sum_{k \in N} Z_k = \Omega \quad (21)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (23)$$

Esta formulación tiene  $(2n^3 + n^2 + 2n)$  variables de las cuales  $n$  son binarias. Tiene también  $(5n^2 + 2n + 1)$  restricciones.

### III.3.1.2. Modelos CMAHLP, UMAHLP y UMA $\rho$ HMP borrosos

#### III.3.1.2.1. El problema CMAHLP borroso

##### ✓ Formulación del modelo

El modelo crisp desarrollado en [10] se presenta de la siguiente:

*Problema CMAHLP-2 crisp*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (1)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (2)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (3)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (4)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l, \quad (6)$$

$$X_{hh}^i = W_{ih} H_h \quad (7)$$

$$Z_k = \sum_{i \in N} Z_{ik} \quad \forall k \in N, \quad (8)$$

$$Z_k \leq \Gamma_k H_k \quad \forall k \in N, \quad (9)$$

$$\sum_{k \in N} Z_k = \Omega \quad (10)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (11)$$

En la función objetivo de esta formulación suponemos la incertidumbre en los datos siguientes:

- $C_{ij}$  el coste por unidad del flujo de  $i$  a  $j$  por todo  $i, j \in N$ . Así suponemos que  $\exists C \supset \mathcal{C}^i = \left\{ \left( C_{ij} \mid \mu_{\mathcal{C}^i}(C_{ij}) \right) \right\}, \forall i, j \in N$ , donde  $\mu_{\mathcal{C}^i}(C_{ij}) \in [0, 1]$  es el grado de pertenencia del coste  $C_{ij}$  al subconjunto borroso  $\mathcal{C}^i$ , o el grado de cumplimiento de  $C_{ij}$ ;
- $Z_{ik}$  el flujo del nodo  $i$  al centro  $k$  por todo  $i, k \in N$ . Así suponemos que  $\exists Z \supset \mathcal{Z}^i = \left\{ \left( Z_{ik} \mid \mu_{\mathcal{Z}^i}(Z_{ik}) \right) \right\}, \forall i, k \in N$ , donde  $\mu_{\mathcal{Z}^i}(Z_{ik}) \in [0, 1]$  es el grado de pertenencia del flujo  $Z_{ik}$  al subconjunto borroso  $\mathcal{Z}^i$ , o el grado de cumplimiento de  $Z_{ik}$ ;
- $Y_{kl}^i$  el flujo del nodo  $i$  vía los centros  $k$  y  $l$  por todo  $i, k, l \in N$ . Así suponemos que  $\exists Y \supset \mathcal{Y}^i = \left\{ \left( Y_{kl}^i \mid \mu_{\mathcal{Y}^i}(Y_{kl}^i) \right) \right\}, \forall i, k, l \in N$ , donde  $\mu_{\mathcal{Y}^i}(Y_{kl}^i) \in [0, 1]$  es el grado de pertenencia del flujo  $Y_{kl}^i$  al subconjunto borroso  $\mathcal{Y}^i$ , o el grado de cumplimiento de  $Y_{kl}^i$ ;
- $X_{lj}^i$  el flujo del nodo  $i$  distribuido por el centro  $l$  al nodo  $j$  por todo  $i, l, j \in N$ . Así suponemos que  $\exists X \supset \mathcal{X}^i = \left\{ \left( X_{lj}^i \mid \mu_{\mathcal{X}^i}(X_{lj}^i) \right) \right\}, \forall i, l, j \in N$ , donde  $\mu_{\mathcal{X}^i}(X_{lj}^i) \in [0, 1]$  es el grado de pertenencia del flujo  $X_{lj}^i$  al subconjunto borroso  $\mathcal{X}^i$ , o el grado de cumplimiento de  $X_{lj}^i$ ;
- $F_k$  el coste fijo de establecer un centro al centro  $k$  por todo  $k \in N$ . Así suponemos que  $\exists F \supset \mathcal{F}^i = \left\{ \left( F_k \mid \mu_{\mathcal{F}^i}(F_k) \right) \right\}, \forall k \in N$ , donde  $\mu_{\mathcal{F}^i}(F_k) \in [0, 1]$  es el grado de pertenencia del coste  $F_k$  al subconjunto borroso  $\mathcal{F}^i$ , o el grado de cumplimiento de  $F_k$ .

Con estos datos imprecisos, presentamos el modelo del problema borroso de localización de centro capacitado con asignación múltiple que denominamos *CMAHLP-2 borroso*:

*Problema CMAHLP-2 borroso*

$$(Z^* | \lambda^*) \cong \min \sum_i \left[ \lambda \sum_k (C_{ik} | \mu_{\mathcal{C}^0}(C_{ik})) (Z_{ik} | \mu_{\mathcal{Z}^0}(Z_{ik})) + \right. \\ \left. \alpha \sum_k \sum_l (C_{kl} | \mu_{\mathcal{C}^0}(C_{kl})) (Y_{kl}^i | \mu_{\mathcal{Y}^0}(Y_{kl}^i)) + \delta \sum_l \sum_j (C_{lj} | \mu_{\mathcal{C}^0}(C_{lj})) (X_{lj}^i | \mu_{\mathcal{X}^0}(X_{lj}^i)) \right] + \quad (1)$$

$$\sum_k (F_k | \mu_{\mathcal{F}^0}(F_k)) H_k$$

$$\text{s.a.} \quad \sum_k Z_{ik} \cong \sum_j W_{ij} \quad \forall i, \quad (2)$$

$$\sum_l X_{lj}^i \cong W_{ij} \quad \forall i, j, \quad (3)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} \cong 0 \quad \forall i, k, \quad (4)$$

$$Z_{ik} \cong \sum_j W_{ij} H_k \quad \forall i, k, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l, \quad (6)$$

$$X_{hh}^i \cong W_{ih} H_h \quad (7)$$

$$Z_k \cong \sum_i Z_{ik} \quad \forall k \in N, \quad (8)$$

$$Z_k \cong \sum_i W_{ij} H_k \quad \forall k \in N, \quad (9)$$

$$\sum_{k \in N} Z_k \cong \Omega \quad (10)$$

$$\sum_i X_{lj}^i \cong \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (11)$$

El modelo se resuelve en dos etapas. Primero se calcula  $Z^*$  usando la versión crisp del problema, es decir el modelo *CMAHLP-2 crisp*, luego se usa el modelo siguiente para calcular el grado de cumplimiento de la solución crisp:

*Grado de cumplimiento CMAHLP-2*

$$\lambda^* = \frac{1}{2} \left( \frac{n}{3} \frac{1}{3} \left( \prod_{k=1}^p (\mu_{\mathcal{C}^0}(C_{ik}) \wedge \mu_{\mathcal{Z}^0}(Z_{ik})) + \right. \right. \\ \left. \prod_{k=1}^p \prod_{l=1}^p (\mu_{\mathcal{C}^0}(C_{kl}) \wedge \mu_{\mathcal{Y}^0}(Y_{kl}^i)) + \prod_{l=1}^p \prod_{j=1}^n (\mu_{\mathcal{C}^0}(C_{lj}) \wedge \mu_{\mathcal{X}^0}(X_{lj}^i)) \right) + \quad (1)$$

$$\left. \prod_{k=1}^p (\mu_{\mathcal{F}^0}(F_k)) H_k \right)$$

$$\text{s.a.} \quad \mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{Z}^0}(Z_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{Y}^0}(Y_{kl}^i), \mu_{\mathcal{C}^0}(C_{lj}), \mu_{\mathcal{X}^0}(X_{lj}^i), \mu_{\mathcal{F}^0}(F_k) \in [0, 1], \\ \forall i, j, k, l, \quad (2)$$

Puesto que  $\lambda^* \in [0,1]$ , afirmamos que el CMAHLP borroso generaliza el CMAHLP crisp.

### ✓ Algoritmo de cálculo del grado de cumplimiento CMAHLP-2

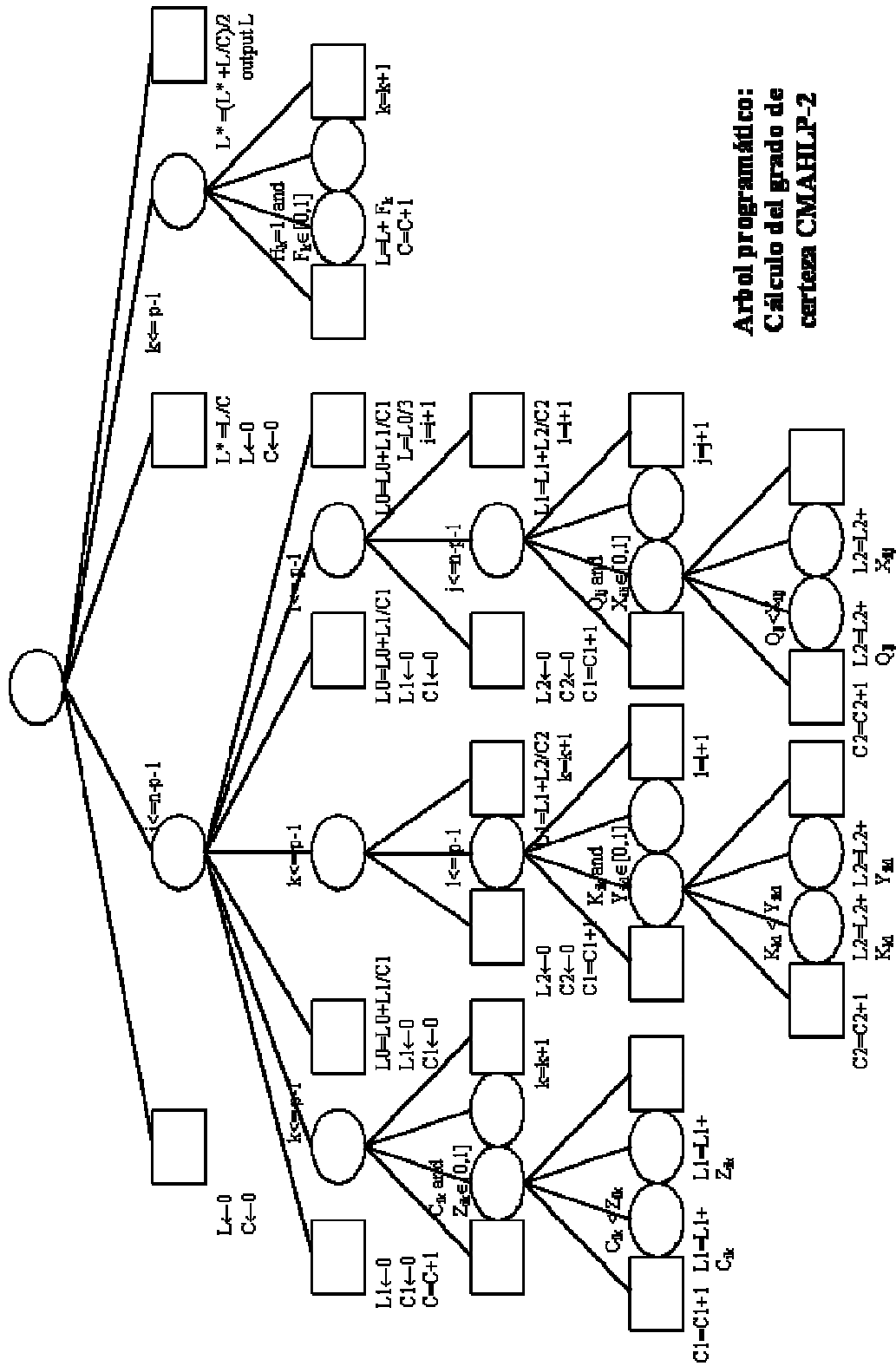
Datos de entrada:

- La matriz  $C_{(n-p) \times p} = [\mu_{C^0}(C_{ik})]_{(n-p) \times p}$
- La matriz  $K_{p \times p} = [\mu_{C^0}(C_{kl})]_{p \times p}$
- La matriz  $Q_{p \times (n-p)} = [\mu_{C^0}(C_{lj})]_{p \times (n-p)}$
- La matriz  $X_{(n-p) \times p \times (n-p)} = [\mu_{X^0}(X_{lj}^i)]_{(n-p) \times p \times (n-p)}$
- La matriz  $Z_{(n-p) \times p} = [\mu_{Z^0}(Z_{ik})]_{(n-p) \times p}$
- El vector  $F_p = [\mu_{F^0}(F_k)]_p$

VARIABLES UTILIZADAS:

- L, L0, L1 y L2 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C, C1 y C2 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- i, j, k y l son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red.
- p almacena el número de nodos centros.

El árbol programático del cálculo del grado de cumplimiento de la solución del problema CMAHLP-2 se presenta como sigue



**Arbol programático:  
Cálculo del grado de  
carteza CMAHLP-2**



## ✓ Pseudocódigo: Grado de cumplimiento CMAHLP-2

```

L←0
C←0
for i=0 to n-p-1
  L0←0
  L1←0
  C1←0
  C=C+1
  for k=0 to p-1
    if  $C_{ik}, Z_{ik} \in [0, 1]$ 
      C1=C1+1
      if  $C_{ik} < Z_{ik}$ 
        L1=L1+Cik
      else
        L1=L1+Zik
    k=k+1
  L0=L1/C1
  L1←0
  C1←0
  for k=0 to p-1
    L2←0
    C2←0
    C1=C1+1
    for l=0 to p-1
      if  $K_{kl}, Y_{ikl} \in [0, 1]$ 
        C2=C2+1
        if  $K_{kl} < Y_{ikl}$ 
          L2=L2+Kkl
        else
          L2=L2+Yikl
      l=l+1
    L1=L1+L2/C2
    k=k+1
  L0=L0+L1/C1
  L1←0
  C1←0
  for l=0 to p-1
    L2←0
    C2←0
    C1=C1+1
    for j=0 to n-p-1
      if  $Q_{lj}, X_{ilj} \in [0, 1]$ 
        C2=C2+1
        if  $Q_{lj} < X_{ilj}$ 

```

```

                                L2=L2+Q1j
                                else
                                L2=L2+Xi1j
                                j=j+1
                                L1=L1+L2/C2
                                l=l+1
                                L0=L0+L1/C1
                                L=L+L0/3
                                i=i+1
L* = L/C
L←0
C←0
for k=0 to p-1
    if Fk∈ [0,1]
        L=L+Fk
        C=C+1
    k=k+1
L* = (L* + L/C)/2
output L*
```

Tiempo computacional:  $n^2 + np + n + 1$  sea  $O(n^2)$

### ✓ Implementación del algoritmo en C++

```

//=====
//Fuzzy CMAHLP-2: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Monday,November 22nd, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main() {
    int n,p,C=0,C1,C2;
    double L=0,L0,L1,L2;
    cout << "Degree of Certainty (CMAHLP-2)\n\n";
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of hubs: ";
    cin >> p;

    //Declaring input data
    double C[n-p][p],Z[n-p][p],K[p][p],Y[n-p][p][p];
    double Q[p][n-p],X[n-p][p][n-p],F[p];
```

```

//Entering input data
    //Will be written later
//Processing data
for (int i=0; i<=n-p-1;){
    L0=0;
    L1=0;
    C1=0;
    ++C;
    for (int k=0; k<=p-1;){
        if (C[i][k]>=0 && C[i][k]<=1 &&
            Z[i][k]>=0 && Z[i][k]<=1){
            ++C1;
            if (C[i][k]< Z[i][k])
                L1 += C[i][k];
            else
                L1 += Z[i][k];
        }
        ++k;
    }
    L0 = L1/C1;
    L1=0;
    C1=0;
    for (int k=0; k<=p-1;){
        L2=0;
        C2=0;
        ++C1;
        for (int l=0; l<=p-1;){
            if (K[k][l]>=0 && K[k][l]<=1 &&
                Y[i][k][l]>=0 && Y[i][k][l]<=0){
                ++C2;
                if (K[k][l]<Y[i][k][l])
                    L2 += K[k][l];
                else
                    L2 += Y[i][k][l];
            }
            ++l;
        }
        L1 += L2/C2;
        ++k;
    }
    L0 += L1/C1;
    L1=0;
    C1=0;
    for (int l=0; l<=p-1;){
        L2=0;
        C2=0;
        ++C1;
        for (int j=0; j<=n-p-1;){

```

```

        if (Q[l][j]>=0 && Q[l][j]<=1 &&
            X[i][l][j]>=0 && X[i][l][j]<=0){
            ++C2;
            if (Q[l][j]<X[i][l][j])
                L2 += Q[l][j];
            else
                L2 += X[i][l][j];
        }
        ++j;
    }
    L1 += L2/C2;
    ++l;
}
L0 += L1/C1;
L += L0/3;
++i;
}
double Lambda = L/C;
L=0;
C=0;
for (int k=0; k<=p-1;){
    if(F[k]>=0 && F[k]<=1){
        L += F[k];
        ++C;
    }
    ++k;
}
Lambda = (Lambda + L/C)/2;
cout << "\nThe degree of certainty is " << Lambda;
return 0;
}

```

### III.3.1.2.2. Problema UMAHLP borroso

Desarrollamos el modelo UMAHLP borroso a partir del modelo UMHLP crisp siguiente desarrollado en [10]:

*Problema UMAHLP-1,2,3*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (1)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (2)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (3)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (4)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l, \quad (5)$$

$$Z_{hh} = O_h H_h \quad (6)$$

$$X_{hh}^i = W_{ih} H_h \quad (7)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, i \neq k, \quad (8)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j \quad (9)$$

La función objetivo de este modelo siendo igual a la del CMAHLP-2, suponemos la imprecisión en los mismos datos como en este último modelo. Consecuentemente, así formulamos el modelo UMAHLP borroso:

*UMAHLP borroso:*

$$\begin{aligned} (Z^* | \lambda^*) \cong \min \sum_i \left[ \chi \sum_k (C_{ik} | \mu_{\mathcal{C}_0}(C_{ik})) (Z_{ik} | \mu_{\mathcal{Z}_0}(Z_{ik})) + \right. \\ \left. \alpha \sum_k \sum_l (C_{kl} | \mu_{\mathcal{C}_0}(C_{kl})) (Y_{kl}^i | \mu_{\mathcal{Y}_0}(Y_{kl}^i)) + \delta \sum_l \sum_j (C_{lj} | \mu_{\mathcal{C}_0}(C_{lj})) (X_{lj}^i | \mu_{\mathcal{X}_0}(X_{lj}^i)) \right] + \\ \sum_k (F_k | \mu_{\mathcal{F}_0}(F_k)) H_k \quad (1) \end{aligned}$$

$$\text{s.a.} \quad \sum_k Z_{ik} \cong \sum_j W_{ij} \quad \forall i, \quad (2)$$

$$\sum_l X_{lj}^i \cong W_{ij} \quad \forall i, j, \quad (3)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} \cong 0 \quad \forall i, k, \quad (4)$$

$$X_{ij}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l, \quad (5)$$

$$Z_{hh} \cong O_h H_h \quad (6)$$

$$X_{hh}^i \cong W_{ih} H_h \quad (7)$$

$$Z_{ik} \cong \sum_j W_{ij} H_k \quad \forall i, k, i \neq k, \quad (8)$$

$$\sum_i X_{lj}^i \cong \sum_i W_{ij} H_l \quad \forall j, l \neq j \quad (9)$$

El modelo se resuelve en dos pasos. Primero se calcula el coste óptimo crisp  $Z^*$  usando el modelo sin tomar en cuenta los grados de pertenencia, es decir el UMAHLP crisp. Luego se calcula el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el paso 1 usando los grados de pertenencia de la función objetivo de la siguiente manera:

*Grado de cumplimiento de la solución UMAHLP:*

$$\lambda^* = \frac{1}{2} \left[ \prod_{i=1}^{n-p} \frac{1}{3} \left( \prod_{k=1}^p (\mu_{\mathcal{C}^0}(C_{ik}) \wedge \mu_{\mathcal{Z}^0}(Z_{ik})) + \prod_{k=1}^p (\mu_{\mathcal{C}^0}(C_{kl}) \wedge \mu_{\mathcal{Y}^0}(Y_{kl}^i)) + \prod_{l=1}^{n-p} \prod_{j=1}^p (\mu_{\mathcal{C}^0}(C_{lj}) \wedge \mu_{\mathcal{X}^0}(X_{lj}^i)) \right) + \prod_{k=1}^p (\mu_{\mathcal{F}^0}(F_k)) H_k \right] \quad (1)$$

$$\text{s.a.} \quad \mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{Z}^0}(Z_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{Y}^0}(Y_{kl}^i), \mu_{\mathcal{C}^0}(C_{lj}), \mu_{\mathcal{X}^0}(X_{lj}^i), \mu_{\mathcal{F}^0}(F_k) \in [0, 1] \quad \forall i, j, k, l, \quad (2)$$

Puesto que  $\lambda^* \in [0, 1]$ , afirmamos que el UMAHLP borroso generaliza el crisp.

Este modelo siendo igual al del CMAHLP-2 para el cálculo del grado de cumplimiento, se resuelva con el mismo algoritmo presentado para el CMAHLP-2.

### III.3.1.2.3. El problema UMApHMP borroso

#### ✓ Formulación delo modelo

Desarrollamos el modelo UMApHMP borroso partiendo del siguiente modelo crisp desarrollado en [10]:

*Problema UMApHMP-1,2,3,5*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] \quad (1)$$

$$\text{s.a.} \quad \sum_k H_k = p \quad (2)$$

$$\sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (5)$$

$$X_{ij}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (8)$$

$$Z_{hh} = O_h H_h \quad (9)$$

$$X_{hh}^i = W_{ih} H_h \quad (10)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, i \neq k \quad (11)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (12)$$

En la función objetivo de esta formulación supongamos la incertidumbre en los mismos datos como en el problema CMAHLP-2 borroso con la diferencia que aquí no utilizaremos la variable  $F_k$ , el coste fijo de establecer un centro al nodo  $k$  por todo  $k \in N$ .

El modelo del problema borroso de localización de p-mediana central no capacitado con asignación múltiple:

UMApHMP borroso:

$$(Z^* | \lambda^*) \equiv \min \sum_i \left[ \chi \sum_k (C_{ik} | \mu_{\mathcal{C}^0}(C_{ik})) (Z_{ik} | \mu_{\mathcal{Z}^0}(Z_{ik})) + \alpha \sum_k \sum_l (C_{kl} | \mu_{\mathcal{C}^0}(C_{kl})) (Y_{kl}^i | \mu_{\mathcal{Y}^0}(Y_{kl}^i)) + \delta \sum_l \sum_j (C_{lj} | \mu_{\mathcal{C}^0}(C_{lj})) (X_{lj}^i | \mu_{\mathcal{X}^0}(X_{lj}^i)) \right]$$

$$\text{s.a.} \quad \sum_k H_k = p \quad (2)$$

$$\sum_k Z_{ik} \cong \sum_j W_{ij} \quad \forall i, \quad (3)$$

$$\sum_l X_{lj}^i \cong W_{ij} \quad \forall i, j, \quad (4)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} \cong 0 \quad \forall i, k, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \cong 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l. \quad (6)$$

$$Z_{hh} \cong O_h H_h \quad (7)$$

$$X_{hh}^i \cong W_{ih} H_h \quad (8)$$

$$Z_{ik} \cong \sum_j W_{ij} H_k \quad \forall i, k, i \neq k \quad (9)$$

$$\sum_i X_{lj}^i \cong \sum_i W_{ij} H_l \quad \forall j, l \neq j. \quad (10)$$

El modelo se utiliza en dos pasos. Primero se calcula  $Z^*$  crisp es decir el modelo que acabamos de presentar sin tomar en cuenta los grados de pertenencia. Luego se calcula el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el primer paso usando la función objetivo del modelo que acabamos de presentar tomando en cuenta solamente los grados de pertenencia de la siguiente manera:

Grado de cumplimiento de la solución UMapHMP:

$$\lambda^* = \mathbf{M}_i \frac{1}{3} \left( \mathbf{M}_k (\mu_{\mathcal{C}^0}(C_{ik}) \wedge \mu_{\mathcal{Z}^0}(Z_{ik})) + \mathbf{M}_k \mathbf{M}_l (\mu_{\mathcal{C}^0}(C_{kl}) \wedge \mu_{\mathcal{Y}^0}(Y_{kl}^i)) + \right. \quad (1)$$

$$\left. \mathbf{M}_l \mathbf{M}_j (\mu_{\mathcal{C}^0}(C_{lj}) \wedge \mu_{\mathcal{X}^0}(X_{lj}^i)) \right)$$

$$\text{s.a.} \quad \mu_{\mathcal{C}^0}(C_{ik}), \mu_{\mathcal{Z}^0}(Z_{ik}), \mu_{\mathcal{C}^0}(C_{kl}), \mu_{\mathcal{Y}^0}(Y_{kl}^i), \mu_{\mathcal{C}^0}(C_{lj}), \mu_{\mathcal{X}^0}(X_{lj}^i) \in [0, 1] \quad \forall i, j, k, l. \quad (2)$$



Puesto que  $\lambda^* \in [0,1]$ , afirmamos que el UMApHMP borroso generaliza el UMApHMP crisp.

### ✓ Algoritmo de cálculo del grado de cumplimiento UMApHMP

Datos de entrada:

- La matriz  $C_{(n-p) \times p} = [\mu_{\lambda^*}(C_{ik})]_{(n-p) \times p}$
- La matriz  $K_{p \times p} = [\mu_{\lambda^*}(C_{kl})]_{p \times p}$
- La matriz  $Q_{p \times (n-p)} = [\mu_{\lambda^*}(C_{lj})]_{p \times (n-p)}$
- La matriz  $X_{(n-p) \times p \times (n-p)} = [\mu_{\lambda^*}(X_{lj}^i)]_{(n-p) \times p \times (n-p)}$
- El vector  $Z_{(n-p) \times p} = [\mu_{\lambda^*}(Z_{ik})]_{(n-p) \times p}$

Variables utilizadas:

- L, L0, L1 y L2 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C, C1 y C2 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- i, j, k y l son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red.
- p almacena el número de nodos centros.

El algoritmo se presenta como sigue:



## ✓ Pseudocódigo: Grado de cumplimiento UMAPHMP

```

L←0
C←0
for i=0 to n-p-1
  C=C+1
  L0=0
  L1=0
  C1=0
  for k=0 to p-1
    if  $C_{ik}, Z_{ik} \in [0, 1]$ 
      C1=C1+1
      if  $C_{ik} < Z_{ik}$ 
        L1=L1+Cik
      else
        L1= L1+Zik
    k=k+1
  L0=L1/C1
  L1=0
  C1=0
  for k=0 to p-1
    C1=C1+1
    L2=0
    C2=0
    for l=0 to p-1
      if  $K_{kl}, Y_{ikl} \in [0, 1]$ 
        C2=C2+1
        if  $K_{kl} < Y_{ikl}$ 
          L2=L2+ Kkl
        else
          L2=L2+ Yikl
      l=l+1

```

```

        L1=L1+L2/C2
        k=k+1
L0=L0+L1/C1
L1=0
C1=0
for l=0 to p-1
    C1=C1+1
    L2=0
    C2=0
    for j=0 to n-p-1
        if  $Q_{1j}, X_{i1j} \in [0, 1]$ 
            C2=C2+1
            if  $Q_{1j} < X_{i1j}$ 
                L2=L2+ $Q_{1j}$ 
            else
                L2=L2+ $X_{i1j}$ 
        j=j+1
    L1=L1+L2/C2
    l=l+1
L0 = L0+L1/C1
L = L+L0/3
i=i+1
output L/C

```

Tiempo computacional :  $np(n+p+1)$

✓ **Implementación del algoritmo en C++**

```
//=====
//Fuzzy UMapHMP: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Monday,November 22nd, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main(){
    int n,p,C=0,C1,C2;
    double L=0,L0,L1,L2;
    cout << "Degree of Certainty (UMApHMP)\n\n";
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of hubs: ";
    cin >> p;

    //Declaring input data
    double C[n-p][p],Z[n-p][p],K[p][p],Y[n-p][p][p];
    double Q[p][n-p], X[n-p][p][n-p];

    //Entering input data
        //will be written later
    //Processing input data
    for (int i=0; i<=n-p-1;){
        ++C;
        L0=0;
        L1=0;
        C1=0;
        for (int k=0; k<=p-1;){
```

```

    if (C[i][k] >= 0 && C[i][k] <= 1 &&
        Z[i][k] >= 0 && Z[i][k] <= 1) {
        ++C1;
        if (C[i][k] < Z[i][k])
            L1 += C[i][k];
        else
            L1 += Z[i][k];
    }
    ++k;
}
L0=L1/C1;
L1=0;
C1=0;
for (int k=0; k<=p-1;){
    ++C1;
    L2=0;
    C2=0;
    for (int l=0; l<=n-p;){
        if (K[k][l] >= 0 && K[k][l] <= 1 &&
            Y[i][k][l] >= 0 && Y[i][k][l] <= 1) {
            ++C2;
            if (K[k][l] < Y[i][k][l])
                L2 += K[k][l];
            else
                L2 += Y[i][k][l];
        }
        ++l;
    }
    L1 += L2/C2;
    ++k;
}
L0 += L1/C1;
L1=0;

```

```

C1=0;
for (int l=0; l<=p-1;){
    ++C1;
    L2=0;
    C2=0;
    for (int j=0; j<=n-p-1;){
        if (Q[l][j]>=0 && Q[l][j]<=1 &&
            X[i][l][j]>=0 && X[i][l][j]<=1){
            ++C2;
            if (Q[l][j]<X[i][l][j])
                L2 += Q[l][j];
            Else
                L2 += X[i][l][j];
        }
        ++j;
    }
    L1 += L2/C2;
    ++l;
}
L0 += L1/C1;
L += L0/3;
++i;
}
cout << "\nThe degree of certainty is " << L/C;
return 0;
}

```

### III.3.2. Problema CMAHLP

#### III.3.2.1. Problema CMAHLP crisp

Además de las definiciones y formulaciones presentadas en [10], se define el *CMAHLP* más precisamente de la manera siguiente [32]: Dado un grafo completo  $G=(N,A)$  donde  $N=\{1,2,\dots,n\}$  es el conjunto de nodos y  $A=N\times N$ , un conjunto de aristas dirigidas. Hay un requisito del envío de tráfico de  $W_{ij}$  asociado a cada par ordenado de nodos  $(i,j)\in A$ ;  $W_{ij}$  es el volumen del envío de tráfico asociado. No se asume que  $W_{ij}=W_{ji}$  ni que  $W_{ii}=0$ . También dado  $C_{ij}$ , el coste de enviar una unidad de tráfico directamente de  $i$  a  $j$ .  $C_{ij}$  es siempre proporcional a la distancia entre  $i$  y  $j$  y es generalmente euclideo. Cada flujo tiene tres componentes separados: la colección (nodo origen – centro), transferencia (inter – centro) y distribución (centro – nodo destino). Cada componente tiene el coeficiente de coste por unidad, respectivamente  $\chi$ ,  $\alpha$ ,  $\delta$ . Así, el coste del flujo por unidad para la colección es  $\chi C_{ik}$ , donde  $i$  es un nodo no centro y  $k$  es un centro.

Similarmente, la transferencia inter-centro tiene el coste  $\alpha C_{kl}$  donde  $k, l$  son ambos centros y la distribución tiene el coste  $\delta C_{lj}$  donde  $l$  es un centro y  $j$  es un nodo no centro. A menudo, debido a las eficiencias de transferencia inter-centro,  $\alpha < 1$  es un factor de descuento de la transferencia, mientras que  $\chi, \delta = 1$ . En cualquier caso, se asume que  $\chi, \delta > \alpha$ .

No hay un algoritmo de solución o estudio computacional del *CMAHLP* en la literatura. El problema de  $p$ -mediana centro no capacitada con asignación múltiple (*UMApHMP*), que es quizás el problema más relacionado, se ha investigado profundamente. El *UMApHMP* fue estudiado por Aykin (1993) que examinó un modelo con capacidades sobre las aristas utilizando un enfoque exacto y heurístico. Campbell [14] desarrolló las formulaciones para varias variantes de *UMApHMP*. En [15], Campbell define dos heurísticas (ALLFLO y MAXFLO) para solucionar este problema. En [94], Skorin-Kapov et al. facilitan un programa lineal entero mixto que ajustan las formulaciones dadas por Campbell y utiliza pocas variables. Ernst y Krishnamoorthy [38] dan una idea de una heurística utilizando caminos más cortos y obtienen soluciones exactas utilizando dos métodos, la enumeración explícita y el



"Branch and Bound". Además, Ernst y Krishnamoorthy [38] desarrollan un programa lineal entero mixto con  $O(n^3)$  variables.

Además de las formulaciones presentadas en [10], en [32] se estudia el Problema de Localización de Centro Capacitado con Asignación múltiple (que se refiere como *CMAHLP*). La elección del problema se motiva por una aplicación de reparto postal. En el reparto postal, el volumen del correo que los centros pueden clasificar se limita por una restricción del tiempo, que limita el volumen del correo que se puede clasificar en los centros de clasificación de correos. Por eso, se introduce las restricciones de capacidad en los nodos centros. Estas restricciones se aplican solamente sobre el tráfico que llega al centro directamente desde los nodos no centros, como es donde la clasificación tiene lugar.

Una variante de este problema se presenta cuando existen las capacidades sobre ambas, la colección y la distribución. En el *CMAHLP* hay también un coste asociado al establecimiento de un nodo como centro. Consecuentemente, el número de los centros para cualquier problema dado necesita derivarse por un modelo matemático de decisión.

Un programa lineal entero mixto para el *CMAHLP* fue dado por Campbell [14], que incluía restricciones como  $X_{ijkm} \leq Y_k$  y  $X_{ijkm} \leq Y_m \quad \forall i, j, k, m$ . Estas restricciones pueden remplazarse por sus agregaciones, tal como presentadas en las formulaciones de *UMApHMP* de Skorin-kapov et al. [94]. El MILP para el *CMAHLP* dado por Campbell restringe que la suma del tráfico que entra en cualquier centro vía la colección y la transferencia sea menos que la capacidad de ese centro. En la versión de [32] se aplica una restricción sobre el volumen de tráfico que entra en un centro vía la colección. Para el modelo considerado en [32], se remplace simplemente

$$\sum_{i,j} W_{ij} \left[ \sum_m (X_{ijkm} + X_{ijmk}) - X_{ijkk} \right] \leq \Gamma_k Y_k \quad \forall k \in n \quad (1)$$

por la restricción siguiente

$$\sum_i \sum_j W_{ij} \sum_m X_{ijkm} \leq \Gamma_k Y_k \quad \forall k \in N \quad (2)$$

En [32] la formulación de Campbell se denota como *CMAHLP-C* y se presenta como sigue:

*Formulación CMAHLP-C*

$$\min \sum_i \sum_j \sum_k \sum_m W_{ij} C_{ijkm} X_{ijkm} + \sum_k F_k Y_k \quad (3)$$

$$\text{s.a.} \quad \sum_k \sum_m X_{ijkm} = 1 \quad \forall i, j \in N, \quad (4)$$

$$\sum_m X_{ijkm} \leq Y_k \quad \forall i, j, k \in N, \quad (5)$$

$$\sum_k X_{ijkm} \leq Y_m \quad \forall i, j, m \in N, \quad (6)$$

$$\sum_i \sum_j W_{ij} \sum_m X_{ijkm} \leq \Gamma_k Y_k \quad \forall k \in N, \quad (7)$$

$$Y_k \in \{0, 1\} \quad \forall k \in N, \quad (8)$$

$$0 \leq X_{ijkm} \leq 1 \quad \forall i, j, k, m \in N \quad (9)$$

donde la variable  $X_{ijkm}$  denota la fracción del flujo que viaja del nodo  $i$  al nodo  $j$  vía los centros localizados en  $k$  y  $m$ , la variable  $Y_k$ ,  $\forall k \in N$  se define por

$$Y_k = \begin{cases} 1 & \text{si el nodo } k \text{ es un centro} \\ 0 & \text{si no} \end{cases}$$

El coste por unidad del flujo  $X_{ijkm}$  se define por

$$C_{ijkm} = \chi C_{ik} + \alpha C_{km} + \delta C_{mj} \quad \forall i, j, k, m \in N$$

$F_k$  es el coste fijo asociado al establecimiento de un centro en el nodo  $k$  y  $\Gamma_k$  es su capacidad.

La ecuación (4) asegura que el flujo total ( $W_{ij}$ ) de  $i$  a  $j$  está transferido, mientras que las ecuaciones (5) y (6) garantizan que las transferencias ocurren solamente vía los centros. La ecuación (7) asegura que las capacidades se están tomando en cuenta.

Con el *CMAHLP* los centros están capacitados y hay un coste fijo asociado al establecimiento de un nodo dado como centro. Así, el modelo no solo decide sobre la localización óptima de los centros sino también sobre el número de centros requeridos. Se nota también que si la localización de los centros se conoce a priori, entonces el *CMAHLP* se reduce a un problema de flujo multiproducto.

En [32] se presenta una nueva formulación MILP para el *CMAHLP*. Esta formulación utiliza más pocas variables y restricciones que la de Campbell presentada arriba o el *UMApHMP* muy relacionado presentado por Ernst y Krishnamoorthy [38]. Como punto de partida, se refiere a la formulación *UMApHMP* publicada por Ernst y Krishnamoorthy [38]. Se definen los parámetros del problema como sigue:

$C_{ij}$	coste por unidad de flujo de $i$ a $j$ ;
$W_{ij}$	flujo de $i$ a $j$ ;
$F_k$	coste de establecimiento de un centro en el nodo $k$ ;
$\chi, \alpha, \delta$	coeficientes respectivamente de la colección, transferencia y distribución;
$N$	número de nodos;
$\Gamma_k$	capacidad del centro $k$ ;
$p$	número de nodos que se deben escoger como centros;

Las variables de la formulación son:

$Z_{ik}$	flujo del nodo $i$ al centro $k$ ;
$Y_{kl}^i$	flujo del nodo $i$ vía los centros $k$ y $l$ ;
$X_{lj}^i$	flujo del nodo $i$ vía el centro $l$ ;
$H_k = \begin{cases} 1 & \text{si el nodo } k \text{ es un centro} \\ 0 & \text{si no} \end{cases}$	

Con cambios de notación menores y las definiciones de arriba, Ernst y Krishnamoorthy [38] formularon el *UMApHMP* como sigue:

*Formulación UMApHMP-N*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] \quad (10)$$

$$\text{s.a.} \quad \sum_k H_k = p \quad (11)$$

$$\sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (12)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (13)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (14)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (15)$$

$$X_{lj}^i \leq W_{ij} H_l \quad \forall i, l, k, \quad (16)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (17)$$

La ecuación (11) especifica el número de centros mientras que las ecuaciones (129 y (14) representan las ecuaciones de divergencia para un problema de flujo en la red. Las ecuaciones (15) y (16) aseguran que ningún flujo viaja directamente entre nodos no centros.

La formulación *UMApMHP-N* tiene  $(2n^3 + n^2 + n)$  variables de las cuales  $n$  son binarias y  $(n^3 + 3n^2 + n + 1)$  son restricciones lineales.

La formulación *CMAHLP* que se desarrolla en [32] determina implícitamente el número de centros requeridos para la solución óptima. Este se realiza añadiendo un coste fijo para establecer un centro en el nodo  $k$ . Así, la función objetivo vuelve a ser

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (18)$$

También la ecuación (11) ya no es necesaria y se puede borrar. Para tomar en cuenta las capacidades de los centros, se añade el siguiente conjunto de restricciones:

$$\sum_i Z_{ik} \leq \Gamma_k H_k \quad \forall k \quad (19)$$

Este basta para especificar completamente el *CMAHLP*. No obstante, en la práctica, agregando la ecuación (16) de la siguiente manera

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall l, j \quad (20)$$

hace que la solución óptima se encuentre más eficientemente. Esta nueva formulación tiene  $(4n^2 + 2n)$  restricciones y  $(2n^3 + n^2 + n)$  variables. Así, una formulación MILP del *CMAHLP* es

*Formulación CMAHLP-N*

$$\min \sum_i \left[ \chi \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^i + \delta \sum_l \sum_j C_{lj} X_{lj}^i \right] + \sum_k F_k H_k \quad (21)$$

$$\text{s.a.} \quad \sum_k Z_{ik} = \sum_j W_{ij} \quad \forall i, \quad (22)$$

$$\sum_l X_{lj}^i = W_{ij} \quad \forall i, j, \quad (23)$$

$$\sum_i Z_{ik} \leq \Gamma_k H_k \quad \forall k, \quad (24)$$

$$\sum_l Y_{kl}^i + \sum_j X_{kj}^i - \sum_l Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k, \quad (25)$$

$$Z_{ik} \leq \sum_j W_{ij} H_k \quad \forall i, k, \quad (26)$$

$$\sum_i X_{lj}^i \leq \sum_i W_{ij} H_l \quad \forall l, j, \quad (27)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (28)$$

Esta formulación tiene substancialmente más pocas variables que una formulación del *CMAHLP* que se basa en [14]. Campbell muestra el uso de  $O(n^4)$  variables mientras que el *CMAHLP-N* utiliza  $O(n^3)$  variables. Pero deseando resolver problemas más grandes, se explora una segunda formulación en la cual se reduce el número de variables haciendo la substitución de la forma

$$Z_{ik} = \sum_l Y_{kl}^i \quad \forall i, k \quad (29)$$

Se reduce también el número de restricciones borrando las ecuaciones (22) y (26) y substituyendo la ecuación (27) por

$$\sum_i \sum_k Y_{kl}^i \leq \sum_i \sum_j W_{ij} H_l \quad \forall l \quad (30)$$

Entonces, la nueva formulación de [32] se presenta como sigue:

*Formulación CMAHLP-F*

$$\min \sum_i \sum_k \sum_l (\chi C_{ik} + \alpha C_{kl}) Y_{kl}^i + \sum_i \sum_j \sum_l \delta C_{lj} X_{lj}^i + \sum_k F_k H_k \quad (31)$$

$$\text{s.a.} \quad \sum_l X_{lj}^i = W_{ij} \quad \forall i, j \quad (32)$$

$$\sum_i \sum_l Y_{kl}^i \leq \Gamma_k H_k \quad \forall k, \quad (33)$$

$$\sum_i \sum_k Y_{kl}^i \leq \sum_i \sum_j W_{ij} H_l \quad \forall l, \quad (34)$$

$$\sum_k Y_{kl}^i = \sum_j X_{lj}^i \quad \forall i, l, \quad (35)$$

$$X_{lj}^i, Y_{kl}^i \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (36)$$

Esta formulación tiene  $(2n^3 + n)$  variables y  $(2n^2 + n)$  restricciones, que es aproximadamente la mitad del número de restricciones del *CMAHLP-N*.

**III.3.2.2. Problema CMAHLP borroso****✓ Formulación del modelo**

Además del *CMAHLP-2 borroso* que hemos desarrollado más arriba, presentamos otro que desarrollamos a partir de la siguiente formulación crisp desarrollada en [32]:

*Formulación CMAHLP-F*

$$\min \sum_i \sum_k \sum_l (\chi C_{ik} + \alpha C_{kl}) Y_{kl}^i + \sum_i \sum_j \sum_l \delta C_{lj} X_{lj}^i + \sum_k F_k H_k \quad (1)$$

$$\text{s.a.} \quad \sum_l X_{lj}^i = W_{ij} \quad \forall i, j \quad (2)$$

$$\sum_i \sum_l Y_{kl}^i \leq \Gamma_k H_k \quad \forall k, \quad (3)$$

$$\sum_i \sum_k Y_{kl}^i \leq \sum_i \sum_j W_{ij} H_l \quad \forall l, \quad (4)$$

$$\sum_k Y_{kl}^i = \sum_j X_{lj}^i \quad \forall i, l, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i \geq 0 \quad H_k \in \{0,1\} \quad \forall i, j, k, l. \quad (6)$$

En la función objetivo de esta formulación, suponemos la incertidumbre en los mismos datos como en el modelo precedente con la excepción que  $Y_{kl}^i$  siendo el flujo del nodo  $i$  vía los

centros  $k$  y  $l$  por todo  $i, k, l \in N$  y que en esta formulación  $\sum_l Y_{kl}^i = Z_{ik}, \forall i, k$ , se utilizará

$\sum_l Y_{kl}^i$  en vez de  $Z_{ik}$ .

Con esta precisión, presentamos el segundo modelo del problema borroso de localización de centro capacitado con asignación múltiple que denominamos *CMAHLP-F borroso*:

*CMAHLP-F borroso*

$$(Z^* | \lambda^*) \cong \min \sum_i \sum_k \sum_l \left( \chi(C_{ik} | \mu_{\varrho_0}(C_{ik})) + \alpha(C_{kl} | \mu_{\varrho_0}(C_{kl})) \right) Y_{kl}^i + \sum_i \sum_j \sum_l \delta(C_{lj} | \mu_{\varrho_0}(C_{lj})) X_{lj}^i + \sum_k (F_k | \mu_{\rho_0}(F_k)) H_k \quad (1)$$

$$\text{s.a.} \quad \sum_l X_{lj}^i \cong W_{ij} \quad \forall i, j \quad (2)$$

$$\sum_i \sum_l Y_{kl}^i \leq \Gamma_k H_k \quad \forall k, \quad (3)$$

$$\sum_i \sum_k Y_{kl}^i \leq \sum_i \sum_j W_{ij} H_l \quad \forall l, \quad (4)$$

$$\sum_k Y_{kl}^i \cong \sum_j X_{lj}^i \quad \forall i, l, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i \geq 0 \quad H_k \in \{0, 1\} \quad \forall i, j, k, l. \quad (6)$$

Primero calculamos el coste óptimo  $Z^*$  utilizando el modelo sin los grados de pertenencia, es decir el modelo CMAHLP-F crisp, luego utilizamos los grados de pertenencia en la fórmula siguiente para calcular el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el paso 1:

*Grado de cumplimiento CMAHLP-F*

$$\lambda^* = \frac{1}{3} \left[ \text{MMM}_{i \ k \ l} \left( \frac{1}{2} (\mu_{\varrho_0}(C_{ik}) + \mu_{\varrho_0}(C_{kl})) \wedge \mu_{\varrho_0}(Y_{kl}^i) \right) + \text{MMM}_{i \ j \ l} \left( \mu_{\varrho_0}(C_{lj}) \wedge \mu_{\varrho_0}(X_{lj}^i) \right) + \text{M}_k \mu_{\rho_0}(F_k) H_k \right] \quad (1)$$

$$\text{s.a.} \quad \mu_{\varrho_0}(C_{ik}), \mu_{\varrho_0}(C_{kl}), \mu_{\varrho_0}(Y_{kl}^i), \mu_{\varrho_0}(C_{lj}), \mu_{\varrho_0}(X_{lj}^i), \mu_{\rho_0}(F_k) \in [0, 1] \quad \forall i, j, k, l. \quad (2)$$

Si  $\lambda^* = 1$ , entonces se habrá tratado de un CMAHLP-F crisp; si, por el contrario  $0 < \lambda^* < 1$ , entonces se habrá tratado de un CMAHLP-F borroso. Así pues, afirmamos que el CMAHLP borroso es una generalización del CMAHLP-F crisp.

#### ✓ Algoritmo de cálculo del grado de cumplimiento CMAHLP-F

Datos de entrada:

- La matriz  $C_{(n-p) \times p} = [\mu_{\mathcal{C}}^{\%}(C_{ik})]_{(n-p) \times p}$
- La matriz  $K_{p \times p} = [\mu_{\mathcal{C}}^{\%}(C_{kl})]_{p \times p}$
- La matriz  $Q_{p \times (n-p)} = [\mu_{\mathcal{C}}^{\%}(C_{lj})]_{p \times (n-p)}$
- La matriz  $X_{(n-p) \times p \times (n-p)} = [\mu_{\mathcal{X}}^{\%}(X_{lj}^i)]_{(n-p) \times p \times (n-p)}$
- El vector  $F_p = [\mu_{\mathcal{F}}^{\%}(F_k)]_p$

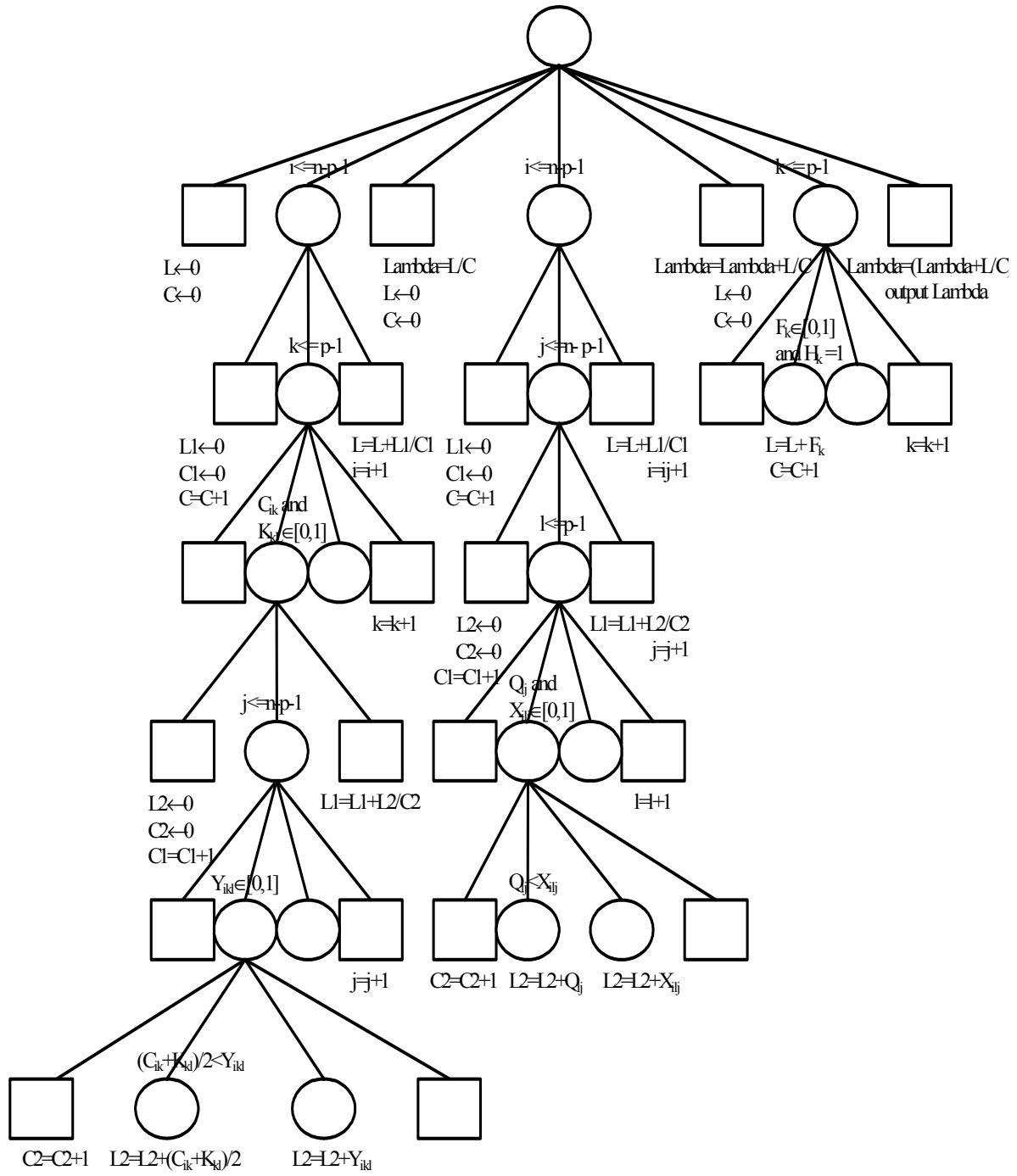
VARIABLES UTILIZADAS:

- L, L0, L1 y L2 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C, C1 y C2 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- i, j, k y l son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red.
- p almacena el número de nodos centros.

El árbol programático del cálculo del grado de cumplimiento de una solución CAMHLP-F se

Presenta como sigue:





✓ **Pseudocódigo: Grado de cumplimiento CMAHLP-F**

```

L←0
C←0
for i=0 to n-p-1
    L1←0
    C1←0
    C=C+1
    for k=0 to p-1
        if  $C_{ik}, K_{k1} \in [0, 1]$ 
            L2←0
            C2←0
            C1=C1+1
            for l=0 to p-1
                if  $Y_{ikl} \in [0, 1]$ 
                    C2=C2+1
                    if  $(C_{ik}+K_{k1})/2 < Y_{ikl}$ 
                         $L2=L2+(C_{ik}+K_{k1})/2$ 
                    Else
                         $L2=L2+Y_{ikl}$ 
                l=l+1
            L1=L1+L2/C2
        k=k+1
    L=L+L1/C1
    i=i+1
Lambda=L/C
L←0
C←0
for i=0 to n-p-1
    L1←0
    C1←0
    C=C+1

```

```

for j=0 to n-p-1
    L2←0
    C2←0
    C1=C1+1
    for l=0 to p-1
        if  $Q_{1j}, X_{i1j} \in [0, 1]$ 
            C2=C2+1
            If  $Q_{1j} < X_{i1j}$ 
                L2=L2+ $Q_{1j}$ 
            else
                L2=L2+ $X_{i1j}$ 
        l=l+1
    L1=L1+L2/C2
    j=j+1
L=L+L1/C1
i=i+1
Lambda=Lambda+L/C
L←0
C←0
for k=0 to p-1
    if  $F_k \in [0, 1]$ 
        L=L+ $F_k$ 
        C=C+1
    k=k+1
Lambda= (Lambda+L/C) /3
output Lambda

```

Tiempo computacional del algoritmo :  $p(2(n)^2 + 1)$

✓ **Implementación en C++**

```
//=====
//Fuzzy CMAHLP-F: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Tuesday,November 23rd, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main(){
    int n,p,C=0,C1,C2;
    double L=0,L1,L2;
    cout << "Degree of Certainty (CMAHLP-F)\n\n";
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of hubs: ";
    cin >> p;

    //Declaring input data
    double C[n-p][p],K[p][p],Q[p][n-p];
    double X[n-p][p][n-p],F[p];

    //Entering input data
        // will be written later
    //Processing data
    for (int i=0; i<=n-p-1;){
        L1=0;
        C1=0;
        ++C;
        for (int k=0; k<=p-1;){
            if (C[i][k]>=0 && C[i][k]<=1 &&
```

```

K[k] [l] >= 0 && K[k] [l] <= 1) {
L2=0;
C2=0;
++C1;
for (int l=0; l<=p-1;) {
    if (Y[i] [k] [l] >= 0 && Y[i] [k] [l] <= 1) {
        ++C2;
        if ((C[i] [k] + K[k] [l]) / 2 <
            Y[i] [k] [l])
            L2 += (C[i] [k] + K[k] [l]) / 2;
        else
            L2 += Y[i] [k] [l];
    }
    ++l;
}
L1 += L2 / C2;
}
++k;
}
L += L1 / C1;
++i;
}
Lambda = L / C;
L=0;
C=0;
for (int i=0; i<=n-p-1) {
    L1=0;
    C1=0;
    ++C;
    for (int j=0; j<=n-p-1) {
        L2=0;
        C2=0;
        ++C1;

```

```

for (int l=0; l<=p-1;){
    if (Q[l][j]>=0&&Q[l][j]<=0&&
        X[i][l][j]>=0&& X[i][l][j]<=0){
        ++C2;
        if (Q[l][j]<X[i][l][j])
            L2+=Q[l][j];
        else
            L2+=X[i][l][j];
    }
    ++l;
}
L1 += L2/C2;
++j;
}
L += L1/C1;
++i;
}
Lambda += L/C;
L=0;
C=0;
for (int k=0; k<=p-1;){
    if (F[k]>=0 && F[k]<=1){
        L += F[k];
        ++C;
    }
    ++k;
}
Lambda = (Lambda+L/C)/3;
cout << "\nThe degree of certainty is " << Lambda;
return 0;
}

```

### III.3.3. Problema UMA $p$ HMP

#### III.3.3.1. UMA $p$ HMP crisp

Además de lo visto en [10] sobre UMA $p$ HMP, en [38] se considera una red en la cual algún producto se intercambia entre nodos, posiblemente vía otros nodos que facilitan este flujo. En las situaciones donde se puede ganar un descuento del volumen combinando flujos de numerosas fuentes, se debe organizar la red para que incluya un conjunto de centros, que actúan como centros de consolidación para numerosos nodos. Estos centros están completamente interconectados mientras que los nodos no centros deben enviar todo su tráfico indirectamente vía uno o muchos centros. Tales redes de transporte se presentan en la industria de línea aérea, los sistemas de reparto postal y de telecomunicación. Cuando se diseña una tal red, se debe escoger un número fijo  $p$  de los nodos como centros y asignar (o conectar) los nodos restantes a uno o muchos de los centros escogidos de modo que se minimicen los costes de funcionamiento de la red resultante. Este tipo de problema de diseño se refiere como un problema de  $p$ -mediana centro.

Se asume que se conocen los flujos no negativos asociados a cada par de nodos origen – destino (O – D). El coste de funcionamiento de cualquier red depende de la distancia entre nodos en la red, el montante del flujo que debe desplazarse a través de estas distancias y el tipo de conexión entre los nodos (si el flujo se transfiere entre los centros, se colecciona desde o se distribuye a un nodo no centro).

En este problema de  $p$ -Mediana Centro no Capacitado con Asignación Múltiple no hay restricciones de capacidad en los centros o sobre los flujos en las aristas. También cada nodo no centro puede asignarse a múltiples centros en vez de enviar y recibir todo su flujo a o de un único centro. Mientras que en el caso de asignación única, cada nodo no centro se asigna exactamente a un centro, el caso de asignación múltiple permite que cada nodo no centro se asigne a muchos o a todos los centros de modo que se minimice todo el coste para satisfacer la demanda del flujo. Por ejemplo, en la aplicación de línea aérea, los pasajeros de un nodo no centro pueden elegir volar vía diferentes nodos centros dependiendo de su destino final. En el ejemplo de reparto postal, el correo de un nodo no centro puede rutarse vía diferentes nodos centros dependiendo de su último destino. Puesto que las restricciones de asignación única se

relajan en el caso de asignación múltiple, la solución resultante será menor que en el caso de única asignación.

El Problema de  $p$ -Mediana Centro no Capacitado con Asignación Múltiple es un problema de decisión de localización–asignación. Dado un grafo completo  $G=(N,A)$ , donde  $N=\{1,\dots,n\}$  es el conjunto de nodos y  $A=N\times N$  es el conjunto de aristas. Para cada par de nodos  $(i,j)$ , se tiene una demanda no negativa  $W_{ij}$  del flujo. En la literatura se asume generalmente que los volúmenes del flujo son simétricos, es decir  $W_{ij}=W_{ji}$ . Mientras que este no es una restricción en [38], se muestra como esta propiedad se puede explotar si existe. Hay también distancias  $d_{ij}$  entre los pares de nodos  $(i,j)$ . Dado  $p$ , el número de centros a localizar en una red, el Problema de  $p$ -Mediana Centro no Capacitado con Asignación Múltiple es un problema de encontrar las localizaciones óptimas de centros (un subconjuntos de nodos), así como la asignación de todos los nodos no centros restantes a los centros.

En el ejemplo de reparto postal, cada flujo  $W_{ij}$  de  $i$  a  $j$  tiene tres componentes separados asociados a el, denominados colección, transferencia, y distribución. La colección representa el movimiento del correo desde los distritos postales a su centro de clasificación de correo al que se asignan (centro). Los centros en esta aplicación actúan como centros de consolidación y de clasificación para el flujo de correo. La transferencia es el movimiento de correo entre los centros. La distribución ocurre desde el centro terminal al distrito postal de destino. Cada una de estas actividades tiene un coste por unidad de flujo que es proporcional a la distancia. Los coeficientes de coste se denotan respectivamente como  $\chi, \alpha$  y  $\delta$ . En la literatura,  $\chi$  y  $\delta$  son generalmente iguales (normalizados a uno) mientras que  $\alpha$  es un factor de descuento de volumen que es menos que uno.

El Problema de  $p$ -Mediana Centro no Capacitado con Asignación Múltiple puede formularse como un Programa Mezclado Lineal Entero. Se utiliza, como punto de partida, la formulación publicada por Campbell en [14]: Sea el conjunto de  $n$  nodos de los cuales  $p$  se deben escoger como centros. Sea  $H_k$  una variable binaria para cada  $k \in N$  que tiene un valor de uno si  $k$  es un centro. Para cada  $i, j, k, l \in N$ , sea  $X_{ijkl}$  una variable que es la fracción del flujo total  $W_{ij}$  desde el nodo  $i$  al nodo  $j$ , rutado vía  $k$  y  $l$ . El coste del flujo por unidad junto con esta ruta se



da por  $C_{ijkl}$ . Definiendo  $d_{ij}$  como la distancia entre los nodos  $i$  y  $j$ , el coste se puede escribir como

$$C_{ijkl} = \chi d_{ik} + \alpha d_{kl} + \delta d_{lj}$$

donde  $\chi, \alpha$  y  $\delta$  son costes respectivos de colección, transferencia y distribución. En la literatura  $\alpha < 1$  es el factor de descuento y  $\chi = \delta = 1$ . No obstante, para las aplicaciones de reparto postal, se puede tener tres constantes distintas. Con estas definiciones, el problema UMApHMP se puede estatuar como sigue:

(Problema UMApHMP-LP)

$$\min \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \sum_{j \in N} W_{ij} C_{ijkl} X_{ijkl}$$

$$\text{sujeto a: } \sum_{k \in N} H_k = p \quad (1)$$

$$\sum_{k \in N} \sum_{l \in N} X_{ijkl} = 1 \quad \forall i, j \in N, \quad (2)$$

$$\sum_{l \in N} X_{ijkl} \leq H_k \quad \forall i, j, k \in N, \quad (3)$$

$$\sum_{k \in N} X_{ijkl} \leq H_l \quad \forall i, j, l \in N, \quad (4)$$

$$H_k \in \{0, 1\} \quad \forall k \in N, \quad (5)$$

$$X_{ijkl} \geq 0 \quad \forall i, j, k, l \in N. \quad (6)$$

Esta formulación fue utilizada por Skorin-Kapov et al. [93] para resolver el problema de  $p$ -mediana centro no capacitado con asignación múltiple exactamente incorporando la relajación de la programación lineal de esta formulación en una búsqueda sobre un árbol.

Se debe observar que las únicas variables enteras en el Problema de  $p$ -Mediana Centro no Capacitado con Asignación Múltiple son las  $H_k$ . Para los problemas de asignación múltiple, Los  $X$  pueden ser fraccional. Dado que  $X_{ijkl}$  no tiene por que ser entera, el lema siguiente es fácil comprobarse.

**Lema 1:** La formulación UMApHMP-LP tiene una solución óptima en la cual todos los flujos  $X_{ijkl}$  son sea cero sea uno. (En otras palabras, existe una solución en la cual todo flujo entre cualquier par de nodos se transfiere por un solo camino). Además, dado un conjunto óptimo

de centros, esta solución debe encontrarse solucionando el problema de camino más corto entre cada par de nodos.

**Prueba:** Puesto que  $W_{ij} \geq 0$ ,  $G$  es un grafo completo y  $p \geq 1$ , siempre existirá una solución factible al  $UMApHMP-LP$ . Si  $X_{ijkl}$  es fraccional para algún  $i, j, k$  y  $l$ , entonces existe al menos un otro  $X_{ijmq}$  fraccional para algún  $m, q \in N$ , puesto que la ecuación (2) está válida. Ahora si  $C_{ijmq} < C_{ijkl}$ , entonces la solución no es óptima puesto que una solución más barata pudiera encontrarse poniendo

$$X_{ijmq} \leftarrow X_{ijmq} + X_{ijkl} \text{ y } X_{ijkl} \leftarrow 0.$$

Un argumento similar es válido si  $C_{ijmq} > C_{ijkl}$ . Finalmente si  $C_{ijmq} = C_{ijkl}$ , el flujo podría enviarse vía uno de los conjuntos de centros  $m, q$  o  $k, l$ . Este muestra que para un conjunto de centros  $H$  dado, la solución óptima puede obtenerse escogiendo algún  $k, l \in H$  de modo que  $C_{ijkl} = \min_{m, q \in H} C_{ijmq}$  y poniendo  $X_{ijkl}$  a uno.

Tales  $k$  y  $l$  se pueden encontrar solucionando el problema del camino más corto en un grafo dirigido  $G'$ , que se construye de manera siguiente: De cada  $i \in N \setminus H$  a cada  $k \in H$ , hay una arista de longitud  $\chi d_{ik}$ . Entre cualesquiera dos nodos  $k, l \in H$  existen aristas de longitud  $\alpha d_{kl}$  en ambas direcciones. De cualquier centro  $l \in H$  a cualquier otro nodo  $j \in N \setminus H$  existe una arista de longitud  $\delta d_{lj}$ .

Ernst y Krishnamoorthy [38] utilizan el Lema 1 para desarrollar una heurística para el  $UMApHMP$ . Dado  $H$ , un conjunto de  $p$  localizaciones de centros, se puede solucionar la parte asignación del problema  $UMApHMP$  muy eficientemente solucionando un problema de camino más corto de todos los pares. Esto se puede hacer utilizando la versión ligeramente modificada del algoritmo de Floyd-Warshall [27]. Para un conjunto de centros  $H$  dado, el algoritmo de camino más corto (SPA), calcula para cada  $i, j \in N$ :

$$c[i, j] = \sum_{k \in N} \sum_{l \in N} C_{ijkl} X_{ijkl}.$$

SPA tiene una complejidad polinomial de  $O(pn^2)$  y en la práctica, se puede ejecutar muy eficientemente. El coste óptimo para el conjunto de centros dado puede evaluarse trivialmente como  $\sum_{i,j \in N} W_{ij}c[i, j]$ .

Así, dado  $H$  generado de alguna manera, se puede óptimamente solucionar la parte asignación del problema usando SPA. Se puede utilizar SPA de dos maneras diferentes para generar soluciones factibles al  $UMApHMP$ , como se ha descrito arriba.

Dado un conjunto  $H$  de centros, una solución factible al  $UMApHMP$  puede ser obtenida usando SPA. Ernst y Krishnamoorthy [38] utilizan esta idea para desarrollar un algoritmo heurístico para el problema. Se refieren a esta heurística como SPBH (Shortest Path Based Heuristic). Se empiezan con una configuración inicial aleatoria  $H$ . Se exploran todas las maneras posibles de cambiar  $H$  reemplazando uno de sus miembros por un nodo en  $N \setminus H$ , hasta que se obtenga una solución mejor. Esta nueva solución se usa como punto de partida de la iteración siguiente. Si no es posible obtener mejor solución de esta manera, se empieza con una configuración aleatoria inicial diferente. Se repite este proceso para  $\gamma$  iteraciones, donde  $\gamma$  depende del tamaño del problema, y se retiene la mejor solución. En general aumentar el valor de  $\gamma$  aumenta la probabilidad de encontrar el óptimo global. Por otra parte si se elige un valor grande de  $\gamma$  para problemas grandes, el tiempo computacional puede ser excesivo. No obstante se debe notar que el algoritmo es todavía polinomial.

Ernst y Krishnamoorthy [38] utilizan en el lema 1 el hecho de que generalmente  $p = n$ , para desarrollar un algoritmo combinatorio de enumeración explícita de para el  $UMApHMP$ . Emplean el SPA sobre cada combinación posible de centros  $H \subseteq N$  y la solución óptima es la de coste mínimo. Denotan este algoritmo por EE. Claro, con un tal enfoque está garantizado encontrar la solución óptima. No obstante tiene una complejidad combinatoria  $O(C_p^n pn^2)$ . El algoritmo EE es todavía razonable para pequeños problemas, particularmente si  $p$  es muy pequeño. Por ejemplo, si se deben localizar exactamente dos centros, la complejidad en términos de los nodos es  $O(n^4)$ . Para grandes problemas, obviamente, este enfoque no es apropiado puesto que hay  $C_p^n$  maneras posibles de localizar los centros.

Para solucionar grandes problemas, particularmente para  $p$  muy grande, se necesita otros enfoques en vez de solamente la enumeración. El problema  $UMApHLP-LP$  supone  $(n^4 + n)$  variables de las cuales  $n$  son binarias. Requiere  $(1 + n^2 + 2n^3)$  restricciones lineales. No obstante, como  $n$  aumenta, esta formulación vuelve a ser intratable debido al crecimiento explosivo en el número de variables y restricciones. Además solucionar la relajación de LP toma montantes excesivos de tiempo computacional (tiempos CPU de más 3 – 4 horas para problemas con  $n = 25$ ).

Así se propone una nueva formulación del problema cuya motivación es el deseo de solucionar grandes problemas. La idea básica escondida detrás de esta formulación denotada por  $UMApHMP-N$ , es eliminar las variables  $X_{ijkl}$  tratando las transferencias intercentros como un problema de flujo multiproducto. Cada producto representa el flujo de tráfico originando desde un nodo particular.

$Y_{kl}^i$  se define como el montante total del flujo del producto  $i$ , (es decir, el tráfico emanando del nodo  $i$ ) que transita entre los centros  $k$  y  $l$ . También  $Z_{ik}$  se define como el flujo del nodo  $i$  al centro  $k$  y  $X_{lj}^i$  como el flujo del producto  $i$  fluyendo del centro  $l$  al nodo  $j$ . Para simplificar la notación, se define

$$O_i = \sum_{j \in N} W_{ij} \text{ y } D_i = \sum_{j \in N} W_{ji}.$$

La nueva formulación es, entonces:

(Problema  $UMApHMP-N$ )

$$\min \sum_{i \in N} \left[ \sum_{k \in N} \chi d_{ik} Z_{ik} + \sum_{k \in N} \sum_{l \in N} \alpha d_{kl} Y_{kl}^i + \sum_{l \in N} \sum_{j \in N} \delta d_{kl} X_{lj}^i \right]$$

$$\text{sujeto a: } \sum_{k \in N} H_k = p, \quad (7)$$

$$\sum_{k \in N} Z_{ik} = O_i \quad \forall i \in N, \quad (8)$$

$$\sum_{l \in N} X_{lj}^i = W_{ij} \quad \forall i, j \in N, \quad (9)$$

$$\sum_{l \in N} Y_{kl}^i + \sum_{j \in N} X_{kj}^i - \sum_{l \in N} Y_{lk}^i - Z_{ik} = 0 \quad \forall i, k \in N, \quad (10)$$

$$Z_{ik} \leq O_i H_k \quad \forall i, k \in N, \quad (11)$$

$$X_{lj}^i \leq W_{ij} H_l \quad \forall i, j, l \in N, \quad (12)$$

$$X_{lj}^i, Y_{kl}^i, Z_{ik} \geq 0 \quad \forall i, j, k, l \in N, \quad (13)$$

$$H_k \in \{0, 1\} \quad \forall k \in N. \quad (14)$$

Las ecuaciones (8) – (10) representan ecuaciones de divergencia para un problema de flujo de la red para cada producto  $i$ . Hay  $n$  redes (una para cada producto) con  $1+n+n$  nodos. El primer nodo es siempre el nodo fuente. Corresponde al nodo  $i$  y tiene un suministro de  $O_i$ . La demanda en cada nodo de destino se ha dado como  $W_{ij}$  en las  $n$  ecuaciones siguientes. Finalmente hay  $n$  nodos de transporte, uno correspondiendo a cada nodo centro posible.

En comparación con la formulación *UMApHMP-LP*, esta formulación disminuye el tamaño del problema. La formulación *UMApHMP-N* implica  $2n^3 + n^2 + n$  variables de las cuales  $n$  son binarias. Requiere  $1+n+3n^2+n^3$  restricciones lineales. En otras palabras el tamaño del problema se reduce por un factor de  $n$  en el número de variables y por casi un factor de 2 en el número de restricciones.

Obviamente el número de restricciones es todavía bastante grande. Así, sería preferible reducirlo todavía. Una manera de hacerlo es agregar algunas restricciones de la forma (12). Una formulación alternativa se obtiene reemplazando estas restricciones por

$$\sum_{i \in N} X_{lj}^i \leq D_j H_l \quad \forall l, j \in N \quad (15)$$

Se refiere a esta formulación como *UMApHMP-R*. Usando (15) en vez de (12) el número de restricciones se reduce por un factor de  $n$ . Desafortunadamente tiene también el efecto de hacer que la relajación del programa lineal *UMApHMP-R* sea más débil que la de *UMApHMP-N*.

Según [38], el problema *UMApHMP* es simétrico si  $W_{ij} = W_{ji}$ ,  $\forall i, j \in N$  (y por eso  $O_i = D_i$ ,  $\forall i \in N$ ) y si los costes de colección y distribución son idénticos ( $\chi = \delta$ ). Si el

problema es simétrico, se puede simplificar todavía notando que la ecuación siguiente es válida:

$$Z_{ik} = \sum_{j \in N} X_{ki}^j \quad \forall i, k \in N. \quad (16)$$

Ernst y Krisnamoorthy [38] utilizan la ecuación (16) para eliminar las variables  $Z$  completamente. Utilizando la desigualdad (15), obtienen la formulación reducida siguiente:

(problema *UMApHMP-S*)

$$\min \sum_{i \in N} \left[ \sum_{k \in N} \sum_{l \in N} \alpha d_{kl} Y_{kl}^i + \sum_{l \in N} \sum_{j \in N} (\delta + \chi) d_{lj} X_{lj}^i \right]$$

$$\text{sujeto a: } \sum_{k \in N} H_k = p, \quad (17)$$

$$\sum_{l \in N} X_{lj}^i = W_{ij} \quad \forall i, j \in N, \quad (18)$$

$$\sum_{l \in N} Y_{kl}^i + \sum_{j \in N} X_{kj}^i - \sum_{l \in N} Y_{lk}^i - \sum_{j \in N} X_{ki}^j = 0 \quad \forall i, k \in N, \quad (19)$$

$$\sum_{i \in N} X_{lj}^i \leq D_j H_l \quad \forall j, l \in N, \quad (20)$$

$$X_{lj}^i, Y_{kl}^i \geq 0 \quad \forall i, j, k, l \in N, \quad (21)$$

$$H_k \in \{0, 1\} \quad \forall k \in N. \quad (22)$$

El problema *UMApHMP-S* tiene  $2n^3 + n$  variables y  $3n^2 + 1$  restricciones.

### III.3.3.2. UMApHMP borroso

#### ✓ Formulación del modelo

A partir de este último modelo UMApHMP-S desarrollamos otro modelo UMApHMP borroso, además del desarrollado más arriba a partir del UMApHMP 1,2,3,5 desarrollado en [10]. En la función objetivo del UMApHMP-S suponemos la imprecisión en los datos siguientes:

- $d_{ij}$ , la longitud del camino más corto entre los nodos  $i$  y  $j$ , para todo  $i, j \in N$ . Así suponemos que  $\exists D \supset \mathcal{D} = \left\{ \left( d_{ij} \mid \mu_{\mathcal{D}}(d_{ij}) \right) \right\}, \forall i, j \in N$ , el subconjunto difuso de longitudes  $d_{ij}$  precisas de caminos más cortos, donde  $\mu_{\mathcal{D}}(d_{ij})$  es el grado de cumplimiento de  $d_{ij}$  o su grado de pertenencia al subconjunto borroso  $\mathcal{D}$ .
- $Y_{kl}^i$ , el flujo del nodo  $i$  transferido del centro  $k$  al centro  $l$ , para todo  $i, k, l \in N$ . Así suponemos que  $\exists Y \supset \mathcal{Y} = \left\{ \left( Y_{kl}^i \mid \mu_{\mathcal{Y}}(Y_{kl}^i) \right) \right\}, \forall i, k, l \in N$ , el subconjunto borroso des flujos  $Y_{kl}^i$  precisos, donde  $\mu_{\mathcal{Y}}(Y_{kl}^i)$  es el grado de cumplimiento de  $Y_{kl}^i$  o su grado de pertenencia al subconjunto difuso  $\mathcal{Y}$ .
- $X_{lj}^i$ , el flujo del nodo  $i$  distribuido por el centro  $l$  al nodo  $j$ , para todo  $i, l, j \in N$ . En esta formulación  $Z_{ik} = \sum_{j \in N} X_{kj}^i, \forall i, k \in N$ . Como para los datos precedentes, se supone que  $\exists X \supset \mathcal{X} = \left\{ \left( X_{lj}^i \mid \mu_{\mathcal{X}}(X_{lj}^i) \right) \right\}, \forall i, l, j \in N$ , el subconjunto difuso de los flujos  $X_{lj}^i$  precisos, donde  $\mu_{\mathcal{X}}(X_{lj}^i)$  es el grado de cumplimiento del flujo  $X_{lj}^i$  o su grado de pertenencia al subconjunto difuso  $\mathcal{X}$ .

La definición de estos datos imprecisos nos permiten desarrollar el modelo UMApHMP basado en el modelo UMApHMP-S, de la manera siguiente:

UMApHMP borroso:

$$(Z^* | \lambda^*); \min \sum_{i \in N} \left[ \sum_{k \in N} \sum_{l \in N} \alpha(d_{kl} | \mu_{\beta_0}(d_{kl})) (\mu_{\psi_0}(Y_{kl}^i) | \mu_{\psi_0}(Y_{kl}^i)) + \sum_{l \in N} \sum_{j \in N} (\delta + \chi)(d_{lj} | \mu_{\beta_0}(d_{lj})) (\mu_{\chi_0}(X_{lj}^i) | \mu_{\chi_0}(X_{lj}^i)) \right] \quad (1)$$

$$\text{sujeto a: } \sum_{k \in N} H_k = p, \quad (2)$$

$$\sum_{l \in N} X_{lj}^i = W_{ij} \quad \forall i, j \in N, \quad (3)$$

$$\sum_{l \in N} Y_{kl}^i + \sum_{j \in N} X_{kj}^i - \sum_{l \in N} Y_{lk}^i - \sum_{j \in N} X_{ki}^j = 0 \quad \forall i, k \in N, \quad (4)$$

$$\sum_{i \in N} X_{lj}^i \leq D_j H_l \quad \forall j, l \in N, \quad (5)$$

$$X_{lj}^i, Y_{kl}^i \geq 0 \quad \forall i, j, k, l \in N, \quad (6)$$

$$H_k \in \{0, 1\} \quad \forall k \in N. \quad (7)$$

El modelo se resuelve en dos etapas. Primero se calcula el coste óptimo crisp  $Z^*$  usando el modelo sin tomar en cuenta los grados de pertenencia presentes en la función objetivo. Luego se calcula el grado de cumplimiento  $\lambda^*$  de la solución crisp obtenida en el primer paso usando los grados de pertenencia presentes en la función objetivo de la siguiente manera:

Grado de cumplimiento UMApHMP:

$$\lambda^* = \mathbf{M}_{i \in N} \left( \frac{\mathbf{M}_{k \in N} \mathbf{M}_{l \in N} \mu_{\beta_0}(d_{kl}) \wedge \mu_{\psi_0}(Y_{kl}^i) + \mathbf{M}_{l \in N} \mathbf{M}_{j \in N} \mu_{\beta_0}(d_{lj}) \wedge \mu_{\chi_0}(X_{lj}^i)}{2} \right) \quad (1)$$

$$\text{s.a. } \mu_{\beta_0}(d_{kl}), \mu_{\psi_0}(Y_{kl}^i), \mu_{\beta_0}(d_{lj}), \mu_{\chi_0}(X_{lj}^i) \in [0, 1], \quad \forall i, j, k, l \in N, \quad (2)$$

Puesto que  $\lambda^* \in [0, 1]$ , podemos afirmar que el UMApHMP borroso es una generalización del UMApHMP crisp.



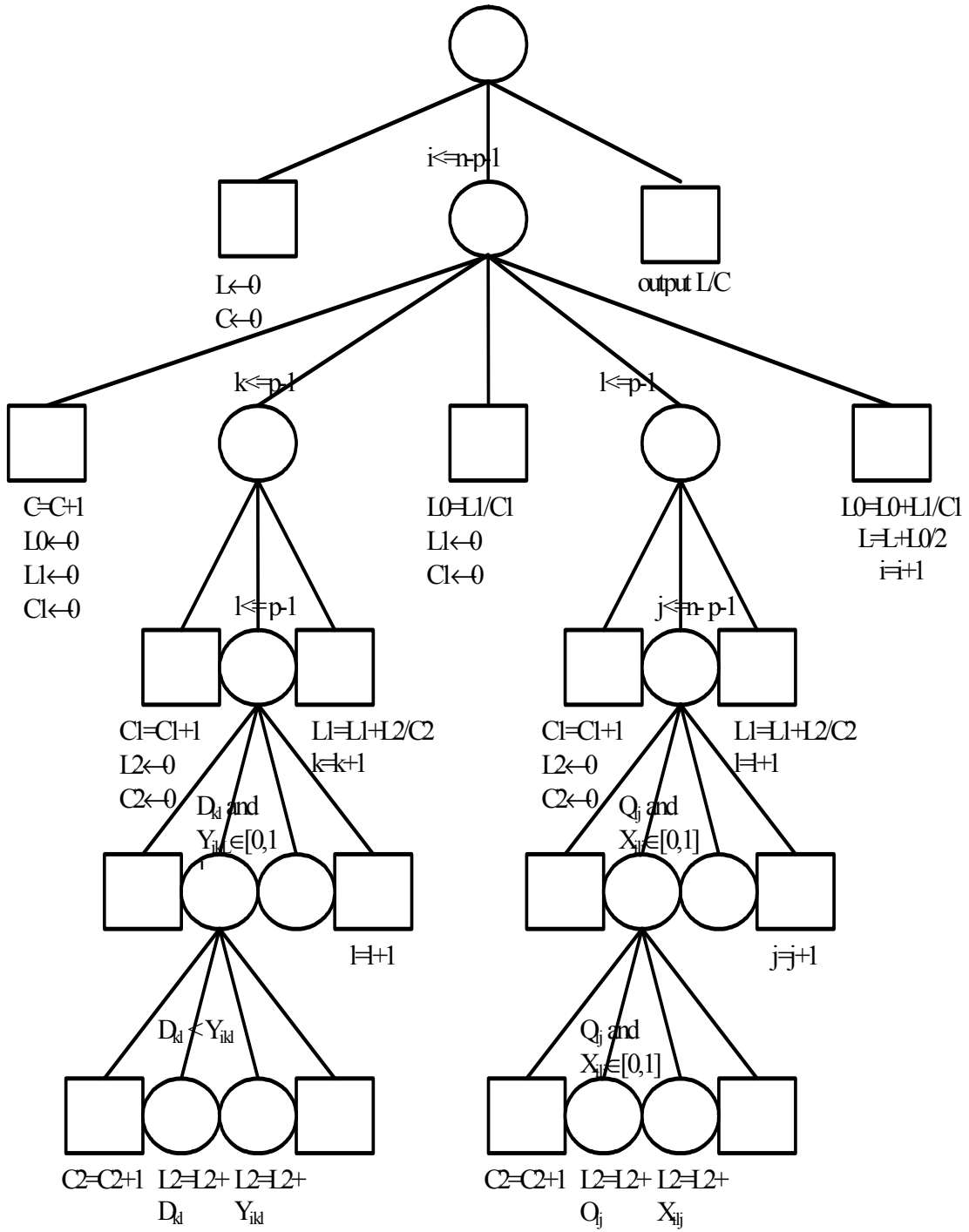
### ✓ Algoritmo de resolución

Datos de entrada:

- La matriz  $D_{p \times p} = [\mu_{\beta^0}(d_{ij})]_{p \times p}$
- La matriz  $Y_{(n-p) \times p \times p} = [\mu_{\gamma^0}(Y_{kl}^i)]_{(n-p) \times p \times p}$
- La matriz  $Q_{p \times (n-p)} = [\mu_{\delta^0}(C_{lj})]_{p \times (n-p)}$
- La matriz  $X_{(n-p) \times p \times (n-p)} = [\mu_{\chi^0}(X_{lj}^i)]_{(n-p) \times p \times (n-p)}$

Variables utilizadas:

- L, L0, L1 y L2 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C, C1 y C2 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- i, j, k y l son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red.
- p almacena el número de nodos centros.



Tiempo computacional del algoritmo:  $np(n+p) \circ O(n^2)$

✓ **Pseudocódigo: Grado de cumplimiento UMApHMP**

```

L←0
C←0
for i=0 to n-p-1
    C=C+1
    L0←0
    L1←0
    C1←0
    for k=0 to p-1
        C1=C1+1
        L2←0
        C2←0
        for l=0 to p-1
            if  $D_{kl}, Y_{ikl} \in [0, 1]$ 
                C2=C2+1
                if  $D_{kl} < Y_{ikl}$ 
                     $L2 = L2 + D_{kl}$ 
                else
                     $L2 = L2 + Y_{ikl}$ 
            l=l+1
        L1=L1+L2/C2
        k=k+1
    L0=L1/C1
    L1←0
    C1←0
    for l=0 to p-1
        C1=C1+1
        L2←0
        C2←0
        for j=0 to n-p-1
            if  $Q_{1j}, X_{i1j} \in [0, 1]$ 

```

```

        C2=C2+1
        if  $Q_{1j} < X_{i1j}$ 
            L2=L2+ $Q_{1j}$ 
        else
            L2=L2+ $X_{i1j}$ 
        j=j+1
        L1=L1+L2/C2
        l=l+1
    L0=L0+L1/C1
    L1=L1+L0/2
    i=i+1
output L/C

```

### ✓ Implementación en C++

```

//=====
//Fuzzy UMAHLP: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Tuesday, November 23rd, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main(){
    int n,p,C=0,C1,C2;
    double L=0,L0,L1,L2;
    cout << "Degree of Certainty (UMAHLP)\n\n";
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of hubs: ";
    cin >> p;

```

```

//Declaring input data
double D[p] [p] ,Y[n-p] [p] [p] ,Q[p] [n-p] ;
double X[n-p] [p] [n-p] ;

//Entering input data
    //will be written later

//Processing input data
for (int i=0; i<=n-p-1;){
    ++C;
    L0=0;
    L1=0;
    C1=0;
    for (int k=0; k<=p-1;){
        ++C1;
        L2=0;
        C2=0;
        for (int l=0; l<=p-1;){
            if (D[k] [l] >=0 && D[k] [l] <=1 &&
                Y[i] [k] [l] >=0 && Y[i] [k] [l] <=1){
                ++C2;
                if (D[k] [l] < Y[i] [k] [l])
                    L2 += D[k] [l];
                else
                    L2 += Y[i] [k] [l];
            }
            ++l;
        }
        L1 += L2/C2;
        ++k;
    }
    L0=L1/C1;
}

```

```

L1=0;
C1=0;
for (int l=0; l<=p-1;){
    ++C1;
    L2=0;
    C2=0;
    for (int j=0; J<=n-p-1;){
        if (Q[l][j]>=0 && Q[l][j]<=1 &&
            X[i][l][j]>=0 && X[i][l][j]<=1){
            ++C2;
            if (Q[l][j]<X[i][l][j])
                L2 += Q[l][j];
            else
                L2 += X[i][l][j];
        }
        ++j;
    }
    L1 += L2/C2;
    ++l;
}
L0 += L1/C1;
L += L0/2;
++i;
}
cout << "\nThe degree of certainty is " << L/C;
return 0;
}

```

### III.3.4. Problema UMApHLP

#### III.3.4.1. UMApHLP crisp

En el problema de localización de  $p$ -centro con asignación múltiple, se considera un problema menos restringido, en el cual no hay restricción sobre el número de centros a los cuales los nodos no centros pueden asignarse. Se presenta la formulación de Skorin-Kapov et al (1996) con la variables siguientes [95]:  $x_{ijkm}$  = el flujo fraccional del nodo  $i$  al nodo  $j$ , enviado por los centros  $k$  y  $m$  en este orden;  $y_k = 1$  si el nodo  $k$  es un centro, y 0 en el caso contrario. Los datos de entrada dados son los siguientes:  $n$  = el numero de nodos (localizaciones);  $p$  = el número requerido de centro que deben abrirse;  $W_{ij}$  = el flujo del nodo  $i$  al nodo  $j$ ;  $c_{ij}$  = el coste del flujo por unidad del nodo  $i$  al nodo  $j$ ;  $\alpha \leq 1$  es el factor de descuento sobre el coste por unidad del flujo entre los centros. Así, el coste por unidad del flujo del nodo  $i$  al nodo  $j$ , enviado por los centros  $k$  y  $m$  en este orden se da por:  $C_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$ . Se asume que  $c_{ii} = 0$ ,  $i = 1, \dots, n$ . Entonces, la formulación de Skorin-Kapov et al. (1996) es la siguiente:

$$(p\text{-MA}) \min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{m=1}^n W_{ij} C_{ijkm} x_{ijkm}$$

$$\text{s.a} \quad \sum_{k=1}^n y_k = p, \quad (1)$$

$$\sum_{k=1}^n \sum_{m=1}^n x_{ijkm} = 1, \quad i, j = 1, \dots, n \quad (2)$$

$$\sum_{m=1}^n x_{ijkm} \leq y_k, \quad i, j, k = 1, \dots, n \quad (3)$$

$$\sum_{k=1}^n x_{ijkm} \leq y_m, \quad i, j, m = 1, \dots, n \quad (4)$$

$$y_k \in \{0, 1\}, \quad k = 1, \dots, n$$

$$x_{ijkm} \geq 0, \quad i, j, k, m = 1, \dots, n.$$

la función objetivo debe minimizar el coste total de transporte sujeto a: tener exactamente  $p$  centros (1); el flujo entre cada par  $(i, j)$  origen-destino (o-d) podría rutarse (2) y puede rutarse solamente vía los centros (3, 4).

### III.3.4.2. UMApHLP difuso

#### ✓ Formulación del modelo

Partiendo de la formulación  $p$ -MA presentada en [95], formulamos el UMApHLP borroso suponiendo la incertidumbre en los datos siguientes:

- $W_{ij}$ , el flujo del nodo  $i$  al nodo  $j$  para todo  $i, j \in N$ . Por eso, suponemos que  $\exists W \supset \mathcal{W}^{\circ} = \left\{ \left( W_{ij} \mid \mu_{\mathcal{W}^{\circ}}(W_{ij}) \right) \right\}, \forall i, j \in N$ , el subconjunto difuso de los flujos  $W_{ij}$  precisos para todo  $i, j \in N$ , donde  $\mu_{\mathcal{W}^{\circ}}(W_{ij})$  es el grado de cumplimiento de  $W_{ij}$  o su grado de pertenencia al conjunto borroso  $\mathcal{W}^{\circ}$ .
- $C_{ijkm}$ , el coste de enviar una unidad del flujo desde el nodo  $i$  hasta el nodo  $j$  vía los centros  $k$  y  $m$  para todo  $i, j, k, m \in N$ . Por eso, suponemos que  $\exists C \supset \mathcal{C}^{\circ} = \left\{ \left( C_{ijkm} \mid \mu_{\mathcal{C}^{\circ}}(C_{ijkm}) \right) \right\}, \forall i, j, k, m \in N$ , el subconjunto borroso de los costes  $C_{ijkm}$  precisos para todo  $i, j, k, m \in N$ , donde  $\mu_{\mathcal{C}^{\circ}}(C_{ijkm})$  es el grado de cumplimiento de  $C_{ijkm}$  o su grado de pertenencia al conjunto borroso  $\mathcal{C}^{\circ}$ .

Así, nuestro modelo UMApHLP borroso se presenta de la siguiente manera:

UMApHLP borroso:

$$\left( Z^* \mid \lambda^* \right) \cong \min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{m=1}^n \left( W_{ij} \mid \mu_{\mathcal{W}^{\circ}}(W_{ij}) \right) \left( C_{ijkm} \mid \mu_{\mathcal{C}^{\circ}}(C_{ijkm}) \right) x_{ijkm} \quad (1)$$

$$\text{s.a} \quad \sum_{k=1}^n y_k = p, \quad (2)$$

$$\sum_{k=1}^n \sum_{m=1}^n x_{ijkm} = 1, \quad i, j = 1, \dots, n \quad (3)$$

$$\sum_{m=1}^n x_{ijkm} \leq y_k, \quad i, j, k = 1, \dots, n \quad (4)$$

$$\sum_{k=1}^n x_{ijkm} \leq y_m, \quad i, j, m = 1, \dots, n \quad (5)$$

$$y_k \in \{0, 1\}, \quad k = 1, \dots, n \quad (6)$$

$$x_{ijkm} \geq 0, \quad i, j, k, m = 1, \dots, n. \quad (7)$$



El modelo se resuelve en dos pasos. Al paso 1 se calcula el coste óptimo crisp  $Z^*$  usando el UMApHLP crisp es decir el modelo presentado sin tomar en cuenta los grados de pertenencia presentes en la función objetivo. Luego, en el paso 2, se calcula el grado de cumplimiento de la solución crisp obtenida en el paso 1 usando los grados de pertenencia presentes en la función objetivo de la siguiente manera:

*Grado de cumplimiento UMApHLP:*

$$\lambda^* = \prod_{i=1}^{n-p} \prod_{j=1}^{n-p} \prod_{k=1}^p \prod_{m=1}^p (\mu_{\psi_0}(W_{ij}) \wedge \mu_{\varrho_0}(C_{ijkm})) \quad (1)$$

$$\text{s.a.} \quad \mu_{\psi_0}(W_{ij}), \mu_{\varrho_0}(C_{ijkm}) \in [0,1] \quad i, j, k, m = 1, \dots, n. \quad (2)$$

Puesto que  $\lambda^* \in [0,1]$ , afirmamos que el UMApHLP borroso generaliza el UMApHLP crisp.

### ✓ Algoritmo de resolución

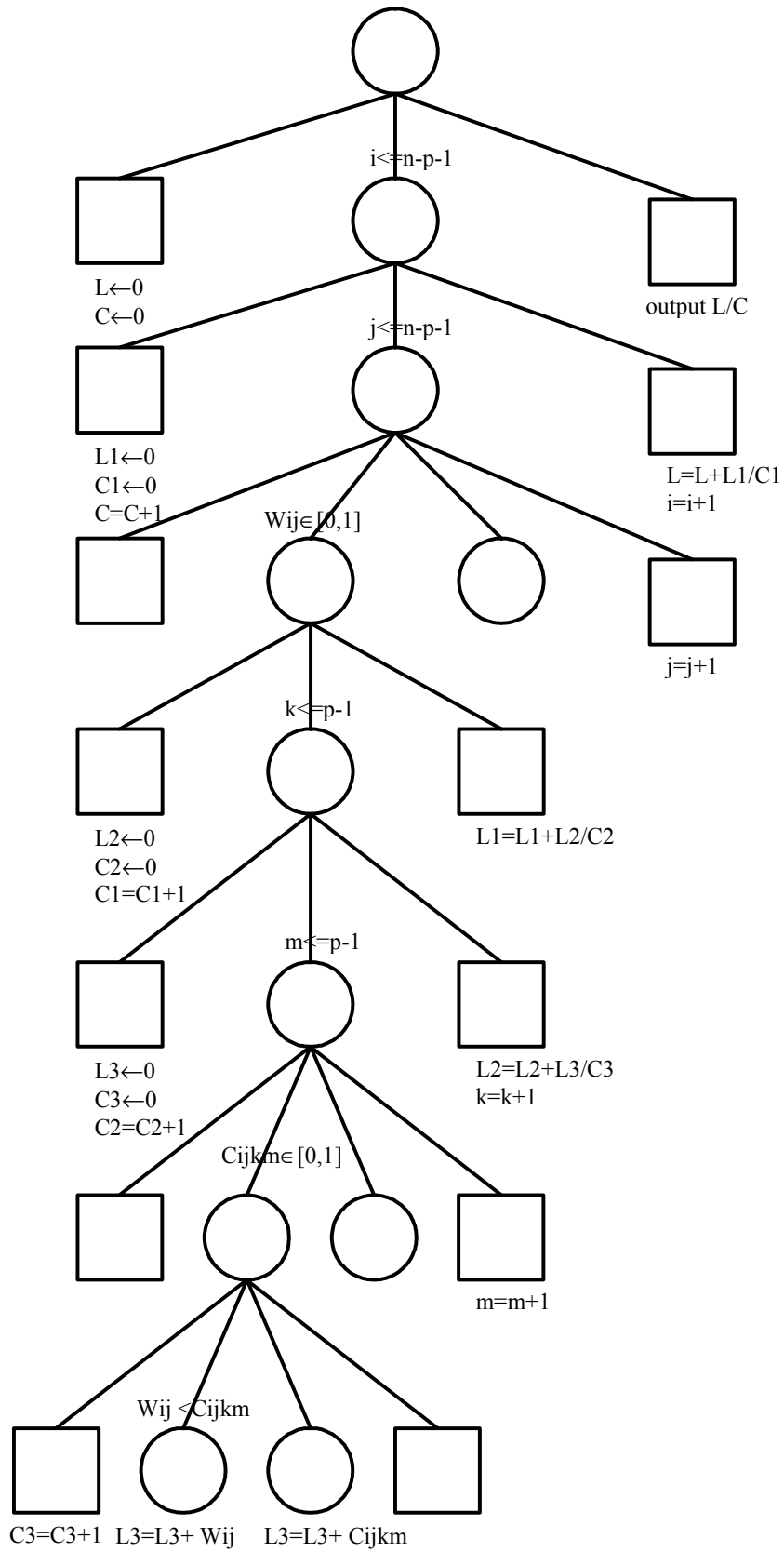
Datos de entrada:

- La matriz  $W_{(n-p) \times p} = [\mu_{\psi_0}(W_{ij})]_{(n-p) \times p}$
- La matriz  $C_{(n-p) \times (n-p) \times p \times p} = [\mu_{\varrho_0}(C_{ijkm})]_{(n-p) \times (n-p) \times p \times p}$

Variables utilizadas:

- L, L1, L2 y L3 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C, C1, C2 y C3 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- i, j, k y m son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red.
- p almacena el número de nodos centros.

El árbol programático del problema se presenta como sigue:



✓ **Pseudocódigo: Grado de cumplimiento UMApHLP**

```

L←0
C←0
for i=0 to n-p-1
    L1←0
    C1←0
    C=C+1
    for j=0 to n-p-1
        if  $W_{ij} \in [0, 1]$ 
            L2←0
            C2←0
            C1=C1+1
            for k=0 to p-1
                L3←0
                C3←0
                C2=C2+1
                for m=0 to p-1
                    if  $C_{ijkm} \in [0, 1]$ 
                        C3=C3+1
                        if  $W_{ij} < C_{ijkm}$ 
                            L3=L3+Wij
                        else
                            L3=L3+Cijkm
                    m=m+1
                L2=L2+L3/C3
            k=k+1
        L1=L1+L2/C2
    j=j+1
    L=L+L1/C1
    i=i+1
output L

```

✓ **Implementación en C++**

```
//=====
//Fuzzy UMapHLP: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Tuesday, November 23rd, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main() {
    int n,p,C=0,C1,C2,C3;
    double L=0,L1,L2,L3;
    cout << "Degree of Certainty (UMApHLP)\n\n";
    cout << "Number of nodes: ";
    cin >> n;
    cout << "\nNumber of hubs: ";
    cin >> p;
    //Declaring input data
    double W[n-p][p],C[n-p][n-p][p][p];
    //Entering input data
        //will be written later
    //Processing input data
    for (int i=0; i<=n-p-1;){
        L1=0;
        C1=0;
        ++C;
        for (int j=0; j<=n-p-1;){
            if (W[i][j]>=0 && W[i][j]<=1){
                L2=0;
                C2=0;
                ++C1;
            }
        }
    }
}
```

```

for (int k=0; k<=p-1;){
    L3=0;
    C3=0;
    ++C2;
    for (int m=0; m<=p-1;){
        if (C[i][j][k][m]>=0 &&
            C[i][j][k][m]<=1){
            ++C3;
            if
                (W[i][j]<C[i][j][k][m])
                    L3+=W[i][j];
            else
                L3+=C[i][j][k][m];
        }
        ++m;
    }
    L2 += L3/C3;
    ++k;
}
L1 += L2/C2;
}
++j;
}
L += L1/C1
++i;
}
Cout << "\nThe degree of certainty is " << L/C;
return 0;
}

```

Tiempo computacional del algoritmo:  $O(n^2 p^2)$

### III.3.5. Problema de localización de centro no capacitado con asignación múltiple (UMAHLP)

#### III.3.5.1. Aspecto crisp

Además de todo lo visto en [10] sobre UMAHLP, H.W. Hamacher et al.[49] examinan el poliedro de la factibilidad del problema de la localización de centro no capacitado (Uncapacitated Hub Location problem – UHL) con asignación múltiple. En particular, determinan la dimensión y derivan algunas clases de caras para este poliedro, obteniendo una nueva formulación del UHL cuyas restricciones son todas definiciones de caras.

Sea  $K$  un conjunto de productos y  $H$  un conjunto de nodos centros potenciales. Para cada producto  $k \in K$  y cada par ordenado de centros  $(i, j) \in H \times H$ , sean  $C_{ijk}$  los costes del transporte para encaminar el producto  $k$  vía los centros  $i$  y  $j$  (en esta dirección). Por otra parte,  $F_j$  representa los costes fijos de establecer el nodo  $j$  ( $j \in H$ ) como nodo centro. Sea  $Y_j$  ( $j \in H$ ) igual a 1, si el nodo  $j$  se establece como un nodo centro y 0 en el caso contrario; y sea  $X_{ijk} \geq 0$  ( $i, j \in H, k \in K$ ) la fracción del producto  $k$  que se encamina vía el primer nodo centro  $i$  y el segundo nodo centro  $j$ . Se desea determinar qué nodos centros deben abrirse y a qué centros se debe asignar cada producto de modo que los costes totales se minimicen bajo la restricción de que todos los productos tienen que encaminarse vía uno o dos centros. El problema de UHL con la asignación múltiple se puede modelar como el programa mixto lineal entero (MILP) siguiente de D. Skorin Kapov et al. [94]:

(UHL)

$$\min \sum_{i \in H} \sum_{j \in H} \sum_{k \in K} C_{ijk} X_{ijk} + \sum_{j \in H} F_j Y_j$$

$$\text{sujeto a } \sum_{i \in H} \sum_{j \in H} X_{ijk} = 1 \quad \text{para todo } k \in K \quad (1)$$

$$\sum_{j \in H} X_{ijk} \leq Y_i \quad \text{para todo } i \in H, k \in K, \quad (2)$$

$$\sum_{i \in H} X_{ijk} \leq Y_j \quad \text{para todo } j \in H, k \in K, \quad (3)$$

$$X_{ijk} \geq 0 \quad \text{para todo } i, j \in H, k \in K, \quad (4)$$

$$0 \leq Y_j \leq 1 \quad \text{para todo } j \in H \quad (5)$$

$$Y_j \in \phi \quad \text{para todo } j \in H \quad (6)$$

En la función objetivo se reducen al mínimo los costes totales (de transporte más los costes fijos). Cada producto  $k$  tiene que enviarse vía uno o dos nodos centros  $i$  y  $j$  (1). Si el producto  $k$  se transporta vía el primer nodo  $i$ , y cualquier otro segundo nodo, entonces  $i$  debe abrirse como nodo centro (2). Análogamente, si el producto  $k$  se envía a través del nodo  $j$  como segundo nodo, entonces  $j$  debe ser un centro (3). Existe siempre una solución óptima de UHL en la cual se valoran como enteras todas las variables  $X_{ijk}$  porque no hay restricciones de capacidad en los centros.

Sea  $q := |\mathcal{K}|$  y  $n := |\mathcal{H}|$  asumimos que ambos,  $q$  y  $n$ , son mayores o iguales a 2. La formulación UHL implica  $n^2q + n$  variables,  $n$  de ellas son binarias. Hay  $(2n+1)q$  restricciones lineales que se satisfarán.

Por simplicidad, sea  $X := (X_{ijk})_{i,j \in \mathcal{H}, k \in \mathcal{K}}$  y  $Y := (Y_j)_{j \in \mathcal{H}}$ .

Además sea  $X_{UHL}$  el conjunto de soluciones factibles de UHL, es decir

$$X_{UHL} := \{(X, Y) \in \mathbb{R}^{n^2q+n} : (X, Y) \text{ satisface (1) – (6)}\},$$

sea  $\overline{X}_{UHL}$  el conjunto de soluciones a la relajación de UHL, es decir

$$\overline{X}_{UHL} := \{(X, Y) \in \mathbb{R}^{n^2q+n} : (X, Y) \text{ satisface (1) – (5)}\},$$

sea  $L_{UHL}$  el conjunto de puntos enteros factibles de UHL, es decir

$$L_{UHL} := \{(X, Y) \in X_{UHL} : X_{ijk} \in \{0, 1\} \forall i, j \in \mathcal{H}, k \in \mathcal{K}\}$$

y sea  $P_{UHL}$  el poliedro obtenido como capsula convexa de  $L_{UHL}$ , es decir

$$P_{UHL} := \text{conv}(L_{UHL}).$$

La localización de la facilidad no capacitada (Uncapacitated Facility Location – UFL) se puede modelar como el siguiente programa mixto lineal entero:

(UFL)

$$\begin{aligned} \min \quad & \sum_{j \in \mathcal{H}} \sum_{k \in \mathcal{K}} c_{jk} x_{jk} + \sum_{j \in \mathcal{H}} f_j y_j \\ \text{sujeto a} \quad & \sum_{j \in \mathcal{H}} x_{jk} = 1 \quad \forall k \in \mathcal{K}, \end{aligned} \quad (7)$$

$$x_{jk} \leq y_j \quad \forall j \in H, k \in K, \quad (8)$$

$$x_{jk} \geq 0 \quad \forall j \in H, k \in K, \quad (9)$$

$$0 \leq y_j \leq 1 \quad \forall j \in H, \quad (10)$$

$$y_j \in \{0, 1\} \quad j \in H, \quad (11)$$

donde  $H$  es un conjunto de facilidades,  $K$  el conjunto de los clientes,  $c_{jk}$  los costes de transporte para que una facilidad  $j$  sirva al cliente  $k$ ,  $f_j$  los costes fijos de establecer una facilidad en el nodo  $j$ ,  $x_{jk}$  la fracción de la demanda del cliente  $k$  servida por la facilidad  $j \in H$ ; y donde  $y_j = 1$  si la facilidad  $j$  esta abierta, y  $y_j = 0$  en el caso contrario.

Como más arriba, se asume que  $n = |H|$  y  $q = |K|$  son ambos mayores o igual a 2.

Sea  $x := (x_{jk})_{j \in H, k \in K}$ ,  $y := (y_j)_{j \in H}$ ,  $X_{UFL} := \{(x, y) \in \mathbb{R}^{nq+n} : (x, y) \text{ satisface (7) - (11)}\}$ ,

$L_{UFL} := \{(x, y) \in X_{UFL} : x_{jk} \in \{0, 1\} \forall j \in H, k \in K\}$  y  $P_{UFL} := \text{conv}(L_{UFL})$ .

La dimensión de  $P_{UHL}$  puede ser derivada directamente demostrando que las  $q$  restricciones de la igualdad en (7) son linealmente independientes y que cada otra igualdad satisfecha por todos los puntos en  $P_{UFL}$  es una combinación lineal de igualdades de (7)

Entonces el UHL puede formularse como el siguiente programa mixto lineal entero más fuerte (stronger MILP), denominado FACET-UHL.

(FACET-UHL)

$$\min \sum_{e \in E} \sum_{k \in K} C_{ek}^0 X_{ek} + \sum_{j \in H} F_j Y_j$$

sujeto a

$$\sum_{e \in E} X_{ek} = 1 \quad \text{para todo } k \in K \quad (12)$$

$$\sum_{e \in E: e \ni j} X_{ek} \leq Y_j \quad \text{para todo } j \in H, k \in K \quad (13)$$

$$X_{ek} \geq 0 \quad \text{para todo } e \in E, k \in K \quad (14)$$

$$Y_j \leq 1 \quad \text{para todo } j \in H, k \in K \quad (15)$$



$$Y_j \in \mathcal{C} \quad \text{para todo } j \in H \quad (16)$$

Otra ventaja de FACET-UHL es su número más pequeño de variables  $X_{ek}$ , que resulta en una dimensión más pequeña para el poliedro entero factible.

### III.3.5.2. Aspecto difuso

#### ✓ Formulación del modelo

Además del UMAHLP borroso que hemos desarrollado antes a partir del UMAHLP crisp desarrollado en [10], presentamos otro UMAHLP borroso a partir del FACET-UHL crisp desarrollado en [49]. Pero, en vez de  $\mathcal{C}_{ek}^{\theta}$  escribiremos  $C_{ek}$  para que no se confunda con un subconjunto borroso:

*FACET-UHL crisp*

$$\min \sum_{e \in E} \sum_{k \in K} C_{ek} X_{ek} + \sum_{j \in H} F_j Y_j$$

sujeto a

$$\sum_{e \in E} X_{ek} = 1 \quad \text{para todo } k \in K \quad (1)$$

$$\sum_{e \in E: e \ni j} X_{ek} \leq Y_j \quad \text{para todo } j \in H, k \in K \quad (2)$$

$$X_{ek} \geq 0 \quad \text{para todo } e \in E, k \in K \quad (3)$$

$$Y_j \leq 1 \quad \text{para todo } j \in H, k \in K \quad (4)$$

$$Y_j \in \mathcal{C} \quad \text{para todo } j \in H \quad (5)$$

En la función objetivo de este modelo suponemos la incertidumbre en los datos siguientes:

- $C_{ek}$ , el coste de transporte del producto  $k$  definido como  $C_{ek} := \min\{C_{ijk}, C_{jik}\}$  si  $e = \{i, j\}$ , y  $C_{ek} := C_{iik}$  si  $e = \{i\}$ , para todo  $e \in E$  y  $k \in K$ , sabiendo que  $E := \{S \subseteq H : 1 \leq |S| \leq 2\}$ . Así podemos suponer que  $\exists C \supset \mathcal{C}_{ek} = \{(C_{ek} | \mu_{\mathcal{C}_{ek}}(C_{ek}))\}$ ,

$\forall e \in E, \forall k \in K$ , el subconjunto borroso de los costes  $C_{ek}$  precisos, donde  $\mu_{\mathcal{C}^i}(C_{ek})$  es el grado de cumplimiento de  $C_{ek}$  o su grado de pertenencia al subconjunto borroso  $\mathcal{C}^i$ .

- $F_j, \forall j \in H$ , coste fijo para establecer el nodo  $j$  como centro. Suponemos pues que  $\exists F \supset \mathcal{F}^i = \{(F_j | \mu_{\mathcal{F}^i}(F_j))\}, \forall j \in H$ , el subconjunto borrosos de los costes  $F_j$  precisos, donde  $\mu_{\mathcal{F}^i}(F_j)$  es el grado de cumplimiento de  $F_j$  o su grado de pertenencia al subconjunto difuso  $\mathcal{F}^i$ .

Así, el FACET-UHL difuso (UMAHLP difuso) se presentará como sigue:

*FACET-UHL borroso:*

$$(Z^* | \lambda^*) \cong \min \sum_{e \in E} \sum_{k \in K} (C_{ek} | \mu_{\mathcal{C}^i}(C_{ek})) X_{ek} + \sum_{j \in H} (F_j | \mu_{\mathcal{F}^i}(F_j)) Y_j \quad (1)$$

sujeto a

$$\sum_{e \in E} X_{ek} = 1 \quad \text{para todo } k \in K \quad (2)$$

$$\sum_{e \in E: e \ni j} X_{ek} \leq Y_j \quad \text{para todo } j \in H, k \in K \quad (3)$$

$$X_{ek} \geq 0 \quad \text{para todo } e \in E, k \in K \quad (4)$$

$$Y_j \leq 1 \quad \text{para todo } j \in H, k \in K \quad (5)$$

$$Y_j \in \mathcal{C} \quad \text{para todo } j \in H \quad (6)$$

El modelo se resuelve en dos pasos. En el paso 1 se calcula el coste óptimo crisp usando el FACET-UHL crisp, es decir, el modelo que acabamos de presentar sin tomar en cuenta los grados de pertenencia presentes en la función objetivo. En el paso 2, se calcula el grado de cumplimiento de la solución crisp obtenida en el paso 1, usando los grados de pertenencia presentes en la función objetivo de la siguiente manera:

Grado de cumplimiento FACET-UHL:

$$\lambda^* = \frac{\sum_{e \in E} \sum_{k \in K} \mu_{\rho^0}(C_{ek}) \cdot X_{ek} + \sum_{j \in H} \mu_{\rho^0}(F_j) \cdot Y_j}{2} \quad (1)$$

$$\text{s.a.} \quad \sum_{e=1}^{n^2} X_{ek} = 1 \quad \forall k \in K \quad (2)$$

$$X_{ek}, Y_j \in \{0, 1\} \quad \forall e \in E, \forall j \in H, \forall k \in K \quad (3)$$

$$\mu_{\rho^0}(C_{ek}), \mu_{\rho^0}(F_j) \in [0, 1] \quad \forall e \in E, \forall j \in H, \forall k \in K \quad (4)$$

Puesto que  $\lambda^* \in [0, 1]$ , podemos afirmar que el FACET-UHL borroso generaliza el crisp.

#### ✓ Algoritmo de cálculo de grado de cumplimiento del FACET-UHL

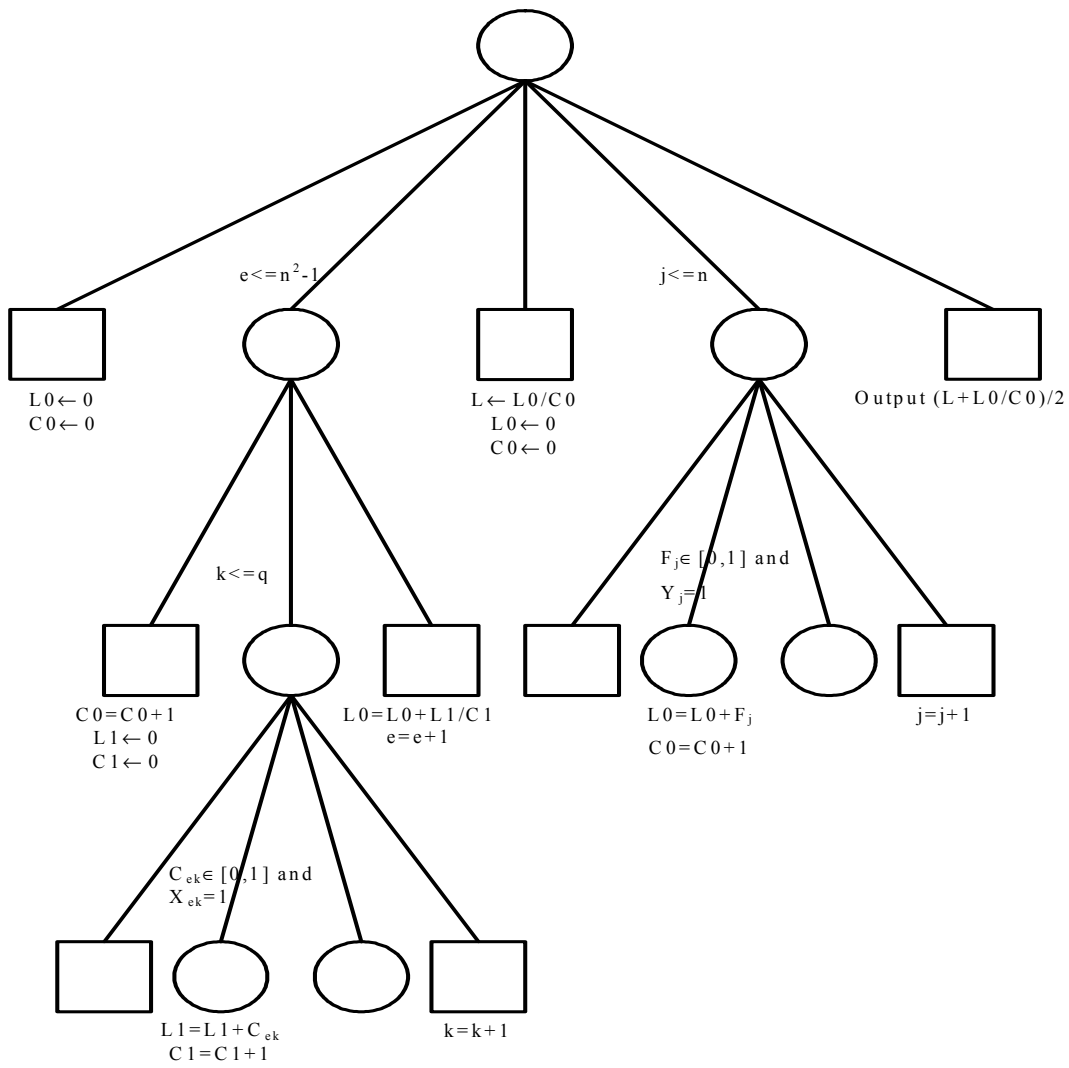
Datos de entrada:

- La matriz  $C_{n^2 \times q} = [\mu_C(C_{ek})]_{n^2 \times q}$
- La matriz  $X_{n^2 \times q} = [X_{ek}]_{n^2 \times q}$
- El vector  $F_n = [\mu_{\rho^0}(F_j)]_n$
- El vector  $Y_n = [Y_j]_n$

VARIABLES UTILIZADAS:

- L, L0 y L1 almacenan los resultados intermedios calculados en los diferentes bucles del algoritmo, desde los bucles más exteriores a los bucles más interiores respectivamente. El resultado final se almacena en la variable L.
- C0 y C1 son contadores que sirven como denominadores en los cálculos del resultado, desde los bucles más exteriores a los bucles más interiores.
- e, j y k son los índices de acceso a los datos de entrada.
- n almacena el número de los nodos de la red (nodos centros potenciales)  $i, j \in H$ .
- q almacena el número de productos  $k \in K$ .

Arbol programático:



✓ **Pseudocódigo: Grado de cumplimiento FACET-UHL**

```

L0←0
C0←0
for e=0 to n2-1
    C0=C0+1
    L1←0
    C1←0
    for k=0 to q-1
        if Cek∈ [0,1] and Xek=1
            L1=L1+Cek
            C1=C1+1
        k=k+1
    L0=L0+L1/C1
    e=e+1
L←L0/C0
L0←0
C0←0
for j=0 to n-1
    if Fj∈ [0,1] and Yj=1
        L0=L0+Fj
        C0=C0+1
    j=j+1
output (L+L0/C0)/2

```

Tiempo computacional del algoritmo:  $n^2q+n \circ O(n^2)$

## ✓ Implementación en C++

```

//=====
//Fuzzy FACET-UHL: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Friday,November 27th, 2004
//Department of Computer Science and
//Artificial Intelligence
//University of Granada
//=====
#include <iostream.h>
main(){
    int n,q,C0=0,C1;
    double L0=0,L1;
    cout << "Degree of Certainty (FACET-UHL)\n\n";
    cout << "Number of potential hub nodes (i or j): ";
    cin >> n;
    cout << "\nNumber products (k): ";
    cin >> q;

    //Declaring input data
    double C[n*n][q],F[n];
    bool X[n*n][q],Y[n];

    //Entering input data
        //WILL BE WRITTEN LATER

    //Processing input data
    for(int e=0; e<=n*n-1;){
        ++C;
        L1=0;
        C1=0;
        for(int k=0; k<=q-1;){
            if(X[e][k] && C[e][k]>=0 && C[e][k]<=0){
                L1 += C[e][k];
                ++C1;
            }
            ++k;
        }
        L0 += L1/C1;
        ++e;
    }
    double L=L0/C0;
    for(int j=0; j<=n-1;){
        if (Y[j] && F[j]>=0 && F[j]<=1){
            L0 += F[j];
            ++C0;
        }
    }
}

```

```

        ++j;
    }
    cout << "\nThe degree of certainty is ";
    cout << (L+L0/C0)/2;
    return 0;
}

```

#### V.4. Conclusión

En este capítulo, hemos investigado los problemas de centro y routing cuyo objetivo es localizar un número limitado de centros de modo que el flujo originado a cualquier nodo no centro envíe primero a un o dos nodos centros antes de que esté distribuido a su destino. Esto se debe a la casi imposibilidad o al carácter costoso de dedicar conexiones exclusivas a cada par de nodos en la red.

Hemos visto que los problemas de centro se subdividen en varios tipos. Primero, hemos visto los problemas con asignación única que son: el problema de p-mediana central no capacitada con asignación (USApHMP), el problema de asignación de p-centro (pHAP), el problema de localización de p-centro no capacitado con sola asignación (USApHLP) con una variante en que se toma en cuenta los efectos de la congestión (USApHLPC). Luego, hemos visto los problemas con asignación múltiple que son: el problema de localización de centro capacitado con asignación múltiple (CMAHLP), el problema de localización de centro no capacitado con asignación múltiple (UMAHLP), el problema de p-mediana central no capacitada con asignación múltiple (UMApHMP) y el problema de localización de p-centro no capacitado con asignación múltiple (UMApHLP).

Además de investigar sobre los problemas existentes, hemos planteado modelos para los problemas siguientes:

1. Problema borroso de la p-mediana central no capacitada con asignación única (USApHMP borroso);
2. Problema borroso de localización de p-centro no capacitado con asignación única (USApHLP borroso);
3. Problema borroso de localización de p-centro no capacitado con asignación única con congestión (USApHLPC borroso);

4. Problema borroso de asignación de p-centro (pHAP borroso);
5. Problema borroso de p-mediana central no capacitada con asignación múltiple (UMApHMP borroso);
6. Problema borroso de localización de p-centro no capacitado con asignación múltiple (UMApHLP borroso);
7. Problema borroso de localización de centro no capacitado con asignación múltiple (UMAHLP borroso);
8. Problema borroso de localización de centro capacitado con asignación múltiple (CMAHLP borroso).

La diferencia entre los modelos crisp y los borrosos ha consistido en que los modelos borrosos toman en cuenta el carácter impreciso de los datos acompañándolos de sus grados de pertenencia en la función objetivo para permitir el cálculo del grado de cumplimiento de la solución final mientras que en los modelos crisp, los grados de pertenencia se suponen que son iguales a 1, no hay necesidad de figurarlos en la función objetivo del modelo, tampoco hay necesidad de calcular el grado de cumplimiento de la solución. Puesto que el grado de cumplimiento se encuentra siempre en el intervalo  $[0,1]$ , hemos afirmado a lo grande de este capítulo que los problemas borrosos generalizan los crisp y hemos desarrollado algoritmos de cálculo del grado de cumplimiento en cada problema, cuyos tiempos computacionales son generalmente de  $O(n^2)$ .

Desde el punto de vista práctico, todos los modelos son útiles para la optimización de las redes de transporte en general tanto en una situación determinista como como en la borrosa.



## **CAPÍTULO IV. OPTIMIZACIÓN DE LA RED DE TRANSPORTE DE LA CIUDAD DE KINSHASA**

### **IV.1. Introducción**

En esta parte, primero diseñaremos el grafo de la ciudad de Kinshasa a partir del plano disponible. El diseño consistirá en fijar los vértices o nodos en la red de transporte de Kinshasa, luego atribuimos un peso a cada vértice, que representa la población en el nodo. Las aristas son las rutas asfaltadas existentes, puesto que nuestro estudio busca optimizar la red existente. A cada arista asignaremos un peso que representa su longitud. Puesto que la distribución de la población en los diferentes nodos que tendremos que fijar y las longitudes de las aristas son aproximativas, es decir inciertas, tendremos que indicar el nivel de certeza de dichos pesos acompañando cada uno de ellos de su grado de pertenencia que fijamos a partir de la observación. Así habremos obtenido un grafo difuso sobre el cual experimentamos algunos de nuestros modelos.

Como dijimos en la introducción general, en la actualidad la ciudad de Kinshasa dispone solamente de un Mercado Central rodeado de centros comerciales en el norte de Kinshasa (zona administrativa de Gombe). Existen algunos pequeños mercados como el Marché Gambela, Marché Matete etc., pero no atienden a todas las necesidades de la población que está obligado a suministrarse al Mercado Central (Marché Central de Kinshasa) cuando se trata de compras importantes. Eso dificulta el transporte. Puesto que la solución a esta dificultad es abrir centros comerciales de la misma importancia como el del norte en toda la ciudad y agregar a cada uno de ellos una estación de autobús, nuestro objetivo en este capítulo consistirá primero en determinar donde localizar de manera óptima estas facilidades mediante el modelo de la p-mediana. Luego, entre los nodos de facilidad que abremos obtenido, se tratará de escoger un o dos centros de transporte. Como no disponemos de datos sobre los flujos entre cada par de nodos, localizaremos los centros de transporte mediante el modelo de la p-mediana aplicado sobre los nodos de facilidad obtenidos.

## IV.2. Diseño del grafo de Kinshasa

### IV.2.1. Los vértices

Para fijar los vértices, hemos considerado conveniente repartir la ciudad en las zonas administrativas ya existentes y en cada una de ellas colocar uno o mas nodos dependiendo de la población. Estos nodos están situados en aquellos sitios en los que la gente ya acostumbra a coger los autobuses a pesar de que no existe ninguna estación.

A un vértice se le asigna como peso, la población de la zona administrativa en la que se encuentra. Si se encuentra más de un vértice en una zona administrativa, ésta les compartirá igualmente su población. Si un vértice pertenece a muchas zonas administrativas, su peso valdrá la suma de las poblaciones de las zonas administrativas.

En la tabla que hemos presentado en el apartado I.5. sobre fijación de la tarifa de transportes en común y de recorridos en coches en Kinshasa en 1999 hemos presentado 14 puntos de partida de autobuses en negrita a pesar de que, salvo el Mercado Central, los demás no son centros de facilidad, solo que allí la gente se acostumbra a coger autobuses. Escogemos 13 de ellos más 3 otros como nodos potenciales de facilidad:Marché Centale:

- Royal
- IPN
- Delvaux
- Rond-Point Ngaba
- Lemba
- N'djili Ste-Thérèse
- Kingasani Pascal
- Victoire
- Kintambo
- Pont-Matete
- Marché Selembao
- Marché Matete
- Bon Marché

- FIKIN
- Kalembelembe

Para fijar los pesos a estos nodos, tenemos que saber qué zona administrativa se asigna a cada uno de ellos según su situación geográfica. Como dijimos más arriba, a un nodo se puede asignar más una zona administrativa como más de un nodo pueden asignarse una misma zona administrativa. La tabla siguiente presenta la asignación de las 24 zonas administrativas a los nodos que hemos escogido:

### Asignación zona administrativa – nodo de facilidad

Zona administrativa	Nodos asignados
1. Bandalungwa	Kintambo, Marché Selembao
2. Barumbu	Bon Marché, Marché Central
3. Bumbu	Marché Selembao, Victoire
4. Gombe	Marché Central, Kintambo, Royal
5. Kalamu	Victoire
6. Kasavubu	Victoire
7. Kimbanseke	Kingasani Pascal, N'djili Ste-Thérèse
8. Kinshasa	Marché Central
9. Kintambo	Kintambo
10. Kisenso	Marché Matete
11. Lemba	Lemba (Terminus), FIKIN, RP Ngaba
12. Limete	Pont-Matete, FIKIN, Victoire
13. Lingwala	Kalembelembe, Victoire, Royal
14. Makala	Rond-Point Ngaba, Marché Selembao
15. Maluku	Kingasani Pascal
16. Masina	Kingasani Pascal
17. Matete	Marché Matete, Pont Matete
18. Mont-Ngafula	Rond-Point Ngaba, IPN
19. N'djili	N'djili Ste-Thérèse
20. Ngaba	Rond-Point Ngaba, Lemba terminus
21. Ngaliema	IPN, Delvaux, Kintambo
22. Ngiringiri	Victoire
23. N'sele	Kingasani Pascal, N'djili Ste-Thérèse
24. Selembao	Marché Selembao

### Repartición de pesos de las zonas administrativas a los nodos

Zona admin.	Peso de la zona	Nodo(s) asignado(s)	Peso atribuido al nodo
Bandalungwa	132774	1. Kintambo 2. Marché Selembao	66387 66387
Barumbu	89024	1. Bon Marché 2. Marché Central	44512 44512
Bumbu	235344	1. Marché Selembao 2. Victoire	117672 117672
Gombe	19676	1. Marché Central 2. Kintambo 3. Royal	6559 6559 6558
Kalamu	162139	1. Victoire	162139
Kasavubu	80270	1. Victoire	80270
Kimbanseke	570101	1. Kingasani Pascal 2. N'djili Ste-Thérèse	285050 285051
Kinshasa	91607	1. Marché Central	91607
Kintambo	64269	1. Kintambo	64269
Kisenso	229267	1. Marché Matete	229267
Lemba	200288	1. Lemba Terminus 2. FIKIN 3. Rond-Point Ngaba	66763 66763 66762
Limete	223440	1. Pont-Matete 2. FIKIN 3. Victoire	74480 74480 74480
Lingwala	54549	1. Kalembelembe 2. Victoire 3. Royal	18183 18183 18183
Makala	150747	1. Rond-Point Ngaba 2. Marché Selembao	75374 75373
Maluku	159129	1. Kingasani Pascal	159129
Masina	294548	1. Kingasani Pascal	294548
Matete	170046	1. Marché Matete 2. Pont-Matete	85023 85023
Mont Ngafula	175230	1. Rond-Point Ngaba 2. IPN	87615 87615
N'djili	284230	1. N'djili Ste-Thérèse	284230
Ngaba	104449	1. Rond-Point Ngaba 2. Lemba Terminus	52225 52224
Ngaliema	453703	1. IPN 2. Delvaux 3. Kintambo	151235 151234 151234
Ngiringiri	98098	1. Victoire	98098
N'sele	103214	1. Kingasani Pascal	103214
Selembao	195212	1. Marché Selembao	195212
<b>Total</b>	<b>4.341.354</b>		<b>4.341.354</b>

Así, el peso de un nodo sera la suma de los pesos de las zonas administrativas asignadas al nodo.

Esto aparece claramente en la tabla siguiente:

**Pesos de nodos**

<b>Nodo</b>	<b>Zona adm. asignada</b>	<b>Peso atribuido al nodo</b>	<b>Peso total del nodo</b>
Bon Marché	Barumbu	44512	44512
Delvaux	Ngaliema	151234	151234
FIKIN	Lemba Limete	66763 74480	141243
IPN	Mont-Ngafula Ngaliema	87615 151235	238850
Kalembelembe	Lingwala	18183	18183
Kingasani Pascal	Kimbanseke Maluku Masina N'sele	285050 159129 294548 103214	841941
Kintambo	Bandalungwa Gombe Kintambo Ngaliema	66387 6559 64269 151234	288449
Lemba Terminus	Lemba Ngaba	66763 52224	118987
Marché Central	Barumbu Gombe Kinshasa	44512 6559 91607	142678
Marché Matete	Kisenso Matete	229267 85023	314290
Marché Selembao	Bandalungwa Bumbu Makala Selembao	66387 117672 75373 195212	454644
N'djili Ste-Thérèse	Kimbanseke N'djili	285051 284230	569281
Pont-Matete	Limete Matete	74480 85023	159503
Rond-Point Ngaba	Lemba Makala Mont-Ngafula Ngaba	66762 75374 87615 52225	281976
Royal	Gombe Lingwala	6558 18183	24741
Victoire	Bumbu Kalamu Kasavubu Limete Lingwala Ngiringiri	117672 162139 80270 74480 18183 98098	550842
<b>Total</b>		<b>4.341.354</b>	<b>4.341.354</b>

Los pesos de los nodos son imprecisos debido a la antigüedad de los datos. Estos provienen del censo de 1999. Además la repartición equitativa de la población de una zona administrativa a los nodos más cercanos no siempre refleja la realidad. Expresaremos esta imprecisión indicando al lado de cada peso su grado de pertenencia estimado a partir de la observación que hicimos sobre el terreno durante nuestra investigación en Kinshasa en verano 2003.

### Nodos, pesos y grados de pertenencia

Nodo	Peso	Grado de cumplimiento
Bon Marché	44512	0,7
Delvaux	151234	0,8
FIKIN	141243	0,8
IPN	238850	0,7
Kalembelembe	18183	0,8
Kingasani Pascal	841941	0,7
Kintambo	288449	0,7
Lemba	118987	0,6
Marché Centale (ZANDO)	142678	0,6
Marché Matete	314290	0,8
Marché Selembao	454644	0,8
N'djili Ste-Thérèse	569281	0,8
Pont-Matete	159503	0,6
Rond-Point Ngaba	281976	0,7
Royal	24741	0,5
Victoire	550842	0,7

### Ordenación de los nodos

Los nodos retenidos se ordenan y se denominan definitivamente de la manera siguiente :PASCAL

1. N'DJILI
2. MATETE
3. FIKIN
4. LEMBA
5. RP NGABA
6. IPN
7. DELVAUX
8. KINTAMBO
9. ROYAL
10. SELEMBAO
11. VICTOIRE
12. KALEMBELEMBE
13. ZANDO
14. BON MARCHE
15. PONT-MATETE

Presentamos las distancias entre los nodos adyacentes (longitudes des las aristas) en la tabla siguiente:

#### IV.2.2. Las aristas

Vértice <i>i</i>	Vértice <i>j</i>	peso en Km	Grado de cumplimiento
1. PASCAL	2. N'DJILI	6	0.7
1. PASCAL	3. MATETE	8	0.7
1. PASCAL	4. FIKIN	7.5	0.7
2. N'DJILI	3. MATETE	8	0.7
2. N'DJILI	4. FIKIN	7.5	0.7
3. MATETE	4. FIKIN	2.5	0.7
3. MATETE	5. LEMBA	3	0.7
3. MATETE	6. RP NGABA	5.5	0.7
4. FIKIN	5. LEMBA	1.5	0.7
4. FIKIN	12. VICTOIRE	6.5	0.7
4. FIKIN	14. ZANDO	10.5	0.7
5. LEMBA	6. RP NGABA	2.5	0.7
6. RP NGABA	7. IPN	15	0.7
6. RP NGABA	12. VICTOIRE	7.5	0.7
7. IPN	8. DELVAUX	3	0.7
7. IPN	11. SELEMBAO	3.5	0.7
8. DELVAUX	9. KINTAMBO	5	0.7
9. KINTAMBO	10. ROYAL	3.5	0.7
9. KINTAMBO	12. VICTOIRE	7.5	0.7
9. KINTAMBO	13. KALEMBELEMBE	5.5	0.7
10. ROYAL	11. SELEMBAO	9.5	0.7
10. ROYAL	13. KALEMBELEMBE	4	0.7
10. ROYAL	14. ZANDO	3.5	0.7
11. SELEMBAO	12. VICTOIRE	7	0.7
11. SELEMBAO	13. KALEMBELEMBE	8	0.7

12. VICTOIRE	13. KALEMBELEMBE	3	0.7
12. VICTOIRE	14. ZANDO	5.5	0.7
12. VICTOIRE	15. BON MARCHE	5.5	0.7
12. VICTOIRE	16. PONT-MATETE	4	0.7
13. KALEMBELEMBE	14. ZANDO	4	0.7
13. KALEMBELEMBE	15. BON MARCHE	3.5	0.7
14. ZANDO	15. BON MARCHE	2	0.7
15. BON MARCHE	16. PONT-MATETE	8	0.7

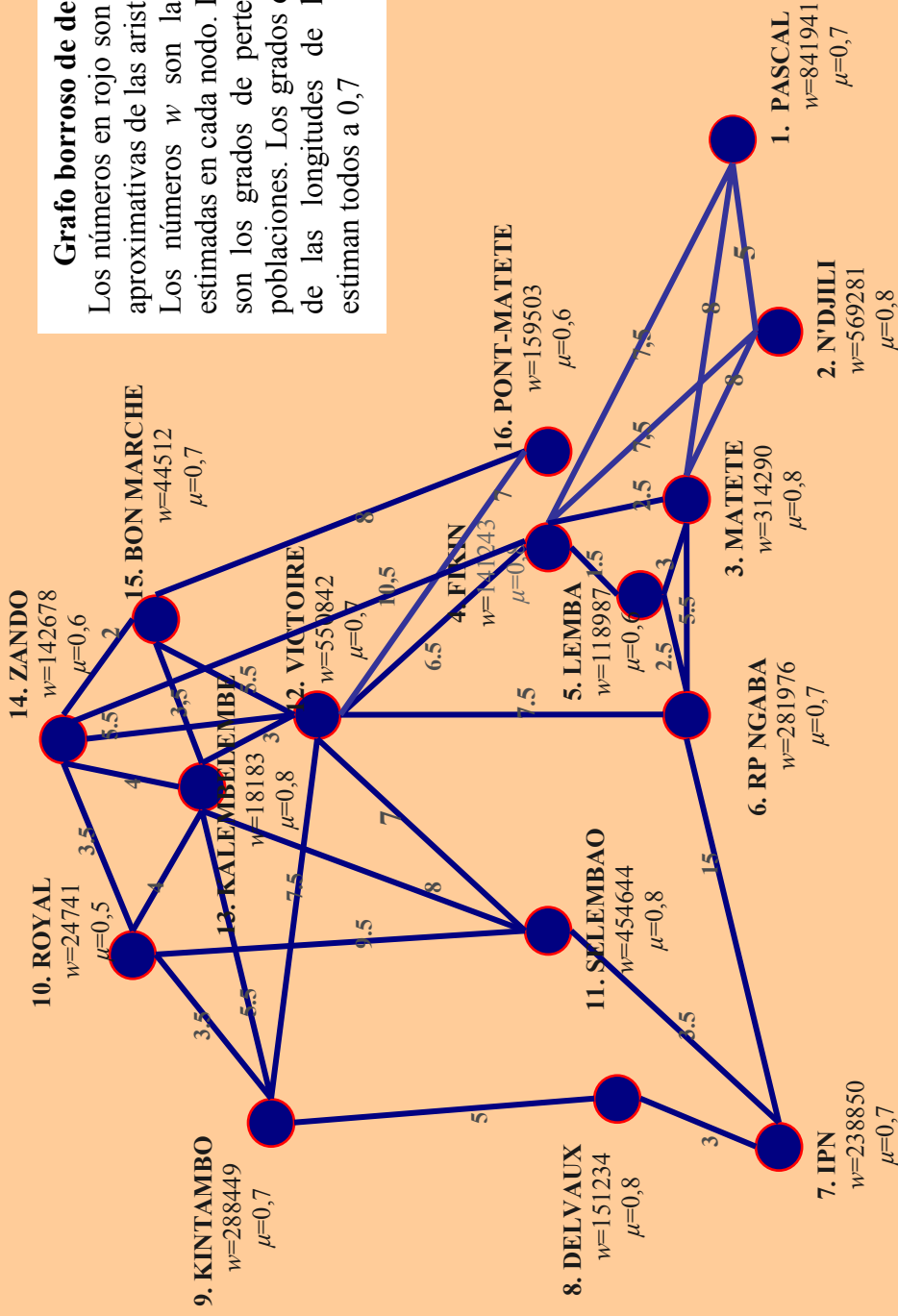
El Grafo de Kinsasa se presenta como sigue:



**Grafo borroso de Kinshasa**

Los números en rojo son las longitudes aproximativas de las aristas en km.

Los números  $w$  son las poblaciones estimadas en cada nodo. Los números  $\mu$  son los grados de pertenencia de las poblaciones. Los grados de pertenencia de las longitudes de las aristas se estiman todos a 0,7



### IV.3. Breve presentación del software GAMS

Utilizaremos el software GAMS para resolver los modelos crisp de localización que necesitamos para optimizar la red de transporte de la ciudad de Kinshasa. Luego calcularemos el grado de cumplimiento de las soluciones obtenidas utilizando los programas en C++ desarrollados en este trabajo.

Sus inventadores, Brooke, Kendrick, Meeraus y Raman describen este software en un manual de usuario [12] que se encuentra en el sitio web del Gams, [www.gams.com](http://www.gams.com). En el mismo sitio se consigue también la versión Demo del Gams. El GAMS (General Algebraic Modeling System) es un software que

- facilita un lenguaje de alto nivel para la representación compacta de modelos matemáticos largos y complejos,
- permite hacer cambios en las especificaciones del modelo de manera simple y segura,
- permite sentencias inambiguas de relaciones algebraicas
- permite descripciones del modelo independientes de los algoritmos de solución

Algunas características básicas del GAMS son:

#### a) Principios generales del GAMS

- Todos los métodos algorítmicos podrían ser disponibles sin cambiar la representación del modelo del usuario. La introducción de nuevos métodos, o nuevas implementaciones de métodos existentes, podrían ser posibles sin requerir cambios en los modelos existentes. Optimizaciones lineales, no lineales, enteras, mezcladas enteras no lineales (MILP) y problemas complementarios mezclados pueden acomodarse.
- El problema de optimización podría expresarse independientemente de los datos que utiliza. Esta separación de la lógica y de los datos permite que un problema aumente en tamaño sin causar un aumento en la complejidad de la representación.
- El uso del modelo de datos relacional requiere que la asignación de recursos del ordenador sea automática. Esto significa que los modelos largos y complejos pueden

construirse sin que el usuario se preocupe sobre los detalles el almacenamiento de las matrices etc.

### **b) Documentación**

La representación del modelo GAMS es en una forma que la gente y los ordenadores pueden leer fácilmente. Esto significa que el mismo programa GAMS es la documentación del modelo. No obstante, el diseño del GAMS incorpora las características dirigidas especialmente a las necesidades de documentación del usuario:

- Una representación del modelo GAMS es concisa, y hace plenamente uso de la elegancia de la representación matemática.
- Todas las transformaciones de los datos se especifican concisamente y algebraicamente. Esto quiere decir que todos los datos pueden introducirse en su forma más elemental y que todas las transformaciones operadas en la construcción del modelo son disponibles para la inspección.
- Texto explicatorio puede formar parte de la definición de todos los símbolos y se reproduce cada vez que los valores asociados salen en la pantalla.
- Toda información necesitada para comprender el modelo se encuentra en un documento.

### **c) Portabilidad**

El sistema GAMS se ha diseñado para que los modelos puedan resolverse en diferentes tipos de ordenadores sin ningún cambio. Un modelo desarrollado en un pequeño ordenador personal puede más tarde resolverse en un superordenador. Uno puede escribir un modelo que utilizan luego otras personas físicamente lejos del escritor del modelo. En contraste a lo dicho previamente, solo un documento necesita moverse – la instrucción GAMS del modelo. Este contiene todas las especificaciones de datos y lógica necesarias para resolver el modelo.

#### d) Interfaz usuario

La portabilidad tiene también implicaciones sobre la interfaz usuario. El sistema básica del GAMS está dirigido a fichero, y no existe un editor especial ni rutinas gráficas para la entrada y salida de datos. En vez de aburrir el usuario con un conjunto de comandos, GAMS ofrece una arquitectura abierta en que cada usuario puede utilizar un editor de su preferencia. La interfaz usuario básica facilita la integración del GAMS con una variedad de entornos usuarios de desarrollo existentes y futuros.

#### e) Librería del modelo

Los elementos diseño a partir de previos proyectos sirven como fuente de ideas para nuevos desarrollos.

### IV.4. Resultado del Cálculo de la p-mediana crisp

El resultado de la p-mediana crisp sobre el grafo de la ciudad de Kinshasa presentado más arriba se presenta como sigue, con  $p = 2$  hasta  $p = 15$

P	mediana	radio	tiempo/seg	tipo
2.00	2.14635E+7	12.50	4.00	OPTIMA
	VERT-SOL 1 12			
3.00	1.57017E+7	8.00	1.00	OPTIMA
	VERT-SOL 1 11 12			
4.00	1.10622E+7	7.50	1.00	OPTIMA
	VERT-SOL 1 5 11 12			
5.00	8215840.00	7.50	1.00	OPTIMA
	VERT-SOL 1 2 5 11 12			
6.00	5739028.00	7.00	1.00	OPTIMA
	VERT-SOL 1 2 5 9 11 12			
7.00	4529928.50	7.00	1.00	OPTIMA
	VERT-SOL 1 2 3 6 9 11 12			
8.00	3391485.50	7.00	1.00	OPTIMA
	VERT-SOL 1 2 3 6 7 9 11 12			
9.00	2274964.50	5.50	1.00	OPTIMA
	VERT-SOL 1 2 3 6 7 9 11 12 16			
10.00	1334443.50	3.50	2.00	OPTIMA
	VERT-SOL 1 2 3 6 7 9 11 12 14 16			
11.00	862349.00	3.50	1.00	OPTIMA
	VERT-SOL 1 2 3 4 6 7 9 11 12 14 16			
12.00	408647.00	3.50	1.00	OPTIMA
	VERT-SOL 1 2 3 4 6 7 8 9 11 12 14 16			
13.00	230166.50	3.50	2.00	OPTIMA

	VERT-SOL	1	2	3	4	5	6	7	8	9	11	12	14	16
14.00		141142.50		3.50		1.00		OPTIMA						
	VERT-SOL	1	2	3	4	5	6	7	8	9	11	12	14	15
16														
15.00		54549.00		3.00		1.00		OPTIMA						
	VERT-SOL	1	2	3	4	5	6	7	8	9	10	11	12	14
15	16													

En este conjunto de soluciones, es el Gobierno quien debe elegir según sus posibilidades económicas. Pero hemos encuadrado la alternativa con 10 nodos para poder calcular el grado de cumplimiento de esta solución suponiendo que coincide con la opción del gobierno. Esta solución crisp nos sugiere localizar las facilidades con un coste óptimo o distancia total de recorrida de 1.334.443,5 km en los nodos siguientes:

1. PASCAL
2. N'DJILI
3. MATETE
6. RP NGABA
7. IPN
9. KINTAMBO
11. SELEMBAO
12. VICTOIRE
14. ZANDO
16. PONT-MATETE

Estos son los nodos en que se se deben construir centros de comerciales de la misma importancia como ZANDO al norte de Kinshasa para que toda la población de Kinshasa no tenga que ir al norte (ZANDO) para aprovisionarse. En estos nodos igualmente se deben construir grandes estaciones o "parkings" de autobús para conectar los centros de compra, Las grandes empresas pueden también abrir sucursales o oficinas en estos puntos para que sus empleados lejanos del norte no tengan problemas de transporte.

#### IV.5. Cálculo del grado de cumplimiento de la solución crisp obtenida

La solución crisp obtenida nos permite hacer una diferencia entre el conjunto de  $n-p$  nodos de demanda y el de  $p$  nodos de facilidad. Como se trata de dos conjuntos diferentes, una nueva numeración de los nodos se impone dentro de cada conjunto. Las estructuras de datos requeridas en los algoritmos presentados más arriba siendo los vectores y matrices, la nueva numeración irá de 0 a  $n-p-1$  para el conjunto de nodos de demanda y de 0 a  $p-1$  para el conjunto de nodos de facilidad.

Para calcular el grado de cumplimiento de la solución crisp obtenida con el programa correspondiente presentado en el capítulo IV, necesitamos el vector  $W[n-p]$  para los grados de pertenencia de los pesos de los  $n-p$  nodos, la matriz  $D[n-p][p]$  para los grados de pertenencia de las distancias entre los  $n-p$  nodos de demanda y los  $p$  nodos de facilidad y la matriz booleana  $X[n-p][p]$  de las asignaciones de los  $n-p$  nodos de demanda a los  $p$  nodos de facilidad.

Los  $n-p$  nodos de demanda y los grados de pertenencia de sus pesos se presentan como sigue, después de haberlos numerado de nuevo:

Nodo de demanda	Grado de cumplimiento del peso
0. FIKIN	0,8
1. LEMBA	0,6
2. DELVAUX	0,8
3. ROYAL	0,5
4. KALEMBELEMBE	0,8
5. BON MARCHE	0,7

Así la matriz  $W[6] = \{0.8, 0.6, 0.8, 0.5, 0.8, 0.7\}$

Ordenaremos los  $p$  nodos de facilidad de la manera siguiente:

1. PASCAL
2. N'DJILI
3. MATETE
4. RP NGABA
5. IPN
6. KINTAMBO
7. SELEMBAO
8. VICTOIRE
9. ZANDO
10. PONT-MATETE

Sabiendo que cada uno de los  $n-p$  nodos de demanda se asigna a uno de los  $p$  nodos de facilidad más cercano, construiremos la matriz de las asignaciones a partir de la tabla siguiente donde las filas representan los nodos de demanda, las columnas representan los nodos de facilidad y la intersección de la fila y la columna toma el valor 1 si el nodo correspondiente de la demanda se asigna al nodo correspondiente de la facilidad.

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	1	0

Esta matriz nos enseña, por ejemplo, que el nodo de demanda (0) FIKIN se asigna al nodo de facilidad (1) MATETE, el nodo de demanda (4) KALEMBELEMBE se asigna al nodo de facilidad (7) VICTOIRE, etc.

Todas las distancias tienen el grado de cumplimiento 0,7. Así, la matriz  $D[n-p][p]$  será obviamente:

	0	1	2	3	4	5	6	7	8	9
0	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
1	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
2	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
3	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
4	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
5	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7

A partir de la nueva situación impuesta por la solución óptima crisp, utilizaremos el programa C++ presentado en el capítulo IV al que añadimos algunas sentencias para la entrada de datos y que se presenta ahora como sigue:

```
//=====
//Fuzzy p-Median: Degree of certainty
//Author: David Kutangila Mayoya
//Date: Wednesday, November 24th, 2004
//Department of Computer Sciences and Artificial
//Intelligence
//University of Granada (Spain)
//=====
#include <iostream.h>
#include <stdlib.h>

main(){
    double L=0,L1;
    int C=0,C1, n, p;
    cout << "Fuzzy P-Median\n";
    cout << "\nNumber of nodes: ";
    cin >> n;
    cout << "\nNumber of facility nodes: ";
    cin >> p;
```



```

//Declaring input data
double W[n-p];
double D[n-p][p];
bool X[n-p][p];

//Entering demnads degrees of certainty
cout << "\nEnter demands degrees of certainty,";
for (int i=0; i<=n-p-1; i++){
    cout << "\nNode " << i << ": ";
    cin >> W[i];
}
//Entering distances degrees of certainty
cout << "\nEnter d[i][j]'s degrees of certainty,";
for (int i=0; i<=n-p-1; i++){
    for (int j=0; j<=p-1; j++){
        cout << "\nd[" << i << "],[" << j << "]: ";
        cin >> D[i][j];
    }
}

//Entering allocations values
cout << "\nEnter 1 if node I is allocated to j,";
cout << "\nand 0 otherwise\n";
for (int i=0; i<= n-p-1; i++){
    for (int j=0; j<=p-1; j++){
        cout << "\nd[" << i << "],[" << j << "]: ";
        cin >> X[i][j];
    }
}

//Processing data
for (int i=0; i<=n-p-1;){

```

```

    if (W[i]>=0 && W[i]<=1) {
        C1=0;
        L1=0;
        ++C;
        for (int j=0; j<=p-1;){
            if ((X[i][j])&&
                D[i][j]>=0 && D[i][j]<=1) {
                if (W[i]<D[i][j])
                    L1+=W[i];
                else L1+=D[i][j];
                ++C1;
            }
            ++j;
        }
        L += L1/C1;
    }
    ++i;
}
L /= C;
cout << "\nThe degree of certainty is " << L << endl;
system("PAUSE");
return 0;
}

```

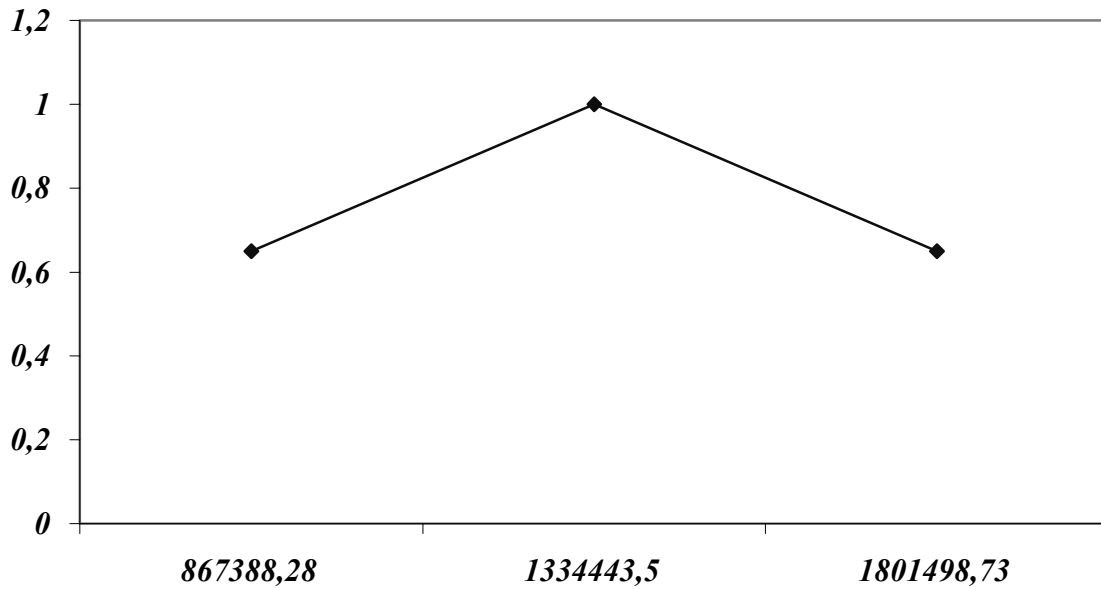
La ejecución de este programa nos da el resultado de 0.65 como grado de cumplimiento de la solución obtenida.

#### IV.6. Determinación del intervalo de coste óptimo

El grado de cumplimiento 0,65 indica que el coste de 1.334.443,5 km no es preciso. Si expresamos este grado en término de porcentaje, podemos decir que el coste se sitúa entre  $0,35 \times$

1.334.443,5 km más bajo y  $0,35 \times 1.334.443,5$  km más alto, es decir entre 867.388,28 km y 1.801.498,73 km.

***Intervalo del coste óptimo***



**IV.7. Centros de transporte**

Los 10 nodos de facilidad incorporarán los pesos de los nodos de demanda que se asignan a ellos y se presentarán de la manera siguiente:

***Nodos de facilidad y pesos***

Nodos de facilidad	Pesos	Grados de pertenencia
1. PASCAL	841941	0,70
2. N'DJILI	569281	0,80
3. MATETE	455533	0,80
4. RP NGABA	400963	0,65
5. VICTOIRE	569025	0,75
6. PONT MATETE	159503	0,60
7. ZANDO	211931	0,60
8. KINTAMBO	288449	0,70
9. IPN	390084	0,75
10. SELEMBAO	454644	0,8

3. MATETE ha incorporado el peso de FIKIN, 4. RP NGABA el peso de LEMBA, 5. VICTOIRE el peso de KALEMBELEMBE, 7. ZANDO los pesos de ROYAL y BON MARCHE, 9. IPN el peso de DELVAUX. Los grados de pertenencia se han calculado por la media aritmética.

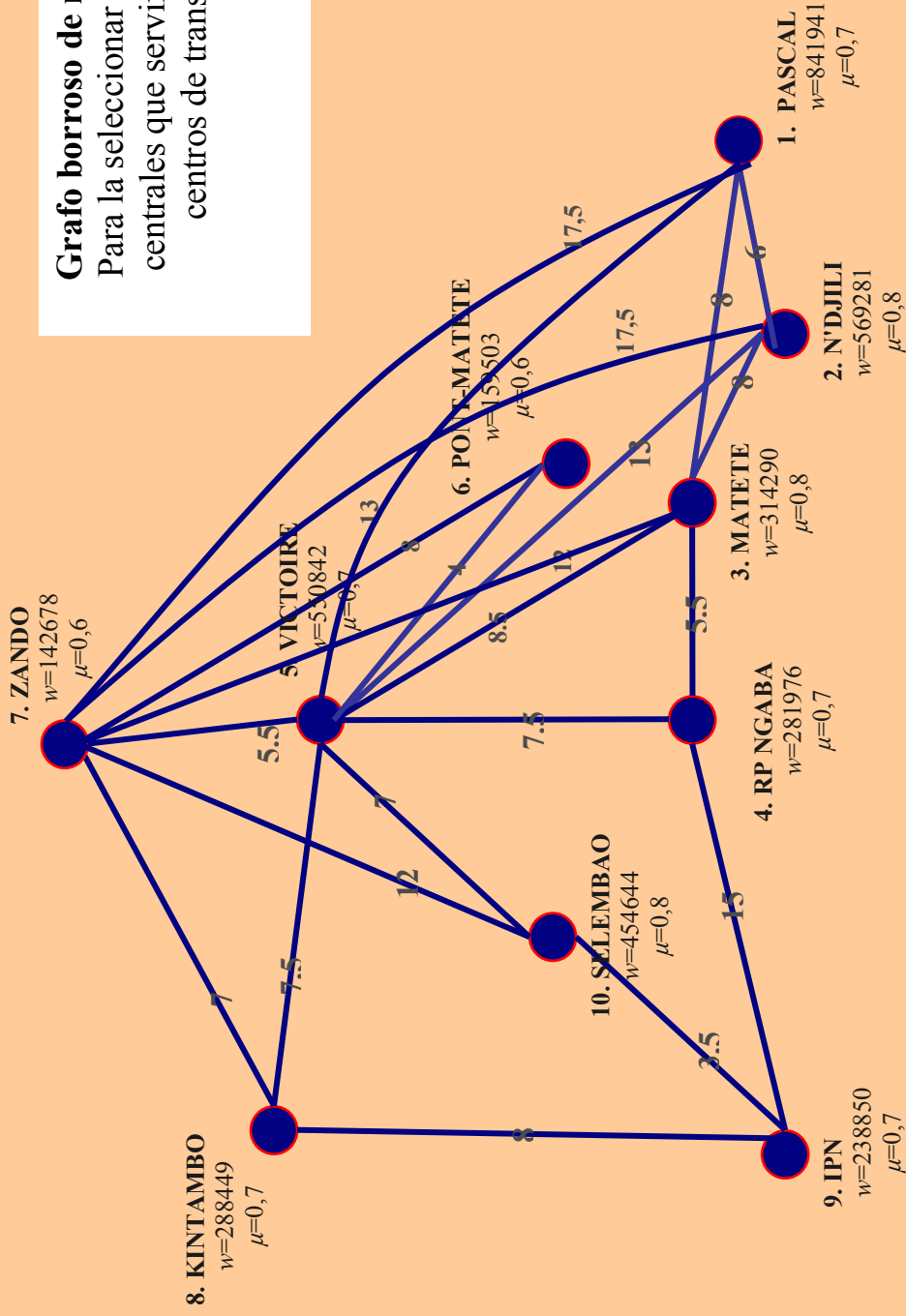
*Aristas entre los nodos de facilidad*

Arista	Longitud	Arista	Longitud	Arista	Longitud
1 – 2	6,00	3 – 4	5,50	5 – 8	7,50
1 – 3	8,00	3 – 5	8,50	5 – 10	7,00
1 – 5	13,00	3 – 7	12,00	6 – 7	8,00
1 – 7	17,50	4 – 5	7,50	7 – 8	7,00
2 – 3	8,00	4 – 9	15,00	7 – 10	12,00
2 – 5	13,00	5 – 6	4,00	8 – 9	8,00
2 – 7	17,50	5 – 7	5,00	9 - 10	3,50

Las aristas tienen todas el grado de pertenencia 0,7.

El grafo de los nodos de facilidad en el cual seleccionamos algunos centros de transporte se presenta como sigue:

**Grafo borroso de medianas**  
 Para la selección de medianas  
 centrales que servirán como  
 centros de transporte



Calculando la p-mediana sobre este nuevo grafo, con  $p = 2$  hasta 9, tenemos los resultados siguientes:

P	mediana	radio	tiempo/seg	tipo
2.00	2.12066E+7	10.50	4.00	OPTIMA
VERT-SOL 1 5				
3.00	1.52935E+7	8.00	1.00	OPTIMA
VERT-SOL 1 5 10				
4.00	1.08473E+7	7.50	1.00	OPTIMA
VERT-SOL 1 3 5 10				
5.00	7431625.00	7.50	1.00	OPTIMA
VERT-SOL 1 2 3 5 10				
6.00	5226328.50	7.50	1.00	OPTIMA
VERT-SOL 1 2 3 4 5 10				
7.00	3062961.00	5.00	2.00	OPTIMA
VERT-SOL 1 2 3 4 5 8 10				
8.00	1697667.00	5.00	1.00	OPTIMA
VERT-SOL 1 2 3 4 5 8 9 10				
9.00	638012.00	4.00	1.00	OPTIMA
VERT-SOL 1 2 3 4 5 7 8 9 10				

En este conjunto de soluciones, la localización óptima de 3 medianas centrales que servirán como centros de transporte se efectuará en los nodos:

1. PASCAL 5. VICTOIRE y 10. SELEMBAO con un coste óptimo o distancia total de recorrida de 15.295.700 km.

El grado de cumplimiento de esta solución es 0,66, sabiendo que las asignaciones de los demás nodos a los centros de transporte más cercanos se han hecho de manera siguiente:

- Al centro 1. PASCAL se han asignado los nodos 2. N'DJILI y 3. MATETE
- Al centro 5. VICTOIRE se han asignado los nodos 4. RP NGABA, 6. PONT METETE, 7. ZANDO y 8. KINTAMBO
- Al centro 10. SELEMBAO se ha asignado el nodo 9. IPN.

Dado el plano de rutas y la configuración geográfica de la ciudad de Kinshasa, podemos forzar la localización del tercer centro en el nodo 9. IPN en vez de 10. SELEMBAO.

Pero, dado la configuración geográfica y la red de transporte, podemos forzar la localización del tercer centro en el nodo 9. IPN en vez de 10. SELEMBAO

#### **IV.8. Determinación de líneas de autobuses fijas**

Gracias a las localizaciones de centros de transporte, podemos determinar rutas de modo que haya por lo mucho un trasbordo en el desplazamiento entre cualesquiera dos estaciones de autobús o cualesquiera dos centros comerciales.

Determinamos las rutas de manera siguiente:

- Línea 1: desde Masina Pascal hasta Binza IPN, pasando por Marché Matete y Rond-Point Ngaba;
- Línea 2: desde Masina Pascal hasta Victoire, pasando por Marché Matete y luego tomando el bulevar Lumumba.
- Línea 3: desde Victoire hasta IPN pasando por Marché Selembao, siempre que se mantenga el trozo de camino entre Selembao y IPN.
- Línea 4: desde IPN hasta Royal pasando por Kintambo Magasin
- Línea 5: desde Masina Pascal hasta Kimbanseke pasando por N'Djili Sainte-Thérèse;
- Línea 6: desde Victoire hasta Kimwenza pasando por Rond-Point Ngaba
- Línea 7: desde Victoire hasta Pont-Matete pasando por Kingabwa ICOM
- Línea 8: desde Victoire hasta Zando
- Línea 9: desde Victoire hasta 8. Kintambo;
- Línea 10: desde Victoire hasta Royal
- Línea 11: desde Masina Pascal hasta Maluku pasando por el aeropuerto Internacional de Kinshasa.

## **CONCLUSIÓN GENERAL Y DESARROLLO DE LÍNEAS DE INVESTIGACIÓN FUTURAS**

Una de las causas importantes de la drama del transporte público en Kinshasa detallada en la introducción general de esta memoria es la mala configuración de esta ciudad: concentración de actividades importantes como centros comerciales, empresas públicas y multinacionales etc. en el norte de la ciudad que es la zona administrativa de Gombe. Esto justifica los traficos desde toda la ciudad hacia el norte cada mañana y desde el norte hacia toda la ciudad por la tarde, con todas las consecuencias posibles como la congestión, viajes largos y costosos etc.

En este trabajo, hemos reflexionado sobre la mejor localización de los centros comerciales en cada uno de los cuales se agrega una estaciona de autobús en toda la ciudad de modo que se minimice la distancia total de viaje de todos los habitantes de Kinshasa desde sus casas a su centro comercial o estación de autobús más cercano. Este problema se refiere como el problema de la p-mediana. El carácter difuso de este problema se debe a dos hechos: primero la caducidad de los datos sobre la población y segundo la repartición arbitraria de esta población en los diferentes nodos retenidos en el diseño del grafo de Kinshasa.

Suponiendo que en una ciudad como Kinshasa, las actividades como las que se encuentran en el norte se pueden localizar en 10 otros puntos de la ciudad, tuvimos que escoger 3 de ellos como centros de transporte o centros de consolidación de tráfico. Esto se hace mediante un problema de centro sobre el que hemos reflexionado en el capítulo 3. Pero en el caso de que no se dispone de los datos sobre los tráfico entre cada par de nodos, este problema se resuelve calculando la p-mediana central como lo hemos hecho.

Los resultados obtenidos nos conducen a hacer siguientes propuestas para resolver el problema del transporte público en Kinshasa

- construir centros comerciales públicos del mismo tamaño que el mercado central Zando o construir y arrendar grandes edificios de venta, abrir oficinas o sucursales de empresas públicas y estaciones o parkings de autobús en los puntos siguientes:
  - Masina Pascal;
  - N'Djili Sainte-Thérèse;
  - Marché Matete (agrandar el mercado existentes);



- Rond-Point Ngaba;
  - Victoire (agrandar el mercado Gambela existente);
  - Pont-Matete (Precisamente en Kingabwa ICOM);
  - Zando (mantener el mercado existente);
  - Kintambo Magasin;
  - Binza IPN;
  - Marché Selembao (agrandar el mercado existente);
- Una gran ciudad como Kinshasa tiene solamente un centro de consolidación de tráfico, el parking de Victoire. Así sugerimos establecer como centros de transporte además del Victoire los puntos siguientes:
- Masina Pascal;
  - Binza IPN
- Sugerimos las líneas principales siguientes para el transporte público permitiendo el desplazamiento entre dos puntos cualesquiera de Kinshasa con un trasbordo por lo mucho:
- Línea 1: desde Masina Pascal hasta Binza IPN, pasando por Marché Matete y Rond-Point Ngaba;
  - Línea 2: desde Masina Pascal hasta Victoire, pasando por Marché Matete y luego tomando el bulevar Lumumba.
  - Línea 3: desde Victoire hasta IPN pasando por Marché Selembao, siempre que se mantenga el trozo de camino entre Selembao y IPN.
  - Línea 4: desde IPN hasta Royal pasando por Kintambo Magasin
  - Línea 5: desde Masina Pascal hasta Kimbanseke pasando por N'Djili Sainte-Thérèse;
  - Línea 6: desde Victoire hasta Kimwenza pasando por Rond-Point Ngaba
  - Línea 7: desde Victoire hasta Pont-Matete pasando por Kingabwa ICOM
  - Línea 8: desde Victoire hasta Zando
  - Línea 9: desde Victoire hasta 8. Kintambo;
  - Línea 10: desde Victoire hasta Royal
  - Línea 11: desde Masina Pascal hasta Maluku pasando por el aeropuerto Internacional de Kinshasa.

Por el presente, dado que a nivel nacional no existen carreteras para conectar a todas las provincias de la República Democrática del Congo, nuestra investigación posterior concernirá la factibilidad de los modelos de centro y routing en la construcción de esas carreteras.

## ANEXOS

### ANEXO I. ELEMENTOS DE CONJUNTOS BORROSOS [59]

#### I.1. Introducción

En la introducción general dijimos que la metodología extremadamente útil para solucionar una variedad de problemas como el del transporte público en Kinshasa es la proporcionada por los conjuntos y sistemas borrosos. Después de la descripción realizada sobre el ámbito en que hemos desarrollado el trabajo, debe haber quedado clara la utilidad de considerar las metodologías borrosas. Así, en este anexo presentamos los conceptos básicos sobre los subconjuntos borrosos a saber: la noción de pertenencia, el concepto de subconjunto borroso y algunas operaciones sencillas sobre los subconjuntos borrosos que son necesarios puesto que en lo largo de nuestro trabajo hemos hablado de los problemas borrosos de localización y de los grafos borrosos.

#### I.2. La noción de pertenencia

Sea  $E$  un conjunto y  $A$  un subconjunto de  $E$ :  $A \subset E$ .

Se suele indicar por el símbolo  $\in$  que un elemento  $x$  de  $E$  pertenece a  $A$ :  $x \in A$ .

Se puede también utilizar, para indicar esta pertenencia, otro concepto, el de “funcion de pertenencia”  $\mu_A(x)$  cuyo valor indica si sí o no,  $x$  pertenece a  $A$ :

$$\begin{aligned}\mu_A(x) &= 1 \text{ si } x \in A \\ &= 0 \text{ si } x \notin A\end{aligned}$$

Sea este ejemplo de un conjunto finito de 5 elementos:

$$E = \{x_1, x_2, x_3, x_4, x_5\}$$

y

$$A = \{x_2, x_3, x_5\}$$

Se escribirá:

$$\mu_A(x_1) = 0, \mu_A(x_2) = 1, \mu_A(x_3) = 1, \mu_A(x_4) = 0, \mu_A(x_5) = 1.$$

Este nos permitirá presentar  $A$  acompañando los elementos de  $E$  del valor de la función característica:

$$A = \{(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 0), (x_5, 1)\}.$$

### I.3. Concepto de subconjunto borroso

Consideremos el subconjunto  $A$  de  $E$  definido en la sección precedente. Los cinco elementos de  $E$  pertenecen o no pertenecen a  $A$ , es el uno o el otro. La función característica toma solamente el valor 0 o el valor 1.

Imaginemos ahora que esta función característica pueda tomar un cualquier valor en el intervalo  $[0,1]$ . Así, un elemento  $x_i$  de  $E$  podría no pertenecer a  $A$  ( $\mu_A(x_i) = 0$ ), pertenecer un poco a  $A$  ( $\mu_A(x_i)$  cerca de 0), pertenecer bastante a  $A$  ( $\mu_A(x_i)$  no muy cerca de 0 y no muy cerca de 1), pertenecer fuertemente a  $A$  ( $\mu_A(x_i)$  cerca de 1).

El concepto matemático definido por la expresión siguiente:

$$A_{\%} = \{(x_1 | 0,2), (x_2 | 0), (x_3 | 0,3), (x_4 | 1), (x_5 | 0,8)\}$$

donde  $x_i$  es un elemento del referencial  $E$  y donde el número puesto después de la barra es el valor de la función característica de este elemento, se denominará “subconjunto borroso” de  $E$  y se denotará por

$$A_{\%} \subset E \text{ o también } A_{\%} \subsetneq E.$$

La pertenencia a un subconjunto borroso se podría denotar por:

$$x \in_{0,2 \%} A, y \in_{1 \%} A, z \in_{0 \%} A$$

El símbolo  $\in_1$  pudiendo equivalerse a  $\in$  para indicar la pertenencia y  $\in_0$  a  $\notin$  para indicar la no pertenencia.

Así, el subconjunto borroso definido más arriba contiene un poco de  $x_1$ , no contiene  $x_2$ , contiene un poquito más de  $x_3$ , contiene todo el  $x_4$ , y contiene una gran parte de  $x_5$ . Este permitirá construir una estructura matemática con la cual se podrá manipular conceptos mal definidos pero cuya pertenencia a subconjuntos ha podido ser jerarquizada. Así, se puede considerar, en el conjunto de los hombres, el subconjunto borroso de hombres más altos; en el conjunto de colores básicos, el subconjunto borroso de colores verde oscuro; un subconjunto borroso de buenas decisiones en el conjunto de decisiones, etc. Se verá cómo manipular estos

conceptos que parecen particularmente bien adaptados a la imprecisión que reina en las ciencias humanas.

Demos ahora una definición rigurosa de este concepto. Sea  $E$  un conjunto numerable o no y  $x$  un elemento de  $E$ , entonces un “subconjunto borroso”  $A_{\alpha}$  de  $E$  es un conjunto de parejas

$$\{(x | \mu_{A_{\alpha}}(x))\}, \forall x \in E.$$

donde  $\mu_{A_{\alpha}}(x)$  es el “grado de pertenencia” de  $x$  en  $A_{\alpha}$ . Así, si  $\mu_{A_{\alpha}}(x)$  toma sus valores en un conjunto  $M$  denominado “conjunto de pertenencia”, se puede decir que  $x$  toma sus valores en  $M$  por la función  $\mu_{A_{\alpha}}(x)$ , sea:

$$x \xrightarrow{\mu_{A_{\alpha}}} M$$

Esta función será también llamada “función de pertenencia”.

Para considerar las funciones binarias booleanas como casos particulares de estas funciones de pertenencia, Kaufmann reemplaza la definición precedente por la siguiente:

Sea  $E$  un conjunto numerable o no y  $x$  un elemento de  $E$ , entonces un “subconjunto borroso”  $A_{\alpha}$  de  $E$  es un conjunto de parejas

$$\{(x, \mu_{A_{\alpha}}(x))\}, \forall x \in E.$$

donde  $\mu_{A_{\alpha}}(x)$  es una “función característica de pertenencia” que toma sus valores en un conjunto totalmente ordenado  $M$  que indica el “grado” o “nivel” de pertenencia. Si  $M = \{0,1\}$ , el conjunto borroso  $A_{\alpha}$  vuelve a ser un “subconjunto no borroso” o “subconjunto vulgar”. Las funciones  $\mu_{A_{\alpha}}(x)$  son entonces funciones binarias booleanas.

Así, la noción de subconjunto borroso está vinculada a la noción de conjunto y permite estudiar, mediante estructuras matemáticas, conceptos imprecisos.

Un conjunto y un subconjunto se denotan con una letra y un subconjunto borroso se denota con una letra con signo  $\alpha$  : debajo de dicha letra.

Así:

$$\frac{A}{\%}, \frac{X}{\%}, \frac{a}{\%}, \frac{p}{\%}$$

representarán unos conjuntos borrosos. Cuando tales subconjuntos vuelvan a ser vulgares, desaparecerá el signo : suplementario.

La pertenencia y la no pertenencia se indican por:

$$\in \text{ y } \notin,$$

La “pertenencia borrosa” y la “no pertenencia borrosa” se representarán por:

$$\in_{\%} \text{ y } \notin_{\%} \text{ si es necesario.}$$

En algunos casos donde el conjunto  $M$  totalmente ordenado, en el cual  $\mu_{\frac{A}{\%}}(x)$  toma sus valores, es el intervalo cerrado  $[0,1]$ , puede ser cómodo acompañar el símbolo  $\in$  por un número de  $[0,1]$  puesto abajo. Así:

$$\begin{aligned} x \in_{\frac{1}{\%}} A \text{ significa } x \in_{\frac{1}{\%}} A, \text{ es decir "x pertenece a } A\text{"}. \\ x \in_{\frac{0}{\%}} A \text{ significa } x \notin_{\frac{0}{\%}} A, \text{ es decir "x no pertenece a } A\text{"}. \\ x \in_{\frac{0,8}{\%}} A \text{ significa que } x \text{ pertenece a } A \text{ con el grado } 0,8. \end{aligned}$$

Ejemplo: Sea un conjunto finito:

$$E = \{a, b, c, d, e, f\}$$

y el conjunto ordenado:

$$M = \{0, 1/2, 1\}.$$

Eso es un subconjunto borroso de  $E$ :

$$\frac{A}{\%} = \{(a|0), (b|1), (c|\frac{1}{2}), (d|0), (e|\frac{1}{2}), (f|0)\}.$$

Se escribirá:

$$a \in_{\frac{0}{\%}} A, \quad b \in_{\frac{1}{\%}} A, \quad c \in_{\frac{1}{2}\%} A, \quad d \in_{\frac{0}{\%}} A \text{ etc.}$$

## I.4. Algunas operaciones sencillas sobre los subconjuntos borrosos

### I.4.1. Inclusión

Sean  $E$  un conjunto y  $M$  su conjunto asociado de pertenencia,  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  dos subconjuntos borrosos de  $E$ ; Diremos que  $A_{\frac{\circ}{\circ}}$  está incluido en  $B_{\frac{\circ}{\circ}}$  si:

$$\forall x \in E : \mu_{\frac{\circ}{\circ}}^A(x) \leq \mu_{\frac{\circ}{\circ}}^B(x).$$

Se escribirá entonces:

$$A_{\frac{\circ}{\circ}} \subset B_{\frac{\circ}{\circ}}$$

Y, si es necesario, para evitar una confusión:

$$A_{\frac{\circ}{\circ}} \subset_{\frac{\circ}{\circ}} B_{\frac{\circ}{\circ}};$$

lo que precisará que se trata bien de una inclusión en el sentido de la teoría de los subconjuntos borrosos.

La inclusión estricta, correspondiendo al caso donde al menos una relación es estricta, se denotará:

$$A_{\frac{\circ}{\circ}} \subset\subset B_{\frac{\circ}{\circ}} \text{ o } A_{\frac{\circ}{\circ}} \subset\subset_{\frac{\circ}{\circ}} B_{\frac{\circ}{\circ}}$$

Ejemplo: Sean

$$E = \{x_1, x_2, x_3, x_4\}, M = [0, 1].$$

$$A_{\frac{\circ}{\circ}} = \{(x_1 | 0, 4), (x_2 | 0, 2), (x_3 | 0), (x_4 | 1)\}.$$

$$B_{\frac{\circ}{\circ}} = \{(x_1 | 0, 3), (x_2 | 0), (x_3 | 0), (x_4 | 0)\}.$$

Tenemos:

$$B_{\frac{\circ}{\circ}} \subset A_{\frac{\circ}{\circ}} \text{ porque } 0,3 < 0,4, 0 < 0,2, 0 = 0, 0 < 1.$$

Así,  $E$  está incluido en él mismo en el sentido de la teoría de los conjuntos borrosos:

$$E \subset_{\frac{\circ}{\circ}} E.$$

Esta propiedad queda verdad cualquier sea  $E$ .

#### **I.4.2. Igualdad**

Sean  $E$  un conjunto y  $M$  su conjunto asociado de pertenencia,  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  dos subconjuntos borrosos de  $E$ ; Diremos que  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  son iguales si y solamente si:

$$\forall x \in E : \mu_{\frac{\circ}{\circ}}^A(x) = \mu_{\frac{\circ}{\circ}}^B(x).$$

Se escribirá entonces:

$$A_{\frac{\circ}{\circ}} = B_{\frac{\circ}{\circ}}.$$

Si un  $x$  de  $E$  es tal como la igualdad  $\mu_{A_{\frac{\circ}{\circ}}}(x) = \mu_{B_{\frac{\circ}{\circ}}}(x)$  no está satisfecha; diremos que  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  son desiguales, y se denotará por:

$$A_{\frac{\circ}{\circ}} \neq B_{\frac{\circ}{\circ}}$$

### I.4.3. Complementación

Sean  $E$  un conjunto y  $M$  su conjunto asociado de pertenencia,  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  dos subconjuntos borrosos de  $E$ ; Diremos que  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  son complementarios si:

$$\forall x \in E : \mu_{B_{\frac{\circ}{\circ}}}(x) = 1 - \mu_{A_{\frac{\circ}{\circ}}}(x).$$

Se escribirá entonces:

$$B_{\frac{\circ}{\circ}} = \overline{A_{\frac{\circ}{\circ}}} \text{ o } \overline{A_{\frac{\circ}{\circ}}} = B_{\frac{\circ}{\circ}}.$$

Obviamente, tenemos siempre:

$$\overline{\left(\overline{A_{\frac{\circ}{\circ}}}\right)} = A_{\frac{\circ}{\circ}}.$$

Ejemplo:

$$E = \{x_1, x_2, x_3, x_4, x_5, x_6\}, M = [0, 1].$$

$$A_{\frac{\circ}{\circ}} = \{(x_1 | 0, 13), (x_2 | 0, 61), (x_3 | 0), (x_4 | 0), (x_5 | 1), (x_6 | 0, 03)\}.$$

$$B_{\frac{\circ}{\circ}} = \{(x_1 | 0, 87), (x_2 | 0, 39), (x_3 | 1), (x_4 | 1), (x_5 | 0), (x_6 | 0, 97)\}.$$

Tenemos:

$$\overline{A_{\frac{\circ}{\circ}}} = B_{\frac{\circ}{\circ}}$$

### I.4.4. Intersección

Sean  $E$  un conjunto y  $M$  su conjunto asociado de pertenencia,  $A_{\frac{\circ}{\circ}}$  y  $B_{\frac{\circ}{\circ}}$  dos subconjuntos borrosos de  $E$ ; la intersección se definirá:

$$A_{\frac{\circ}{\circ}} \cap B_{\frac{\circ}{\circ}}$$

por el subconjunto borroso más grande contenido, a la vez, en  $A_{\frac{\circ}{\circ}}$  y en  $B_{\frac{\circ}{\circ}}$ . Es decir:



$$\forall x \in E : \mu_{\frac{A}{\%} \cap \frac{B}{\%}}(x) = \min(\mu_{\frac{A}{\%}}(x), \mu_{\frac{B}{\%}}(x)).$$

Ejemplo:

$$E = \{x_1, x_2, x_3, x_4, x_5\}, M = [0, 1].$$

$$\frac{A}{\%} = \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 1), (x_4 | 0), (x_5 | 0, 5)\}.$$

$$\frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 3), (x_3 | 1), (x_4 | 0, 1), (x_5 | 0, 5)\}.$$

$$\frac{A}{\%} \cap \frac{B}{\%} = \{(x_1 | 0, 2), (x_2 | 0, 3), (x_3 | 1), (x_4 | 0), (x_5 | 0, 5)\}.$$

Esto se puede también escribir:

$$\forall x \in \frac{A}{\%} : x \in \frac{A}{\%} \text{ y } x \in \frac{B}{\%} \Rightarrow x \in \frac{A}{\%} \cap \frac{B}{\%}$$

Lo que permite introducir un “y” borroso, o simbólicamente “ $\frac{y}{\%}$ ”.

#### I.4.5. Unión

Sean  $E$  un conjunto y  $M$  su conjunto asociado de pertenencia,  $\frac{A}{\%}$  y  $\frac{B}{\%}$  dos subconjuntos borrosos de  $E$ ; la unión se definirá:

$$\frac{A}{\%} \cup \frac{B}{\%},$$

por el subconjunto borroso más pequeño que contenga, a la vez,  $\frac{A}{\%}$  y  $\frac{B}{\%}$ . Es decir:

$$\forall x \in E : \mu_{\frac{A}{\%} \cup \frac{B}{\%}}(x) = \max(\mu_{\frac{A}{\%}}(x), \mu_{\frac{B}{\%}}(x)).$$

Ejemplo:

$$E = \{x_1, x_2, x_3, x_4, x_5\}, M = [0, 1].$$

$$\frac{A}{\%} = \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 1), (x_4 | 0), (x_5 | 0, 5)\}.$$

$$\frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 3), (x_3 | 1), (x_4 | 0, 1), (x_5 | 0, 5)\}.$$

$$\frac{A}{\%} \cup \frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 7), (x_3 | 1), (x_4 | 0, 1), (x_5 | 0, 5)\}.$$

Esto se puede también escribir:

$$\forall x \in \frac{A}{\%} : x \in \frac{A}{\%} \text{ y/o } x \in \frac{B}{\%} \Rightarrow x \in \frac{A \cup B}{\%}$$

Lo que permite introducir un “y/o” borroso, o simbólicamente “y/o”.

#### I.4.6. Suma disyuntiva

La suma disyuntiva de dos subconjuntos borrosos se definirá a partir de la unión y de la intersección de la siguiente manera:

$$\frac{A}{\%} \oplus \frac{B}{\%} = (\frac{A}{\%} \cap \overline{\frac{B}{\%}}) \cup (\overline{\frac{A}{\%}} \cap \frac{B}{\%})$$

Esta operación corresponderá al “o disyuntivo borroso” o “o” y se leerá “o”, y podrá escribirse “o” sin riesgo de equivocación.

Ejemplo:

$$\frac{A}{\%} = \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 1), (x_4 | 0), (x_5 | 0, 5)\}$$

$$\frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 3), (x_3 | 1), (x_4 | 0, 1), (x_5 | 0, 5)\}$$

$$\overline{\frac{A}{\%}} = \{(x_1 | 0, 8), (x_2 | 0, 3), (x_3 | 0), (x_4 | 1), (x_5 | 0, 5)\}$$

$$\overline{\frac{B}{\%}} = \{(x_1 | 0, 5), (x_2 | 0, 7), (x_3 | 0), (x_4 | 0, 9), (x_5 | 0, 5)\}$$

$$\frac{A}{\%} \cap \overline{\frac{B}{\%}} = \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 0), (x_4 | 0), (x_5 | 0, 5)\}$$

$$\overline{\frac{A}{\%}} \cap \frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 3), (x_3 | 0), (x_4 | 0, 1), (x_5 | 0, 5)\}$$

$$\frac{A}{\%} \oplus \frac{B}{\%} = \{(x_1 | 0, 5), (x_2 | 0, 7), (x_3 | 0), (x_4 | 0, 1), (x_5 | 0, 5)\}$$

#### I.4.7. Diferencia

La diferencia se definirá por la relación:

$$\frac{A}{\%} - \frac{B}{\%} = \frac{A}{\%} \cap \overline{\frac{B}{\%}}$$

Ejemplo:

$$\begin{aligned}
A_{\%} &= \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 1), (x_4 | 0), (x_5 | 0, 5)\}. \\
B_{\%} &= \{(x_1 | 0, 5), (x_2 | 0, 3), (x_3 | 1), (x_4 | 0, 1), (x_5 | 0, 5)\}. \\
\overline{A}_{\%} &= \{(x_1 | 0, 8), (x_2 | 0, 3), (x_3 | 0), (x_4 | 1), (x_5 | 0, 5)\}. \\
\overline{B}_{\%} &= \{(x_1 | 0, 5), (x_2 | 0, 7), (x_3 | 0), (x_4 | 0, 9), (x_5 | 0, 5)\}. \\
A_{\%} - B_{\%} &= A_{\%} \cap \overline{B}_{\%} = \{(x_1 | 0, 2), (x_2 | 0, 7), (x_3 | 0), (x_4 | 0), (x_5 | 0, 5)\}.
\end{aligned}$$

Excepto el caso particular:

$$A_{\%} - B_{\%} \neq B_{\%} - A_{\%}$$

### I.5. Algunas operaciones de la lógica borrosa

Sean  $a = \mu_A(x)$ ,  $b = \mu_B(x)$ , dos variables booleanas.

Se sabe que la correspondencia entre las operaciones de la teoría de los conjuntos y las del álgebra binaria de Boole se presenta de la manera siguiente:

Operaciones sobre subconjuntos vulgares	Operaciones correspondientes sobre variables booleanas
$A \cap B$	$a \cdot b$
$A \cup B$	$a + b$
$\overline{A} = C_E A$	$\overline{a}$
$A \oplus B = (\overline{A} \cap B) \cup (A \cap \overline{B})$	$a \oplus b = \overline{a} \cdot b + a \cdot \overline{b}$

Estas principales correspondencias concernirán también las variables borrosas definidas por la función de pertenencia  $M = [0, 1]$ .

Sean  $a_{\%} = \mu_{A_{\%}}(x)$ ,  $b_{\%} = \mu_{B_{\%}}(x)$ , dos variables borrosas. Definiremos las operaciones siguientes sobre  $a_{\%}, b_{\%}, \dots$

$$\begin{aligned}
 a \wedge b &= \min(a, b) \\
 a \vee b &= \max(a, b) \\
 \bar{a} &= 1 - a \\
 a \oplus b &= (a \wedge \bar{b}) \vee (\bar{a} \wedge b)
 \end{aligned}$$

sabiendo que:

$$\begin{aligned}
 &\left. \begin{aligned} a \wedge b &= b \wedge a \\ a \vee b &= b \vee a \end{aligned} \right\} \text{comutatividad} \\
 &\left. \begin{aligned} (a \wedge b) \wedge c &= a \wedge (b \wedge c) \\ (a \vee b) \vee c &= a \vee (b \vee c) \end{aligned} \right\} \text{asociatividad} \\
 &\left. \begin{aligned} a \wedge a &= a \\ a \vee a &= a \end{aligned} \right\} \text{idempotencia} \\
 &\left. \begin{aligned} a \wedge (b \vee c) &= (a \wedge b) \vee (a \wedge c) \\ a \vee (b \wedge c) &= (a \vee b) \wedge (a \vee c) \end{aligned} \right\} \text{distributividad} \\
 &a \wedge 0 = 0 \\
 &a \vee 0 = a \\
 &a \wedge 1 = a \\
 &a \vee 1 = 1 \\
 &\bar{\bar{a}} = a \\
 &\left. \begin{aligned} \overline{a \wedge b} &= \bar{a} \vee \bar{b} \\ \overline{a \vee b} &= \bar{a} \wedge \bar{b} \end{aligned} \right\} \text{Teoremas de De Morgan generalizados en el caso donde } M = [0, 1]
 \end{aligned}$$

Todas estas propiedades son las del álgebra binario de Boole, salvo las dos siguientes:

$$a \cdot \bar{a} = 0 \quad \text{y} \quad a + \bar{a} = 1$$

para las cuales las expresiones correspondientes siguientes no son satisfechas:

$$\begin{aligned}
 a \wedge \bar{a} &\neq 0, \quad \text{salvo para } a = 0 \quad \text{o} \quad a = 1 \\
 a \vee \bar{a} &\neq 1, \quad \text{salvo para } a = 0 \quad \text{o} \quad a = 1
 \end{aligned}$$

Las variables  $a, b, \dots$  pueden someterse a otras operaciones diferentes de  $(\wedge), (\vee)$  y  $(\bar{\quad})$  denominadas operaciones mixtas:

- Multiplicación de dos variables borrosas:

$$a_{\circ} \cdot b_{\circ} = a_{\circ} \times b_{\circ}$$

Ejemplo: si  $a_{\circ} = 0,25$  y  $b_{\circ} = 0,50$ , entonces  $a_{\circ} \cdot b_{\circ} = 0,25 \times 0,50 = 0,125$

- Suma de dos variables borrosas:

$$a_{\circ} \hat{+} b_{\circ} = a_{\circ} + b_{\circ} - a_{\circ} \cdot b_{\circ}$$

Ejemplo: si  $a_{\circ} = 0,25$  y  $b_{\circ} = 0,50$ ,

$$\text{entonces } a_{\circ} \hat{+} b_{\circ} = 0,25 + 0,50 - 0,25 \times 0,50 = 0,625$$

### I.6. Subconjuntos borrosos de nivel $\alpha$ ( $\alpha$ cortes)

Sea  $\alpha \in [0,1]$ , el subconjunto vulgar  $A_{\alpha} = \{x \mid \mu_{A_{\circ}}(x) \geq \alpha\}$  se denominará “subconjunto vulgar de nivel  $\alpha$ ”.

Sea:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$A_{\circ} =$	0,8	0,1	1	0,3	0,6	0,2	0,5

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$A_{0,3} =$	1	0	1	1	1	0	1

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$A_{0,55} =$	1	0	1	0	1	0	0

**Teorema de descomposición:** Todo subconjunto borroso  $A_{\circ}$  se puede descomponer de la forma siguiente donde aparecen los productos de subconjuntos vulgares (sus funciones características) por los coeficientes  $\alpha_i$ :

$$A_{\circ} = \max_{\alpha_i} [\alpha_1 \cdot A_{\alpha_1}, \alpha_2 \cdot A_{\alpha_2}, \dots, \alpha_n \cdot A_{\alpha_n}] \quad 0 < \alpha_i \leq 1 \quad i = 1, 2, \dots, n$$

Este teorema se refiere también como teorema de representación.

La demostración es inmediata:

$$\begin{aligned} \mu_{A_{\alpha_i}}(x) &= 1 & \text{si } \mu_{\frac{A}{\alpha}}(x) \geq \alpha_i \\ &= 0 & \text{si } \mu_{\frac{A}{\alpha}}(x) < \alpha_i \end{aligned}$$

La función de pertenencia de  $\frac{A}{\alpha}$  se puede escribir:

$$\begin{aligned} \mu(x) &= \max_{\alpha_i \in \{ \alpha_i \mid \mu_{\frac{A}{\alpha}}(x) \geq \alpha_i \}} [\alpha_i, \mu_{\frac{A}{\alpha}}(x)] \\ &= \max_{\alpha_i \leq \mu_{\frac{A}{\alpha}}(x)} [\alpha_i] \\ &= \mu_{\frac{A}{\alpha}}(x). \end{aligned}$$

## ANEXO II. CONCEPTOS SOBRE GRAFOS [23]

### II.1. Introducción

En la introducción general dijimos que los objetivos de nuestro estudio eran, además de establecer una serie de centros comerciales repartidos por toda la ciudad, establecer una red de autobuses, siendo la metodología para alcanzar estos objetivos el problema de la  $p$ -mediana. Hablar de la red significa hablar del grafo, y el problema de la  $p$ -mediana como lo hemos visto se aplica sobre un grafo. Así, en este anexo, abordamos algunos conceptos sobre grafos, es decir, los conceptos básicos y algunas definiciones, las características de los grafos no dirigidos, las características de los grafos dirigidos, los grafos triangulados, los grafos de aglomerados, la representación de grafos y algunos algoritmos para grafos y por fin la noción de grafo borroso.

### II.2. Conceptos básicos y definiciones

Un grafo es un par de conjuntos  $G = (X, L)$  donde  $X = \{X_1, X_2, \dots, X_n\}$  es un conjunto finito de elementos (nodos), y  $L$  es un conjunto de aristas, es decir, un subconjunto de pares ordenados de elementos distintos de  $X$ . Los términos grafo y red se emplean como sinónimos.

Dado un grafo  $G = (X, L)$ , si  $L_{ij} \in L$  y  $L_{ji} \notin L$ , la arista  $L_{ij}$  entre los nodos  $X_i$  y  $X_j$  se denomina dirigida y se denota mediante  $X_i \rightarrow X_j$ .

Dado un grafo  $G = (X, L)$ , si  $L_{ij} \in L$  y  $L_{ji} \in L$ , la arista  $L_{ij}$  se denomina no dirigida y se denota mediante  $X_i - X_j$  o  $X_j - X_i$ .

Un grafo en el cual todas las aristas son dirigidas se denomina grafo dirigido, y un grafo en el que todas sus aristas son no dirigidas se denomina no dirigido.

Dado un grafo  $G = (X, L)$  y un nodo  $X_i$ , el conjunto adyacente del nodo  $X_i$  es el conjunto de nodos que son directamente alcanzables desde  $X_i$ , es decir

$$Ady(X_i) = \{X_j \in X \mid L_{ij} \in L\}.$$

Un camino del nodo  $X_i$  al nodo  $X_j$  es una sucesión de nodos  $(X_{i_1}, \dots, X_{i_r})$ , comenzando en  $X_{i_1} = X_i$  y finalizando en  $X_{i_r} = X_j$ , de forma que existe una arista del nodo  $X_{i_k}$  al nodo  $X_{i_{k+1}}$ ,  $k = 1, \dots, r-1$ , es decir

$$X_{i_{k+1}} \in Ady(X_{i_k}), \quad k = 1, \dots, r-1.$$

Un camino  $(X_{i_1}, \dots, X_{i_r})$  se dice que es cerrado si el nodo inicial coincide con el final, es decir,  $X_{i_1} = X_{i_r}$ .

### II.3. Características de los grafos no dirigidos

Un grafo no dirigido se denomina completo si tiene una arista entre cada par de nodos. Un subconjunto de nodos  $S$  de un grafo  $G$  se denomina completo si existe una arista en  $G$  para cada par de nodos en  $S$ .

Un conjunto completo de nodos  $C$  se denomina un conglomerado si no es subconjunto propio de otro conjunto completo, es decir, si es maximal.

Un bucle es un camino cerrado en un grafo no dirigido.

El conjunto de nodos adyacentes a un nodo  $X_i$  en un grafo no dirigido se denomina conjunto de vecinos de  $X_i$ .

$$Vec(X_i) = \{X_j \mid X_j \in Ady(X_i)\}.$$

La unión de los conjuntos de vecinos de los nodos de un conjunto dado,  $S$ , excluyendo los nodos de  $S$ , se denomina la frontera de  $S$  y se denota por  $Frn(S)$ .

$$Frn(S) = \left( \bigcup_{X_i \in S} Vec(X_i) \right) \setminus S$$



donde  $X \setminus S$  es el conjunto de nodos de  $X$  excluyendo los de  $S$ .

Un grafo no dirigido se denomina conexo si existe al menos un camino entre cada par de nodos. En caso contrario, el grafo se denomina inconexo.

Un grafo conexo no dirigido se denomina un árbol si existe un único camino entre cada par de nodos.

Un grafo conexo se denomina múltiplemente conexo si contiene al menos un par de nodos que estén unidos por más de un camino o, equivalentemente, si contiene al menos un bucle.

#### II.4. Características de los grafos dirigidos

Cuando existe una arista dirigida,  $X_i \rightarrow X_j$ , del nodo  $X_i$  al nodo  $X_j$ , entonces se dice que el nodo  $X_i$  es un padre del nodo  $X_j$ , y que el nodo  $X_j$  es un hijo de  $X_i$ .

El conjunto formado por un nodo y sus padres se denomina familia del nodo.

Un nodo  $X_j$  se denomina ascendiente del nodo  $X_i$  si existe un camino de  $X_j$  a  $X_i$ .

Un conjunto de nodos  $S$  se denomina un conjunto ancestral si contiene los ascendientes de todos sus nodos.

Un nodo  $X_j$  se denomina descendiente del nodo  $X_i$  si existe un camino de  $X_i$  a  $X_j$ .

Dado un conjunto  $X = \{X_1, \dots, X_n\}$  de nodos, una ordenación,  $\alpha$ , es una bisección que asigna un número del conjunto  $\{1, \dots, n\}$  a cada nodo:

$$\alpha: \{1, \dots, n\} \rightarrow \{X_1, \dots, X_n\}.$$

Por tanto,  $\alpha(i)$  denota el  $i$ -ésimo nodo de la numeración. Una numeración puede representarse mediante la sucesión ordenada de nodos  $(\alpha(1), \dots, \alpha(n))$ .

Una numeración de los nodos de un grafo dirigido se denomina ancestral si el número correspondiente a cada nodo es menor que los correspondientes a sus hijos.

Dado un grafo dirigido, el grafo no dirigido obtenido al reemplazar cada arista dirigida del grafo por la correspondiente arista no dirigida se denomina el grafo no dirigido asociado.

El grafo no dirigido asociado al grafo dirigido que se obtiene al añadir una arista entre cada par de nodos con algún hijo común en un grafo no dirigido, se denomina grafo moral asociado a dicho grafo.

Un ciclo es un camino cerrado en un grafo dirigido.

Un grafo dirigido se denomina conexo si el grafo no dirigido asociado es conexo; en caso contrario se denomina inconexo.

Un grafo dirigido conexo se denomina árbol si el grafo no dirigido asociado es un árbol: en caso contrario se denomina múltiplemente conexo.

Un grafo dirigido se denomina cíclico si contiene al menos un ciclo; en caso contrario se denomina grafo dirigido acíclico.

Un árbol dirigido se denomina un árbol simple si cada nodo tiene como máximo un padre: en caso contrario se denomina poliárbol.

## II.5. Grafos triangulados

Una cuerda es una arista que une dos nodos de un bucle y que no pertenece al bucle.

Un grafo se denomina triangulado, o cordal, si cada bucle de longitud mayor a igual que cuatro contiene al menos una cuerda.

Una numeración de nodos de un grafo,  $\alpha$ , se denomina perfecta si el subconjunto de nodos  $Frn(\alpha(i)) \cap \{\alpha(1), \dots, \alpha(i-1)\}$  es completo para  $i = 2, \dots, n$ .

El teorema de triangulación y numeración perfecta se enuncia de la siguiente manera: Un grafo no dirigido admite una numeración perfecta si y solo si es triangulado.

El algoritmo de máxima cardinalidad siguiente genera una numeración de los nodos del grafo sólo en caso de que el grafo esté triangulado:

- Datos: Un grafo no dirigido  $G = (X, L)$  y un nodo inicial  $X_i$ .
- Resultados: Una numeración  $\alpha$  de los nodos de  $X$ .
  1. *Iniciación*: Asignar el primer número al nodo inicial, es decir,  $\alpha(1) = X_i$ .
  2. Repetir la etapa siguiente con  $i = 2, \dots, n$ .
  3. *Iteración  $i$* : En la  $i$ -ésima etapa de iteración, se asigna el número  $i$  a cada nodo que no haya sido numerado previamente y que tenga el máximo número de vecinos numerados. Los empates se resuelven de forma arbitraria.

El pseudocódigo para el algoritmo de máxima cardinalidad sigue:

Búsqueda de Máxima Cardinalidad

Datos: Un grafo  $G = (X, L)$  y un nodo inicial  $X_i$

Resultado: Una numeración  $\alpha$  de los nodos en  $X$ .

---

Etapa Inicial:

$\alpha(1) \leftarrow X_i$

$Numerados \leftarrow \{X_i\}$

Etapa Iterativa:

for  $i = 2$  to  $n$

$X_k \leftarrow$  elige un nodo  $X_k$  en  $X \setminus Numerados$   
con máximo  $|Vec(X_k) \cap Numerados|$

$\alpha(i) \leftarrow X_k$

añade  $X_k$  a  $Numerados$

El teorema de numeración de máxima cardinalidad se enuncia como sigue:

Cualquier numeración de los nodos de un grafo triangulado obtenida aplicando el algoritmo de máxima cardinalidad es una numeración perfecta.

Por tanto, cuando la numeración generada por el algoritmo de máxima cardinalidad no sea perfecta, significará que el grafo no será triangulado. De esta forma, se puede modificar fácilmente el algoritmo de máxima cardinalidad para comprobar si un grafo es triangulado. Cuando el grafo no sea triangulado, entonces el propio algoritmo añade las aristas necesarias para triangularlo. El algoritmo resultando de la modificación del de la máxima cardinalidad se denomina algoritmo de triangulación de máxima cardinalidad:

- Datos: Un grafo no dirigido  $G = (X, L)$  y un nodo inicial  $X_i$
- Resultado: Un conjunto de nuevas aristas  $L'$  tal que  $G = (X, L \cup L')$  sea triangulado.

*Etapas de Iniciación:*

1. Inicialmente la nueva lista de aristas es vacía,  $L' = \emptyset$ .
2. Sea  $i = 1$  y asígnese el primer número de la numeración al nodo inicial  $X_i$ , es decir,  $\alpha(1) = X_i$ .
- 3.

*Etapas de iteración:*

4. Se asigna el número  $i$  a un nodo  $X_k$  no numerado con máximo número de vecinos numerados,  $\alpha(i) = X_k$
5. Si  $Vec(X_k) \cap \{\alpha(1), \dots, \alpha(i-1)\}$  no es un conjunto completo, añadir a  $L'$  las aristas necesarias para completar el conjunto y volver a la etapa 2: en caso contrario, ir a la etapa 5.
6. Si  $i = n$ , el algoritmo finaliza; en caso contrario, asignar  $i = i + 1$  e ir a la etapa 3.

Una numeración de los conglomerados de un grafo no dirigido  $(C_1, \dots, C_m)$  se dice que satisface la propiedad de intersección dinámica, si el conjunto  $C_i \cap (C_1 \cup \dots \cup C_{i-1})$  está contenido en, al menos, uno de los conglomerados  $\{C_1, \dots, C_{i-1}\}$ , para todo  $i = 1, \dots, m$ .

El teorema de cadena de conglomerado dice que un grafo no dirigido tiene asociada una cadena de conglomerados si y solo si es triangulado.

El algoritmo de generación de una cadena de conglomerados sigue:

- Datos: Un grafo triangulado no dirigido  $G = (X, L)$ .
- Resultado: Una cadena de conglomerados  $(C_1, \dots, C_m)$  asociada a  $G$ .
  2. *Iniciación*: Elegir cualquier nodo como nodo inicial y utilizar el algoritmo de máxima cardinalidad para obtener una numeración perfecta de los nodos,  $X_1, \dots, X_n$ .
  3. Determinar los conglomerados del grafo,  $C$ .
  4. Asignar a cada conglomerado el máximo de los números (correspondientes a la numeración perfecta) de sus nodos.
  5. Ordenar los conglomerados,  $(C_1, \dots, C_m)$ , en orden ascendente de acuerdo a los números asignados (deshacer empates de forma arbitraria).

## II.6. Grafos de aglomerados

Un conjunto de nodos de un grafo se denomina un aglomerado.

Supongamos un grafo  $G = (X, L)$  y un conjunto de aglomerados de  $X$ ,  $C = \{C_1, \dots, C_m\}$  tal que  $X = C_1 \cup \dots \cup C_m$ . El grafo  $G' = (C, L')$  se denomina un grafo de aglomerados de  $G$  si las aristas contenidas en  $L'$  sólo unen aglomerados que contengan algún nodo común, es decir,  $(C_i, C_j) \in L' \Rightarrow C_i \cap C_j \neq \emptyset$ .

Un grafo de aglomerados se denomina un grafo de conglomerados asociado a un grafo no dirigido  $G$  si sus aglomerados son los conglomerados de  $G$ .

Un grafo de conglomerados asociado a un grafo no dirigido se denomina un grafo de unión si contiene todas las aristas posibles que unan conglomerados con algún nodo común.

Un grafo de conglomerados se denomina un árbol de unión si es un árbol y todo nodo que pertenezca a dos conglomerados también pertenezca a todos los conglomerados contenidos en el camino que los une.

El teorema de árbol de unión estipula que un grafo no dirigido posee un árbol de unión si y sólo si es triangulado.

El algoritmo siguiente permite construir un árbol de unión asociado a un grafo triangulado:

- Datos: Un grafo triangulado no dirigido  $G = (X, L)$ .
- Resultado: Un árbol de unión  $G' = (C, L')$  asociado a  $G$ .
  1. *Iniciación:* Utilizar el algoritmo de generación de una cadena de conglomerados para obtener una cadena de conglomerados del grafo  $G$ ,  $(C_1, \dots, C_m)$ .
  2. Para cada conglomerado  $C_i \in C$ , escoger un conglomerado  $C_k$  en  $\{C_1, \dots, C_{i-1}\}$  con el máximo número de nodos comunes y añadir la arista  $C_i - C_k$  a  $L'$  (inicialmente vacía). Los empates se deshacen de forma arbitraria.

## II.7. Representación de grafos

Las formas de representación más comunes son:

- Simbólicamente, como un par  $(X, L)$ , donde  $X$  es un conjunto de variables y  $L$  es un conjunto de aristas entre pares de variables.
- Gráficamente, por medio diagrama formado por un conjunto de nodos (uno para cada variable) y un conjunto de líneas o flechas (una para cada arista del conjunto  $L$ ).
- Numéricamente, utilizando ciertos tipos de matrices.

### II.7.1. Representación gráfica de un grafo

Un grafo está formado por un conjunto de nodos y un conjunto de aristas. Una representación gráfica es buena si cumple los siguientes requisitos:

1. Puede ser construida de forma sencilla y rápida utilizando algún algoritmo.
2. Las características topológicas del grafo podrán ser analizadas mediante la representación gráfica.
3. La representación será simple, teniendo un número mínimo de cortes de aristas.

### **II.7.2. La representación circular de un grafo**

Una de las formas más sencillas de representar gráficamente un grafo es dibujar los nodos sobre una circunferencia a distancias iguales. Esta representación tiene una propiedad importante: garantiza que no puede haber más de dos nodos alineados. Por tanto, si las aristas del grafo se representan mediante líneas rectas, esta representación garantiza que no habrá aristas ocultas entre los nodos. Así la representación circular es la representación óptima para grafos con un gran número de aristas.

La representación circular presenta las siguientes ventajas:

- Es fácil de construir.
- Todas las aristas son transparentes en el diagrama.
- Es la más conveniente para grafos completos o casi completos.

La principal desventaja de esta representación es que pueden existir numerosos cortes entre las aristas, complicando el diagrama.

#### **II.7.2.1. La representación multinivel de un grafo**

La idea básica de la representación multinivel es organizar los nodos en distintos niveles, o capas, de tal forma que no existan aristas entre nodos del mismo nivel y que todo nodo en un nivel esté conectado con algún nodo del nivel previo. Así se podrá lograr una representación

clara del grafo situando los nodos en niveles horizontales. Para desarrollar esta idea en detalle, son necesarias algunas definiciones previas:

Dado un grafo  $(X, L)$ , un subconjunto de nodos  $S \subset X$  se denomina totalmente inconexo si no existe ninguna arista entre los nodos de  $S$ , es decir, si  $(X_i, X_j) \in L \Rightarrow X_i \notin S$  o  $X_j \notin S$ .

Una representación multinivel de un grafo no dirigido  $(X, L)$  es una partición

$$X = \bigcup_{k=1}^m S_k$$

donde los niveles  $S_i, i=1, \dots, m$ , son subconjuntos disjuntos y totalmente inconexos de  $X$  tal que

$$\text{si } X_i \in S_k \Rightarrow \exists X_j \in S_{k-1} \text{ con } X_i \in \text{Ady}(X_j).$$

Es decir, no existen aristas entre nodos del mismo nivel y los nodos de un nivel son adyacentes a, al menos, un nodo del nivel anterior.

Algunas ventajas de la representación multinivel son:

- Es muy conveniente para árboles, o grafos con pocas aristas.
- Muestra la estructura ancestral del grafo a través de los distintos niveles de la representación.

El algoritmo de representación multinivel se presenta de la manera siguiente:

- Datos: Un grafo  $(X, \text{Ady})$  de  $n$  nodos y un conjunto de nodos de raíz  $R$ .
  - Resultado: Una representación multinivel  $S$  del grafo.
1. *Iniciación*:  $\text{Asignados} = R, \text{Nivel}(1) = R, \text{Nivel}(k) = \emptyset$  para  $k = 2, \dots, n$ . Tomar  $j = 1$ .
  2. Si  $\text{Nivel}(j) = \emptyset$ , devolver  $S = \{\text{Nivel}(1), \dots, \text{Nivel}(j-1)\}$ , que es una representación multinivel del grafo y terminar; en caso contrario, hacer  $\text{NivelActual} = \text{Nivel}(j)$  e ir a la etapa 3.
  3. Seleccionar  $X_k \in \text{NivelActual}$ :
    - a. Añadir los elementos de  $\text{Ady}(X_k) \setminus \text{Asignados}$  a  $\text{Nivel}(j+1)$  y a  $\text{Asignados}$



- b. Añadir los elementos de  $Ady(X_k) \cap Nivel(j+1)$  y eliminar estos elementos de  $Nivel(j)$  y de  $NivelActual$  e ir a la etapa 4
4. Eliminar  $X_k$  de  $NivelActual$ . Si  $NivelActual = \emptyset$ , tomar  $j = j+1$  e ir a la etapa 2; en caso contrario, ir a la etapa 3.

### II.7.2.2. La representación numérica de grafos

Un grafo también puede ser representado numéricamente utilizando determinados tipos de matrices.

Sea  $G = (X, L)$  un grafo de  $n$  nodos y sea  $A = (a_{ij})$  una matriz  $n \times n$ , donde

$$a_{ij} = \begin{cases} 1, & \text{si } L_{ij} \in L, \\ 0, & \text{en caso contrario.} \end{cases}$$

La matriz  $A$  se denomina matriz de adyacencia del grafo  $G$ .

Mediante sencillas manipulaciones algebraicas de la matriz de adyacencia se pueden obtener algunas características del grafo como, por ejemplo, el número de caminos distintos que unen dos nodos, comprobar si el grafo es conexo, etc.

Notar que:

- la matriz de adyacencia de un grafo no dirigido es simétrica.
- Dado que  $L_{ij} \notin L$  para todos los valores de  $i$ , los elementos diagonales de  $A$  son nulos.
- La matriz de adyacencia de un grafo no dirigido completo debe contener un uno en todos los elementos no diagonales.

La matriz de adyacencia permite comprobar si existe algún camino entre cada par de nodos. También puede calcularse la longitud de todos los caminos que unan cada par de nodos. El teorema siguiente muestra cómo se puede utilizar la matriz de adyacencia para esta tarea.

Teorema de potencias de la matriz de adyacencia: Sea  $A^r$  la  $r$ -ésima potencia de la matriz de adyacencia asociada a un grafo  $G=(X,L)$ . Entonces, el  $ij$ -ésimo elemento de  $A^r$  da el número de caminos de longitud  $r$  del nodo  $X_i$  al nodo  $X_j$ .

La matriz de alcanzabilidad,  $T=(t_{ij})$ , de un grafo dado se define como

$$t_{ij} = \begin{cases} 1, & \text{si existe algún camino del nodo } X_i \text{ al nodo } X_j \\ 0, & \text{en caso contrario} \end{cases}$$

La matriz de alcanzabilidad está claramente relacionada con las potencias de la matriz de adyacencia

Teorema de acotación a la longitud de un camino: Dado un grafo con  $n$  nodos, si existe un camino del nodo  $X_i$  al nodo  $X_j$ , entonces también existe un camino de longitud menor que  $n$  de  $X_i$  a  $X_j$ .

## II.8. Concepto de grafos borrosos [59]

### III.1. Concepto.

Consideremos dos conjuntos  $E_1$  y  $E_2$ , designemos por  $x$  un elemento de  $E_1$  y por  $y$  un elemento de  $E_2$ . El conjunto de parejas  $(x,y)$  define el conjunto-producto  $E_1 \times E_2$ .

El subconjunto borroso  $G_{\sigma}$  tal que:

$$\forall (x,y) \in E_1 \times E_2 : \mu_{G_{\sigma}}(x,y) \in M,$$

donde  $M$  es el conjunto de pertenencia de  $E_1 \times E_2$ , se denomina un “grafo borroso”.

Sean  $E_1 = \{x_1, x_2, x_3\}$  y  $E_2 = \{y_1, y_2\}$

$$E_1 \times E_2 = \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2), (x_3, y_1), (x_3, y_2)\}.$$

Pongamos, para simplificar la escritura:

$$\mu(x_i, y_j) = \mu_{G_{\%}}(x_i, y_j), \quad i=1,2,3, \quad j=1,2,$$

que se denominará “valor” de la pareja  $(x_i, y_j)$ .

Sea, por ejemplo:

$$\begin{aligned} \mu(x_1, y_1) &= 0,3, \quad \mu(x_1, y_2) = 0,7, \quad \mu(x_2, y_1) = 1, \\ \mu(x_2, y_2) &= 0, \quad \mu(x_3, y_1) = 0,5, \quad \mu(x_3, y_2) = 0,2. \end{aligned}$$

Esta función define el subconjunto borroso:

$$G_{\%} = \{((x_1, y_1)|0,3), ((x_1, y_2)|0,7), ((x_2, y_1)|1), ((x_2, y_2)|0), ((x_3, y_1)|0,5), ((x_3, y_2)|0,2)\}.$$

Este subconjunto borroso puede representarse mediante una matriz de la forma siguiente:

Z	$y_1$	$y_2$
$x_1$	0,3	0,7
$x_2$	1	0
$x_3$	0,5	0,2

Así, el grafo  $G_{\%} \subset E_1 \times E_2$  es un “grafo borroso”.

Lo que se acaba de presentar para un conjunto-producto  $E_1 \times E_2$  puede ser generalizado para un conjunto-producto

$$E_1 \times E_2 \times \dots \times E_n.$$

El subconjunto borroso tal que:

$$x^{(i)} \in E_i, \quad i=1,2,\dots,n,$$

$$\forall (x^{(1)}, x^{(2)}, \dots, x^{(n)}) \in E_1 \times E_2 \times \dots \times E_n : \mu(x^{(1)}, x^{(2)}, \dots, x^{(n)}) \in M,$$

donde  $M$  es el conjunto de pertenencia de  $E_1 \times E_2 \times \dots \times E_n$ , se denomina un “grafo borroso”.

### III.2. Relación borrosa

La noción de grafo borroso puede explicarse mediante la noción de “relación borrosa”. Sea  $P$  un conjunto producto de  $n$  conjuntos y  $M$  su conjunto de pertenencia; Una relación  $n$ -aria borrosa es un subconjunto borroso de  $P$  tomando sus valores en  $M$ .

Sean:

$$E_1 = \{x_1, x_2, x_3\}$$

$$E_2 = \{y_1, y_2, y_3, y_4, y_5\}$$

$$\text{y } M = [0, 1].$$

La tabla siguiente expresa una relación 2-aria borrosa:

Z	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
$x_1$	0	0	0,1	0,3	1
$x_2$	0	0,8	0	0	1
$x_3$	0,4	0,4	0,5	0	0,2

Una relación borrosa en  $E_1 \times E_2$  se denotará:

$$x \in E_1, y \in E_2 : x \mathfrak{R}_\mu y$$

Utilizaremos símbolos siguientes:

$\bigvee_x$  para expresar el máximo en relación con un elemento o variable  $x$ ,

$\bigwedge_x$  para expresar el mínimo en relación con un elemento o variable  $x$ ,

Así, escribir:

$$\mu_1(x) = \bigvee_y \mu(x, y)$$

será equivalente a:

$$\mu_1(x) = \max_y \mu(x, y).$$

También, escribir:

$$\mu_2(x) = \wedge_y \mu(x, y)$$

será equivalente a:

$$\mu_2(x) = \min_y \mu(x, y).$$

### III.3. Composición max-min de dos relaciones borrosas

Indiquemos primero que algunas veces utilizaremos la notación:

$\mathfrak{R}_{\frac{0}{j}} \subset X \times Y$  significando:

$$G_j \subset X \times Y$$

donde  $\mathfrak{R}_{\frac{0}{j}}$  es la relación borrosa correspondiendo al grafo  $G_j$ .

Sean  $\mathfrak{R}^f \subset X \times Y$  y  $\mathfrak{R}_{\frac{0}{2}} \subset Y \times Z$ , la composición max-min de  $\mathfrak{R}_{\frac{0}{2}}$  a  $\mathfrak{R}_{\frac{0}{1}}$ , denotada por  $\mathfrak{R}_{\frac{0}{2}} \circ \mathfrak{R}_{\frac{0}{1}}$ , se definirá por la expresión:

$$\begin{aligned} \mu_{\mathfrak{R}_{\frac{0}{2}} \circ \mathfrak{R}_{\frac{0}{1}}}(x, z) &= \vee_y \left[ \mu_{\mathfrak{R}_{\frac{0}{1}}}(x, y) \wedge \mu_{\mathfrak{R}_{\frac{0}{2}}}(y, z) \right] \\ &= \max_y \left[ \min \left( \mu_{\mathfrak{R}_{\frac{0}{1}}}(x, y), \mu_{\mathfrak{R}_{\frac{0}{2}}}(y, z) \right) \right] \end{aligned}$$

donde  $x \in X, y \in Y, z \in Z$

La operación max-min es asociativa; es decir:

$$\left( \mathfrak{R}_{\frac{0}{3}} \circ \mathfrak{R}_{\frac{0}{2}} \right) \circ \mathfrak{R}_{\frac{0}{1}} = \mathfrak{R}_{\frac{0}{3}} \circ \left( \mathfrak{R}_{\frac{0}{2}} \circ \mathfrak{R}_{\frac{0}{1}} \right).$$

Si  $\mathfrak{R}^f$  es una relación definida en  $E \times E$ , sea  $\mathfrak{R}_{\frac{0}{j}} \subset E \times E$ , se podrá escribir:

$$\mathfrak{R}_{\frac{0}{j}} \circ \mathfrak{R}_{\frac{0}{j}} = \mathfrak{R}_{\frac{0}{j}}^2$$

de allí:

$$\mathfrak{R}_{\frac{0}{j}} \circ \mathfrak{R}_{\frac{0}{j}}^2 = \mathfrak{R}_{\frac{0}{j}}^2 \circ \mathfrak{R}_{\frac{0}{j}} = \mathfrak{R}_{\frac{0}{j}}^3;$$

y, generalmente:

$$\underbrace{\mathfrak{R} \circ \mathfrak{R} \circ \dots \circ \mathfrak{R}}_{k \text{ veces}} = \mathfrak{R}^k$$

### III.4. Camino en un grafo borroso finito

Consideremos en el grafo finito  $G \subset E \times E$ , una  $r$ -upla ordenada con o sin repetición:

$$C = (x_{i_1}, x_{i_2}, \dots, x_{i_r}),$$

donde los  $x_{i_k} \in E$   $k = 1, 2, \dots, r$ ,

con la condición:

$$\forall (x_{i_k}, x_{i_{k+1}}): \mu_{\mathfrak{R}}(x_{i_k}, x_{i_{k+1}}) > 0 \quad k = 1, 2, \dots, r-1.$$

Una tal  $r$ -upla ordenada se denominará “camino de  $x_{i_1}$  a  $x_{i_r}$ ” en el grafo  $G$  (diremos también en la relación  $\mathfrak{R}$ )

A cada camino  $(x_{i_1}, x_{i_2}, \dots, x_{i_r})$  será asociado un “valor” definido por:

$$l(x_{i_1}, x_{i_2}, \dots, x_{i_r}) = \mu_{\mathfrak{R}}(x_{i_1}, x_{i_2}) \wedge \mu_{\mathfrak{R}}(x_{i_2}, x_{i_3}) \wedge \dots \wedge \mu_{\mathfrak{R}}(x_{i_{r-1}}, x_{i_r}).$$

Ahora, consideremos todos los caminos posibles existentes de  $x_i$  a  $x_j$ , dos elementos arbitrarios de  $E$ , sea  $C(x_i, x_j)$ , el conjunto vulgar de todos los caminos

$$C(x_i, x_j) = \left\{ C(x_i, x_j) \mid C(x_i, x_j) = (x_{i_1} = x_i, x_{i_2}, \dots, x_{i_{r-1}}, x_{i_r} = x_j) \right\}.$$

Entonces, el “camino más fuerte”  $C^*(x_i, x_j)$  de  $x_i$  a  $x_j$  se definirá por:

$$l^*(x_i, x_j) = \bigvee_{C(x_i, x_j)} l(x_{i_1} = x_i, x_{i_2}, \dots, x_{i_{r-1}}, x_{i_r} = x_j).$$

El número de elementos menos uno que constituyen un camino, se denominará “longitud del camino”.

Veamos algunos teoremas:

**Teorema I:** - sea  $\mathfrak{R} \subset E \times E$ , tenemos entonces:

$$\forall (x, y) \in E \times E: \mu_{\mathfrak{R}}(x, y) = l_k^*(x, y),$$

donde  $l_k^*(x, y)$  es el camino más fuerte que existe de  $x$  a  $y$  y de longitud  $k$ .

**Teorema II:** - sea  $\mathfrak{R}_{0'} \subset E \times E$  y  $\hat{\mathfrak{R}}_{0'}$  la cierra transitiva de  $\mathfrak{R}_{0'}$ , se tiene entonces:

$$\forall (x, y) \in E \times E : \mu_{\hat{\mathfrak{R}}_{0'}}(x, y) = l^*(x, y).$$

**Teorema III:** - sea  $n = \text{card } E$ , si  $k$  es la longitud del camino de  $x_i$  a  $x_j$ , con  $k > n = \text{card } E$ , entonces todos los elementos del camino no son únicos, hay, al menos, lo que se denomina un “circuito” (camino cerrado) en el camino. Si se quita ese (esos) circuito, el camino se queda con una longitud inferior o igual a  $n$ ; se puede entonces enunciar:

$$l_k^*(x, y) = l_{j \leq n}^*(x, y),$$

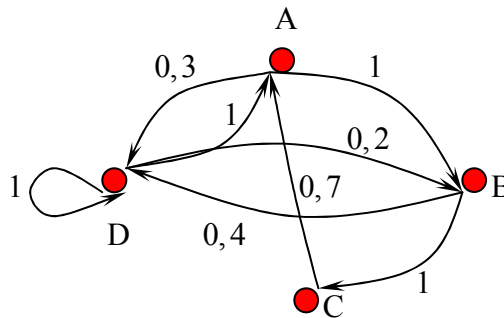
donde  $l_{j \leq n}^*(x, y)$  es el valor del camino más fuerte de longitud inferior o igual a  $n$  de  $x$  a  $y$ .

**Teorema IV:** - Si  $\mathfrak{R}_{0'} \subset E \times E$  y  $n = \text{card } E$ , entonces:

$$\hat{\mathfrak{R}}_{0'} = \mathfrak{R}_{0'} \cup \mathfrak{R}_{0'}^2 \cup \dots \cup \mathfrak{R}_{0'}^n$$

Consideremos la relación  $\mathfrak{R}_{0'}$  representada en la matriz y el grafo siguientes:

$\mathfrak{R}_{0'}$		A	B	C	D
A	0	1	0	0,3	
B	0	0	1	0,4	
C	0,7	0	0	0	
D	1	0,2	0	1	



(No se han representado las parejas  $(x, y)$  tales que  $\mu_{\mathfrak{R}_{0'}}(x, y) = 0$ )

Se utilizarán los resultados siguientes para nuestras explicaciones:

$\mathfrak{R}_{\%}$	A	B	C	D
A	0	1	0	0,3
B	0	0	1	0,4
C	0,7	0	0	0
D	1	0,2	0	1

(a)  $\mathfrak{R}_{\%}$

0,3	0,2	1	0,4
0,7	0,2	0	0,4
0	0,7	0	0,3
1	1	0,2	1

(b)  $\mathfrak{R}_{\%}^2$

0,3	1	1	0,4
0,7	0,2	1	0,4
0,7	0,7	0	0,3
1	1	0,2	1

(c)  $\mathfrak{R}_{\%} \cup \mathfrak{R}_{\%}^2$

0,7	0,3	0,2	0,4
0,4	0,7	0,2	0,4
0,3	0,2	0,7	0,4
1	1	1	1

(d)  $\mathfrak{R}_{\%}^3$

0,7	1	1	0,4
0,7	0,7	1	0,4
0,7	0,7	0,7	0,4
1	1	1	1

(e)  $\mathfrak{R}_{\%} \cup \mathfrak{R}_{\%}^2 \cup \mathfrak{R}_{\%}^3$

0,7	0,7	0,3	0,4
0,4	0,4	0,7	0,4
0,7	0,3	0,2	0,4
1	1	1	1

(f)  $\mathfrak{R}_{\%}^4$

0,7	1	1	0,4
0,7	0,7	1	0,4
0,7	0,7	0,7	0,4
1	1	1	1

(g)  $\mathfrak{R}_{\%} \cup \mathfrak{R}_{\%}^2 \cup \mathfrak{R}_{\%}^3 \cup \mathfrak{R}_{\%}^4 = \hat{\mathfrak{R}}_{\%}$

Sean  $(B, C, A, D)$ , un camino. Calculemos su valor:

$$\begin{aligned}
 l(B, C, D, A) &= \mu_{\mathfrak{R}_{\%}}(B, C) \wedge \mu_{\mathfrak{R}_{\%}}(C, A) \wedge \mu_{\mathfrak{R}_{\%}}(A, D) \\
 &= 1 \wedge 0,7 \wedge 0,3 = 0,3
 \end{aligned}$$

Examinemos todos los caminos que van desde  $B$  hasta  $D$  cuya longitud es inferior o igual a 3, hay solamente: los caminos  $(B, D)$ ,  $(B, D, D)$ ,  $(B, D, D, D)$  para los cuales se tiene:

$$l(B, D) = \mu_{\mathfrak{R}_{\%}}(B, D) = 0,4, \quad l(B, D, D) = \mu_{\mathfrak{R}_{\%}}(B, D) \wedge \mu_{\mathfrak{R}_{\%}}(D, D) = 0,4 \wedge 1 = 0,4$$



$$l(B, D, D, D) = \mu_{\frac{\mathfrak{R}}{\%}}(B, D) \wedge \mu_{\frac{\mathfrak{R}}{\%}}(D, D) \wedge \mu_{\frac{\mathfrak{R}}{\%}}(D, D) = 0,4 \wedge 1 \wedge 1 = 0,4$$

Tenemos pues:

$$l^*(B, D) = l(B, C, D, A) \vee l(B, D) \vee l(B, D, D) \vee l(B, D, D, D) = 0,3 \vee 0,4 \vee 0,4 \vee 0,4 = 0,4$$

Si consultamos  $\hat{\mathfrak{R}}_0$  en la tabla (g) precedente, hallamos:

$$\mu_{\frac{\mathfrak{R}}{\%}}(B, D) = 0,4 \quad (\text{Teorema II})$$

Hay dos caminos de longitud 3 entre  $B$  y  $D$ , que son  $(B, C, D, A)$  y  $(B, D, D, D)$ . Se tiene pues:

$$l_3^*(B, D) = l(B, C, D, A) \vee l(B, D, D, D) = 0,3 \vee 0,4 = 0,4$$

Comprobamos con:

$$\mu_{\frac{\mathfrak{R}^3}{\%}}(B, D) = 0,4 \quad (\text{Teorema I})$$

Consideremos ahora el camino:  $(C, A, B, D, A, D)$ . Este camino posee un circuito que es  $(D, A, D)$ , lo quitaremos, y tendremos:

$$\begin{aligned} l_5^*(C, D) &= l_{j \leq 4}^*(C, D) \\ &= l_1^*(C, D) \vee l_2^*(C, D) \vee l_3^*(C, D) \vee l_4^*(C, D) \\ &= \mu_{\frac{\mathfrak{R}}{\%}}(C, D) \vee \mu_{\frac{\mathfrak{R}^2}{\%}}(C, D) \vee \mu_{\frac{\mathfrak{R}^3}{\%}}(C, D) \vee \mu_{\frac{\mathfrak{R}^4}{\%}}(C, D) \\ &= 0 \vee 0,3 \vee 0,4 \vee 0,4 = 0,4 \end{aligned}$$

Quizás, esperaríamos hallar 0,3 pero el camino más fuerte de longitud 5 entre  $C$  y  $D$  no es  $(C, A, B, D, A, D)$  sino  $(C, A, B, D, D, D)$ , los dos reduciéndose a  $(C, A, B, D)$  cuando se elimina el circuito. Véanse la tablas precedentes en (b).

## ANEXO III. MODELOS GAMS UTILIZADOS\*

### III.1. Implementación en GAMS del algoritmo de Floyd-Warshal (Cálculo de la matriz de distancias de caminos más cortos)

```

$ONTEXT
FICHERO FLOYD.GMS
NECESITA EL FICHERO      ARISTAS.TXT
CALCULA LA MATRIZ DE DISTANCIAS ENTRE VERTICES D(I,J) Y LA ESCRIBE
EN EL FICHERO FLOYD.TXT
$OFFTEXT

OPTION OPTCR = 0.0001;
option optca = 0;
OPTION RESLIM = 10000000;
OPTION ITERLIM =16000000;
option mip=osl2;
option solprint = off;
option limcol = 0;
option limrow = 0;

$offlisting;
$offsymlist;
$offsymxref;
$offinclude;
*para que no incluya el listado de los includes en el lst

*para controlar el tiempo
scalar
ini
fin
dur
hora
minu
seg;
ini=jnow;

```

---

\* Modelos desarrollados por M.J. Canós Darós, Vicente Liern, Carlos Ivorra y Manolo, Departamento de Economía Matemática, Universidad de Valencia.

```

$include "aristas.txt"

Parameter D(i,j);
d(i,j)=c(i,j)$c(i,j)+c(j,i)$c(j,i)+big$(c(i,j)+c(j,i) eq 0);
d(i,i)=0;
loop(k,
    loop(i,
        loop(j,
            if (d(i,j)<(d(i,k)+d(k,j)),
                d(i,j)=d(i,j);
            else
                d(i,j)=d(i,k)+d(k,j);
            );
        ));
*d(j,i)$(ord(j) gt ord(i))=d(i,j);
file dist /floyd.txt/;
put dist;
Put "table d(i,j)"/;
scalar r,s;
r=1;
s=10;
put"":<7;
loop(j$(ord(j) ge r),
    loop(k$[(ord(k) ge r) and (ord(k) le s)], put k.tl:<7; );
    put /;
    loop(i, put i.tl:<7;
        loop(k$[(ord(k) ge r) and (ord(k) le s)],
            put d(i,k):<7:2;);
        put /);
    if((s lt card(i)), put / "+":<7;);
    s=s+10;
    r=r+10;
);
putclose dist;

fin=jnow;
hora=ghour(fin)-ghour(ini);
minu=gminute(fin)-gminute(ini);

```

```

seg=gsecond(fin)-gsecond(ini);
dur=[ghour(fin)*3600+gminute(fin)*60+gsecond(fin)]-
[ghour(ini)*3600+gminute(ini)*60+gsecond(ini)];

file time /time-floyd.txt/;
put time;
put "fichero":<10 "duracion en segundos":<25 /;
put "floyd.gms":<10 dur:<>25:0 "segundos":<8 /;
putclose time;

```

### III.2. Modelo GAMS para el cálculo de la p-mediana crisp

```

*fichero mediana
*calculo de la mediana
*escribe fichero rmed.txt con los radios de la mediana
*escribe el fichero solmediana.txt con las soluciones y tiempo de ejecucion

OPTION OPTCR = 0.0001;
option optca = 0;
OPTION RESLIM = 10000000;
OPTION ITERLIM =16000000;
option mip=osl2;
option solprint = off;
option limcol = 0;
option limrow = 0;

$offlisting;
$offsymlist;
$offsymxref;
$offinclude;
*para que no incluya el listado de los includes en el lst

$include "aristas.txt"
$include "pesos.txt"
$include "floyd.txt";
ALIAS (J, JR);
SCALAR P,ms,rmed ;

```

\*para controlar el tiempo

scalar

ini

fin

dur

hora

minu

seg;

PARAMETER RMEDIANA(JR)

VMEDIANA(JR);

PARAMETER RCMP(JR) RADIO DE LA MEDIANA PONDERADO POR EL CENTRO;

parameter dist(i,j);

SCALAR BIG;

\* COTA valor de la mediana para el p anterior;

BIG=10000;

\*COTA=BIG;

VARIABLES

Z;

POSITIVE VARIABLES

X(I,J) CANTIDAD SERVIDA DESDE EL CENTRO I AL DESTINO J;

BINARY VARIABLES

Y(I) INDICA SI HAY CENTRO DE SERVICIO EN I;

EQUATIONS

EQ1(J)

EQ2(I,J)

EQ3

FUN;

EQ1(J).. SUM(I,X(I,J)) =E= 1;

EQ2(I,J).. X(I,J) =L= Y(I);

EQ3.. SUM(I,Y(I)) =E= P;

FUN.. Z=E=SUM((I,J),WM(J)\*D(I,J)\*X(I,J));

MODEL mediana /ALL/;

FILE SOL /SOL-MED-POND.TXT/;

```

PUT SOL;
*sol.ap=1;
put "P":<10 "mediana":<>10 "radio":<>10 "tiempo/seg":<>10 "tipo":<>10 /;
*PUT "P=" P:<>5:0 /;
putclose sol;

for(p=2 to (card(j)-1) by 1,
  *P=3;
  ini=jnow;
  *PARA QUE NO ARRASTRE VALORES DE LA SOLUCION ANTERIOR
  *X.L(I,J)=0;
  Y.L(I)=0;
  *X.FX(I,J)$ (D(I,J) GT COTA)=0;
  * no es cierto que se puedan eliminar

SOLVE MEDIANA USING MIP MINIMIZING Z;
  IF((mediana.modelstat eq 1),MS=1;);
  IF((mediana.modelstat eq 8),MS=8;);
  IF([(mediana.modelstat eq 4) OR (mediana.modelstat eq 10)],MS=4;);
  IF(((mediana.modelstat NE 1) AND (mediana.modelstat NE 8) AND
      (mediana.modelstat NE 4) AND (mediana.modelstat NE 10)), MS=0;);

*para calcular el tiempo
fin=jnow;
hora=ghour(fin)-ghour(ini);
minu=gminute(fin)-gminute(ini);
seg=gsecond(fin)-gsecond(ini);
dur=[ghour(fin)*3600+gminute(fin)*60+gsecond(fin)]-
[ghour(ini)*3600+gminute(ini)*60+gsecond(ini)];

PUT SOL;
sol.ap=1;

put P:<10:2;
  IF((MS=1 OR MS=8),
    rmed=smax(j,smax(i,d(i,j)$ (x.l(i,j) gt 0)));
    dist(i,j)=d(i,j)$ (x.l(i,j) gt 0);
    RMEDIANA(JR)$ (ORD(JR) EQ P)=RMED;

```

```

                VMEDIANA (JR) $ (ORD (JR) EQ P) =Z.L;
                RCMP (JR) $ (ORD (JR) EQ P) =SMAX ((I,J) $ (x.l(i,j) gt
0) ,WC(J) *D(I,J) *X.L(I,J) );
                );
        IF((MS=1) ,
*           COTA=SMAX ((I,J) ,D(I,J) $ (X.L(I,J) GT 0) );
           PUT z.l:<>10:2 rmed:<>10:2 dur:<>10 "OPTIMA":<>10 /;
           PUT "":<4 "VERT-SOL":<10;
           LOOP(I$ (Y.L(I) GT 0) ,
                PUT I.TL:<5:0;
                );
           PUT /;
        );
        IF((MS=8) ,
*           COTA=SMAX ((I,J) ,D(I,J) $ (X.L(I,J) GT 0) );
           PUT z.l:<>10:3 rmed:<>10:3 dur:<>10 "INTEGER":<>10 /;
           PUT "":<4 "VERT-SOL":<10;
           LOOP(I$ (Y.L(I) GT 0) ,
                PUT I.TL:<5:0;
                );
           PUT /;
        );
        IF((MS=4) ,
           PUT "":<>10 rmed:<>10:3 dur:<>10 "INFEASIBLE":<>10 /;
        );
        IF((MS=0) ,
           PUT "":<>10 rmed:<>10:3 dur:<>10 "NO SOLUTION":<>10 /;
        );

        PUTCLOSE SOL;

*cierra for p
);
FILE RCM /RCMP.TXT/;
PUT RCM;
PUT "PARAMETER RCMP(JR)"/;
PUT "/" /;
LOOP(JR,
        PUT JR.TL:<4:0 "":<5 RCMP(JR):<10:3 /;

```

```

    );
PUT "/;"
PUTCLOSE RCM;

FILE SOL1 /RMED.TXT/;
PUT SOL1;
PUT "PARAMETER RMEDIANA(JR)"/;
PUT "/" /;
LOOP(JR,
    PUT JR.TL:<4:0 "":<5 RMEDIANA(JR):<10:3 /;
    );
PUT "/;" ///;
PUT "PARAMETER VMEDIANA(JR)"/;
PUT "/" /;
LOOP(JR,
    PUT JR.TL:<4:0 "":<5 VMEDIANA(JR):<10:3 /;
    );
PUT "/;" /;
PUTCLOSE SOL1;

file time /time-mediana-pond.txt/;
put time;
put "fichero":<10 "duracion en segundos":<25 /;
put "pmediana":<10 dur:<>25:0 "segundos":<8 /;
putclose time;

```

### III.3. Ficheros txt de datos utilizados en el modelo (Caso de Kinshasa)

- **aristas.txt**

```

set i nodos /1*10/;
alias (i,j,k,l);
scalar big;
big=100;
parameter c(i,j)
/1.2      6
1.3      8
1.5     13

```



1.7	17.5
2.3	8
2.5	13
2.7	17.5
3.4	5.5
3.5	8.5
3.7	12
4.5	7.5
4.9	15
5.6	4
5.7	5
5.8	7.5
5.10	7
6.7	8
7.8	7
7.10	12
8.9	8
9.10	3.5/;

- **pesos.txt**

```
parameter wm(i)
/1      841941
2       569281
3       455533
4       400963
5       569025
6       159503
7       211931
8       288449
9       390084
10      454644/;
parameter wc(i);
wc(i)=1;
```

**BIBLIOGRAFÍA**

- [1] Atlas de Kinshasa
- [2] I. Averbakh y O. Berman, Minmax Regret Median Location on a Network under Uncertainty, *INFORMS Journal on Computing* 12 (2000) 104-110.
- [3] T. Aykin, Networking Policies for Hub-and-Spoke Systems with Application to the Air Transportation System, Working Paper, Rutgers Graduate School of Management, New Ark, NJ, (1993).
- [4] T. Aykin, On the Location of Hub Facilities, *Transportation Science* 22 (1988) 155-157.
- [5] T. Aykin, On a Quadratic Integer Program for the Location of Interacting Hub Facilities, *European Journal of Operational Research* 46 (1990) 409-411.
- [6] T. Aykin, Lagrangian Relaxation Based Approaches to Capacitated Hub-and-Spoke Network Design Problems, *European Journal of Operational Research* 79 (1994) 501-523.
- [7] T. Aykin y G. Brown, Interacting New Facilities and Location-Allocation Problems, *Transportation Science* 26 (1992) 77-99.
- [8] M. Balinski, Integer Programming: Methods, Uses, Computations, *Management Science* 12 (1965) 253-263
- [9] W. Baumol y P. Wolfe, A Warehouse-location problem, *Operations Research* 6 (1958) 252-263.
- [10] N. Boland, M. Krishnamoorthy, A.T. Ernst, Jamie Ebery, Preprocessing and cutting for multiple allocation hub location problems, *European Journal of Operational Research* 155 (2004) 638 – 653
- [11] R.L. Bowerman, P.H. Calamai, G.B. Hall, The demand partitioning method for reducing aggregation errors in p-median problems, *Computers & Operations Research* 26 (1999) 1097 – 1111
- [12] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, GAMS: a user's guide, GAMS Development Corporation (1998) 262 p.
- [13] J.F Campbell, Location-Allocation for Distribution Systems with Transshipments and Transportation Economies of Scale, *Annals of Operations Research* 40 (1992) 77-99 .
- [14] J.F. Campbell, Integer Programming Formulations of Discrete Hub Location Problems, *European Journal of Operational Research* 72 (1994) 387-405

- [15] J.F. Campbell, Hub Location and the  $p$ -Hub Median Problem, *Operations Research* 44 (6) (1996) 923-935.
- [16] M.J. Canós, Reducción de costes de distribución en problemas de localización con demandas inciertas mediante técnicas de gestión borrosas, Tesis Doctoral, Universidad de Valencia.
- [17] M.J. Canós Darós y M. Mocholí Arce, *Técnicas de optimización robusta Aplicadas al problema de la  $p$ -mediana en condiciones de incertidumbre*, en Actas de VI Jornadas de la Asociación Española de Profesores Universitarios de Matemáticas para la Economía y la Empresa, Santiago de Compostela, 1998.
- [18] M.J. Canós, C. Ivorra, V. Liern, An exact algorithm for the fuzzy  $p$ -median problem, *European Journal of Operational Research* 116 (1999) 80 – 86.
- [19] M.J. Canós, C. Ivorra, V. Liern, The fuzzy  $p$ -median problem: A global analysis of the solutions, *European Journal of Operational Research* 130 (2001) 430 – 436
- [20] M.J. Canós, M. Martínez Romero, *Cálculo de escenarios para la mediana robusta cuando la demanda es incierta*, en Actas de IX Jornadas de la Asociación Española de Profesores Universitarios de Matemáticas para la Economía y la Empresa, Las Palmas de gran Canaria, 2001
- [21] M.J. Canós, C. Ivorra, V. Liern, Finding satisfactory near-optimal solutions in location problems, J.L. Verdegay Editor, *Fuzzy Sets Based Heuristics for Optimization*, 126 (2002) 265 – 276.
- [22] M.J. Canós, C. Ivorra, V. Liern, Un algoritmo de intercambio para el problema borroso de la  $p$ -mediana cuando el coste óptimo es desconocido, *Emergent Solutions for Information and Knowledge Economy, Proceedings of the 10<sup>th</sup> SIGEF Congress 2 (October 2003)* 79 – 89
- [23] E. Castillo, J.M. Gutiérrez, A.S. Hadi, Sistemas expertos y modelos de redes probabilísticas. *Academia de ingeniería*, 115 - 174
- [24] S. Chanas, M. Delgado, J.L. Verdegay, M.A. Vila, Fuzzy optimal flow on imprecise structure, *European Journal of Operations Research* 83 (1995) 568 – 580
- [25] B. Chen y C.S. Lin, Minmax-Regret Robust 1-Median Location on a Tree”, *Networks* 31 (1998) 93-103.
- [26] T.H. Cormen, C.E. Leiserson y R.L. Rivest, *Introduction to algorithms*, The MIT Press, London (1994)

- [27] M. Daskin, A Maximum Expected Covering Location Model Formulation, Properties and Heuristic Solution, *Transportation Science* 17 (1983) 48-70.
- [28] P.M. Dearing, R.L. Francis y T.L. Lowe, Convex Location Problems on Tree Networks, *Operations Research* 26 (1976) 628-642.
- [29] M. Delgado, J.L. Verdegay, M.A. Vila, On fuzzy tree definition, *European Journal of Operacional Research* 22 (1985) 243 – 249
- [30] M. Delgado, J.L. Verdegay, M.A. Vila, On valuation and problems in fuzzy graphs: a general approach and some particular cases, *ORSA Journal on Computing* 2 (1990) 74 – 84
- [31] J. Ebery, Solving large single allocation  $p$ -hub problems with two or three hubs, *European Journal of Operational Research* 128 (2001) 447 – 458
- [32] J. Ebery, M. Krishnamoorthy, A. Ernst, N. Boland, The capacitated multiple allocation hub location problem: Formulations and algorithms, *European Journal of Operational Research* 120 (2000) 614 – 631.
- [33] S. Eilon y R.D. Galvao, Single and Double Vertex Substitution in Heuristic Procedures for the  $p$ -Median Problem, *Management Science* 24 (1978) 1763-1766.
- [34] S. Elhedhli, F.X. Hu, Hub-and-spoke network design with congestion, *Computers & Operations research* (article in press)
- [35] S.B. Eom, S.M. Lee y C. Somarajan, A Survey of Decision Support System Applications (1988-1994), *Journal of the Operational Research Society* 49 (1998) 109-120.
- [36] E. Erkut y B.C. Tansel, On Parametric Medians of Trees, *Transportation Science* 26 (1992) 149-156.
- [37] D. Erlenkotter, A Dual-Based Procedure for Uncapacitated Facility Location, *Operations Research* 26 (1978) 992-1009.
- [38] A.T. Ernst, M. Krishnamoorthy, Exact and heuristic algorithms for the uncapacitated multiple allocation  $p$ -hub median problem, *European Journal of Operational Research* 104 (1998) 100 – 112.
- [39] A.T. Ernst y M. Krishnamoorthy, An Exact Solution Approach Based on Shortest-Paths for  $p$ -Hub Median Problems, *INFORMS Journal on Computing*, Forthcoming
- [40] I.R. de Farias, A family of facets for the uncapacitated  $p$ -median polytope, *Operations Research Letters* 28 (2001) 161 – 167

- [41] E. Feldman, F.A. Lehrer y T.L. Ray, Warehouse Location under Continuous Economies of Scale and Applications, Oxford University Press, Nueva York, 1995.
- [42] H. Frank, Optimum Locations on a Graph with Probabilistic Demands, Operations Research 14 (1966) 409-421.
- [43] H. Frank, Optimum Location on Graph with Correlated Normal Demands, Operations Research 15 (1967) 552-557.
- [44] F. García-Lopez, B. Melián-Batista, J.A. Moreno-Pérez, J.M. Moreno-Vega, Parallelization of Scatter Search for the  $p$ -Median Problem, Parallel Computing 29 (2003) 575-589
- [45] A.J. Goldman, Optimal Center Location in Simple Networks, Transportation Science 5 (1971) 212-221
- [46] A.J. Goldman y C.J. Witzgall, A Localization Theorem for Optimal Facility Placement, Transportation Science 4 (1970) 406-409.
- [47] S.L. Hakimi, Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph, Operations Research 12 (1964) 450-459.
- [48] S.L. Hakimi, Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems, Operations Research 13 (1965) 462-475
- [49] H.W. Hamacher, M. Labbé, S. Nickel, T. Sonneborn, Adapting polyhedral properties from facility to hub location problems, Discrete Applied mathematics (article in press)
- [50] P. Hansen, N. Mladenović, Variable neighbourhood search for the  $p$ -median, Location Science 5 (1997) 207 – 226.
- [51] R. Hassin y A. Tamir, Improved Complexity Bounds for Location Problems on the Real Line, Operations Research Letters 10 (1991) 395-402.
- [52] S.A. Helm y M.A. Venkataramanan, Using Simulated Annealing to Solve the  $p$ -Hub Location Problem, IRMIS Working Paper, 9304, School of Business, Indiana University, Bloomington, Indiana, 1993.
- [53] K. Hogan y C. ReVelle, Concepts and Applications of Backup Coverage, Management Science 32 (1986) 1434-1444.
- [54] E. Hoover, The Location of Economic Activity, New York (1948)
- [55] M. Hribar, M.S. Daskin, A dynamic programming heuristic for the  $p$ -median problem, European Journal of Operational Research 101 (2001) 430 – 436.

- [56] W.L. Hsu, The Distance-Denomination Numbers of Trees, *Operations Research Letters* 1 (1982) 96-100.
- [57] B.Y. Kara, B.Ç. Tansel, On the single assignment  $p$ -hub center problem, *European Journal of Operational Research* 125 (2000) 648 – 655
- [58] O. Kariv y S.L. Hakimi, An Algorithmic Approach to Network Location Problems. II: The  $p$ -Medians, *SIAM Journal on Applied Mathematics* 37 (1979) 539-560.
- [59] A. Kauffman, *Introduction à la théorie des sous-ensembles flous*, Masson et Cie, Vol. I (1973).
- [60] L. Kaufman y F. Broeckx, An Algorithm for the Quadratic Assignment Problem Using Benders' Decomposition, *European Journal of Operational Research* 2 (1978) 204-211.
- [61] J.G. Klincewicz, Heuristics for the  $p$ -Hub Location Problem, *European Journal of Operational Research* 53 (1) (1991) 25-37
- [62] J.G. Kincewicz, Avoiding Local Optima in the  $p$ -Hub Location Problem Using Tabu Search and Grasp, *Annals of Operations research* 40 (1992) 283-302.
- [63] P. Kouvelis y G. Yu, *Robust Discrete Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, Holanda ,1997.
- [64] A.A. Kuehn y M.J. Hamburger, A Heuristic Program for Locating Warehouses, *Management Science* 9 (1963) 643-666.
- [65] D. Kutangila, J.L. Verdegay, Fuzzy graphs as basis for solving some public transportation problems in Kinshasa, *Emergent Solutions for Information and Knowledge Economy, Proceedings of the 10<sup>th</sup> SIGEF Congress 2 (October 2003)* 71 – 77.
- [66] D. Kutangila, Optimización borrosa aplicada al problema de la  $p$ -mediana: localización de centros comerciales y estaciones de autobús en la ciudad de Kinshasa, Memoria de Investigación Tutelada, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada (Junio 2004)
- [67] M. Labbé, D. Peeters y R.E. Wendell, Location on Networks, en: M.O. Ball, T.L. Magnanti, C.L. Monma y G.L. Nemhauser (eds.), *Network Routing*, Elsevier Science, Amsterdam, 1995.
- [68] M. Labbé, J.F. Thisse y R.E. Wendell, Sensitivity Analysis in Minisum Facility Location Problems, *Operations Research* 39 (1991) 961-969.

- [69] Y.F. Lan y Y.L Wang, An Optimal Algorithm for Solving the 1-Median Problem on Weighted 4-Cactus Graphs, *European Journal of Operational Research* 122 (2000) 602-610.
- [70] L.A.N. Lorena, E.L.F. Senne, A column generation approach to capacitated p-median problems, *Computers & Operations Research* 31 (2004) 863 – 876
- [71] F.E. Maranzana, On the Location of Supply Points to Minimize Transport Costs, *Operational Research Quarterly* 15 (1964) 261-270
- [72] D.W. Matula y R. Kolde, Efficient Multi-Median Location in Acyclic Networks, *ORSA/TIMS Bulletin*, n °2, Noviembre 1976, (abstract).
- [73] P.B. Mirchandani, Locational Decisions on Stochastic Networks, *Geographical Analysis* 12 (1980) 172-183.
- [74] P.B. Mirchandani y A. Oudjit, Locating 2-Medians on Probabilistic and Deterministic Tree Networks, *Networks* 10 (1980) 329-350.
- [75] P.B. Mirchandani, A. Oudjit y R.T. Wong, Multidimensional extensions and a Nested Dual for the m-Median Problem”, *European Journal of Operational Research* 21 (1985) 121-137.
- [76] J.A. Moreno, J.M. Moreno, J.L. Verdegay, Fuzzy location problems on networks, *Fuzzy Sets and Systems* (article in press)
- [77] J. Morris, On the Extent to Which Certain Fixed Charge Depot Location Problem Can Be Solved by LP, *Journal of the Operational Research Society* 29 (1978) 71-76.
- [78] M. O'Kelly, Activity Levels at Hub Facilities in Interacting Networks, *Geographical Analysis* 18 (1986) 343-356.
- [79] M. O'Kelly, The Location of Interactng Hub Facilities, *Transportation Science* 20 (1986) 92-106.
- [80] M. O'Kelly, A Quadratic Integer Program for the Location of Interacting Hub Facilities, *European Journal of Operational Research* 32 (1987) 393-404
- [81] M. O'Kelly, Hub Facility Location with Fixed Costs, *Papers in Regional Science: The Journal of the RSAI* 71 (1992) 293-306.
- [82] M. O'Kelly, D. Skorin-Kapov y J. Skorin-Kapov, Lower Bounds for the Hub Location Problem, *Management Science* 41 (1995) 713-721.
- [83] W. Pedrycz y F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, The MIT Press, London.

- [84] H. Pirkul, R. Gupta y E. Rolland, Visopt: A Visual Interactive Optimization Tool for  $p$ -Median Problems, *Decision Support Systems* 26 (1999) 209-223.
- [85] C. ReVelle, A perspective of Location Science, *Location Science* 5 (1997) 3 – 13
- [86] C. ReVelle y V. Marianov, A Probabilistic FLEET Model with Individual Vehicle Reliability Requirements, *European Journal of Operational Research* 53 (1991) 93-105.
- [87] C. ReVelle y S. Synder, A Shortest Path Model for the Optimal Timing of Forest Harvest Decisions, *Environment and Planning B* 23 (1996) 165-175.
- [88] E. Roland, D.A. Schilling, J.R. Current, An efficient tabu search procedure for the  $p$ -median problem, *European Journal of Operational Research* 96 (1996) 329 – 342
- [89] K.E. Rosing, E.L. Hillsman y Rosing-Vogelaar, The  $p$ -Median Problem, *Geographical Analysis* 11 (1979) 86-89.
- [90] K.E. Rosing, C.S. ReVelle, D.A. Schilling, A gamma heuristic for the  $p$ -median problem, *European Journal of Operational Research* 117 (1999) 522 – 532.
- [91] D.A. Schilling, K.E. Rosing, C.S. ReVelle, Network distance characteristics that affect computational effort in  $p$ -median location problems, *European Journal of Operational Research* 127 (2000) 525 – 536
- [92] D. Serra, V. Marianov, The  $p$ -median problem in a changing network: the case of Barcelona, *Location Science* 6 (1998) 383 – 394
- [93] D. Skorin-Kapov y J. Skorin-Kapov, On Tabu Search for the Location of Interacting Hub Facilities, *European Journal of Operational Research* 73 (1994) 502-509.
- [94] D. Skorin-Kapov, J. Skorin-Kapov y M. O'Kelly, Tight Linear Programming Relaxations of Uncapacitated  $p$ -Hub Median Problems, *European Journal of Operational Research* 94 (1996) 582-593.
- [95] K. Smith, M. Krishnamoorthy y M. Palaniswami, On the Location of Interacting Hub facilities: Neural Versus Traditional Approaches, *Location Science* 4 (1996) 155-171.
- [96] J. Sohn, S. Park, Efficient solution procedure and reduced size formulations for  $p$ -hub location problems, *European journal of Operational Research* 108 (1998) 118 – 126.
- [97] J. Sohn y S. Park, A Linear Program for the Two-Hub Location Problem, *European Journal of Operational Research* 100 (1997) 617-622.



- [98] A. Tamir, An  $O(pn^2)$  Algorithm for the p-Median and Related Problems on Tree Graphs, *Operations Research Letters* 19 (1996) 59-64.
- [99] M.B. Teitz y P. Bart, Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph, *Operations Research* 16 (1968) 955-961
- [100] URL: <http://www.water.hut.fi/wr/research/glob/publications/Haapala/pdf-files>
- [101] URL: <http://nbochinski.free.fr/galerie/lecaire/notice-lecaire.html>
- [102] S. Versan, J. Quon y A. Charnes, *American Public Works Yearbook* (1962)
- [103] Weber, 1957. *Uber den Standort der Industrien* (On Location of Industries) 1909 (Traducción en inglés por C. Friedrich (1957)). University of Chicago Press, Chicago.
- [104] J.R. Weaver y R.L. Church, Computational Procedures for Location Problems on Stochastic Networks, *Transportation Science* 17 (1983) 168-180.
- [105] G.O. Wesolowsky, Probabilistic Weights in the One-Dimensional Facility Location Problem, *Management Science* 24 (1977) 224-229.
- [106] P. Zhao and R. Batta, "Analysis of Centroid Aggregation for the Euclidian Distance p-Median Problem", *European Journal of Operational Research*, Vol 113, Elsevier(1999), 147-168.